Solent University

Faculty of Business, Law and Digital Technologies

MSc APPLIED AI AND DATA SCIENCE
2021/2022

ROSELINE OMETERE AGEITU

CUSTOMER SEGMENTATION TO IMPROVE SUPPLY CHAIN MANAGEMENT

Supervisor	**:**	**Dr. JARUTAS ANDRITSCH**
Date of submission	**:**	**September, 2022**

Acknowledgements

Firstly, I would like to thank God almighty who gave me the strength and grace to start and finish this project, I also like to say a big thank you to my supervisor Dr. Jarutas who has guided me diligently throughout the process of this research and also a big thank you to all my aimable lectures, who taught me one module or the other, thank you for igniting that shining light in me.

My immerse gratitude goes to my family, my in-laws who never fail to show up when necessary both physically and in all ramifications, I am grateful, finally, I would like to thank my darling husband for his unflinching support, love and care all through the journey, God bless you and also to my children Samuel and Abigail, thank you for not for being the reason I never gave up, I love you both.

Abstract

Growth and survival in today's ever changing business environment requires innovation and the will to adapt. All of these depend on the level at which available resources are utilized to reach, acquire and retain the primary source of business survival – the customer. This study considers supply chain as a critical structure that not only position the customer at the heart of its success; but will ensure that a business can weather the tumultuous global business environment. The ultimate goal of this project is to explore how a properly segmented customer base can improve the level of optimization in a supply chain management. Although Marketing activities have always contributed to supply chain management, designing customer analytic systems with operational needs in mind will greatly improve manufacturing outcomes and minimize information loss that may be encountered while processing customer feedback from a single marketing lens.

The literature was sequentially reviewed to give insight to the field of engagement – Retail, provide a background on how customer satisfaction is the ultimate goal of supply chain management as well as give insight to the available technology driven models for segmenting data and how researchers see each model. This provides a slid basis for the choice of using a hybrid between Recency Frequency Monetary (RFM) segmentation and K-Cluster analysis models for understanding the dataset. This combination worked positively for the set goals because it provided results on product consumption which is critical for our findings to support optimization of a supply chain. This would not be the first time a researcher has combined two models (Dogan, et al, 2018; Fader, et al, 2005; Lumsden et al, 2008), There are also arguments in certain depositions that most of the existing segmentation approaches are vulnerable to outliers and their outcomes may be seriously biased (Wang, 2010).

This study has successfully connected various information gathered from our data analysis to key aspects of supply chain management like; Inventory management,

Contents

List of Tables

List of Figures

1. Introduction

Satisfying customers in a retail business requires a high level of organisation and reliability. The consistent innovation in Information Technology (IT), have continued to disrupt the industry and businesses must adapt to adapting to remain relevant. According to a Journal of Retailing Article;

> *"Innovations in business models are increasingly critical for building sustainable advantage in a marketplace defined by* ***unrelenting change, escalating customer expectations, and intense competition***" (Sorescu, et al, 2011, p.)

Global businesses have continued to recover from the impact of Coronavirus (COVID19) pandemic - with over six million dead and over half a billion infected in less than 36 months (World Health Organisation, 2022). The impact almost halted all forms of physical interactions all over the world including; travel, instore shopping, product manufacturing and distribution, to mention a few. Man-made disruptions such as: violence in some Arab nations (BBC, 2022), the standoff between Rwanda and the Democratic Republic of Congo (UN Security Council, 2022), terrorist activity in West Africa (Obadare, 2022), Russia's invasion of Ukraine (Lock, 2022) the list continues, have created all kinds of unique needs and disruption of the international business environment. These disruptions are very relevant external influences on the supply chain of retail business – we will have a detailed discussion of this impact later in this report, and the business must develop strategies to adapt and thrive or become obsolete. Adaptation just like other interactive processes require information about the problem source to develop solutions for co-existence. In this case, the business needs an understanding of its stakeholder systems, and know what to do with large quantity of information that lay at their disposal (Watson et al, 2012).

The level of success of any business depends on the responsiveness of its supply chain. Hence, supply chains are designed to respond to customer demand. As our business landscape continue to change because of external functions, it is critical that we continue to review our customer engagement process to optimize value and improve retention. One area to look to for such improvement is in the structure of our decisions based on how our customers are wired. Businesses have implemented various Machine Learning solutions to engage and understand customer behaviour but largely targeted for sales and marketing purposes.

This study offers an opportunity to learn more about defining real life datasets, intentionally aggregating them and eventually putting the feedback to use. It would also be my first opportunity to understand business processes for retail businesses. Business innovations have made it possible for small and medium size ventures to participate in international trade, hence the need for an optimized system responding to external influences in an efficient manner

Regardless of point of purchase or utilization, there is a standard operational efficiency required to maintain sustainable growth. The one question that every business owner would ask themselves is, how do I deliver customer satisfaction at the most cost-effective manner? Making such decisions require customer activity data collection, storage, and analysis. For big corporations, they could afford the technology, but still need the "juice" to succeed", and for smaller businesses, it becomes expensive, they resort to having any kind of process that can store information for profit and loss management – other business decisions are taken based on a hunch or replicating best practices. An Investopedia submission described the profit and loss as a summarization of revenue, costs and expenses over a period (Maverick, 2021). Regardless of collation frequency, the content can only be used to determine viability based on concluded events, it does not offer insight into strategic outcomes and is always an after-the-fact information. How then can a business owner be more intentional about what they see in their profit and loss?

In this study we have adopted a real-life data collated by a medium size organisation that specialize in a niche product but supply to a global audience – a larger percentage of their customers are wholesalers, which puts them at a responsibility position of the distribution channel. The data may well be in its raw form, our goal is to develop a python-based algorithm that will help us investigate the state of the business, generate critical data and understand the behaviour of customers based on set out parameters. The outcome will form the basis for recommendations on how to improve loyalty and increase sales. Training a python-based algorithm to sample and analyze existing business activity datasets like; time stamps, purchase volume and preferences, customer demography and traffic, will not only improve the customer satisfaction but also manage waste for the business. It is not enough to have good relationship with your customers – global trade and situations like COVID 19 and other disruptions have also made it important to understand and predict correctly if the business must remain relevant.

1.1 Purpose

The purpose of this study is to use a python algorithm to analyse customer behaviour that will enable an online retail business improve on its supply chain management system.

1.2 Assumptions

In this review we are working with a sample data and choosing this data comes with the following assumptions;

i. Collected data are real life documentations and were collated applying the retailer's data policy.

ii. The business in question is majorly an online retailing business without payment and delivery constraints

iii. The business is consumer focused and has the ability to service international customers

iv. All items described are tangibles who may or may not be in their best sales conditions

2. Literature Review

This chapter explore the contextual background of this study by reviewing existing information on the relationship between the key aspects of the study. From understanding how retailing as an end-to-end process has evolved over time, to presenting the concept of Supply Chain Management and how its goal is customer satisfaction. Also, a review on the use of customer segmentation on other business aspects – especially how data science has made it possible to understand customers better. Finally, a review will be done on some aspects of data analysis to give a proper background on the operations of the information management value chain.

2.1 Evolution of the Retail Business

Retailing simply provides a medium for individuals/entities to purchase at their desired quantities and freedom to choose the end point of the product (Rina, 2013). Over centuries, the way people shop for goods and services – especially for individual use, have continued to evolve, from single pop shop formats in the 1800s to large retailing chains and multi-product single-owner online shops in 2000s. These evolutions have also happened at different timelines for different locations – whereas India waited till about the year 1999 to have a valuation of their retail environment, large retail shops like JC Penny (1902) and Walmart (1970) were already balancing a structured United States Market (KPMG, 2009). The oxford learners dictionary defines retail as "the selling of goods to the public, usually through shops" (Oxford University Press, 2022). While retailing can be described as all the activities involved in retail and the key player becomes the retailer. In this study, our focus business is a hybrid retailer because it is serving at two levels of the retailing structure of occasion gifts – wholesale and last mile retail.

In a 2019 paper, presented at The Institute of Physics (IOP) Conference series, a team led by Irina Krasyuk had an extensive discussion on how technology systems have impacted strategic retail practices (Krasyuk, et al, 20019). It was important to note that technology was not the only driving factor in the evolution of retailing, there were other key factors such as government regulations, competition, investment appetite, consumer culture shift, and businesses consolidation. All these factors make up the global business' environment and each business with international intent must learn how to balance them to remain relevant. The turn of the 21st century saw the online retail ecosystem witness massive leap in the use of Information Technology, and organisations found easier ways to communicate, improve delivery networks, process payment, implement smart storage and inventory management. This resulted in more confidence and increase in number of small businesses embracing online retail - According Market Place Pulse, the number of third-party sellers on Amazon grew from a little less than 3 million in 2017 to over 6 million in 2017 (Kaziukėnas, 2021).

2.2 Supply Chain Management

A supply chain is a system-based network set up to deliver goods and services from raw material stage to the consumer while ensuring the return flow of payment. Bringing this definition into business relationships, Martins Christopher, described a supply chain as a supply chain as an entity network, where different organisations participate in creating value for an ultimate goal customer satisfaction (Christopher, 1998). Such systems require coordination, making sure that each stakeholder deliver as promised through a functional reward and support system. This coordination is termed supply chain management (SCM). Supply chain management can be described as coordinating a network of functions aimed at optimizing the delivery of goods and services from raw material stage to the final consumer stage – with reverse flow of cash across all stages

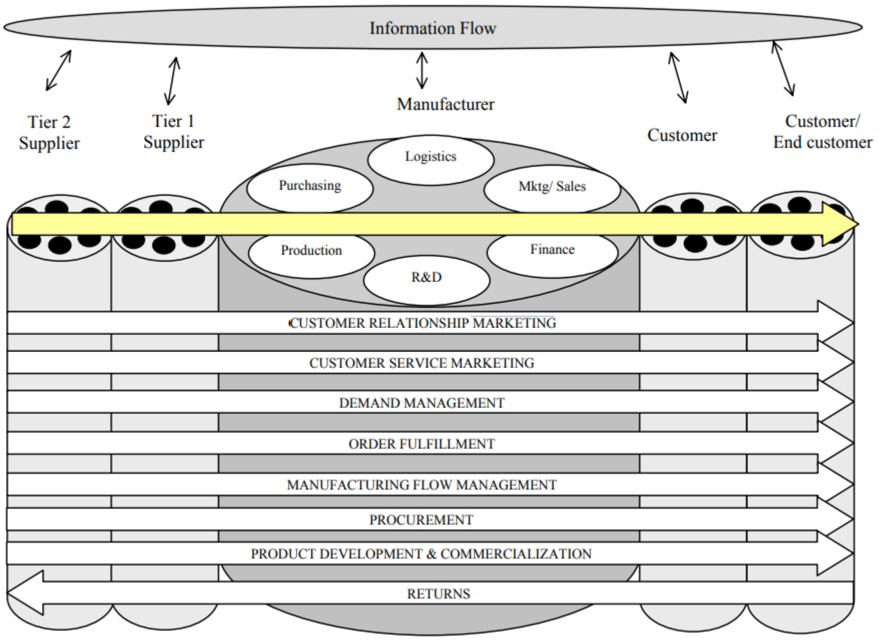


Fig 1. Supply Chain Management Image Source: (Cooper, et al, 1998)

Fig. 1, depicts a practical flow of information in a supply chain. A supply chain structure, a combination of the three pillars of supplier, manufacturer, consumer and other components that ensure consumer demands are met at a competitive price (optimization).

The supplier stage includes all the organisations or individuals who are involved in providing the raw materials for production. Their activation comes from the forward demand by the production facility occasioned by customer demand. In most cases there are secondary suppliers, who sell to the primary supplier.

The manufacturer stage is the point that raw materials are turned into finished products and then shipped off to the consumer point. This stage controls the supplier and responds to the consumer. It has very distinct activities like logistics, and resource management. It is the link between the supplier and the consumer.

The consumer is the stage that creates the demand for products. It is the satisfaction measurement point of a supply chain. This stage includes distribution channels and last mile consumer activities. "In order to achieve this coordination/integration of all the links in the supply chain information is critical" (Giménez and Lourenço, 2004) and the principal source of demand information to be responded to emanates from the customer. Hence the importance of understanding customer behaviour to the success of process. Effective supply chain managements apply "system thinking". Objectively, "systems thinking is a set of synergistic analytic skills used to improve the capability of identifying and understanding systems, predicting their behaviours, and devising modifications to them in order to produce desired" (Arnold and Wade, 2015). Therefore, no function can be considered independent of each other. Segments are differentiated instead of standardizing them.

2.3 Customer Segmentation

The term customer segmentation has become an increasingly popular concept. It is the root of modern-day data driven marketing practices. Adopting a simple definition;

> "*Customer segmentation is the process of examining customer attributes and creating groups based on how they behave, who they are, and their specific characteristics*" (Cousera, 2022)

It is a process that enable the organisation to utilize an optimized approach to better target their customers. Each customer has a different engagement journey, thus applying a single strategy will only result in keeping just a set of customers – dividing your customers in subgroups means you can create unique strategies to satisfy multiple groups at the same time. A business could decide to group its prospective customers based on its value proposition; this will easily be termed "market segmentation". Based on our research data set, our study will rely on internally known information to make decisions. With tens of customers, one could achieve customer segmentation by a simple record keeping, but with hundreds of customers and millions of demand activities, a reliable aid will be required – improvements in data science and machine learning have greatly helped how organisations collect, process and store large amounts of data.

Today there exist various software solutions that enable organisations perform these operations and are generally called Customer Relationship Management (CRM) Solutions. Solutions such as; Salesforce, Oracle, Adobe, Microsoft, SAP and many others function as an integrated solution allowing businesses to manage multiple business operations under one solution. There are also role specific CRMs such as: conversation trackers - Sprout Social, HubSpot etc, and email marketing solutions – Mailchimp, Sendinblue, Bitrix24, Omnisend to mention a few. On the heels of cloud computing, CRM solutions have continued to grow as organisations embrace digital transformation – CRM adoption grew by 15.6% in 2018 and was responsible for approximately 25% of growth in Enterprise software adoption same year (Gartner, 2019). Our concern for this review is understanding how advancement in data science has supported customer segmentation.

2.4 Segmentation Approaches

There have been various approaches to data segmentation, each of them has proved to show strength for different purposes and business sectors. This segment will review four of these approaches based on their historic relevance and their se in our analysis. The review enabled us get conviction on or choice of models and investigate possible loopholes based on existing research. It also helped us to understand the points of convergence for the possibility of applying two modules in order to balance out possible outlier issues.

2.4.1 CHAID

Chi-square Automatic Interaction Detector was a technique introduced by George Kass in 1980 (Kass, 1980) and applies the use of root predictors and nodes to discover the relationship between variables. It is usually very effective when responses to correspondence is low – usually to a very small audience (McCarty and Hastak, 2007). The outcomes are always represented in a decision tree as shown in the fig 2. In this technique, one can easily spot the relationships between split variables and the associated factors within a tree. The tree classification begins by identifying a target variable and then splitting it into nodes, while the nodes are split into child variables. It is widely used in market research for determining relation

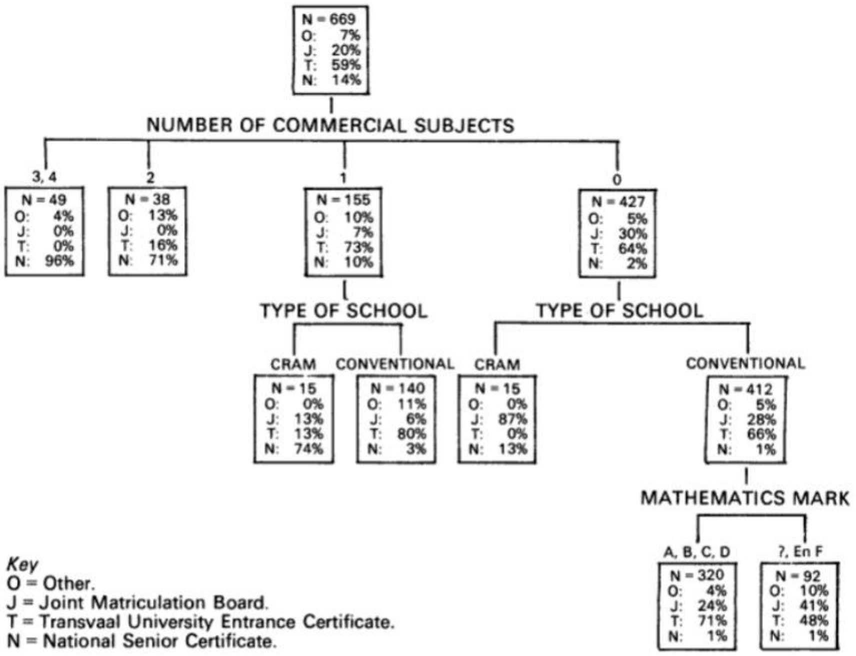


Fig 2. Analysis of South Africa Matriculation Board Image Source: (Kass, 1980)

2.4.2 Logistic Regression

Logistic Regression (LR) is a segmentation model widely used to describe the relationship between a set of outcome variables – usually referred to as "response" and another set of independent variables, referred to as predictor or explanatory. Discovered in the 19th Century by a French mathematician, Pierre François Verhulst, and as applied in the description of growth in human population and the course of autocatalytic chemical reactions (Hosmer, 1989). According to Boateng et al,

"LR is used when the research method is focused on whether or not an event occurred, rather than when it occurred (time course information is not used). It is particularly appropriate for models involving disease state (diseased or healthy) and decision making (yes or no), and therefore is widely used in studies in the health sciences.LR has been extensively applied in medical science research" (Boateng and Abaye, 2019).

They also went on to allude that LR works optimally with large samples of data to provide sufficient numbers for the outcome variables – as the independent variables are increasing, so should the sample data increase. LR is very similar to linear regression in the sense that they share all the basic dimensional attributes, the only distinguishing factor is that while the outcome variables are continuous in linear regression, they are binary or dichotomous in LR.

2.4.3 RFM Segmentation

RFM segmentation is a marketing technique that allows companies to do a behavioural customer segmentation, classifying the customer by considering the client's Recency(R), Frequency(F), and Monetary Value(M) of the spending. Recency is defined as how many days since the customer's last purchase. Frequency is defined as how often the client makes a purchase, and is the total number of yearly number purchases made by a customer. Monetary Value is the total money spent by a customer. RFM Segmentation has been accepted by most scholars as the most widely adopted model in customer segmentation. This not unconnected with its touted simplicity and the ability of decision makers to understand their most valuable customer and plan strategies to retain them (Wang, 2010). RFM represents the key measurement dimensions of the model; “R” for Recency, which measures how recent the customer has purchased, “F” for Frequency that measure how often the customer patronise the business and finally “M” for Monetary, which measures the monetary value of a customer’s purchase over the period. RFM is characterised by the following features;

i. All customers are known and assigned unique IDs for easy identification

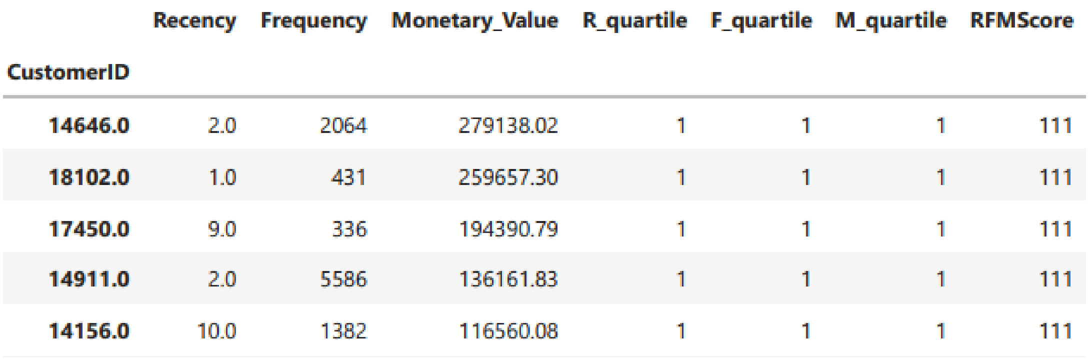
ii. Dataset an only contain existing customers, hence might not be very useful or acquiring new customers

iii. Datasets with incomplete data or negative values must be abandoned

iv. Only tangible and measurable dimensions can be utilized, emotions and abstract dimensions cannot be measured in RFM

v. There must be an infinite time period in review; days, months year.

Wei et al in their detailed review of RFM implementation noted the first step to be; sorting the database into the different dimensions of RFM and then dividing them into five equal quintiles. The study alludes that these quintiles are usually of equal size – each having different response rates. Recency which is noted as the most important dimension is defined by the period between the last purchase, which means that the lower number of days translates to higher recency ((Wei, et al, 2007). The quintiles are then assigned codes from 1 to 5 placing the 20% highest number of at the lowest number of 5 and the next at 4 in a progressing manner. This stacking process places the most visiting subgroup in the highest value of recognition - 5. A review of three studies came to this conclusion. The process is applied to the Frequency, by assigning codes 1-5 (5 being the top subgroup code) based on the number of times the customer has completed purchase and is expected to consider single and repeated incidents. Customers with a high Frequency score are likely to always return to purchase more and could be termed as the most loyal customers. Finally, the database is sorted based on Monetary value, stacking the entries based on the monetary value - in their chosen currency, of total purchases completed by each customer. At the end, customers get assigned three numbers with 555 being the most valued customers and 111 being the least valued customers. It is important to note here that different organisations may choose different conventions when implementing same model. There are instances where quartiles are used – in this case, codes are assigned from 1-4 and subgroups are divided into 25% size quartiles (Jha, et al 2020). The analyst can also choose to assign 1 or 5 as the highest, so long as the convention used are clearly understood by the stakeholders.

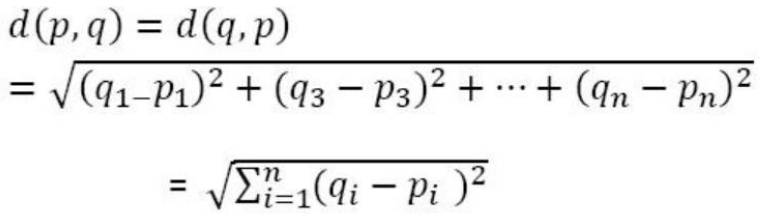


	Recency	Frequency	Monetary_Value	R_quartile	F_quartile	M_quartile	RFMScore
CustomerID							
14646.0	2.0	2064	279138.02	1	1	1	111
18102.0	1.0	431	259657.30	1	1	1	111
17450.0	9.0	336	194390.79	1	1	1	111
14911.0	2.0	5586	136161.83	1	1	1	111
14156.0	10.0	1382	116560.08	1	1	1	111

Table 1. Sample illustration of a completed RFM table using quartiles

2.4.4 K-Means Clustering

Generally, clustering is the process that allows for the division of data points into homogenous sets or clusters. Items are grouped as close as possible in similarity for the purpose of decision making. K-Means is an unsupervised machine learning technique that groups data points based on their closeness to each other. It is the most popular and widely used by marketing practitioners to achieve better engagements with their customers. It is also applied in product clustering for instance; buildings could be clustered based on their location, value, size etc, the ones with the closest similarities are retained in one cluster. According to MacQueen, the K-means process was originally devised in an attempt to establish a feasible method of computing that can achieve optimal partitions (MacQueen, 1967). The K-means algorithm utilizes “Euclidean Distance” to find out value between two points (Gadde, 2021).



$$d(p,q) = d(q,p)$$
$$= \sqrt{(q_{1-}p_1)^2 + (q_3 - p_3)^2 + \cdots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n}(q_i - p_i\ )^2}$$

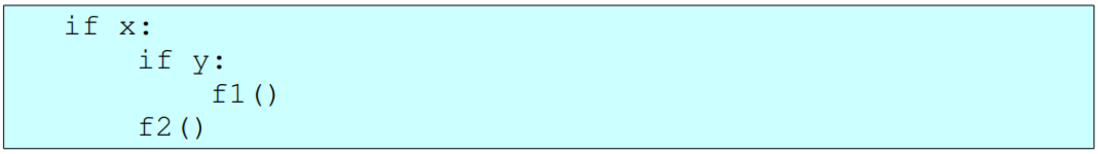
Fig 3. Euclidean Distance Equation Image Source: (Gadde, 2021)

Although K-means has a simple appearance and can be executed very quickly, the behaviour of its algorithm is quite complex (Davidson, 2002). Illustrated a simple application of K-means in helping an organisation to decide the location of a new Pizza franchise, by understanding the areas where Pizza can be ordered and plotting the new location at the universal or closest distance to each point. Those points can *****

2.5 Python

Machine Learning simply put is training machines to enable us unearth information that are not available in the existing data. To achieve this goal, one must be able to interact and speak the language of machines. The process of propagating such language is called programming, a simple definition was used in Southern New Hampshire University Website (SNHU), “Computer programming is the process that professionals use to write code that instructs how a computer, application or software program performs” (SNHU, 2022). When sending instructions, it must be done with a mutual language between the sender and the receiver. In this case, Python is the programming language we will be using to instruct our machine for this purpose.

