SAINT MARY'S UNIVERSITY SINCE 1802

One University. One World. Yours.

MSc in Computing and Data Analytics

MCDA 5580 – Data and Text Mining

Assignment – 1

Submitted to:

Trishla Shah

Prepared by:

Allen Mathew - A00432526

Gyaneshwar Rao - A00433014

Meghashyam - A00432392

Table of Contents

Executive Summary

Data contains all the information of transactions which includes distinct customers who are visiting the store and the detailed information of items purchased. The characteristics of customers and the items are categorized according to the data produced by the business. This characterization is performed by using the unsupervised algorithm which analyses data and groups them to clusters. The nodes are segregated into their respective clusters. Considering these different groups in the customer category, there were groups which bring good revenue to the business while purchasing a few costly items or by purchasing many budgeted items. These customers were contributing largely to the revenue. Few customers are loyal who consistently bought several specific items, which resulted in consistent revenue and few customers were purchasing a specific group of items in large quantities, but they were not visiting the business often. We assume these groups are small business vendors and are potential customer who can bring more interesting patterns of item purchase and revenue. The remaining clusters had the customers who were visiting the store occasionally and there is a need to attract those groups by providing offers and having them visit the store often.

The analysis was also performed on items and the groups that were quite interesting. The more popular groups which brought the highest revenue to the business had the least average price per visit, which indeed indicates that those were the items which were household daily use items and fall into the category of fast-moving consumer goods with least price. The next popular group/cluster has a better average price per visit, but the trend remains as popular groups. Few groups of items were purchased largely in quantity and the remaining groups had the least average prize since the items were returned often. There are few recommendations to navigate investment into quickly consumed groups and to monitor their availability in the store. Further monitoring these groups and understanding developments according to the recommendations must be maintained to improve the business growth. The profiling detailed every category and appropriate recommendations were provided to evaluate the current business processes.

Objective

From the “OnlineRetail” dataset we will be considering 2000 Records each to create the Customer and Product Datasets, which further must go through the cleaning process and data analysis. An unsupervised algorithm (k-means) is performed on the data to identify indigenous clusters in accordance with Customers and Items. The number of clusters is identified using Elbow and Gap Statistics methods. The characteristics of the clusters are studied and compared with each other and relevant information is provided for good business practices.

Data Summary

For our analysis, we will be using the “OnlineRetail” dataset. It consists of 541,909 records which are from December 2011 to December 2012, of the various products bought by customers from multiple countries. The following are the attributes of the data set:

Attributes	**Description**
Invoice Number (InvoiceNo)	It is generally a 6-digit number that uniquely identifies the transaction made by a customer.
Product Item ID (StockCode)	The code is alphanumeric, consisting of 1 - 6 characters. It is used to uniquely identifies the product item.
Description	It is text that is used to describe the stock code. It mainly provides details for the product item.
Quantity	It indicates the number of products brought or returned by the customer.
Unit Price	It is a positive float, that indicates the cost of a single product. But for some records the Unit Price is negative, it was done to “adjust for bad debt” as mentioned in the Description.
CustomerID	It is generally a 6-digit number that uniquely identifies a customer.
Country	It is text that describes the location where the product was bought.
InvoiceDateTime	It consists of the data and time when the product was purchased. In general, the records are between December 2011 to December 2012.

Table 1: Attributes of the “OnlineRetail” Dataset

Observations:

The following table briefly describes the observation made on each of the attributes in the "OnlineRetail" data set:

Attributes	Description
Invoice Number	• All records follow the same pattern, i.e. The code is numeric, consisting of 5 characters. • There are 9,292 records containing Invoice Number 0. • Majority of the records with Invoice Number 0 have a negative value for Quantity.
StockCode	• The code follows 2 distinct patterns: 1. The code is numeric, consisting of 5 characters. 2. The code is alphanumeric, consisting of 5 numeric characters and a single letter. • Apart from the above-mentioned patterns there are 15 unique stock codes that don't follow the above patterns, but they are used to indicate Discount, Bank Charges, Amazon Fee, Samples, Postage, etc.
Description	• For most of the records it displays the title of the product. • It provides describes for the 15 unique stock codes as mentioned in the previous observation. • There are 1,454 records that have no description.
Quantity	• There are 10,624 records that have a negative value. • The Maximum Quantity of a product brought and not returned by a customer is 12,540
Unit Price	• There is 1 record that has a negative value. • The records are between 0.00 to 9.99.
CustomerID	• All records follow the same pattern, i.e. The code is numeric, consisting of 6 characters. • There are 135,080 records containing Customer ID 0. • There are 1,719 records with Customer ID 0 and have a negative value for Quantity. • There are 386 records with Customer ID 0 and Invoice No 0.
Country	• There are 38 distinct counties in the dataset. • Majority of the purchases is done in the United Kingdom. • The least number of purchases is done in Lebanon, RSA and Brazil.
InvoiceDateTime	• All records follow the same pattern of when the product was purchased. • The records are between December 2011 to December 2012.