Python is used for creating backbone structure. Python is intended to be a highly readable language. It is designed to have an uncluttered visual layout, it uses whitespace indentation, rather than curly braces or keywords. Python has a large standard library, commonly cited as one of Python's greatest strengths. It has a rich set of built-in structures which makes data analysis exciting. Python’s data structure is simple but very powerful, an analyst does not need to be very proficient in building software in python to productively do data analysis (McKinny, 2018). Python’s strong structuring constructs (nested code blocks, functions, classes, modules, and packages) and its consistent use of objects and object-oriented programming, it enables the user to implement clear, logical applications for small and large projects (Kuhlman, 20013). Kuhlman also highlighted the fact that Python represents block structures with indentation, where indent one level shows the beginning of a block and outdent one level shows the end of a block.



```
if x:
    if y:
        f1()
    f2()
```

Fig 4. Illustrating Python indentation Image Source: (Kuhlman, 2013)

2.6. Data Mining

Institutions require a fair amount of data to make decisions. It could be a small retailer who remembers that a certain customer needs two books every fortnight or a supplier who remembers that her client prefers the rose fresh and leafy – regardless of the size of the type of business, such individuals act on a pre-stored information. The more information stored; the more queries required for decision making. These complexities gave rise to the advent of data mining (Han and Kamber, 2006). According to their study, data mining is; “the process or method that extracts or “mines” interesting knowledge or patterns from large amounts of data” (Han and Kamber, 2006, p. 3). Each entity could decide how their data get to them and how they keep it. As computing evolves, various means have been used to store data: data warehouse, independent database, online or offline locations to mention a few.

Data mining ensures that an organisation is not limited in the kind of data they collect, at the same time make use of only relevant information – saving time and resources. Usually, an organisation would store historic data in their chosen format and location, then set up a master repository which will contain all the required knowledge base needed for that focus operation. A server is then set up to fetch this information from the data location for comparison with knowledge base – using a match of key words. Pattern evaluation modules guide the search to understand patterns. Functional modules present in data mining engines can perform functions such as; association, classification, cluster deviation analysis, and evolution. Finally, a graphical interface that will enable the user interact visually with the data mining system.

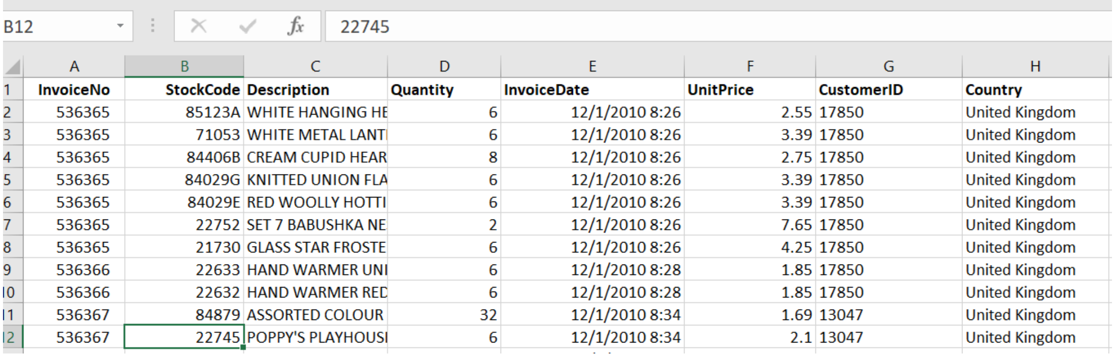
3. METHODOLOGY

For this research methodology a secondary dataset was obtained from the database of an online retail company base in United Kingdom who sells gift items, the research employs qualitative methods to analysis the dataset, Fig 5. gives a diagrammatic expression of our approach. The proposed research methodology includes four major steps, the first phase was sourcing data and exploring analytic models – this was concluded as part of the pilot study, the next phase is data pre-processing phase in this phase we clean up the data and ensure it is readable for a K-means approach. The processed data is then analysed using RFM segmentation, the RFM segmentation is a marketing technique that allows companies to do a behavioural customer segmentation, classifying the customer by considering the client's Recency(R), Frequency(F), and Monetary Value(M) of the spending and K-means cluster, K-Means which is an unsupervised machine learning technique that groups data points based on their closeness to each other. Finally, we present the results for interpretation.

This research complies with all ethical and professional standards in accordance with university guidelines (Appendix B).

3.1 Understanding the Dataset

The data set is an open data collected between January, 2010 and December 2011 by a UK based gift company accessed via University of Chicago Irvine, machine learning repository (UCI, 2012).



	A	B	C	D	E	F	G	H
1	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
2	536365	85123A	WHITE HANGING HE	6	12/1/2010 8:26	2.55	17850	United Kingdom
3	536365	71053	WHITE METAL LANTI	6	12/1/2010 8:26	3.39	17850	United Kingdom
4	536365	84406B	CREAM CUPID HEAR	8	12/1/2010 8:26	2.75	17850	United Kingdom
5	536365	84029G	KNITTED UNION FLA	6	12/1/2010 8:26	3.39	17850	United Kingdom
6	536365	84029E	RED WOOLLY HOTTI	6	12/1/2010 8:26	3.39	17850	United Kingdom
7	536365	22752	SET 7 BABUSHKA NE	2	12/1/2010 8:26	7.65	17850	United Kingdom
8	536365	21730	GLASS STAR FROSTE	6	12/1/2010 8:26	4.25	17850	United Kingdom
9	536366	22633	HAND WARMER UNI	6	12/1/2010 8:28	1.85	17850	United Kingdom
10	536366	22632	HAND WARMER REC	6	12/1/2010 8:28	1.85	17850	United Kingdom
11	536367	84879	ASSORTED COLOUR	32	12/1/2010 8:34	1.69	13047	United Kingdom
12	536367	22745	POPPY'S PLAYHOUSI	6	12/1/2010 8:34	2.1	13047	United Kingdom

Table 2. Dataset Structure Data Source: (UCI, 2012)

The dataset contains 541909 rows and 8 columns;

"Invoice No: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.

Stock Code: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.

Description: Product (item) name. Nominal.

Quantity: The quantities of each product (item) per transaction. Numeric.

Invoice Date: Invoice Date and time. Numeric, the day and time when each transaction was generated.

Unit Price: Unit price. Numeric, Product price per unit in sterling.

Customer ID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

Country: Country name. Nominal, the name of the country where each customer resides" (UCI, 2012)

3.2 Implementation

The major tool for this project is python programming language which is a language that provides great libraries to deal with data science application. As pointed out in 2.5 above, one of the main reasons why Python is widely used in the scientific and research communities is because of its ease of use and simple syntax which makes it easy to adapt for people who do not have an engineering background. Our core editor was Jupyter notebook a web-based editor that provide a way to run python code as fast as it will be in a console while giving the ability to write snippets of code.

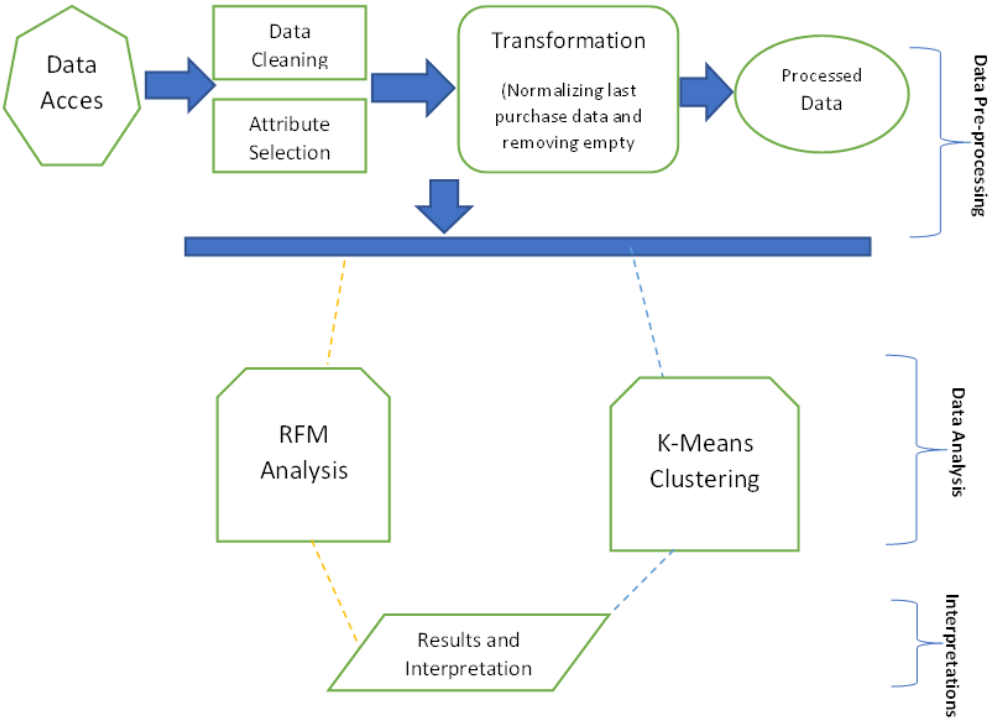


Fig 5. Implementation Approach

After getting the data, the first step was data wrangling and cleaning. Python library called Pandas because it provides us the ability to view the data as tables and perform filtering, sorting and other data cleaning function needed to clean and explore the data. Kuhlman is also of the opinion that Pandas should be viewed more as much about learning techniques for cleaning, exploring and finding aspects as well as the view and display of data. He said, “Pandas contains and provides such a rich set of techniques for working with your data that you should expect to take a reasonable amount of time learning to do the tasks you need, rather than just quickly learn some small set of functions”. (Kuhlman, 2018)

The next step was data exploration and analysis. For the step, we made use of an analytical library called Numerical Python (NumPy). McKinny positioned it as one of the most important foundational packages for numerical computing in Python because most computational packages that provide scientific functions are seen to be using NumPy’s array objects for data exchanged (McKinny, 2018) NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, matrices, etc. We also implemented feature Engineering using “anonymous function. Feature engineering is can be described as a process of extracting and arranging the most important features from raw data for ease of alignment with the machine learning model (Screemany, 2021)

For the visualization of the data, we made use of various libraries like Plotly, Seaborn, Matplotlib, and WordCloud. These libraries provide us with various plot styles and visualizations. Plotly provides an exciting interactive interface which provides for highlighting information – regardless of object size.

Two different customer segmentation algorithms; of which the first one is RFM. RFM segmentation is a marketing technique that allows companies to do a behavioural customer segmentation, classifying the customer considering the client's Recency(R), Frequency(F), and Monetary Value(M) of their spending. Recency is defined as how many days since the customer's last purchase. - Frequency is defined as how often the client makes a purchase, and is the total number of yearly number purchases made by a customer. - Monetary Value is the total money spent by a customer. RFM was reviewed earlier in this study and it was chosen as a first aspect of segmentation to enable of achieve the type of simple alignment towards interpreting our result in combination with K-means.

For Product Description Categorization - The product description feature is a categorical feature, therefore text pre-processing was performed on it using Natural Language Processing (NLPs) techniques like: Stemming, Removing Stop words, and finally, using TF-IDF Vectorizer to convert/encode each word in text to numeric values. In general, NLPs as applied to Artificial Intelligence (AI), is a branch of computing that empowers computers to understand text and voice as much as humans (IBM, 2020)

Truncated SVD algorithm was used to decompose the dimension of resulting sparse matrix. By doing this, I was able to reduce the shape of the features in the data from (3871, 1694) to (3871, 100). In essence, the dataset with a sample size of 3,871 observations and 1694 features to 100 features. Afterwards, I then segmented each of the 3,871 product descriptions in the dataset into 245 clusters using K-means clustering algorithm, The right number of optimum clusters (245) was determined using a metric called Silhouette Score.

A manifold algorithm called t-SNE (t-Distributed Stochastic Neighbour Embedding) was then used to visualize high dimensional data.

To get insight of how and where each product description was categorized, WorldCloud and Matplotlib visualization libraries was combined to randomly show 12 out of the 245 clusters of the product descriptions in our dataset.

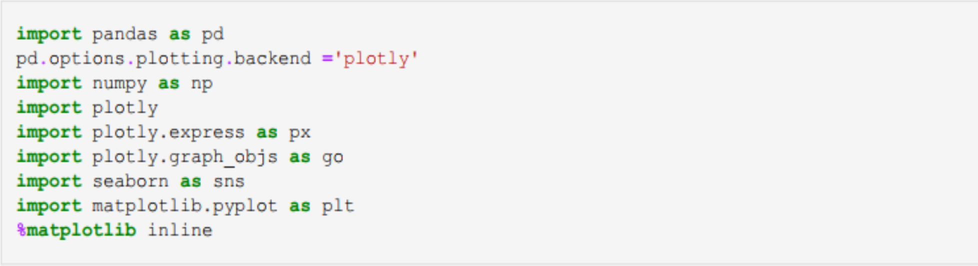
Performing customer segmentation with K-means, the dataset was grouped by Invoice and CustomerID columns respectively, then both data frames were concatenated after the concatenation, the K-means algorithm was used to perform customer segmentation. This time, we used the Elbow method to determine the right/optimum number of clusters to be formed, further use of KElbow Visualizer was use to reverify the right number of cluster (which is 6). Based on the clusters gotten from our clustering algorithm (K-means) Each customer was then profiled so we can identify which cluster each customer actually falls into with ease.

For Cluster Interpretation, Snake Plot and Feature Relative Importance plots was used to interpret each cluster respectively.

4. Results

We have successfully performed a complete analysis using python-based algorithms and will present our findings as follows;

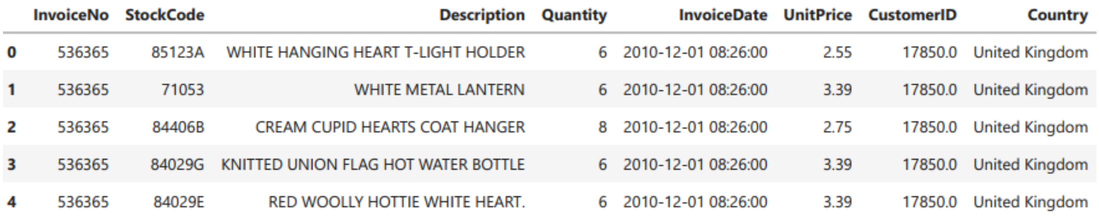
Having imported the relevant base libraries into our programme, we have the following fig



```
import pandas as pd
pd.options.plotting.backend ='plotly'
import numpy as np
import plotly
import plotly.express as px
import plotly.graph_objs as go
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Fig 6. Imported libraries

The dataset was stored as a .xls, but it is preferrable to work with .csv format to allow us have the dataset on my local machine, table XX reflect the data frame to be used;

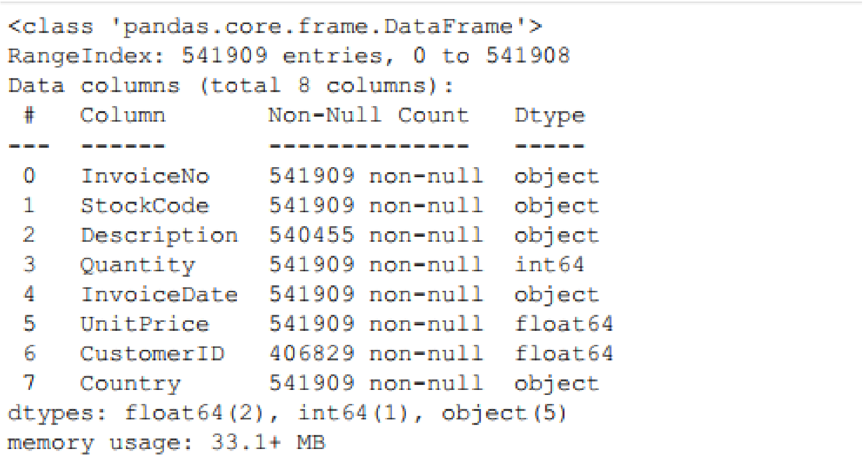


	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Table 3. Dataset in .csv format

The first five instances was shown using the (.head method). It is also noted that Python reads from integer “0” (zero index) which means that our instance number “1” will be reading “0” and replicated across other instances. Our attributes are also consistent with the definitions listed in 3.1 above.

To get further insight into the data set we apply df.info ()



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  object
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

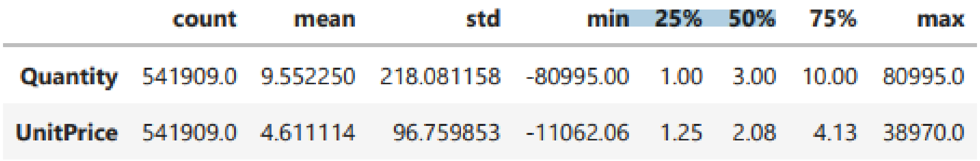
Fig 7. Classifying Column Descriptions

From the information one would notice that all the columns do not have equal number of entries, which could only mean that there are empty cells or missing values.

4.1 Exploratory Data Analysis

4.1.1 Descriptive Statistics

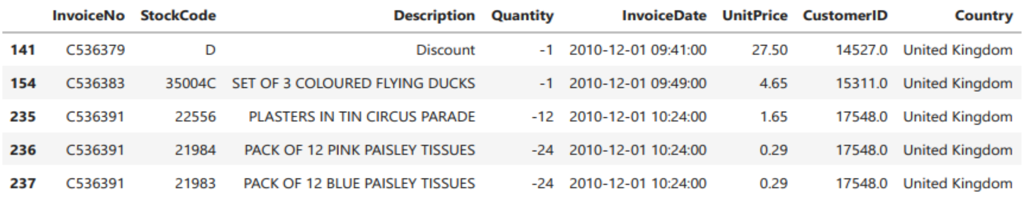
The Exploratory Data Analysis is performed, first point is descriptive statistics. Applying, .describe



	count	mean	std	min	25%	50%	75%	max
Quantity	541909.0	9.552250	218.081158	-80995.00	1.00	3.00	10.00	80995.0
UnitPrice	541909.0	4.611114	96.759853	-11062.06	1.25	2.08	4.13	38970.0

Table 4. Identified columns with Numeric values

We have only two columns present; this is because Quantity and Unit Price are the important numeric columns (.int). It is also realized that the minimum values of both attributes are negative. As our goal is customer segmentation and market basket analysis, it's important that these records are removed, but first we will have to investigate to find out what happened.



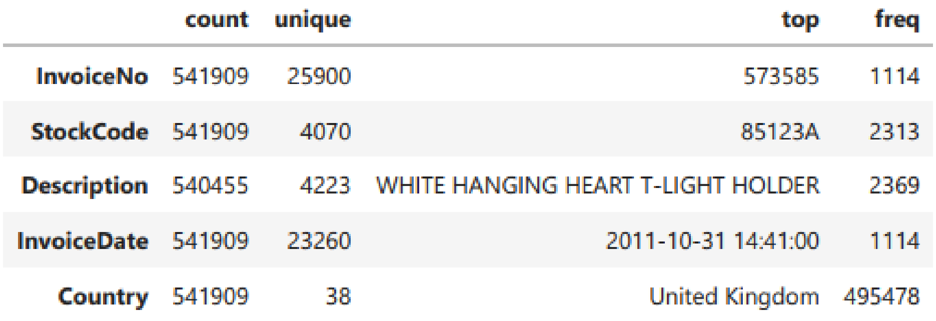
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1	2010-12-01 09:49:00	4.65	15311.0	United Kingdom
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12	2010-12-01 10:24:00	1.65	17548.0	United Kingdom
236	C536391	21984	PACK OF 12 PINK PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom
237	C536391	21983	PACK OF 12 BLUE PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom

Table 5. Outcome of negative value exploration

Interestingly, various things can be pointed out here:

- The stockcode values aren't only numerical, there are special values like “D “which means Discount
- The InvoiceNo aren't also only numerical since there is a C before the other numbers for every negative value in the quantity column, which mean that the order was cancelled.

The InvoiceNo will later be checked for patterns.

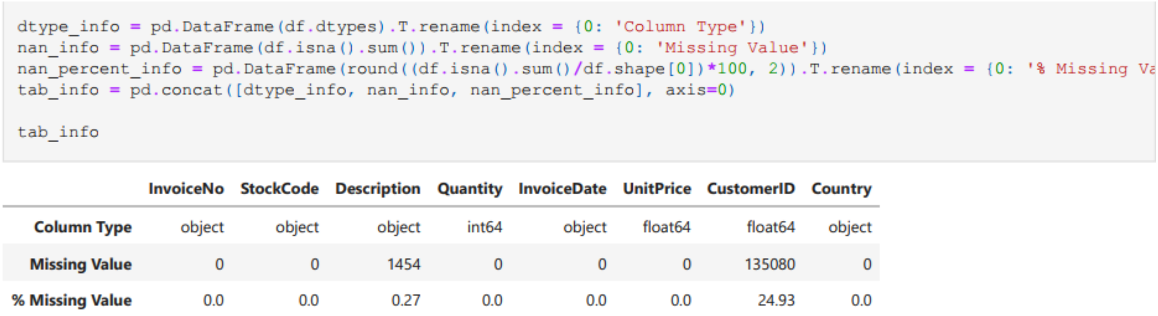


	count	unique	top	freq
InvoiceNo	541909	25900	573585	1114
StockCode	541909	4070	85123A	2313
Description	540455	4223	WHITE HANGING HEART T-LIGHT HOLDER	2369
InvoiceDate	541909	23260	2011-10-31 14:41:00	1114
Country	541909	38	United Kingdom	495478

Table 6. Identifying country with highest data frequency

Of the 38 target markets penetrated by the retail store, it appears the store has more historical data of customers from The United Kingdom. The most purchased product from the store also, appears to be a white hanging t-light holder.

4.1.2 Data Quality Check



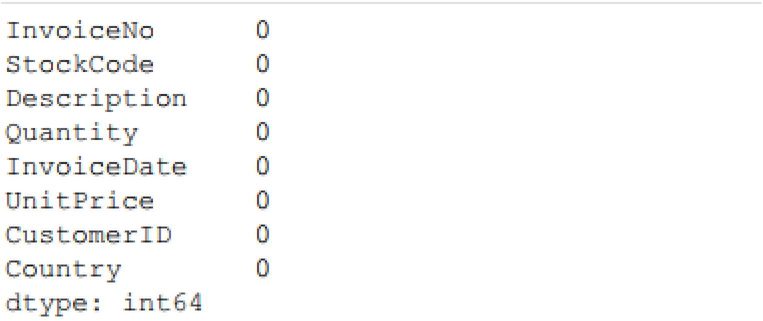
```
dtype_info = pd.DataFrame(df.dtypes).T.rename(index = {0: 'Column Type'})
nan_info = pd.DataFrame(df.isna().sum()).T.rename(index = {0: 'Missing Value'})
nan_percent_info = pd.DataFrame(round((df.isna().sum()/df.shape[0])*100, 2)).T.rename(index = {0: '% Missing Va
tab_info = pd.concat([dtype_info, nan_info, nan_percent_info], axis=0)

tab_info
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
Column Type	object	object	object	int64	object	float64	float64	object
Missing Value	0	0	1454	0	0	0	135080	0
% Missing Value	0.0	0.0	0.27	0.0	0.0	0.0	24.93	0.0

fig 8. Cells with missing values

From the above table, we could tell those two features, “Description” and “CustomerID” have some missing values. About 25% of the entries are not assigned to a particular customer. This can most likely be attributed to the store's negligence in proper information capture on their customers and their purchased product.

After investigations, it’s become evident it's impossible to impute values for the CustomerID.



```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

Table 7. Cells with missing values eliminated

There were also 5225 duplicate rows in the dataset, these were also dropped and only the first in each set of repeats retained.

After removing missing values and dropping duplicate rows in the dataset, the size of the data reduced from $541,909$ rows to $401,604$ rows

4.1.3 Countries

Where Was Each Order Made From?

First, we had to rename certain countries to reflect conventions.

- Rename: EIRE to Ireland, RSA to South Africa, and USA to United States
- Keep: European Community and Unspecified even when they clearly aren't a legit country
- Keep: Channel Islands regardless even though it has no Alpha-3 code and not a country but a British-dependent island off the coast of France.

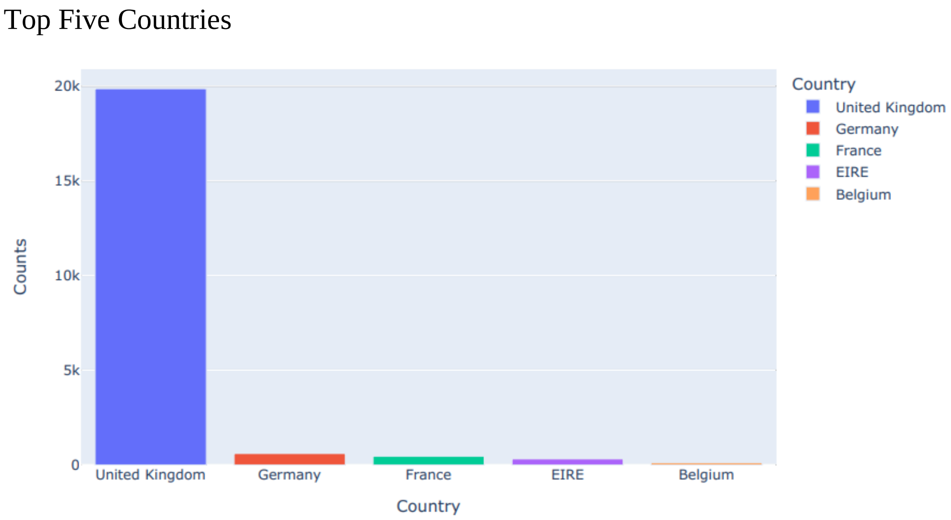


Fig 9. A plot of top order locations

The total number of countries in our dataset is 37. The plot of the top order locations is represented in fig 9

4.1.4 Products and Customers

i. What are the number of users and products in the dataset?

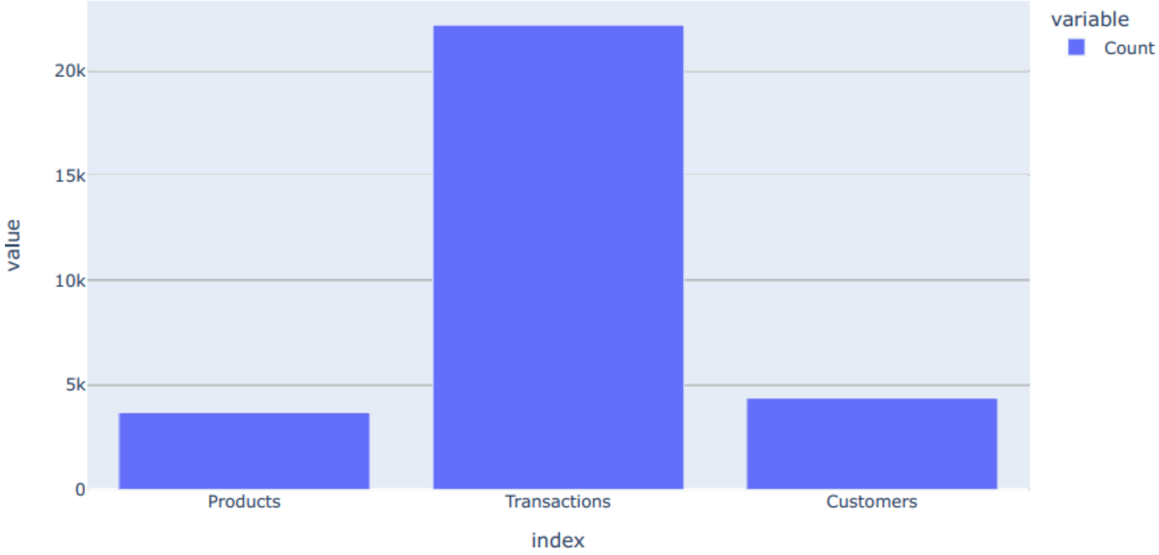
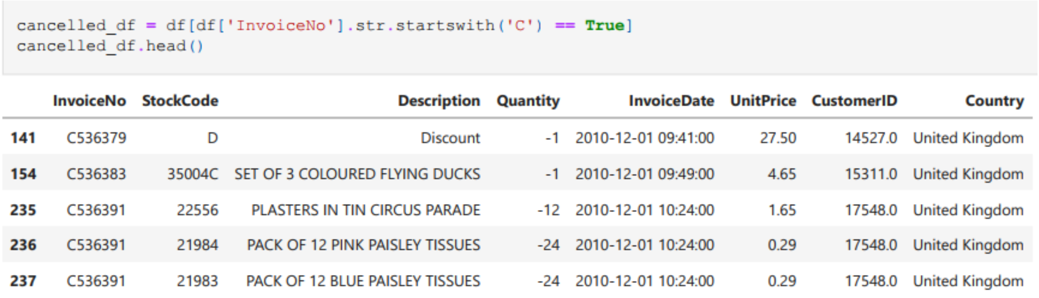


Fig 10. A plot of customers Products, transactions recorded

There are 4372 customers that purchased 4070 different products from the retail store. The total number of transactions carried out amount to 25900 orders.

ii. Order Cancellations



```
cancelled_df = df[df['InvoiceNo'].str.startswith('C') == True]
cancelled_df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1	2010-12-01 09:49:00	4.65	15311.0	United Kingdom
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12	2010-12-01 10:24:00	1.65	17548.0	United Kingdom
236	C536391	21984	PACK OF 12 PINK PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom
237	C536391	21983	PACK OF 12 BLUE PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom

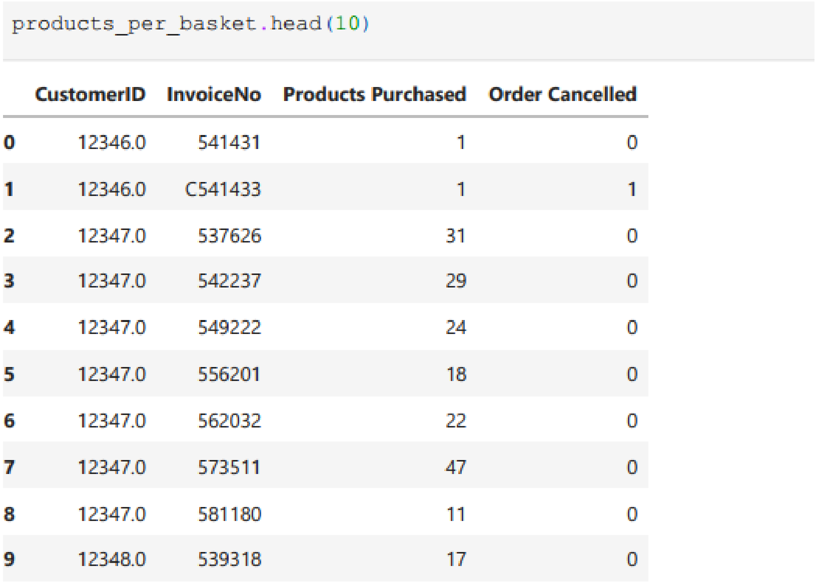
Table 8. Order cancellations

The number of cancelled orders is 3654, which is 16.47% of the entire dataset.

iii. 1.4.3 InvoiceNo + InvoiceDate (Order Cancellation)

- Further investigation on InvoiceNo
- Product Purchase Made Per Transaction
- Who cancelled an order

Here the instance count is increased to “10” to get a wider understand per view. Table 9 below shows canclled orders by customer ID , while fig. 11 show cancelled orders by product volumes. Invoice number cell 2 to 8 shows repeated customer ID, applying slicing to sort data, Table 10 provides that information on top five cancelled products



```
products_per_basket.head(10)
```

	CustomerID	InvoiceNo	Products Purchased	Order Cancelled
0	12346.0	541431	1	0
1	12346.0	C541433	1	1
2	12347.0	537626	31	0
3	12347.0	542237	29	0
4	12347.0	549222	24	0
5	12347.0	556201	18	0
6	12347.0	562032	22	0
7	12347.0	573511	47	0
8	12347.0	581180	11	0
9	12348.0	539318	17	0

Table 9. Cancelled Orders by CustomerID

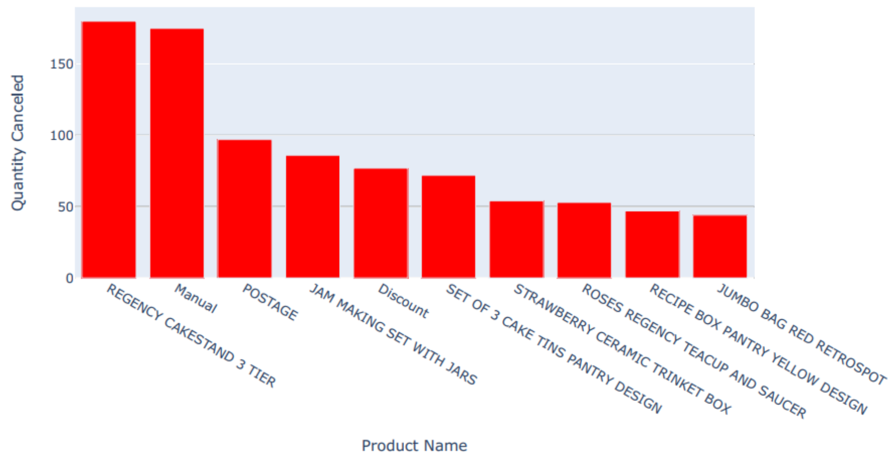
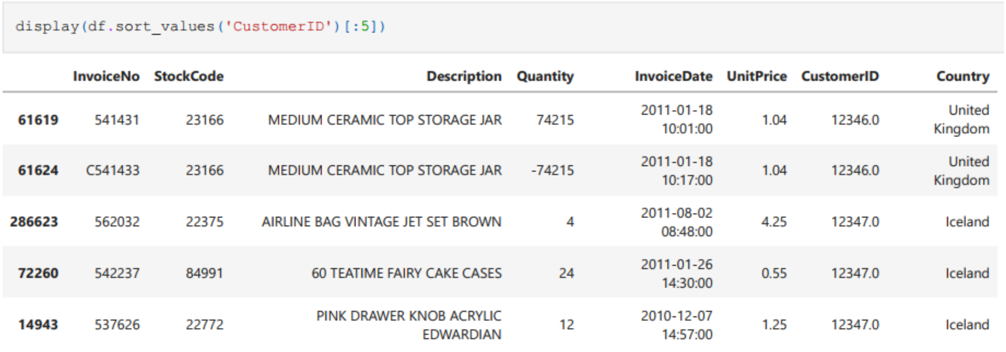


Fig 11. A comparison of top 10 cancelled products



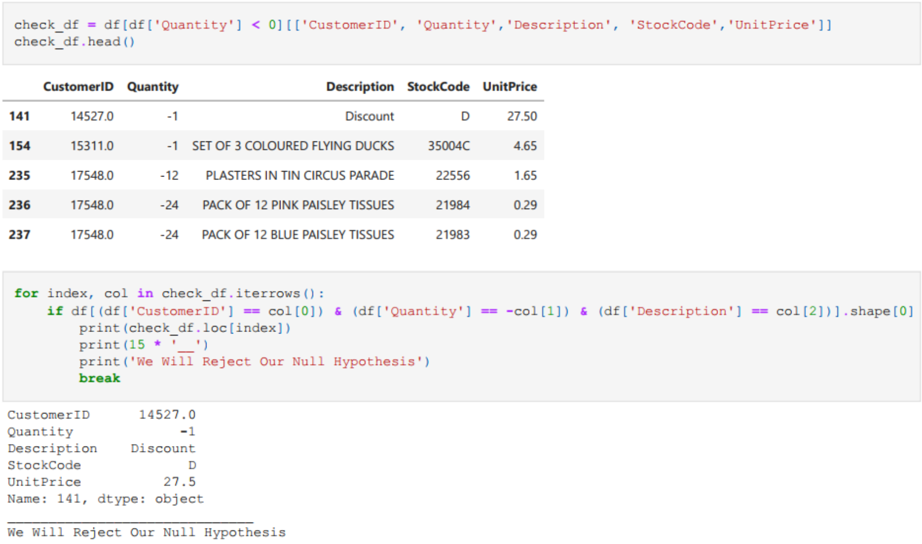
```
display(df.sort_values('CustomerID')[:5])
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
61619	541431	23166	MEDIUM CERAMIC TOP STORAGE JAR	74215	2011-01-18 10:01:00	1.04	12346.0	United Kingdom
61624	C541433	23166	MEDIUM CERAMIC TOP STORAGE JAR	-74215	2011-01-18 10:17:00	1.04	12346.0	United Kingdom
286623	562032	22375	AIRLINE BAG VINTAGE JET SET BROWN	4	2011-08-02 08:48:00	4.25	12347.0	Iceland
72260	542237	84991	60 TEATIME FAIRY CAKE CASES	24	2011-01-26 14:30:00	0.55	12347.0	Iceland
14943	537626	22772	PINK DRAWER KNOB ACRYLIC EDWARDIAN	12	2010-12-07 14:57:00	1.25	12347.0	Iceland

Table 10. Top 5 cancelled products

We have thus, encountered a Null Hypthesis – for each cancelled order in the database, there is another counterpart heirin.

It is important to check if this assertion above holds true for all entries. To do this, we decide to locate the entries that indicate a negative quantity and check if there is systematically an order indicating the same quantity (but positive), with the same value in its (CustomerID, Description and UnitPrice) columns:



```
check_df = df[df['Quantity'] < 0][['CustomerID', 'Quantity','Description', 'StockCode','UnitPrice']]
check_df.head()
```

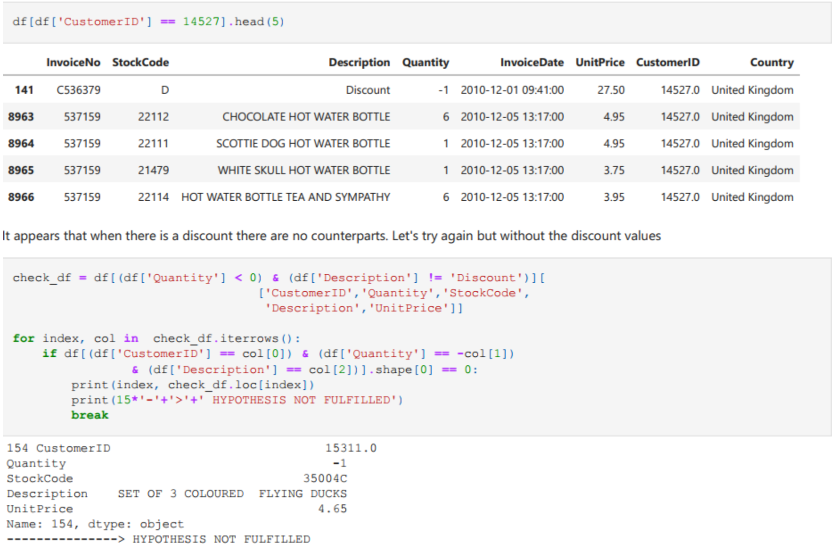
	CustomerID	Quantity	Description	StockCode	UnitPrice
141	14527.0	-1	Discount	D	27.50
154	15311.0	-1	SET OF 3 COLOURED FLYING DUCKS	35004C	4.65
235	17548.0	-12	PLASTERS IN TIN CIRCUS PARADE	22556	1.65
236	17548.0	-24	PACK OF 12 PINK PAISLEY TISSUES	21984	0.29
237	17548.0	-24	PACK OF 12 BLUE PAISLEY TISSUES	21983	0.29

```
for index, col in check_df.iterrows():
    if df[(df['CustomerID'] == col[0]) & (df['Quantity'] == -col[1]) & (df['Description'] == col[2])].shape[0]
        print(check_df.loc[index])
        print(15 * '__')
        print('We Will Reject Our Null Hypothesis')
        break
```

```
CustomerID      14527.0
Quantity             -1
Description    Discount
StockCode             D
UnitPrice          27.5
Name: 141, dtype: object
______________________________
We Will Reject Our Null Hypothesis
```

Fig 12. Null Hypothesis verification A.

It may appear that our hypothesis is not verified, we can conclude that cancellations do not necessarily correspond to orders that would have been made beforehand. Probing further;



```
df[df['CustomerID'] == 14527].head(5)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom
8963	537159	22112	CHOCOLATE HOT WATER BOTTLE	6	2010-12-05 13:17:00	4.95	14527.0	United Kingdom
8964	537159	22111	SCOTTIE DOG HOT WATER BOTTLE	1	2010-12-05 13:17:00	4.95	14527.0	United Kingdom
8965	537159	21479	WHITE SKULL HOT WATER BOTTLE	1	2010-12-05 13:17:00	3.75	14527.0	United Kingdom
8966	537159	22114	HOT WATER BOTTLE TEA AND SYMPATHY	6	2010-12-05 13:17:00	3.95	14527.0	United Kingdom

It appears that when there is a discount there are no counterparts. Let's try again but without the discount values

```
check_df = df[(df['Quantity'] < 0) & (df['Description'] != 'Discount')][
                                    ['CustomerID','Quantity','StockCode',
                                     'Description','UnitPrice']]

for index, col in  check_df.iterrows():
    if df[(df['CustomerID'] == col[0]) & (df['Quantity'] == -col[1])
              & (df['Description'] == col[2])].shape[0] == 0:
        print(index, check_df.loc[index])
        print(15*'-'+'>'+' HYPOTHESIS NOT FULFILLED')
        break
```

```
154 CustomerID                               15311.0
Quantity                                          -1
StockCode                                     35004C
Description    SET OF 3 COLOURED  FLYING DUCKS
UnitPrice                                       4.65
Name: 154, dtype: object
---------------> HYPOTHESIS NOT FULFILLED
```

Fig 13. Null Hypothesis verification B.

Again, our hypothesis is not fulfilled. Hence, cancellations do not necessarily correspond to orders that would have been made beforehand.

i. Top Ordered Products (Not returned)

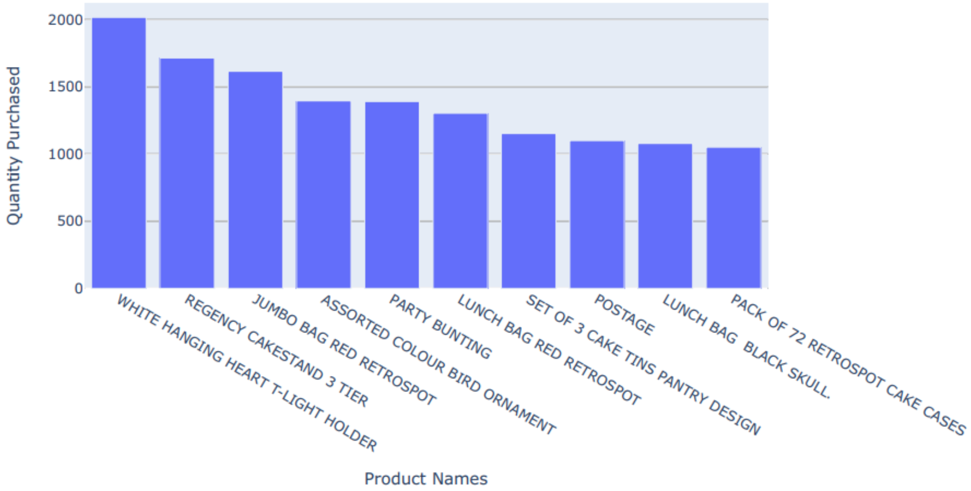
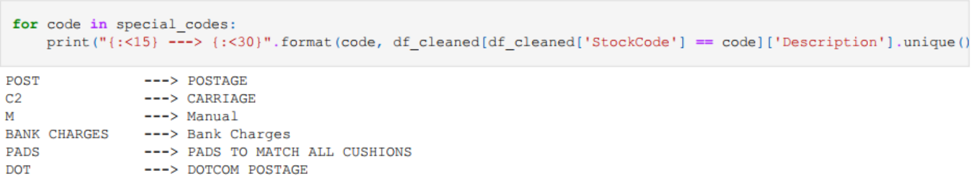


Fig 14. Top ordered products (not returned)

4.1.5 Stockcode

Above, it has been seen that some values of the StockCode variable indicate a particular transaction (i.e., D for Discount). Let us investigate the contents of this variable by looking for the set of codes that would contain only letters:



```
for code in special_codes:
    print("{:<15} ---> {:<30}".format(code, df_cleaned[df_cleaned['StockCode'] == code]['Description'].unique()
```

```
POST            ---> POSTAGE
C2              ---> CARRIAGE
M               ---> Manual
BANK CHARGES    ---> Bank Charges
PADS            ---> PADS TO MATCH ALL CUSHIONS
DOT             ---> DOTCOM POSTAGE
```

Fig 15. Exploring store coded that contain only letters

These result shows specific operations, which do not necessarily characterize our customers, hence they will be dropped from the transactions list. This was achieved by implementing conditional selection to drop all the values that contain; Post, C2, M, Bank Charges, PADS and DOT.

4.2 Feature Engineering

4.2.1 Purchase Date

Our date data type still reflects as “O” which could be because Pandas software is recognising it as a string – but it is a legit date when each customer was issued an invoice which can be very useful to the performance of our model. Hence, the need to convert it to date time object so that Pandas can recognise it as a date - time. A new column was also created in our data frame and names “Purchasedate”. One specific extract was the day of purchase and it was plotted against value.

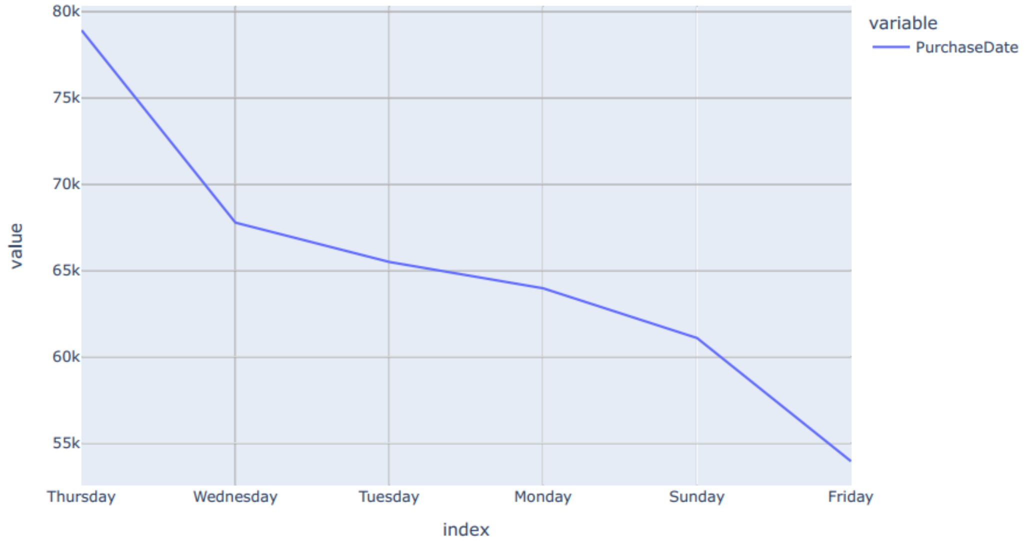
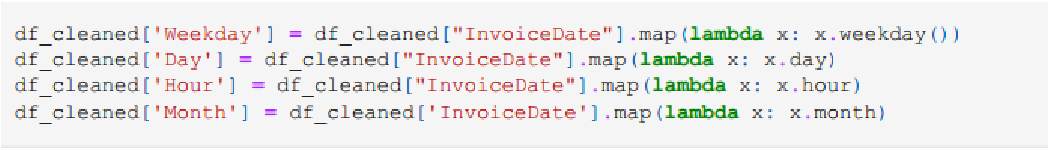


Fig 16. Plotting Purchasedate against value of purchase

Based on the above plot, customers made most of the purchase on Thursdays and the least on Fridays.

4.2.2 Week, Day, Month

Applying the anonymous function, we are able to extract the week, the day and month of purchase from the InvoiceDate as shown below

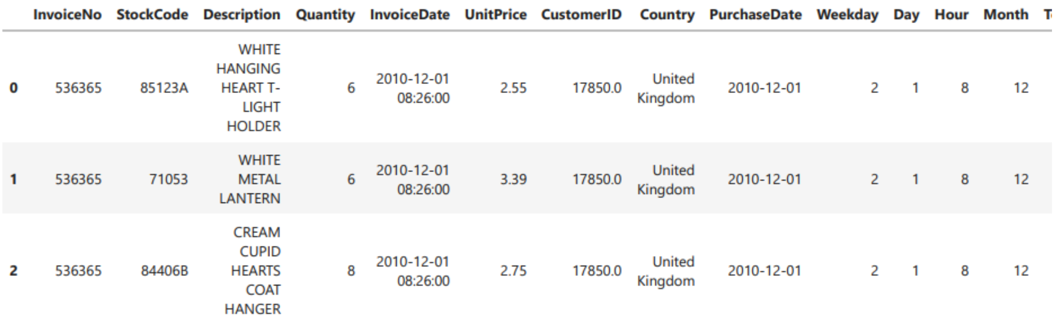


```
df_cleaned['Weekday'] = df_cleaned["InvoiceDate"].map(lambda x: x.weekday())
df_cleaned['Day'] = df_cleaned["InvoiceDate"].map(lambda x: x.day)
df_cleaned['Hour'] = df_cleaned["InvoiceDate"].map(lambda x: x.hour)
df_cleaned['Month'] = df_cleaned['InvoiceDate'].map(lambda x: x.month)
```

Fig 17. Extracting; weekday, day, month and time of purchase

4.2.3 Total Price

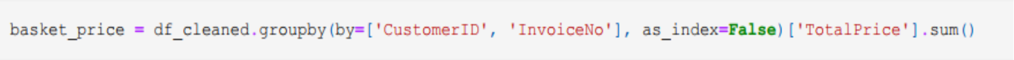
Applying the same function, we can generate total price per product



	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	PurchaseDate	Weekday	Day	Hour	Month	T
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	2010-12-01	2	1	8	12	
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	2010-12-01	2	1	8	12	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	2010-12-01	2	1	8	12	

Table 11. Total Price per product

From table 11, we can see that each entry in the data frame indicates prizes for a single kind of product. Hence, orders made by customers are split on several lines. Let's collect all the purchases made during a single order by each customer to recover the basket



```
basket_price = df_cleaned.groupby(by=['CustomerID', 'InvoiceNo'], as_index=False)['TotalPrice'].sum()
```

Fig 18. Generating total purchase per single order

With the newly engineered Total Price feature, we can also visualize, Revenue per Country (fig. 19) & Invoice Per Country as seen in fig. 20

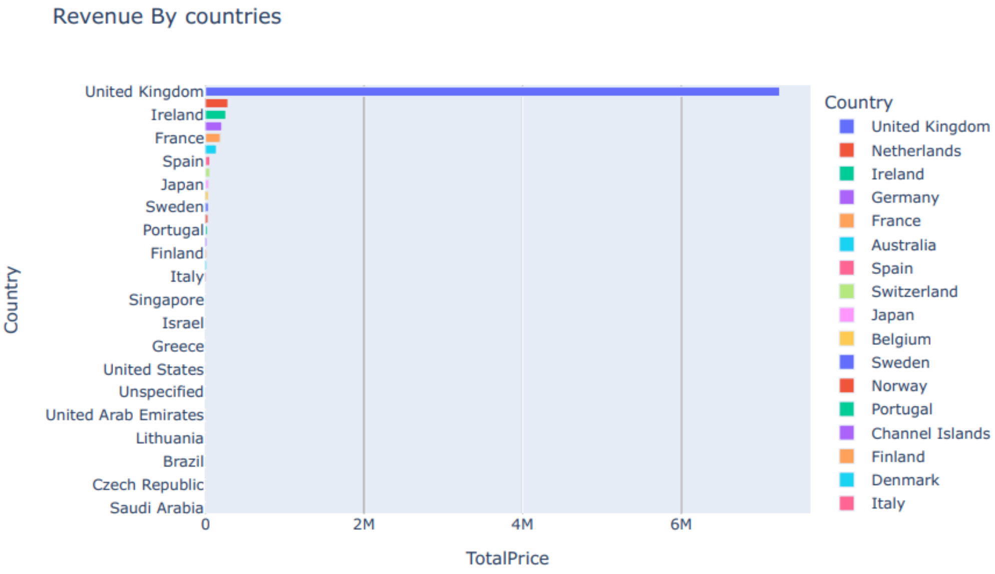


Fig. 19 A plot of total revenue per country

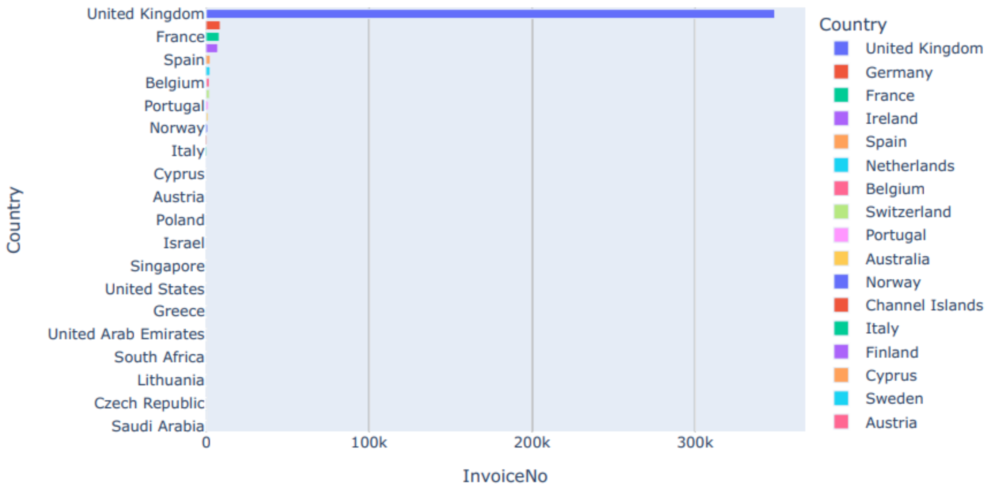
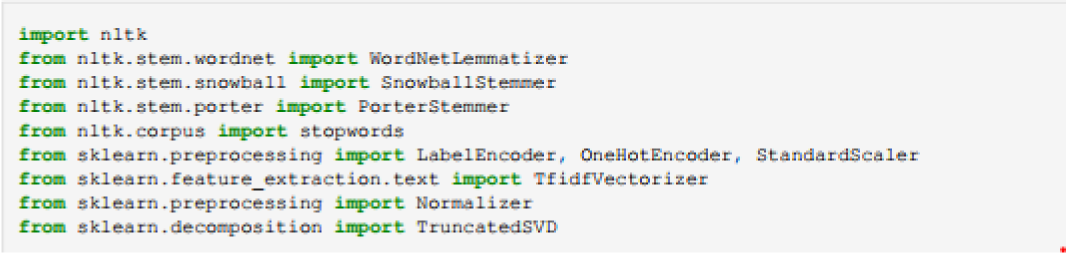


Fig. 20 A plot of total invoice per country

.

Interestingly, Netherlands is the second country by revenue but the fifth in terms of number of invoices generated.

4.3 Data Pre-processing

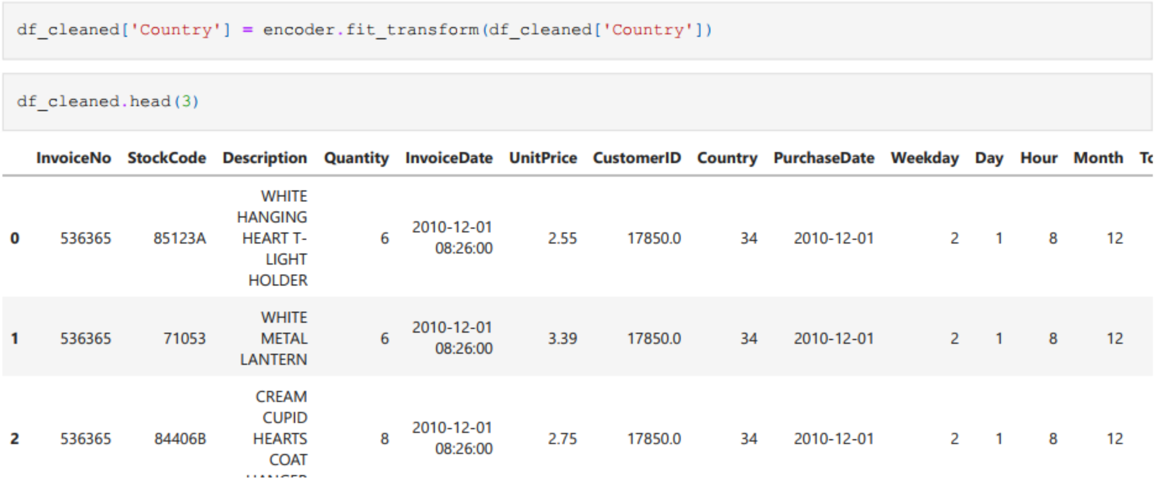


```
import nltk
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import Normalizer
from sklearn.decomposition import TruncatedSVD
```

Fig. 21 Imported libraries form data pre-processing

Label Encoding

We proceed to encode the country feature, assigning each country an integer.



```
df_cleaned['Country'] = encoder.fit_transform(df_cleaned['Country'])
```

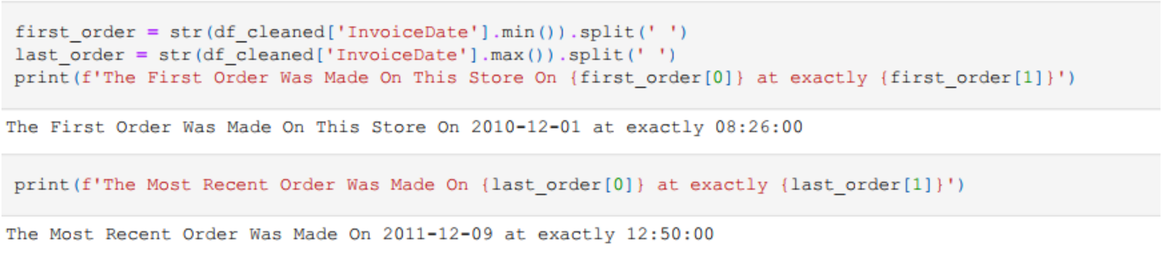
```
df_cleaned.head(3)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	PurchaseDate	Weekday	Day	Hour	Month	Tc
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	34	2010-12-01	2	1	8	12	
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	34	2010-12-01	2	1	8	12	
2	536365	84406B	CREAM CUPID HEARTS COAT	8	2010-12-01 08:26:00	2.75	17850.0	34	2010-12-01	2	1	8	12	

Table 12. Country names converted to numbers (codes)

4.4 RFM Segmentation

As discussed in 2.3.4, this process will help us classify our customers taking into account: **Recency -** how many days since the customer's last purchase, **Frequency -** how often the client makes a purchase, and is the total number of yearly purchases made by a customer, and **Monetary Value** is the total money spent by a customer.



```
first_order = str(df_cleaned['InvoiceDate'].min()).split(' ')
last_order = str(df_cleaned['InvoiceDate'].max()).split(' ')
print(f'The First Order Was Made On This Store On {first_order[0]} at exactly {first_order[1]}')
```

```
The First Order Was Made On This Store On 2010-12-01 at exactly 08:26:00
```

```
print(f'The Most Recent Order Was Made On {last_order[0]} at exactly {last_order[1]}')
```

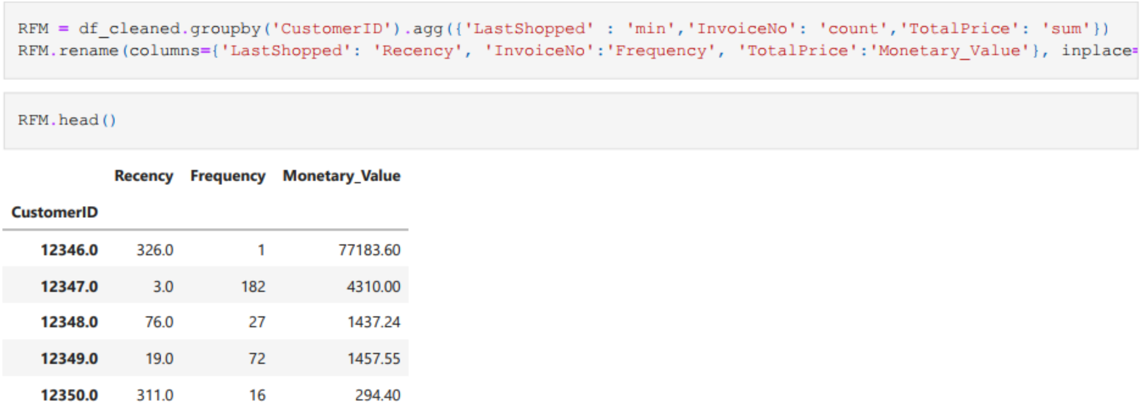
```
The Most Recent Order Was Made On 2011-12-09 at exactly 12:50:00
```

Fig 22. Generating dates for first and last orders placed

Since the last order was placed on 9th December 2011, instead of using today's date to compute recency, 10th December 2011 is used to calculate recency; that is, a day after the last invoice date. entry in the database was recorded instead.

Calculating Frequency and Monetary Dimensions

How often does a customer patronize the store, and what's the monetary value of their purchase?



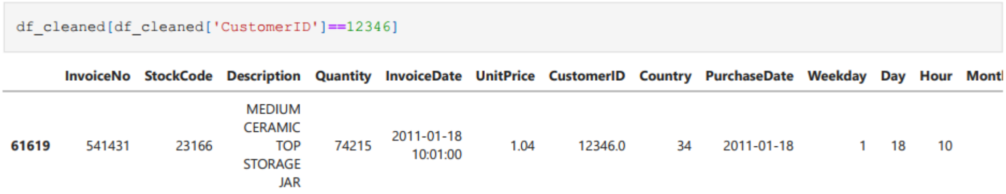
```
RFM = df_cleaned.groupby('CustomerID').agg({'LastShopped' : 'min','InvoiceNo': 'count','TotalPrice': 'sum'})
RFM.rename(columns={'LastShopped': 'Recency', 'InvoiceNo':'Frequency', 'TotalPrice':'Monetary_Value'}, inplace=
```

```
RFM.head()
```

	Recency	Frequency	Monetary_Value
CustomerID			
12346.0	326.0	1	77183.60
12347.0	3.0	182	4310.00
12348.0	76.0	27	1437.24
12349.0	19.0	72	1457.55
12350.0	311.0	16	294.40

Fig 23. Generating the Frequency and Monetary value

A brief detail of the first customer 12346.0



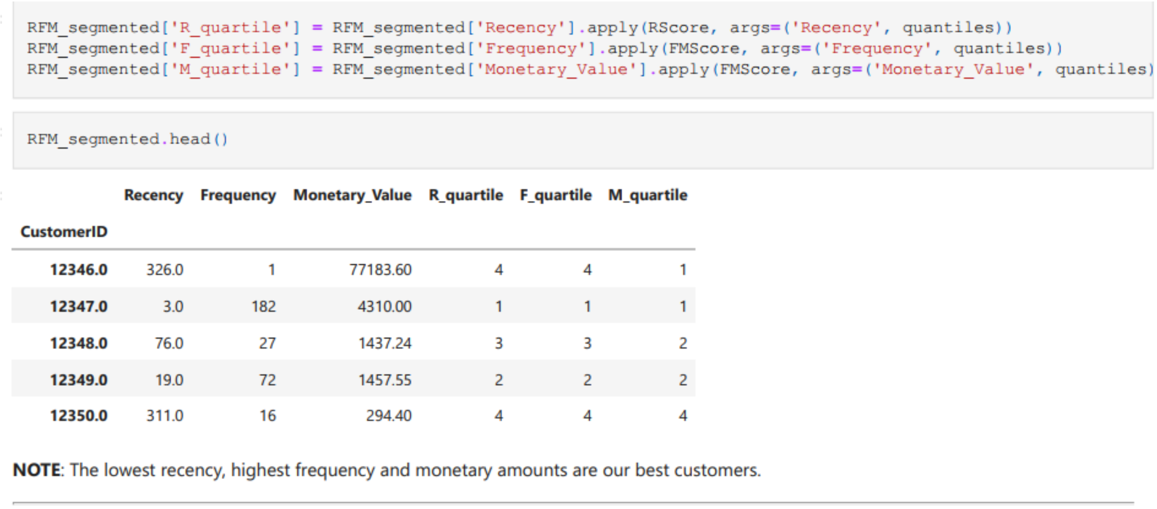
```
df_cleaned[df_cleaned['CustomerID']==12346]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	PurchaseDate	Weekday	Day	Hour	Montl
61619	541431	23166	MEDIUM CERAMIC TOP STORAGE JAR	74215	2011-01-18 10:01:00	1.04	12346.0	34	2011-01-18	1	18	10	

Fig 24. First customer as listed on the RFM table

Interestingly, customer 12346.0 has shopped only once. This customer bought one product at a huge quantity (74,215). Although the unit price is very low though; perhaps the customer purchased those products during a promo or clearance sale

Adding Quantiles



```
RFM_segmented['R_quartile'] = RFM_segmented['Recency'].apply(RScore, args=('Recency', quantiles))
RFM_segmented['F_quartile'] = RFM_segmented['Frequency'].apply(FMScore, args=('Frequency', quantiles))
RFM_segmented['M_quartile'] = RFM_segmented['Monetary_Value'].apply(FMScore, args=('Monetary_Value', quantiles)
```

```
RFM_segmented.head()
```

	Recency	Frequency	Monetary_Value	R_quartile	F_quartile	M_quartile
CustomerID						
12346.0	326.0	1	77183.60	4	4	1
12347.0	3.0	182	4310.00	1	1	1
12348.0	76.0	27	1437.24	3	3	2
12349.0	19.0	72	1457.55	2	2	2
12350.0	311.0	16	294.40	4	4	4

NOTE: The lowest recency, highest frequency and monetary amounts are our best customers.

Fig 25. Generating RFM quantiles – and resulting illustration

We then assign scores to all customers the dataset returned 4335 rows × 7 columns

KNOWING EACH CUSTOMER'S SEGMENT

Now that we've performed customer segmentation using RFM, fig 26 and 27 can readily answer these questions for the retail store:

Who are our best customers?

Which customers are at the verge of churning?

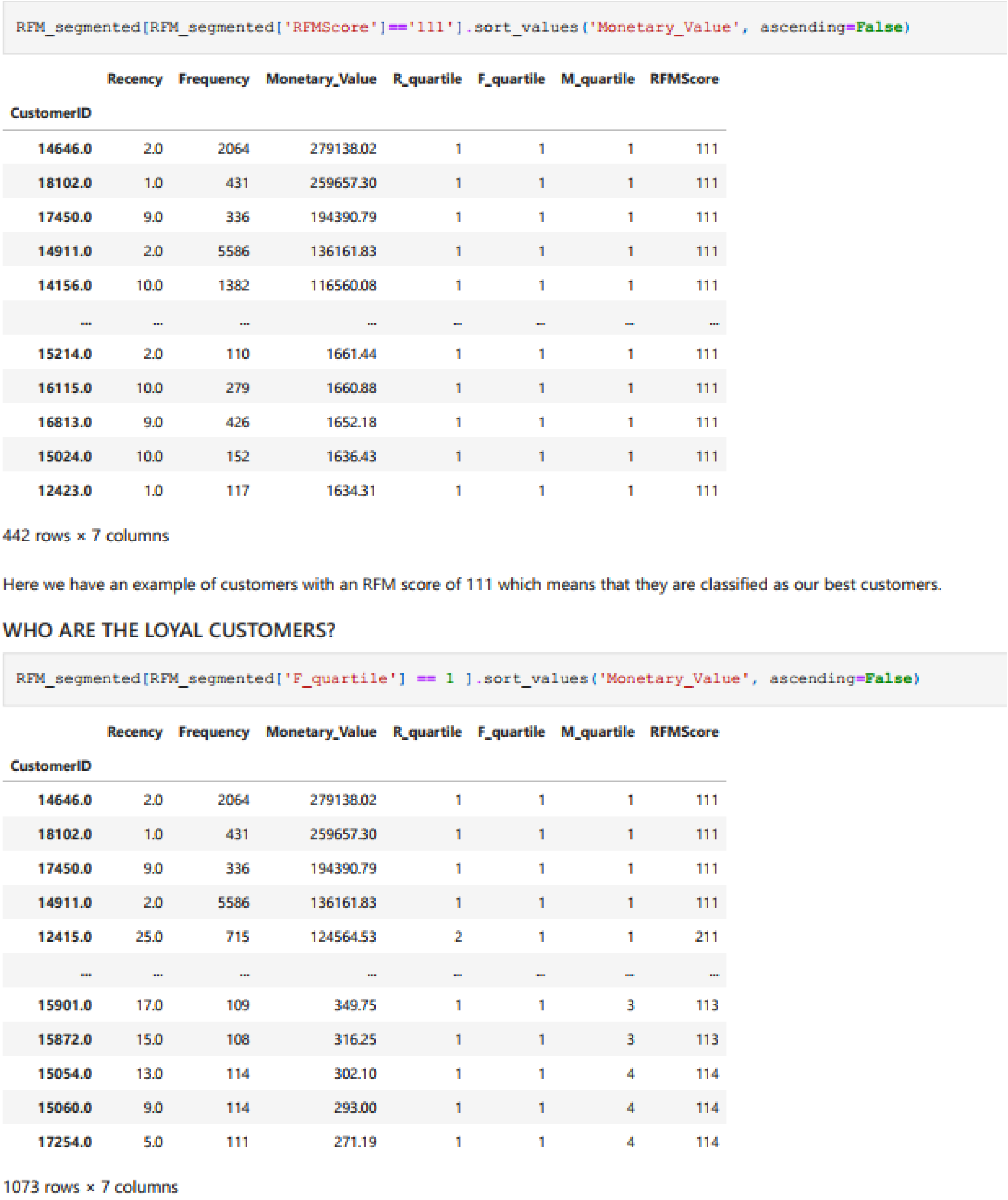
Who are lost customers that we don’t need to pay much attention to?

Who are our loyal customers?

Which customers must we retain?

Who has the potential to be converted into more profitable customers?

Which group of customers is most likely to respond to our current campaign? To all the customers and determine who are our best customers and the most frequent.



```
RFM_segmented[RFM_segmented['RFMScore']=='111'].sort_values('Monetary_Value', ascending=False)
```

CustomerID	Recency	Frequency	Monetary_Value	R_quartile	F_quartile	M_quartile	RFMScore
14646.0	2.0	2064	279138.02	1	1	1	111
18102.0	1.0	431	259657.30	1	1	1	111
17450.0	9.0	336	194390.79	1	1	1	111
14911.0	2.0	5586	136161.83	1	1	1	111
14156.0	10.0	1382	116560.08	1	1	1	111
...	...	...	...	...	...	...	...
15214.0	2.0	110	1661.44	1	1	1	111
16115.0	10.0	279	1660.88	1	1	1	111
16813.0	9.0	426	1652.18	1	1	1	111
15024.0	10.0	152	1636.43	1	1	1	111
12423.0	1.0	117	1634.31	1	1	1	111

442 rows × 7 columns

Here we have an example of customers with an RFM score of 111 which means that they are classified as our best customers.

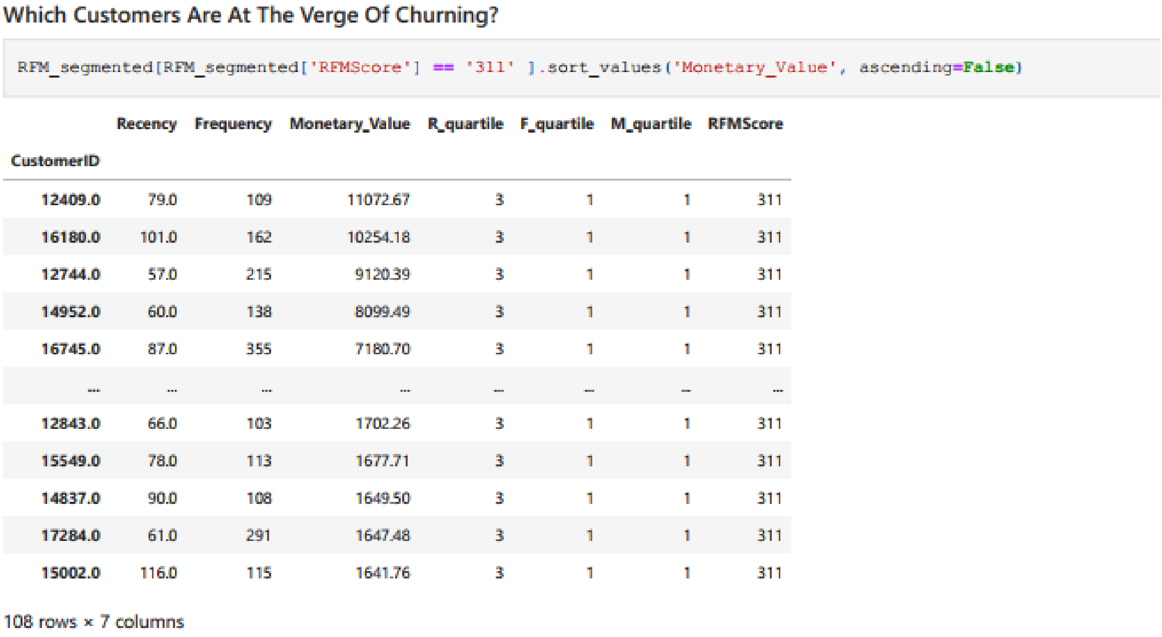
WHO ARE THE LOYAL CUSTOMERS?

```
RFM_segmented[RFM_segmented['F_quartile'] == 1 ].sort_values('Monetary_Value', ascending=False)
```

CustomerID	Recency	Frequency	Monetary_Value	R_quartile	F_quartile	M_quartile	RFMScore
14646.0	2.0	2064	279138.02	1	1	1	111
18102.0	1.0	431	259657.30	1	1	1	111
17450.0	9.0	336	194390.79	1	1	1	111
14911.0	2.0	5586	136161.83	1	1	1	111
12415.0	25.0	715	124564.53	2	1	1	211
...	...	...	...	...	...	...	...
15901.0	17.0	109	349.75	1	1	3	113
15872.0	15.0	108	316.25	1	1	3	113
15054.0	13.0	114	302.10	1	1	4	114
15060.0	9.0	114	293.00	1	1	4	114
17254.0	5.0	111	271.19	1	1	4	114

1073 rows × 7 columns

Fig 26. Sample results from customer group analysis


Which Customers Are At The Verge Of Churning?

```
RFM_segmented[RFM_segmented['RFMScore'] == '311' ].sort_values('Monetary_Value', ascending=False)
```

	Recency	Frequency	Monetary_Value	R_quartile	F_quartile	M_quartile	RFMScore
CustomerID							
12409.0	79.0	109	11072.67	3	1	1	311
16180.0	101.0	162	10254.18	3	1	1	311
12744.0	57.0	215	9120.39	3	1	1	311
14952.0	60.0	138	8099.49	3	1	1	311
16745.0	87.0	355	7180.70	3	1	1	311
...	...	...	...	...	...	...	...
12843.0	66.0	103	1702.26	3	1	1	311
15549.0	78.0	113	1677.71	3	1	1	311
14837.0	90.0	108	1649.50	3	1	1	311
17284.0	61.0	291	1647.48	3	1	1	311
15002.0	116.0	115	1641.76	3	1	1	311

108 rows × 7 columns

Fig 27. Result of customers likely to churn

4.5 Product Description Categories

We are focusing now on cleaning the description column to ensure there no repeated products using stemming library, then remove overly repeated words (0 to 0.30) using “stopword”

Fitting and transformation resulted in 3871 rows and 1694 columns. These columns have seemingly increased much and may lead to a problem of multicollinearity. Hence the need to decompose/truncate the variables which reduced it to; 3871 rows and 100 columns.

Fig xx contains imported libraries that will aid in our product categorization which include, clustering algorithms, visualizations libraries and randomization. We are using silhouette score to determine the number of categories to place our products

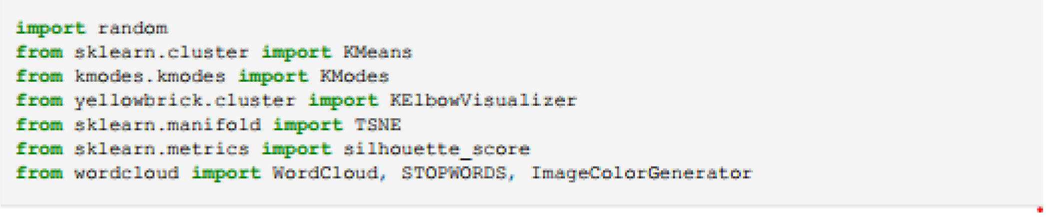
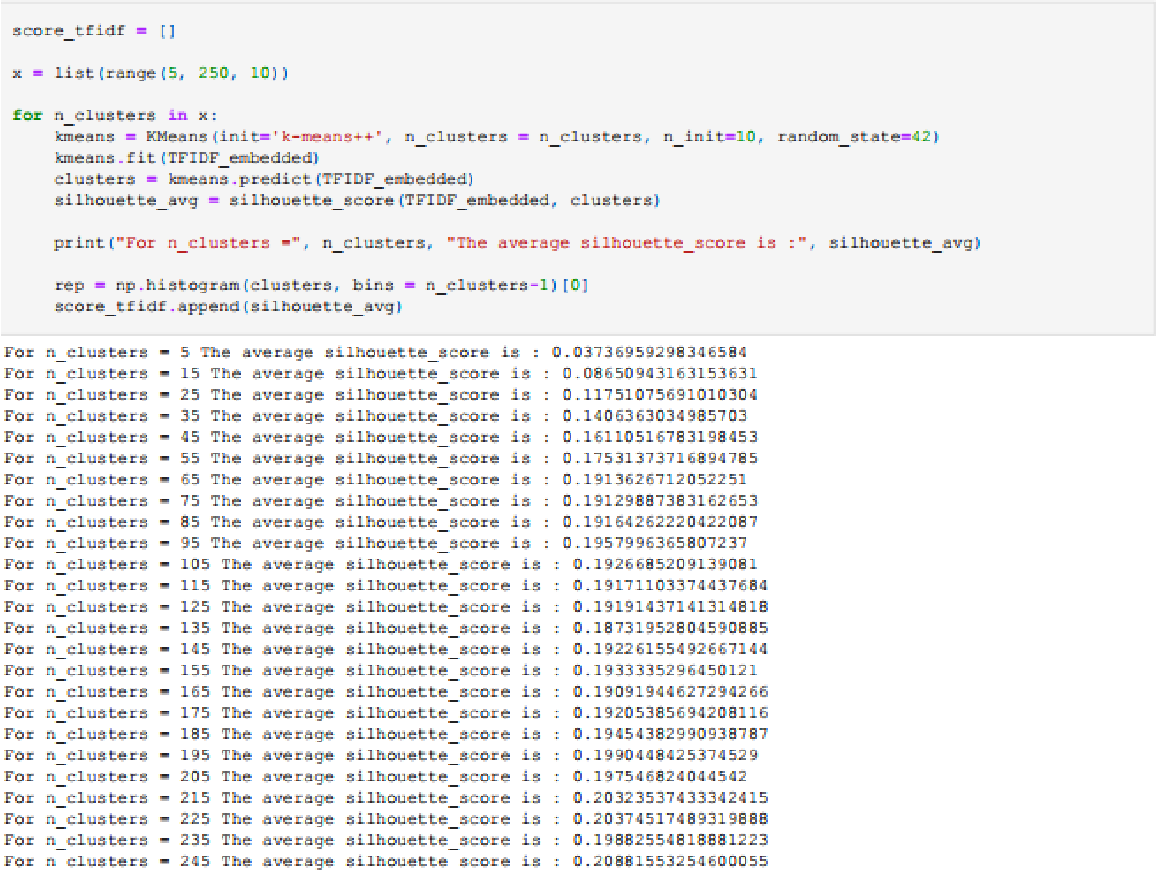

```
import random
from sklearn.cluster import KMeans
from kmodes.kmodes import KModes
from yellowbrick.cluster import KElbowVisualizer
from sklearn.manifold import TSNE
from sklearn.metrics import silhouette_score
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

Fig 28. Imported libraries to support product categorization

In order to achieve a silhouette score for our distribution, which allow us to know the optimum number of clusters to choose.

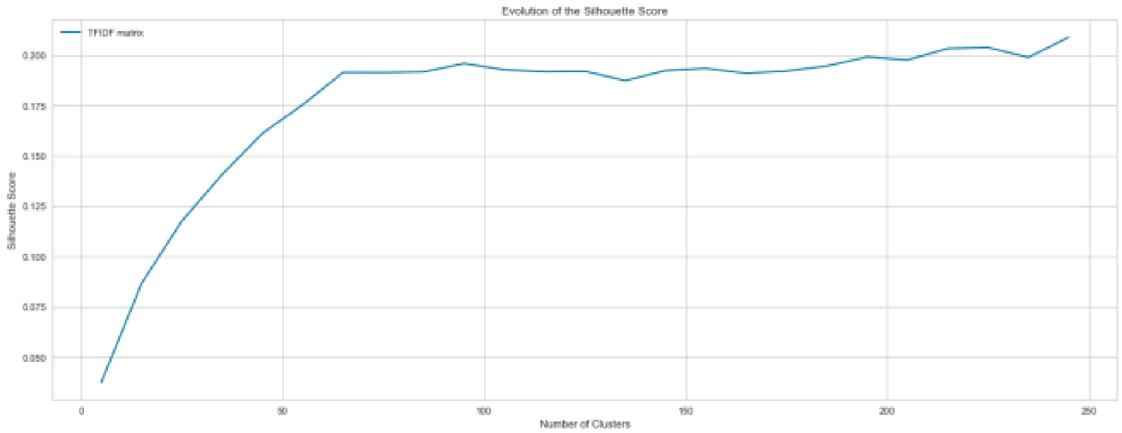


```
score_tfidf = []

x = list(range(5, 250, 10))

for n_clusters in x:
    kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=10, random_state=42)
    kmeans.fit(TFIDF_embedded)
    clusters = kmeans.predict(TFIDF_embedded)
    silhouette_avg = silhouette_score(TFIDF_embedded, clusters)

    print("For n_clusters =", n_clusters, "The average silhouette_score is :", silhouette_avg)

    rep = np.histogram(clusters, bins = n_clusters-1)[0]
    score_tfidf.append(silhouette_avg)
```

```
For n_clusters = 5 The average silhouette_score is : 0.03736959298346584
For n_clusters = 15 The average silhouette_score is : 0.08650943163153631
For n_clusters = 25 The average silhouette_score is : 0.11751075691010304
For n_clusters = 35 The average silhouette_score is : 0.1406363034985703
For n_clusters = 45 The average silhouette_score is : 0.16110516783198453
For n_clusters = 55 The average silhouette_score is : 0.17531373716894785
For n_clusters = 65 The average silhouette_score is : 0.1913626712052251
For n_clusters = 75 The average silhouette_score is : 0.19129887383162653
For n_clusters = 85 The average silhouette_score is : 0.19164262220422087
For n_clusters = 95 The average silhouette_score is : 0.1957996365807237
For n_clusters = 105 The average silhouette_score is : 0.1926685209139081
For n_clusters = 115 The average silhouette_score is : 0.19171103374437684
For n_clusters = 125 The average silhouette_score is : 0.19191437141314818
For n_clusters = 135 The average silhouette_score is : 0.18731952804590885
For n_clusters = 145 The average silhouette_score is : 0.19226155492667144
For n_clusters = 155 The average silhouette_score is : 0.1933335296450121
For n_clusters = 165 The average silhouette_score is : 0.19091944627294266
For n_clusters = 175 The average silhouette_score is : 0.19205385694208116
For n_clusters = 185 The average silhouette_score is : 0.19454382990938787
For n_clusters = 195 The average silhouette_score is : 0.1990448425374529
For n_clusters = 205 The average silhouette_score is : 0.197546824044542
For n_clusters = 215 The average silhouette_score is : 0.20323537433342415
For n_clusters = 225 The average silhouette_score is : 0.20374517489319888
For n_clusters = 235 The average silhouette_score is : 0.19882554818881223
For n clusters = 245 The average silhouette score is : 0.20881553254600055
```

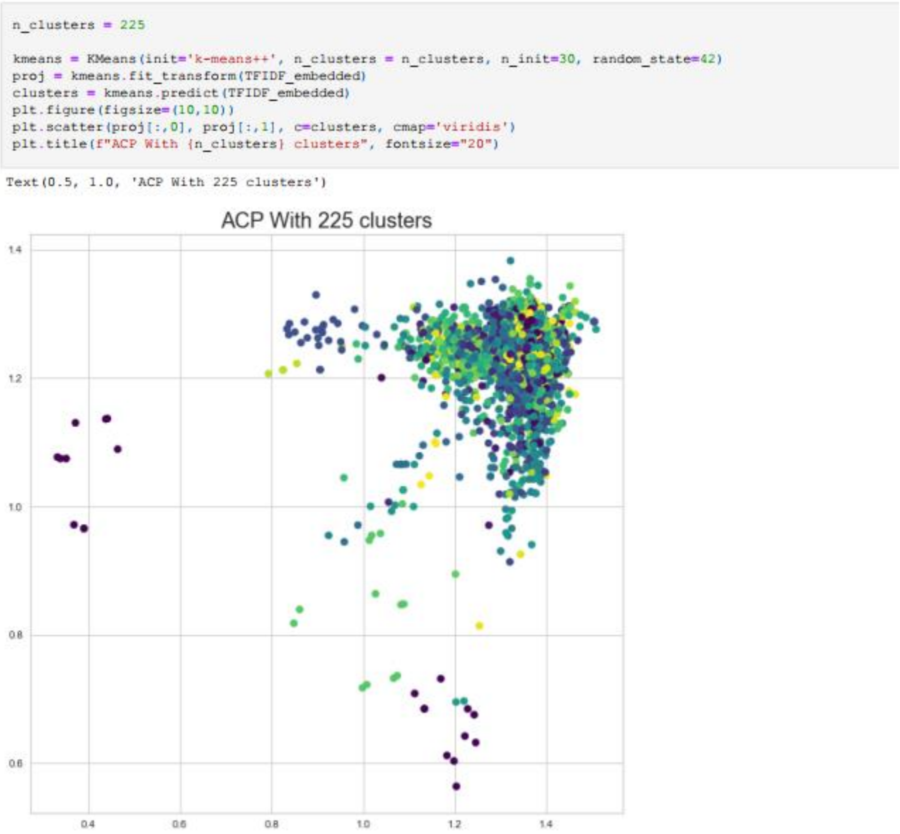
Fig 29. Generating and result of n_clusters

The silhouette curve



The highest value for the silhouette score is when there are 225 clusters. So we'll chose this value.

Fig 30. A plot of the Silhouette



```
n_clusters = 225

kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=30, random_state=42)
proj = kmeans.fit_transform(TFIDF_embedded)
clusters = kmeans.predict(TFIDF_embedded)
plt.figure(figsize=(10,10))
plt.scatter(proj[:,0], proj[:,1], c=clusters, cmap='viridis')
plt.title(f"ACP With {n_clusters} clusters", fontsize="20")
```

```
Text(0.5, 1.0, 'ACP With 225 clusters')
```

Fig 31. Generating the clusters – and ACP with 225 clusters

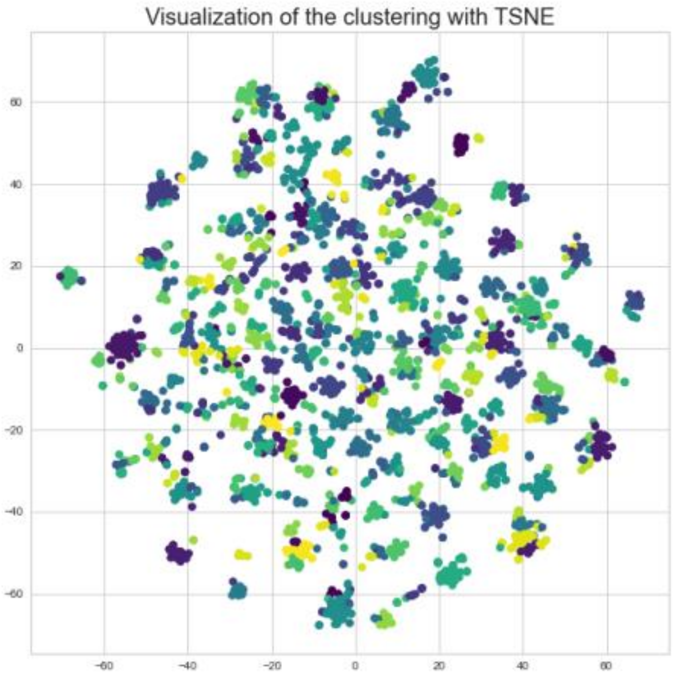
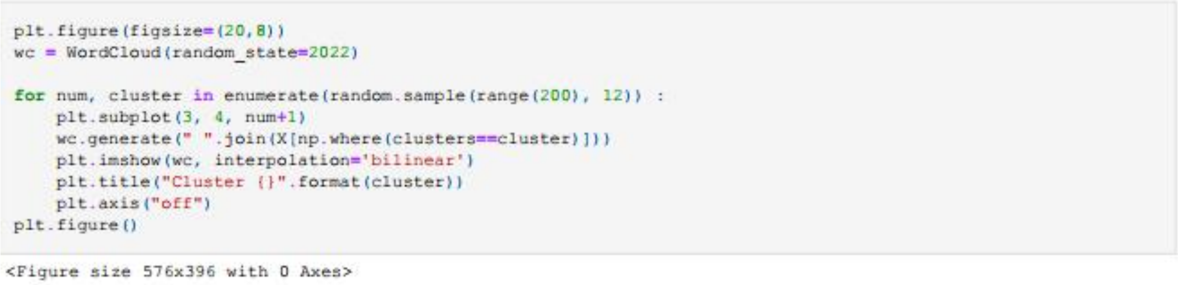


Fig 32. Cluster output represented with TSNE

Randomizing



```
plt.figure(figsize=(20,8))
wc = WordCloud(random_state=2022)

for num, cluster in enumerate(random.sample(range(200), 12)) :
    plt.subplot(3, 4, num+1)
    wc.generate(" ".join(X[np.where(clusters==cluster)]))
    plt.imshow(wc, interpolation='bilinear')
    plt.title("Cluster {}".format(cluster))
    plt.axis("off")
plt.figure()
```

```
<Figure size 576x396 with 0 Axes>
```

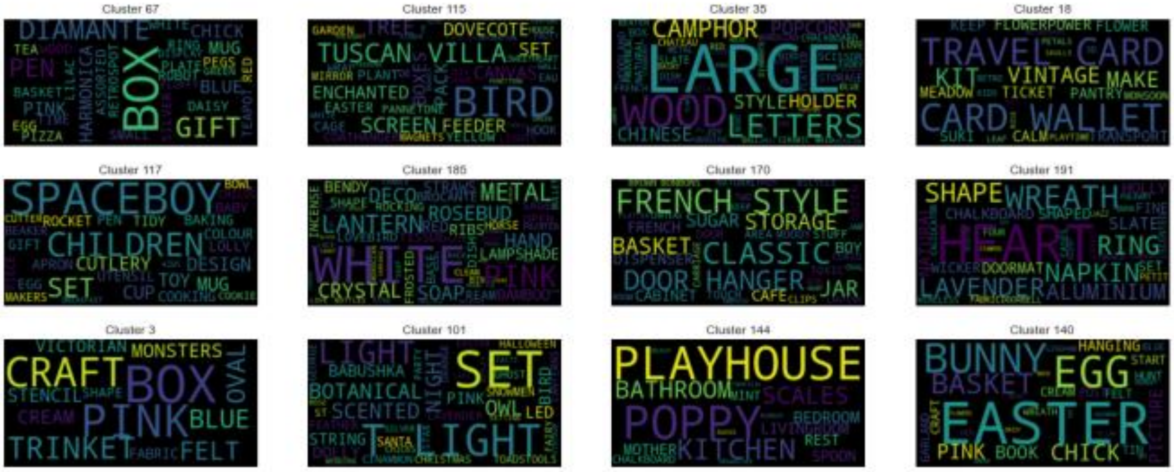
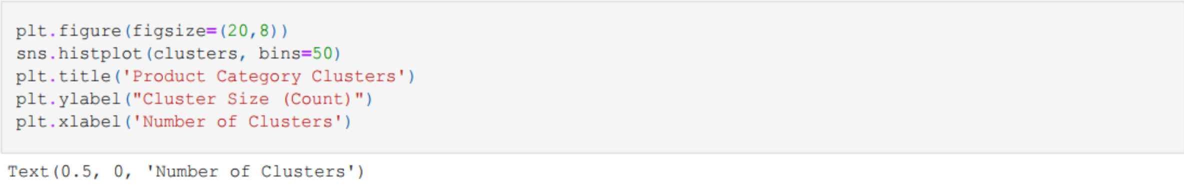


Fig 33. Randomized output presenting with WordCloud

From this representation, we can see that for example, one of the clusters contains objects that could be associated with gifts (keywords: Christmas, decoration, tree, ...). Another cluster would rather contain love and romance items (keywords: Candle, Scented Votive, Dinner, Lavender...). Nevertheless, it can also be observed that many words appear in various clusters and it is therefore difficult to clearly distinguish them.

Note: The WordCloud displayed above will randomly change for each run.

Plotting product category clusters



```
plt.figure(figsize=(20,8))
sns.histplot(clusters, bins=50)
plt.title('Product Category Clusters')
plt.ylabel("Cluster Size (Count)")
plt.xlabel('Number of Clusters')
```

```
Text(0.5, 0, 'Number of Clusters')
```

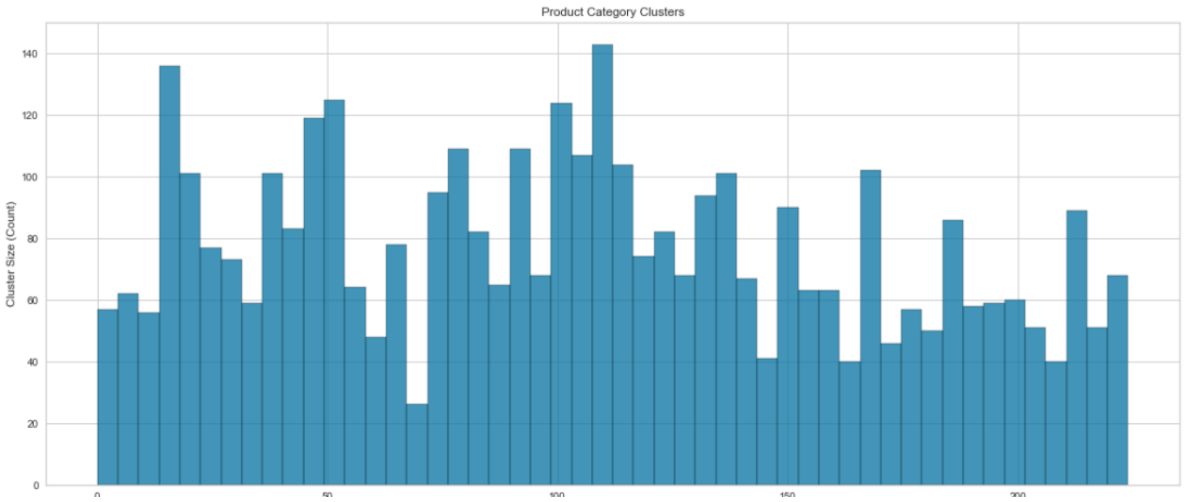
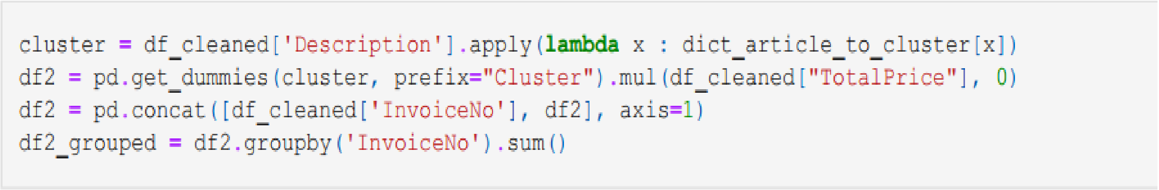


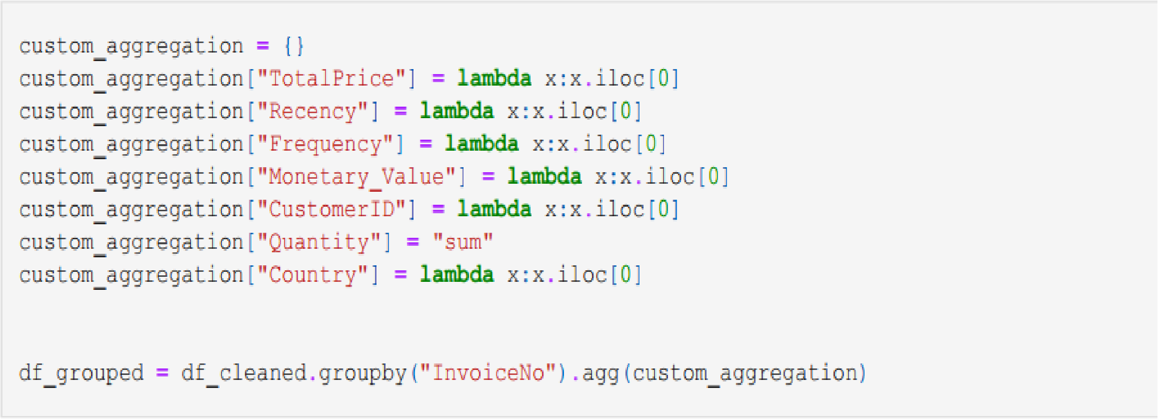
Fig 34. A plot Product Category Clusters

4.6 Creating Customer Categories

Dictionary comprehension is then performed



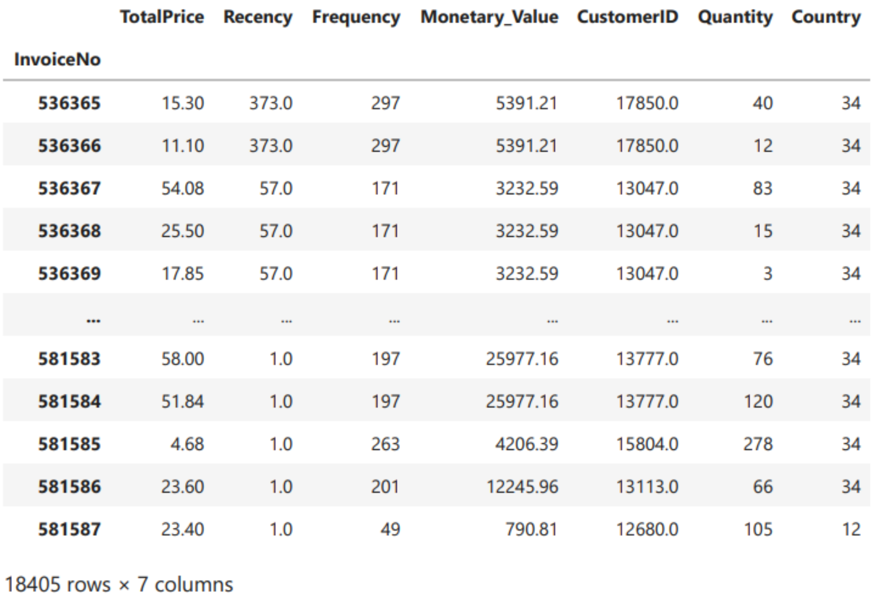
```
cluster = df_cleaned['Description'].apply(lambda x : dict_article_to_cluster[x])
df2 = pd.get_dummies(cluster, prefix="Cluster").mul(df_cleaned["TotalPrice"], 0)
df2 = pd.concat([df_cleaned['InvoiceNo'], df2], axis=1)
df2_grouped = df2.groupby('InvoiceNo').sum()
```



```
custom_aggregation = {}
custom_aggregation["TotalPrice"] = lambda x:x.iloc[0]
custom_aggregation["Recency"] = lambda x:x.iloc[0]
custom_aggregation["Frequency"] = lambda x:x.iloc[0]
custom_aggregation["Monetary_Value"] = lambda x:x.iloc[0]
custom_aggregation["CustomerID"] = lambda x:x.iloc[0]
custom_aggregation["Quantity"] = "sum"
custom_aggregation["Country"] = lambda x:x.iloc[0]


df_grouped = df_cleaned.groupby("InvoiceNo").agg(custom_aggregation)
```

Fig 35. Dictionary comprehension of data frame

4.6.1 Grouped by Invoice

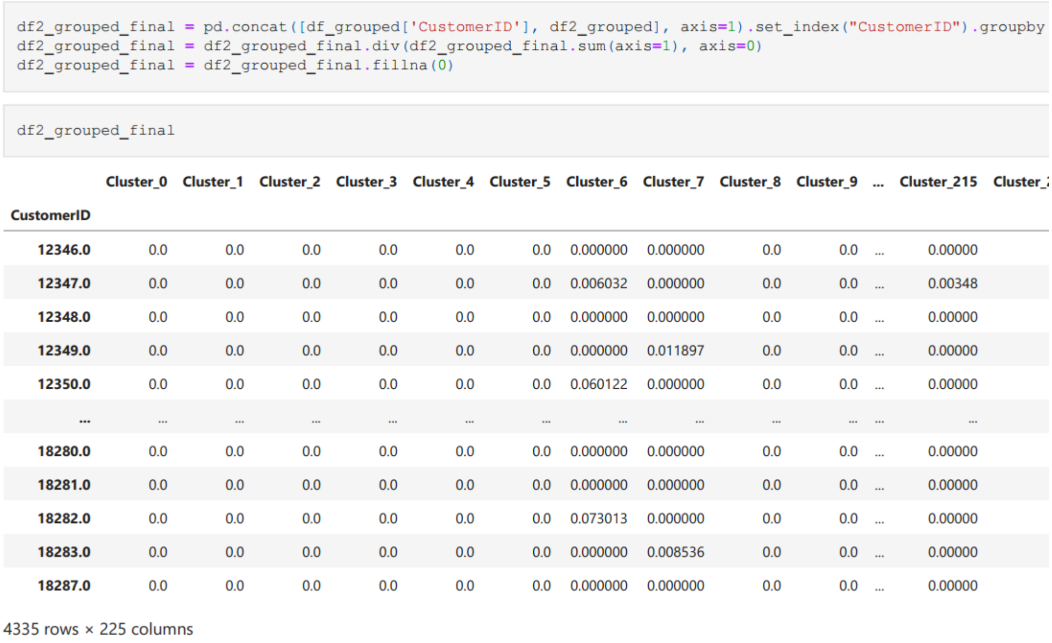


	TotalPrice	Recency	Frequency	Monetary_Value	CustomerID	Quantity	Country
InvoiceNo							
536365	15.30	373.0	297	5391.21	17850.0	40	34
536366	11.10	373.0	297	5391.21	17850.0	12	34
536367	54.08	57.0	171	3232.59	13047.0	83	34
536368	25.50	57.0	171	3232.59	13047.0	15	34
536369	17.85	57.0	171	3232.59	13047.0	3	34
...	...	...	...	...	...	...	...
581583	58.00	1.0	197	25977.16	13777.0	76	34
581584	51.84	1.0	197	25977.16	13777.0	120	34
581585	4.68	1.0	263	4206.39	15804.0	278	34
581586	23.60	1.0	201	12245.96	13113.0	66	34
581587	23.40	1.0	49	790.81	12680.0	105	12

18405 rows × 7 columns

Table 13. Outcomes grouped by InvoiceNo

4.6.2 Grouping by Customer



```
df2_grouped_final = pd.concat([df_grouped['CustomerID'], df2_grouped], axis=1).set_index("CustomerID").groupby
df2_grouped_final = df2_grouped_final.div(df2_grouped_final.sum(axis=1), axis=0)
df2_grouped_final = df2_grouped_final.fillna(0)
```

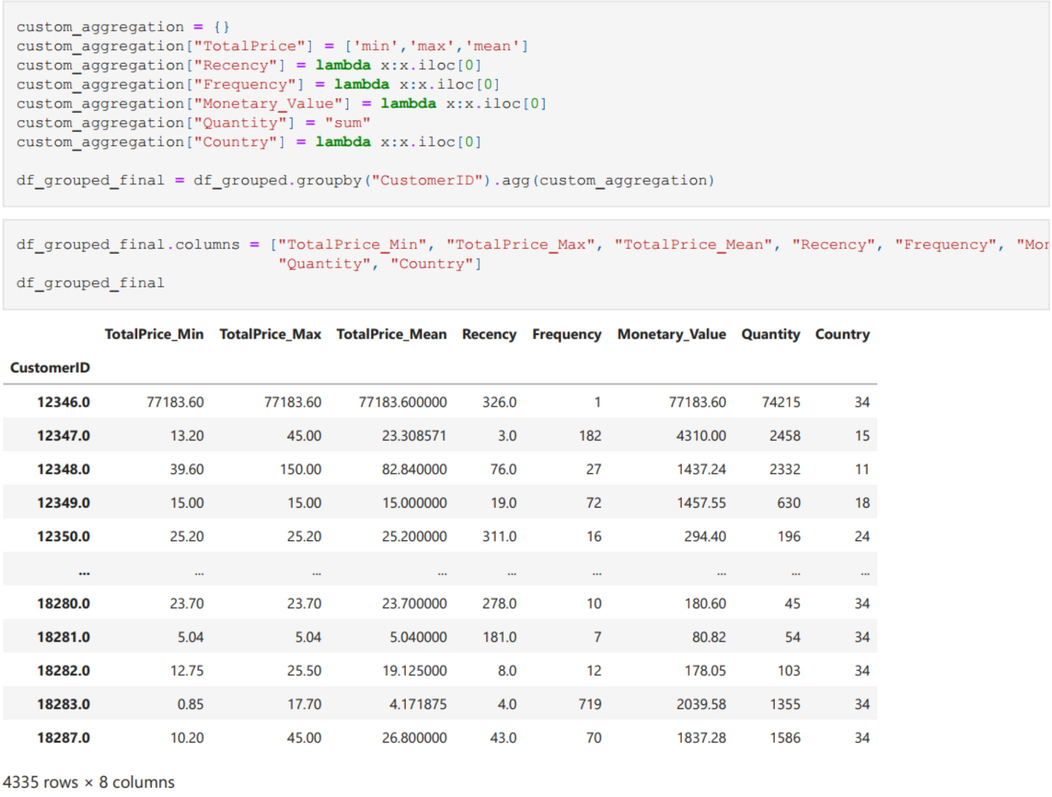
```
df2_grouped_final
```

	Cluster_0	Cluster_1	Cluster_2	Cluster_3	Cluster_4	Cluster_5	Cluster_6	Cluster_7	Cluster_8	Cluster_9	...	Cluster_215	Cluster_
CustomerID													
12346.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.00000	
12347.0	0.0	0.0	0.0	0.0	0.0	0.0	0.006032	0.000000	0.0	0.0	...	0.00348	
12348.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.00000	
12349.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.011897	0.0	0.0	...	0.00000	
12350.0	0.0	0.0	0.0	0.0	0.0	0.0	0.060122	0.000000	0.0	0.0	...	0.00000	
...	...	...	...	...	...	...	...	...	...	...	...	...	
18280.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.00000	
18281.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.00000	
18282.0	0.0	0.0	0.0	0.0	0.0	0.0	0.073013	0.000000	0.0	0.0	...	0.00000	
18283.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.008536	0.0	0.0	...	0.00000	
18287.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.00000	

4335 rows × 225 columns

Fig. 36. Outcomes grouped by Customer

Both outcomes of Group by Invoice and Group by Customer were then merged to achieve fig. 37



```
custom_aggregation = {}
custom_aggregation["TotalPrice"] = ['min','max','mean']
custom_aggregation["Recency"] = lambda x:x.iloc[0]
custom_aggregation["Frequency"] = lambda x:x.iloc[0]
custom_aggregation["Monetary_Value"] = lambda x:x.iloc[0]
custom_aggregation["Quantity"] = "sum"
custom_aggregation["Country"] = lambda x:x.iloc[0]

df_grouped_final = df_grouped.groupby("CustomerID").agg(custom_aggregation)
```

```
df_grouped_final.columns = ["TotalPrice_Min", "TotalPrice_Max", "TotalPrice_Mean", "Recency", "Frequency", "Mor
                            "Quantity", "Country"]
df_grouped_final
```

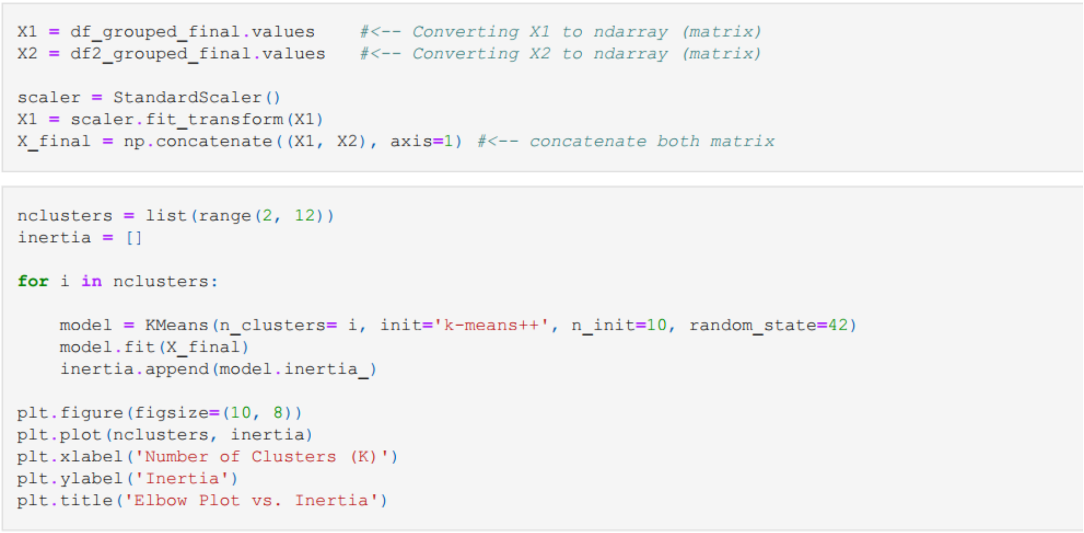
	TotalPrice_Min	TotalPrice_Max	TotalPrice_Mean	Recency	Frequency	Monetary_Value	Quantity	Country
CustomerID								
12346.0	77183.60	77183.60	77183.600000	326.0	1	77183.60	74215	34
12347.0	13.20	45.00	23.308571	3.0	182	4310.00	2458	15
12348.0	39.60	150.00	82.840000	76.0	27	1437.24	2332	11
12349.0	15.00	15.00	15.000000	19.0	72	1457.55	630	18
12350.0	25.20	25.20	25.200000	311.0	16	294.40	196	24
...	...	...	...	...	...	...	...	...
18280.0	23.70	23.70	23.700000	278.0	10	180.60	45	34
18281.0	5.04	5.04	5.040000	181.0	7	80.82	54	34
18282.0	12.75	25.50	19.125000	8.0	12	178.05	103	34
18283.0	0.85	17.70	4.171875	4.0	719	2039.58	1355	34
18287.0	10.20	45.00	26.800000	43.0	70	1837.28	1586	34

4335 rows × 8 columns

Fig. 37. Merging the outcomes in Table 14 and Fig. 36

4.6.3 Clustering with K-Means

Before K-means is applied, it is important to ensure that the dataset follow a standard normal distribution. to, hence the use of scaler to scale X1 and then concatenate X1 and X2 column wise. Then make use of



```
X1 = df_grouped_final.values    #<-- Converting X1 to ndarray (matrix)
X2 = df2_grouped_final.values   #<-- Converting X2 to ndarray (matrix)

scaler = StandardScaler()
X1 = scaler.fit_transform(X1)
X_final = np.concatenate((X1, X2), axis=1) #<-- concatenate both matrix
```

```
nclusters = list(range(2, 12))
inertia = []

for i in nclusters:

    model = KMeans(n_clusters= i, init='k-means++', n_init=10, random_state=42)
    model.fit(X_final)
    inertia.append(model.inertia_)

plt.figure(figsize=(10, 8))
plt.plot(nclusters, inertia)
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia')
plt.title('Elbow Plot vs. Inertia')
```

Fig. 38. Normalizing the dataset for K-means clustering

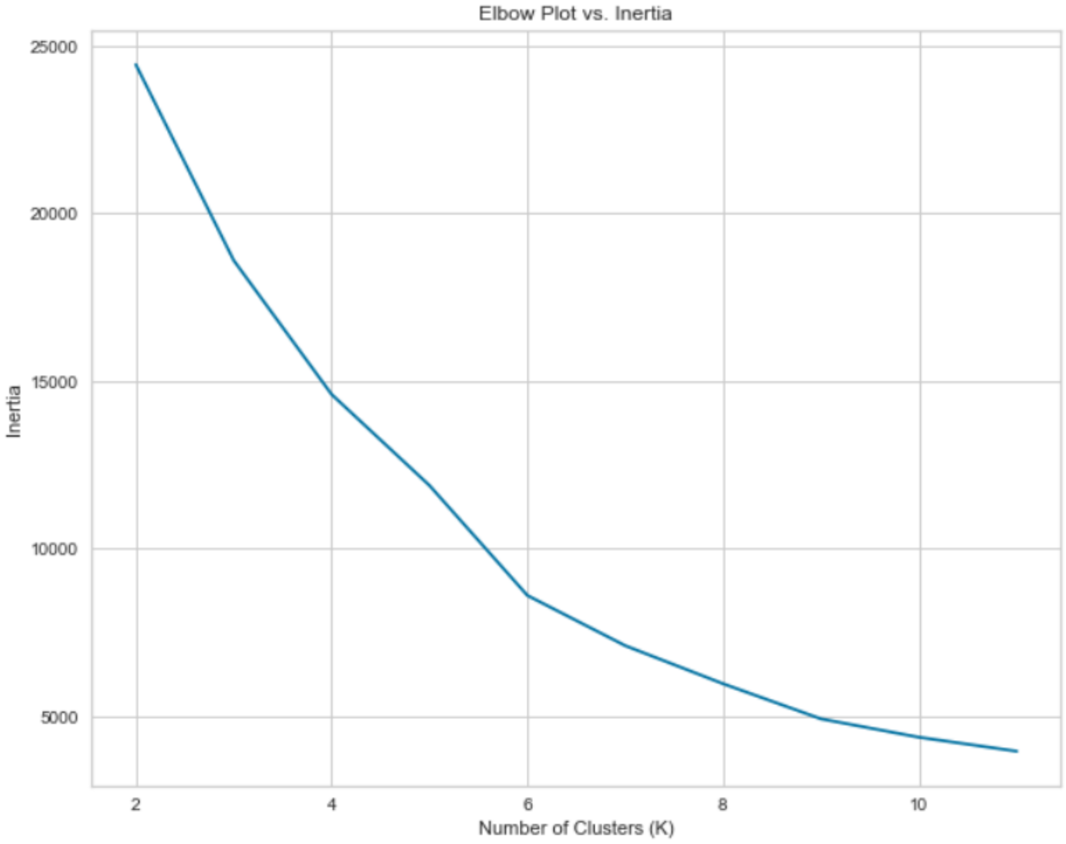
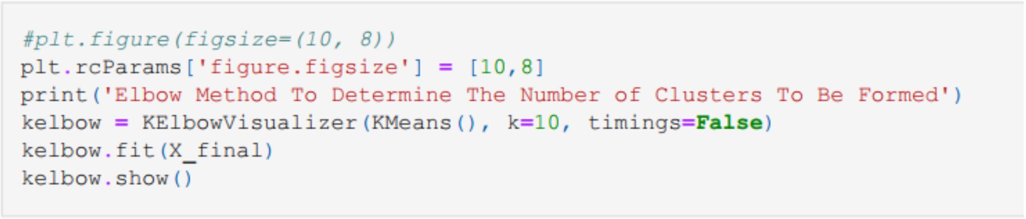


Fig. 39. Elbow plot for standardized dataset

From the above elbow plot, we can see that the number of clusters are 6. To further reverify this claim, we utilize KElbowVisualizer sourced from yellowbrick



```
#plt.figure(figsize=(10, 8))
plt.rcParams['figure.figsize'] = [10,8]
print('Elbow Method To Determine The Number of Clusters To Be Formed')
kelbow = KElbowVisualizer(KMeans(), k=10, timings=False)
kelbow.fit(X_final)
kelbow.show()
```

Fig. 40. Confirmatory exploration using KElbow Visualizer

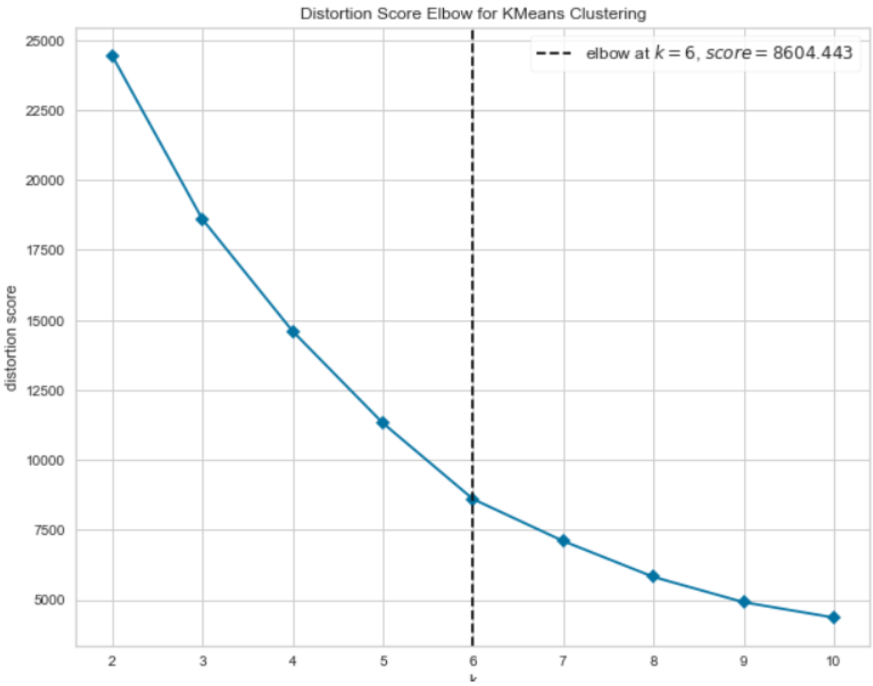
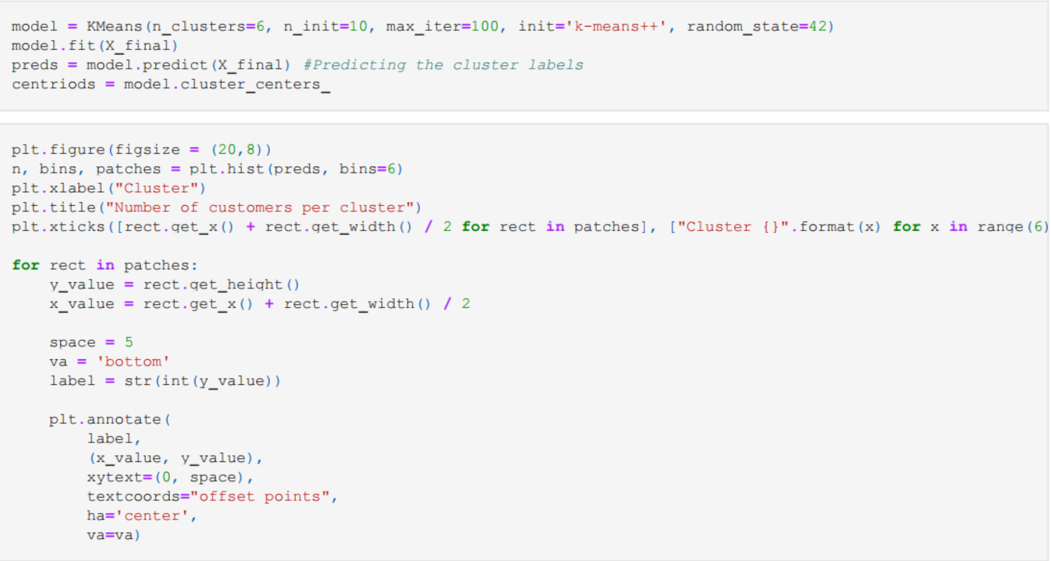


Fig. 41. Confirmatory KElbow visualizer plot

The KElbow Visualizer affirms our earlier position of 6 clusters.

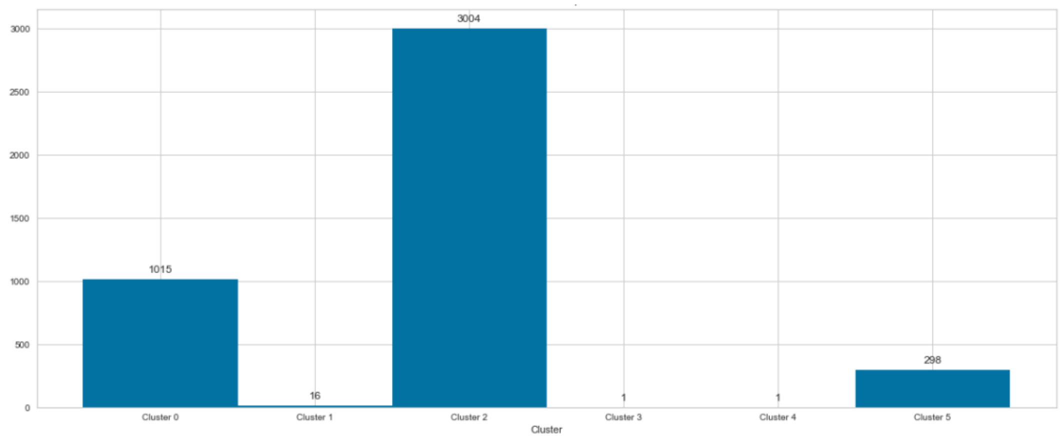


```
model = KMeans(n_clusters=6, n_init=10, max_iter=100, init='k-means++', random_state=42)
model.fit(X_final)
preds = model.predict(X_final) #Predicting the cluster labels
centriods = model.cluster_centers_
```

```
plt.figure(figsize = (20,8))
n, bins, patches = plt.hist(preds, bins=6)
plt.xlabel("Cluster")
plt.title("Number of customers per cluster")
plt.xticks([rect.get_x() + rect.get_width() / 2 for rect in patches], ["Cluster {}".format(x) for x in range(6)

for rect in patches:
    y_value = rect.get_height()
    x_value = rect.get_x() + rect.get_width() / 2

    space = 5
    va = 'bottom'
    label = str(int(y_value))

    plt.annotate(
        label,
        (x_value, y_value),
        xytext=(0, space),
        textcoords="offset points",
        ha='center',
        va=va)
```

Fig. 42. Inputting our generator number of clusters

Since we have established our number of clusters, we then proceed to input the values into our algorithm to determine the exact number of customers in a cluster from the activity in fig. 42 and represented in a plot shown in fig. 43. Machine learning provides the predictions and centroids have been confirmed



b Fig. 43. Representing customers per cluster

Cluster 1 – 1015 customers

Cluster 2 – 16 customers

Cluster 3 – 3004 customers

Cluster 4 – 1 customer

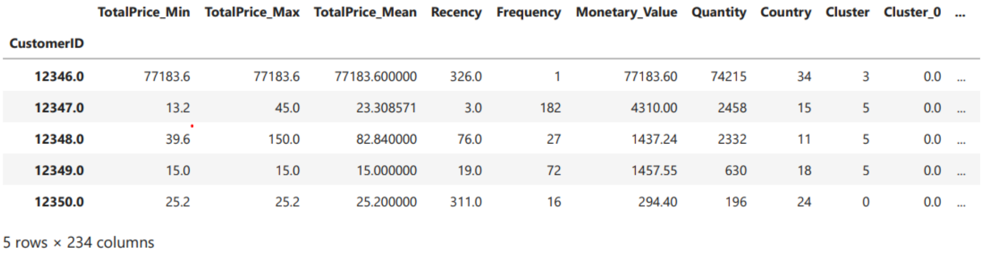
Cluster 5 – 1 customer

Cluster 6 – 298 customers

4.6.4 Profiling

Now that we've formed the clusters, let's now segment the customers in their respective clusters to know who the star customers are and those who needs more attention from the organisation. This is the converging point for RFM Segmentation and K-means Clustering in our analysis.

A new column is then introduced to the data frame called “Clusters” and then a concatenation of df and df 2 is executed to achieve our new data frame as seen in Table 15

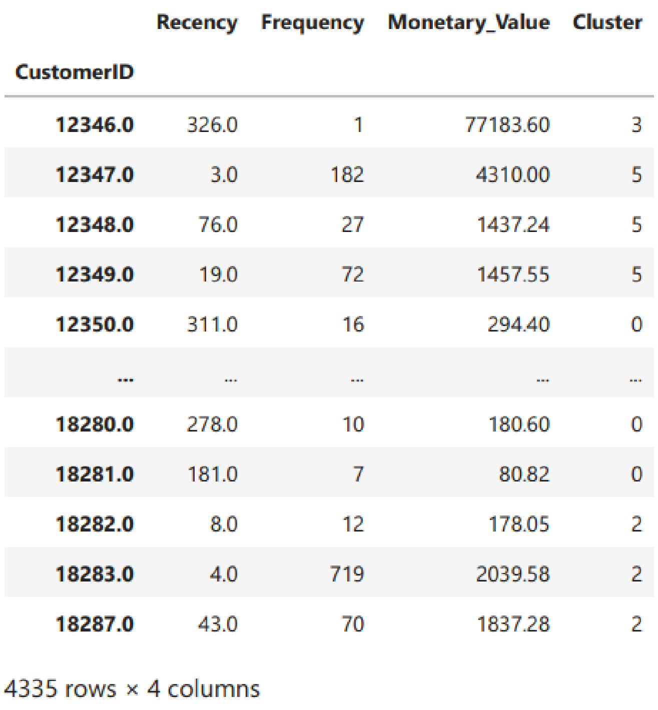


	TotalPrice_Min	TotalPrice_Max	TotalPrice_Mean	Recency	Frequency	Monetary_Value	Quantity	Country	Cluster	Cluster_0	...
CustomerID											
12346.0	77183.6	77183.6	77183.600000	326.0	1	77183.60	74215	34	3	0.0	...
12347.0	13.2	45.0	23.308571	3.0	182	4310.00	2458	15	5	0.0	...
12348.0	39.6	150.0	82.840000	76.0	27	1437.24	2332	11	5	0.0	...
12349.0	15.0	15.0	15.000000	19.0	72	1457.55	630	18	5	0.0	...
12350.0	25.2	25.2	25.200000	311.0	16	294.40	196	24	0	0.0	...

5 rows × 234 columns

Table 14. Data frame with additional “Clusters” column

We are also able to establish table 16, which will become very useful for our cluster interpretation.



	Recency	Frequency	Monetary_Value	Cluster
CustomerID				
12346.0	326.0	1	77183.60	3
12347.0	3.0	182	4310.00	5
12348.0	76.0	27	1437.24	5
12349.0	19.0	72	1457.55	5
12350.0	311.0	16	294.40	0
...	...	...	...	...
18280.0	278.0	10	180.60	0
18281.0	181.0	7	80.82	0
18282.0	8.0	12	178.05	2
18283.0	4.0	719	2039.58	2
18287.0	43.0	70	1837.28	2

4335 rows × 4 columns

Table 15. Merged critical features of RFM and K-means outcomes

Using Plotly to represent these key features, which presents us with a unique image that not only represents the clusters but also tells us about the actual behaviour of the customers in a simplistic way that can easily be understood.

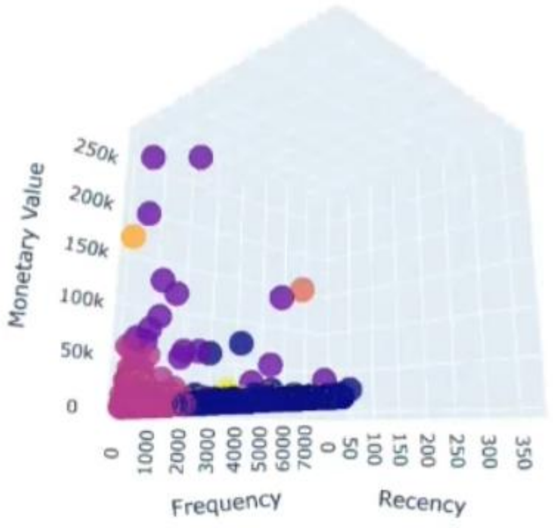
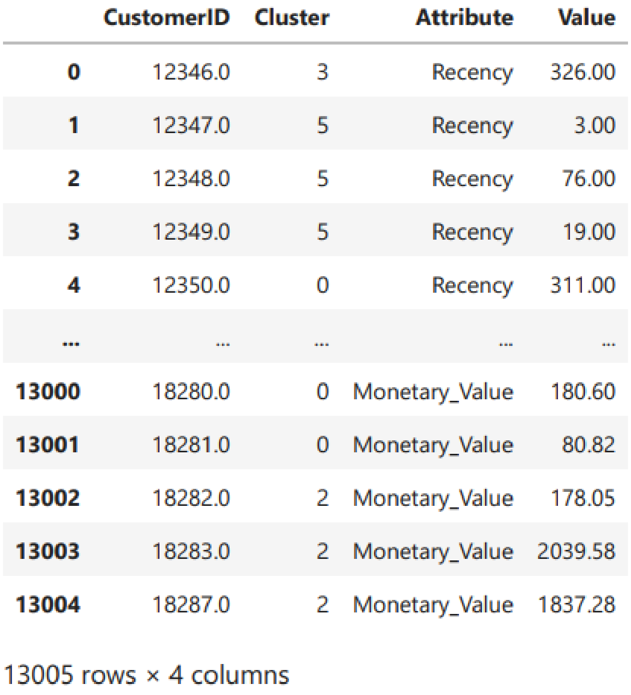


Fig. 44. A 3-Dimensional representation of merged features

4.6.5 Interpreting the Clusters

With the intention to generate a snake plot for this cluster, which will aid in the interpretation, we decide to melt the data frame and reflect the strongest attribute of each customer which generates table 17

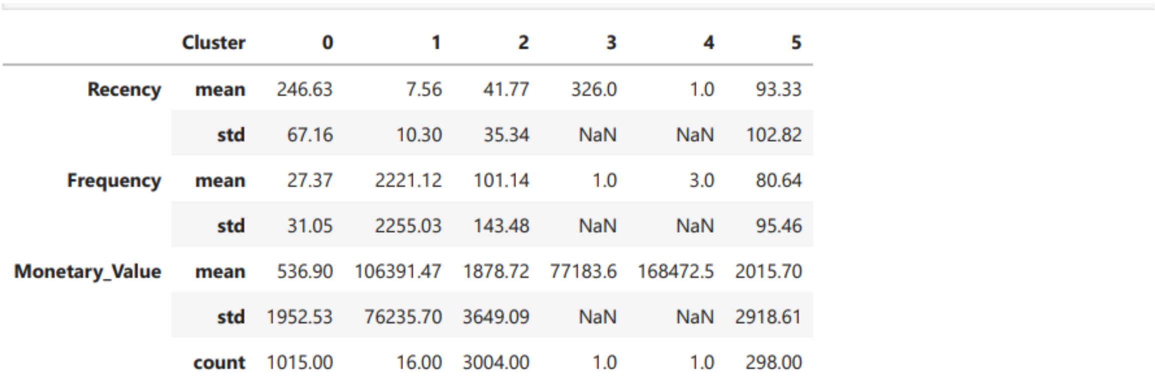


	CustomerID	Cluster	Attribute	Value
0	12346.0	3	Recency	326.00
1	12347.0	5	Recency	3.00
2	12348.0	5	Recency	76.00
3	12349.0	5	Recency	19.00
4	12350.0	0	Recency	311.00
...	...	...	...	...
13000	18280.0	0	Monetary_Value	180.60
13001	18281.0	0	Monetary_Value	80.82
13002	18282.0	2	Monetary_Value	178.05
13003	18283.0	2	Monetary_Value	2039.58
13004	18287.0	2	Monetary_Value	1837.28

13005 rows × 4 columns

Table 16. Indicative prevalent attribute for each customer

As indicated above, customer 12346 belongs to cluster 3 and has a strong attribute of "Recency". We then proceed to discover the mean and standard RFM feature for each cluster.



	Cluster	0	1	2	3	4	5
Recency	**mean**	246.63	7.56	41.77	326.0	1.0	93.33
	std	67.16	10.30	35.34	NaN	NaN	102.82
Frequency	**mean**	27.37	2221.12	101.14	1.0	3.0	80.64
	std	31.05	2255.03	143.48	NaN	NaN	95.46
Monetary_Value	**mean**	536.90	106391.47	1878.72	77183.6	168472.5	2015.70
	std	1952.53	76235.70	3649.09	NaN	NaN	2918.61
	count	1015.00	16.00	3004.00	1.0	1.0	298.00

Table 17. Mean and Standard Deviation of RFM per Cluster

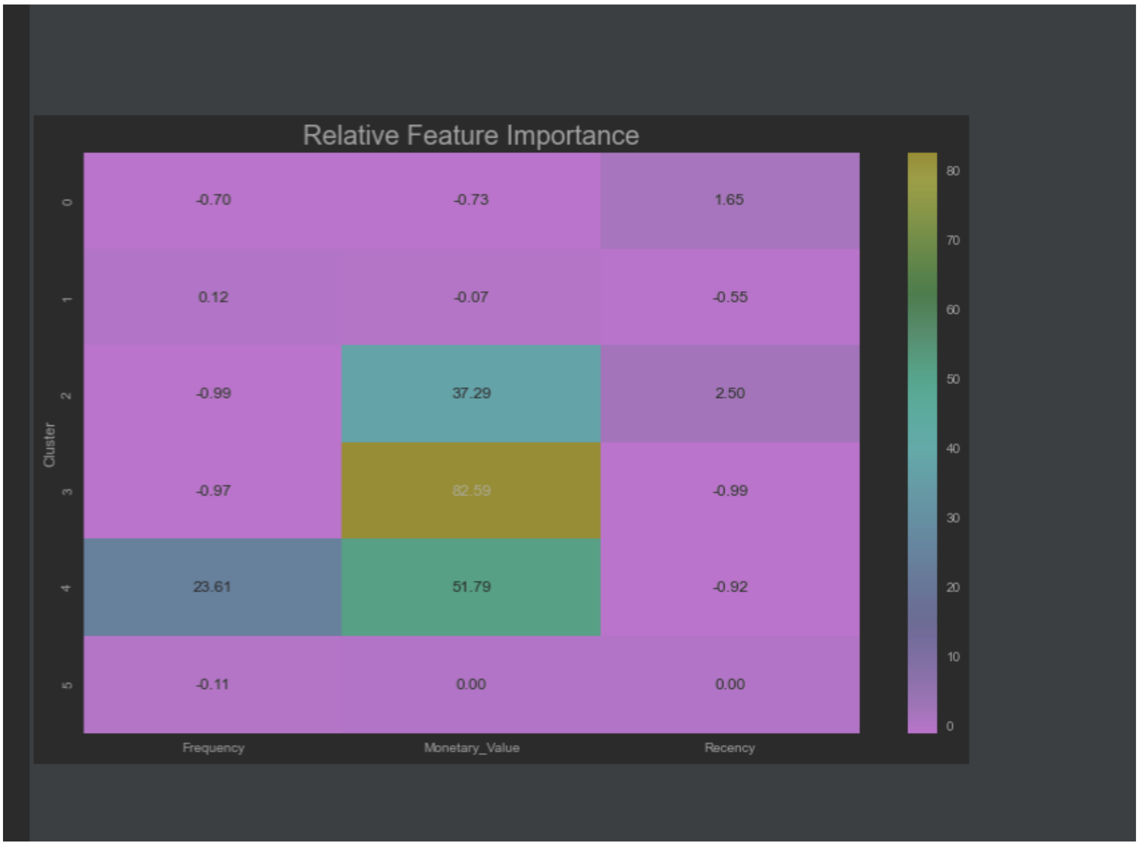


Fig. 45. Heatmap

From the above heatmap we can see the following;

- Customers in cluster 1 have the highest Frequency (Loyal Customers).
- Customers in cluster 4 have the highest Monetary Value (Big Spenders) followed by cluster 1 and cluster 3 respectively.
- Customers in cluster 4 have the lowest Recency (Best Customers) followed by cluster 1.

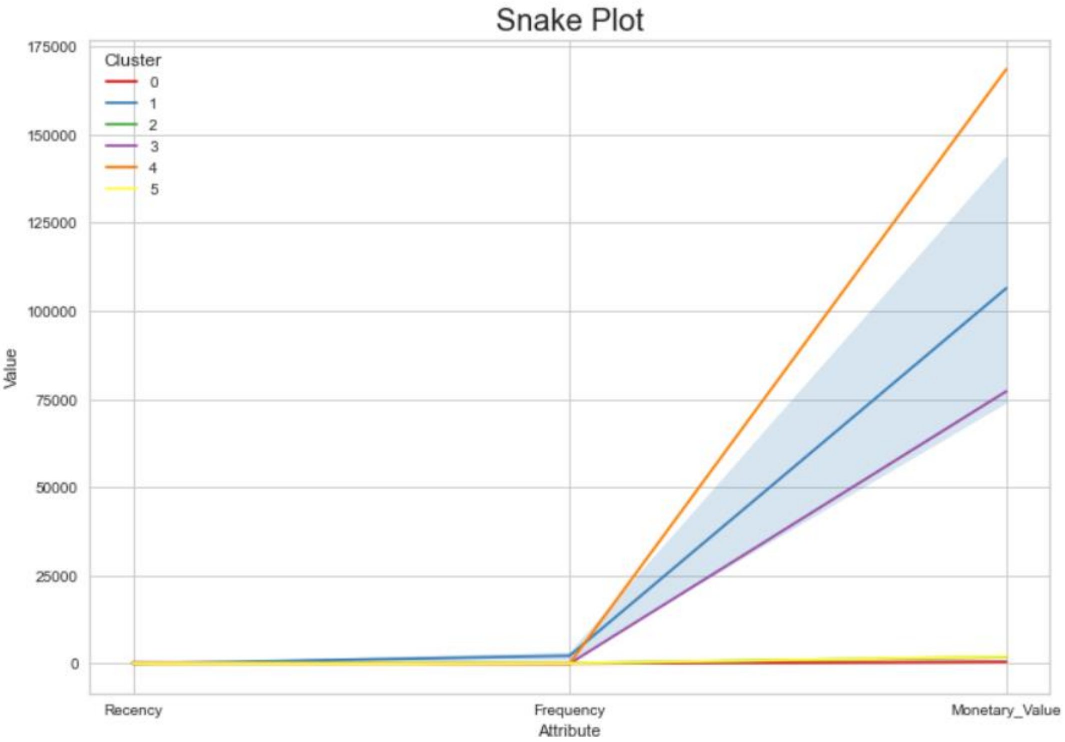


Fig. 46. Snake Plot representing Table 18.

- Cluster 4 is the most important cluster for the business, and it's compound by just 1 customer. They have the lowest Recency mean (1 day has passed since their last purchase), the highest Frequency (On average they've purchased products from the store 3 times), and Monetary Value (On average they spent $168,472.5). The customer(s) in this cluster could be called the Gold Customers, Big Spenders or the best customers.

- Cluster 1 is compound by 16 customers. They have the 2nd lowest Recency mean (7 days have passed since their last purchase), Frequency (On average they've purchased products from the store 2,221 times), and Monetary Value (On average they spent $106,391.47). The customer(s) in this cluster could be called the Silver Customers or Loyal Customers

- Cluster 3 is also compound by 1 customer. They have third lowest Recency mean (326 days; almost a year, have passed since their last purchase), Frequency (On average they've purchased products from the store once), and

Monetary Value (On average they spent $77,183.6) The customer(s) in this cluster could be called the Almost Lost customers

- Cluster 5 is compound by 298 customers, a Recency mean (93 days has passed since their last purchase), Frequency (On average they purchased a product from the store 80 times), and Monetary Value (On average they've spent $2,015.70).

- Cluster 2 is compound by 3004 customers. They have a Recency mean of 41.77 (41 days has passed since their last purchase), Frequency (On average they purchased a product from the store 101 times), and Monetary Value (On average they spend $1,878.72).

- Finally, Cluster 0 could be called lost customers, 255 customers belong to this cluster, they have 2nd-to-the highest Recency (246 days has passed since their last purchase), lowest Frequency (On average they've patronized a store 27 times) and lowest Monetary Value (On average they spend $536.90) The customers in this cluster could be called Lost cheap customers

The result from our clustering model revealed that:

Customers in cluster 5, cluster 2, and cluster 0 needs serious marketing attention.

While customers in cluster 1, cluster 3, and cluster 4 are the best customers the retail store has got.

With proper Supplier Relationship Management (SRM) in place, the retail store can focus more on improving the experience its best customers (cluster 1, cluster 3, cluster 4) get from the products they purchase.

5. Discussion and Recommendations

As organisations struggle to keep up with the ever-changing world of retail business the exercise we have successfully concluded and document can provide great insights on how the business in question can survive and thrive

The first thing to note from our results is that most of the customers are in the United Kingdom and the highest spenders according to Table 14 are all based in the UK with country code 34. This is not surprising because delivery and returns will be easy which will be a result of proximity to source. Location is a critical function in supply chain management. There has been extensive research and documentation on the role of location in Operations. One of such reviews is Melo, et el who alluded that location is critical role in the design of supply chain networks (Melo, et al, 2009)

Regardless of the level of advertisement of promotional efforted that this retail organisation embarks on, their customer segmentation may remain the same unless they consider a review of their supply chain. Based on our review in chapter two, we can see that fig 1. May yet hold the key to realigning the current segmentation of this business – unless it does not have any interest of going global. But it is already a global business with customers in 37 countries. Their ability to review their processes will definitely increase the Frequency feature of the clusters and ultimately increase the number of best customers.

More so, with SRM, the retail store can find new ways to involve key suppliers who can help the company gain a competitive edge. This approach can easily be used to develop a two-way, mutually beneficial relationships with strategic supply partners to deliver greater levels of innovation and competitive advantage that could not easily be achieved by operating independently.

The number of cancelled orders is 3654, which is 16.47% of the entire dataset. If this be the case, feeding this information into the supply chain will provide the required information to determine re-purposing of inventory, rather then have a situation where such products might be sold at a discount – consider the fact that it is a unique gifting product business. This case may not be resolved by marketing efforts, rather a well aligned supply chain.

6. Conclusions

First, I would like to say that this has been an interesting experience, a learning journey into what makes the retail sector thick as well as deeper understanding of the trends that has shaped the application of intelligent computing in the sector. The thought of mainstreaming customer segmentation into the core operations of a retail business instead of treating it as a marketing insight created a new industry understanding.

The decision to apply both RFM Segmentation and K-means as a joint approach is to get more insight and understanding of the dataset, after clustering the customers base on similarities in behaviour using k-means, we can also determine who are best customers, loyal customer etc by incorporating the quartile table from RFM which can be used to personalize market offer.

Also the world is fast-changing and organisations are forced to change and adapt in order to remain competitive, the role of the supply chain and the way organisations are engaging with suppliers is also changing. In order to determine how purchasing and the supply base can add value to the competitiveness of an organisation, the landscape that the organisation operates in, needs to be understood also with proper Supplier Relationship Management (SRM) in place, the retail store can focus more on improving the experience of its best customers (cluster 1, cluster 3, cluster 4) get from the products they purchase.

Furthermore, with customer segmentation analysis which we have done, it is easy to identify customers in cluster 2, cluster 5, & cluster 0 who require extra attention. It can also help create targeted strategies that capture customers' attention and create positive, high-value experiences with the company, also with Supply Relationship Management, the retail store can find new ways to involve key suppliers who can help the company gain a competitive edge. This approach can easily be used to develop a two-way, mutually beneficial relationships with strategic supply partners to deliver greater levels of innovation and competitive advantage that could not easily be achieved by operating independently.

Limitations

Considering the fact that we are conducting a controlled activity based on finalized numbers, our results will be reporting within a set of limitations.

i. Study is limited to an occasion gift company based in the United Kingdom (UCI, 2022)
ii. Our dataset is limited to 8 variables as provided by our dataset in eight columns, any new information provided will have these variables as a decision source.
iii. Recommended outcomes will be limited to international retailing businesses who transact on tangibles.

7. Reference list

Alleman, G. (2014) Performance-Based Project Management: Increasing the Probability of Project Success, AMACOM, ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/UNICAF/detail.action?docID=1407878

Arnold, R. D. and Wade, J. P (2015) A Definition of Systems Thinking: A Systems Approach, Procedia Computer Science, Volume 44, Pages 669-678, USA, Elsevier, ISSN 1877-0509

Bhattacharjee, C. (2013), Retail Management, DMGT550 Retail Management | PDF | Retail | Sales (scribd.com) Lovely Professional University Phagwara

Boateng, E. and Abaye, D. (2019) A Review of the Logistic Regression Model with Emphasis on Medical Research. Journal of Data Analysis and Information Processing, 7, 190-207. Available: https://tinyurl.com/2bzsnedz [assessed: August 30, 2022]

British Broadcasting Corporation (BBC), (2022). Why has the Syrian war lasted 11 years? Available: https://www.bbc.com/news/world-middle-east-35806229 [accessed: 7th July, 2022]

Chen D., Sain, S. L., and Guo, K., (2012) Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining, Journal of Database Marketing and Customer Strategy Management, Vol. 19, No. 3, pp. 197â€ “208 (Published online before print: 27 August 2012. 7 doi: 10.1057/dbm.2012.1).

Chen, P. P-S. (1977), The Entity Relationship Model - Towards a Unified View of Data, Cambridge Massachusetts, Centre for Information Systems Research, Massachusetts Institute of Technology.

Christopher, M. (1998). Logistics and Supply Chain Management: Strategies for reducing cost and improving service. London, Financial Times Pitman Publishing

Cooper, M. C., Lambert D. M., Pagh D. J. (1998). "Supply Chain Management: More than a new name for Logistics." The International Journal of Logistics Management 8(1), p.1-13.

Cousera (2022) What Is Customer Segmentation? + How to Reach Customer Segments. Cousera.org Available: https://www.coursera.org/articles/customer-segmentation [assessed August 25, 2022]

Davidson, I. (2002). Understanding K-means non-hierarchical clustering. Computer Science Department of State University of New York (SUNY), Albany.

Dawson, John. (2010). Retail Trends in Europe, United Kingdon, Springer 10.1007/978-3-540-72003-4_5

Doğan, O., Ayçin, E. and Bulut, Z. A. (2018) Customer Segmentation by Using RFM Model and Clustering Methods: A Case Study in retail Industry. International Journal of Contemporary Economics and Administrative Sciences, Volume :8, Issue: 1, pp.1-19 ISSN: 1925 – 4423

Fader, P. S., Hardie, B. G., & Lee, K. L. (2005). RFM and CLV: Using iso-value curves for customer base analysis. Journal of marketing research, 42(4), 415-430.

FITZ-ENZ, J., and Mattox, J. (2014) Predictive Analytics for Human Resources, https://www.scribd.com/, Wiley ISBN: 9781118940693

Foster, L., Haltiwanger, J., Klimek, S., Krizanc, C.J., & Ohlmacher, S. (2016). "Chapter 1: The evolution of national retail chains: how we got here". In Handbook on the Economics of Retailing and Distribution. Cheltenham, UK: Edward Elgar Publishing. Retrieved Aug 30, 2022, from https://www.elgaronline.com/view/edcoll/9781783477371/9781783477371.00009.xml

Gadde M. (2021) Simple Explanation to Understand K-Means Clustering, Analytics Vidhya. Available: www.analyticsvidhya.com/blog/2021/02/simple-explanation-to-understand-k-means-clustering/ [accessed: September 2, 2022]

Gartner (2019) Gartner Says Worldwide Customer Experience and Relationship Management Software Market Grew 15.6% in 2018. Available: https://tinyurl.com/Gatneer [accessed: August 15, 2022]

Giménez, C. and Lourenço H. R. (2004) e-Supply Chain Management: Review, Implications and Directions for Future Research, Barcelona, Document de Treball, Research Group in Business Logistics GREL- IET, Department of Economics & Business, Universitat Pompeu Fabra, Ramon Trias Fargas.

Gutierrez, D. (2020) Why you should be Using Jypter Notebooks, https://opendatascience.com/why-you-should-be-using-jupyter-notebooks/, Open Data Science Society [Accesses on August 27th, 2022)

Han, J., and Kamber, M. (2006), Data Mining: Concepts and Techniques Solution Manual, Second Edition, University of Illinois at Urban-Champaign, Morgan Kaufmann.

Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2013) Applied Logistic Regression, Third Edition. New York, Wiley

Hosmer, D.W. and Lemeshow, S. (1989) Applied Logistic Regression, First Edition, New York. Wiley

https://reliefweb.int/report/democratic-republic-congo/urgent-action-needed-defuse-violence-democratic-republic-congo-senior-officials-tell-security-council-urging-support-ongoing-regional-efforts 31 May 2022 [Accessed on July 5, 2022]

https://www.bbc.com/news/world-middle-east-35806229, 15th of march 2022 [Accessed on July 5, 2022]

https://www.cfr.org/in-brief/escalating-violence-putting-nigerias-future-line Ebenezer Obadare, 9th of June 2022 [Accessed on July 5, 2022]

IBM Cloud Education (2020) Natural Language Processing (NLP) Available: https://www.ibm.com/cloud/learn/natural-language-processing [assessed: September 2, 2022]

Jha, N., Parekh, D., Mouhoub, M., Makkar, V. (2020). Customer Segmentation and Churn Prediction in Online Retail. In: Goutte, C., Zhu, X. (eds) Advances in Artificial Intelligence. Canadian AI 2020. Lecture Notes in Computer Science, vol 12109. Springer, Cham. https://doi.org/10.1007/978-3-030-47358-7_33

Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. Journal of the Royal Statistical Society. Series C (Applied Statistics), 29(2), 119–127. Wiley

KAZIUKĖNAS, J. (2021) Marketplace Pulse Available: https://www.marketplacepulse.com/articles/amazon-reaches-six-million-third-party-sellers [assessed: August 28, 2022]

Kononow, P. (2017), ER Diagram vs Data Dictionary – Which is Better for Documenting Data Models, https://tinyurl.com/KononowP, Dataedo

KPMG (2009) The Evolution of Retailing: Re-inventing the Customer Experience, Scribd.com. Available: https://www.scribd.com/document/253212986/Evolution-Retailing. [accessed: August 31st, 2022]

Kuhlman D. (2013) A Book of Python, Available: http://www.davekuhlman.org/python_book_01.pdf [assessed September 1, 2022]

Kuhlman D. (2018) A Summary of Tools for data science for Python, Available: http://www.davekuhlman.org/py-datasci-survey.html [assessed September 1, 2022]

Kuhlman, D. (2013) A Python Book: Beginning Python, Advanced Python, and Python Exercise, Revision 1.3a http://www.davekuhlman.org/python_book_01.pdf

Lock, D (2014). The Essentials of Project Management, Taylor & Francis Group, ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/unicaf/detail.action?docID=1784652,
Lock, S. (2022) Russia-Ukraine war latest: what we know on day 190 of the invasion. The Guardian: International Edition. Available: https://www.theguardian.com/world/2022/sep/01/russia-ukraine-war-latest-what-we-know-on-day-190-of-the-invasion [accessed: 1st September, 2022]

Lumsden SA, Beldona S., Morison A.M. (2008). Customer Value in an All-Inclusive Travel Vacation Club: An Application of the RFM Framework, Journal of Hospitality & Leisure Marketing, 16:3, 270-285, DOI: 10.1080/10507050801946858 [assessed: August 30, 2022]

MacQueen, J. (1967) Some methods of Classification and Analysis of Multivariate Observations. Los Angeles. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability '65 and '66, University of California Press.

Maley, C.H. (2012). Project Management Concepts, Methods, and Techniques (1st ed.). New York, Auerbach Publications.
Maverick, J.B. (2021) An Example of a Standard Profit and Loss (P&L) Statement, Investopedia. Available: https://tinyurl.com/InvPandL [assessed: 31st August, 2022]

McCarty, J. A.and Hastak, M. (2007) Segmentation approaches in data-mining: A comparison of RFM, CHAID, and logistic regression, Journal of Business Research, Volume 60, Issue 6, Elsevier ISSN 0148-2963

M.T. Melo, S. Nickel, F. Saldanha-da-Gama,(2009)Facility location and supply chain management – A review, European Journal of Operational Research, Volume 196, Issue 2, Pages 401-412, ISSN 0377-2217,

Obadare, E. (2022), Escalating Violence Is Putting Nigeria's Future on the Line, Council on Foreign Relations. Available: https://www.cfr.org/in-brief/escalating-violence-putting-nigerias-future-line [accessed: 7th July, 2022]

Oxford University Press (2022) Oxford Learner's Dictionaries Available: https://tinyurl.com/oxfretl4 [accessed:1st September, 2022]

Regel, R. (2008) Retail Business Kit for Dummies, 2nd Edition. Indiana, Wiley.

Rina, H. (2013) Retailing: Meaning, Definitions, Evolution, Concept, Features, Theories, Importance, Functions, Channels and Factors, Economics Discussion. Available: https://www.economicsdiscussion.net/marketing-2/retailing/retailing/32357 [accessed 28th August, 2022]

Screemany T. (2021) Introduction to Feature Engineering – Everything You Need to Know! Analytics Vidhya. Available: https://tinyurl.com/yktddy7c [assessed: September 4, 2022]

SNHU (2022) South New Hampshire University, Available: https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming [assessed: September 3, 2022]

Sorescu, A., Frambach, R.T., Singh, J., Rangaswamy, A., Bridges, C. (2011) Innovations in Retail Business Models, Journal of Retailing, Elsevier, Volume 87, Supplement 1 Pages S3-S16, ISSN 0022-4359, (https://www.sciencedirect.com/science/article/pii/S0022435911000340) [Accessed on August 30,2022]

UN Security Council, (2022), 9051ST MEETING (PM) SC/14916. Available: https://press.un.org/en/2022/sc14916.doc.htm [accessed: 1st September, 2022]

University of California Irvin (2012) UCI Machine Learning Repository, Available: https://archive.ics.uci.edu/ml/datasets/Online+Retail# [accessed June 30, 2022]

Wang C-H. (2010). Apply robust segmentation to the service industry using kernel induced fuzzy clustering techniques. Expert Systems with Application, Vol.37. Pages 8395-8400.

Watson M., Lewis S., Cacioppo P, Jayaraman J. (2012) Supply Chain Network Design: Applying Optimization and Analytics to the Global Supply Chain Pearson FT Press. UK

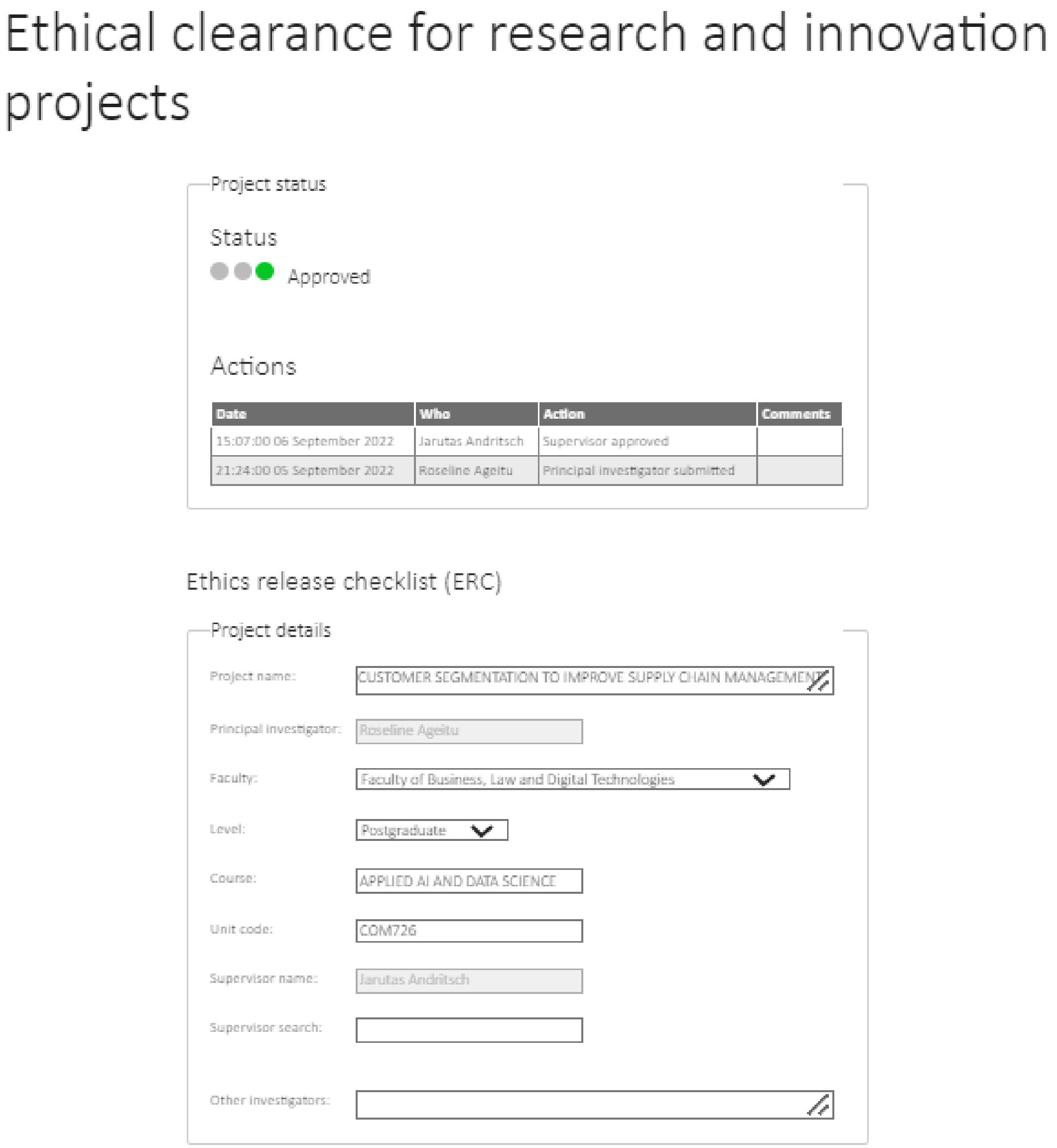
World Health Organisation, (2022). WHO Coronavirus (COVID19) Dashboard. Available: https://covid19.who.int/ [accessed: 31st August, 2022)

8. APPENDIX

Appendix A: Data Sources

https://archive.ics.uci.edu/ml/machine-learningdatabases/00352/Online%20Retail.xlsx

Appendix B: Ethics Release



Ethical clearance for research and innovation projects

Project status

Status

○○● Approved

Actions

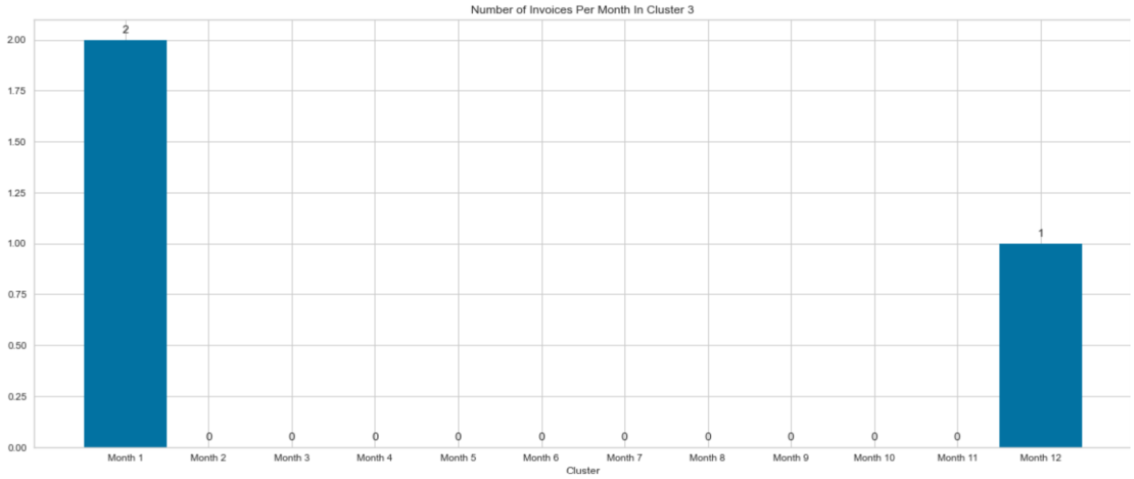
Date	Who	Action	Comments
15:07:00 06 September 2022	Jarutas Andritsch	Supervisor approved	
21:24:00 05 September 2022	Roseline Ageitu	Principal investigator submitted	

Ethics release checklist (ERC)

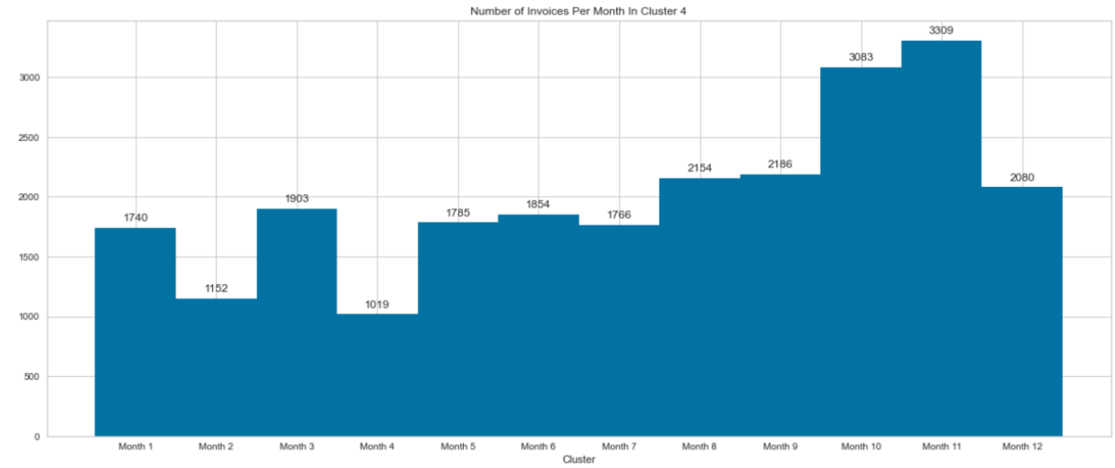
Project details

Project name:	CUSTOMER SEGMENTATION TO IMPROVE SUPPLY CHAIN MANAGEMENT
Principal investigator:	Roseline Ageitu
Faculty:	Faculty of Business, Law and Digital Technologies
Level:	Postgraduate
Course:	APPLIED AI AND DATA SCIENCE
Unit code:	COM726
Supervisor name:	Jarutas Andritsch
Supervisor search:	
Other investigators:	

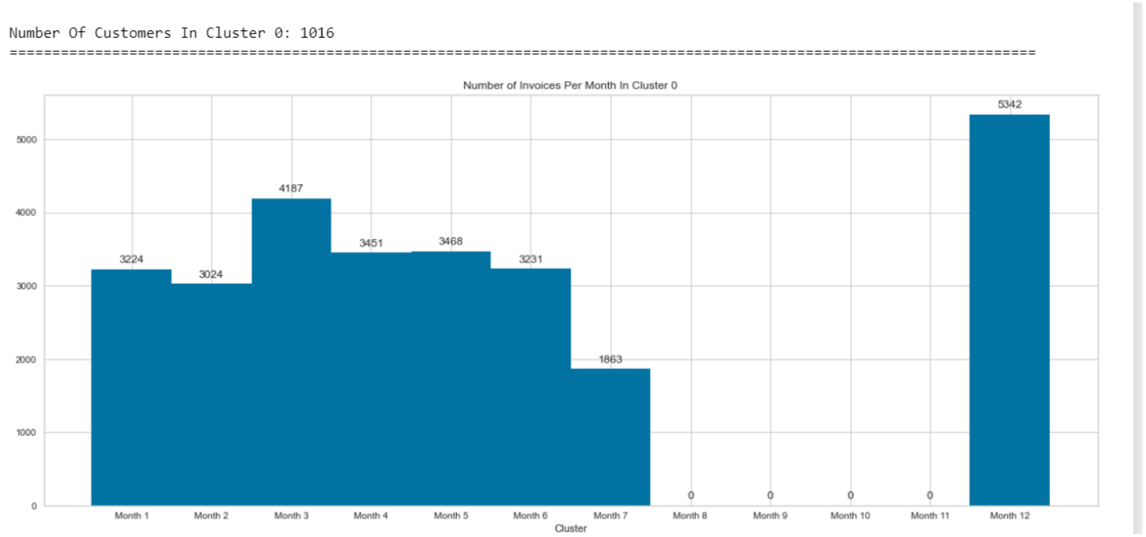
Appendix C: Number of invoices in cluster 3



Appendix D: Number of invoices per mouth in cluster 4



Appendix E: Number of customers in cluster 0



```
Number Of Customers In Cluster 0: 1016
========================================================================================================================
```