Table 2: Observations made on the Attributes of the "OnlineRetail" Dataset

Limitations:

Invoice Number 0, CustomerID 0 and a few StockCode items are not clearly defined in the data set. These attributes are interlinked with other attributes in the dataset like Quantity, UnitPrice, etc. so if we remove them from the dataset we would not get accurate results.

Since Invoice Number 0 mainly contains data on the items that were returned by the customer, so if we remove Invoice Number 0 we would not be able to get the actual quantity purchased by the customer.

To overcome this, we summed the quantity of items purchased by a customer, so if a customer purchased and returned a product then the aggregated quantity would be 0.

Outcome:

From the "OnlineRetail" dataset we will created 2 datasets or tables in MySQL, **ProductCluster** and **CustomerCluster.** Each table will consist a total of **2000 records** from the "OnlineRetail" dataset and the data will be order by the highest revenue earned. Each table will be utilized the following attributes:

1. **Invoice Number**
2. **StockCode**
3. **Quantity**
4. **Unit Price**
5. **CustomerID**

Design/Method/Approach

The following steps were performed for our analysis:

1. **Dataset Selection:**
    - We begin by selecting the "OnlineRetail" dataset that contains all the transaction details.
    - Then we review the source data to understand the content, its structure and its interconnectivity with other attributes within the same dataset.
    - Then we check if the data is correctly formatted and consistent. To do so we perform various analytical checks on the data, such as maximum/minimum, sum and count.
2. **Feature Selection:**
    - Then select and engineer the appropriate features to support the analysis.
    - We use the RFM model to characterize the Customer.
3. **Sample Selection:**
    - We utilize the above features to select the top 2000 customers and top 2000 products based on revenue.
4. **Outlier Removal:**
    - We plot the filtered dataset using 'ggpair'. Then we identify and remove the outliers.
5. **Normalize Data:**
    - The features selected from the filtered dataset vary in magnitude, this in turn could cause issues when we use the k-means algorithm in the next step.
    - If we apply the algorithm to the filtered dataset it would take only the magnitude of the features into consideration and neglect the units.
    - The K-mean algorithm uses the Euclidean distance between two data points. Hence this measure is sensitive to magnitude. (Garbade, 2019)
    - Hence, we should scale and equally weigh all features
6. **Clustering:**
    1. a) To determine appropriate number of clusters for the analysis we use:
        - ○ Elbow Method
        - ○ Gap Statistics
    2. b) Performed clustering using k-means algorithm.
7. **Profiling:**
    - De-normalize the data since we have made the clusters for the datasets.
    - Based on clustering results, we create customer and product profile use tableau, after combining the results with the metadata from the original dataset.

Feature Selection

Most often there tends to be a lot more data than is needed to build a model. Hence feature selection is an effort for reducing the inputs for analysis and processing. It often reduces or combine existing attributes and validates their usefulness in the model. (Brownlee, 2019)

1. Features for Customer:

To properly segment the customers based on their behavior we have used RFM (Recency, Frequency, Monetary) for our analysis (Nair, 2019). To characterize the customer, we have used the following features:

Features	Description
PRODUCT_QUANTITY	• The total number of products the customer has purchased during the time frame (i.e. 1 Year) • By Summing the Quantities, it will adjust the aggregation of buying and returning a product. • It utilizes the 'Quantity' attribute of the dataset.
DISTINCT_PRODUCT	• The number of unique products the customer has purchased during the time frame (i.e. 1 Year) • It utilizes the 'StockCode' attribute of the dataset.
TOTAL_REVENUE	• The total amount spent by the customer during the time frame (i.e. 1 Year) • The product of the unit price and quantity will give the cost of the Product Item. • The sum of all the Product Items purchased by the customer will give the TOTAL_REVENUE. • It utilizes the 'UnitPrice' and 'Quantity' attributes of the dataset.
VISITS	• The total number of times the customer visited the online store during the time frame (i.e. 1 Year) • It utilizes the 'InvoiceNo' attribute of the dataset.
PER_VISIT_COST	• The average amount spent by the customer for every visit to the online store during the time frame (i.e. 1 Year) • It is calculated by dividing the Total Revenue by the Total Number of Visits. • It utilizes the 'InvoiceNo', UnitPrice' and 'Quantity' attributes of the dataset.

Table 3: List of Features for Customer

The above features help us to roughly characterize the customers behavior. It could indicate the loyal customers as well as the customers that need to be engaged. Using these features, we will be able to cluster the customers into distinct clusters.

2. Features for Product:

To characterize the product, we have used the following features:

Features	**Description**
DISTINCT_ CUSTOMERS	• The number of unique customers that purchased the product during the time frame (i.e 1 Year) • It utilizes the ‘CustomerID’ attribute of the dataset.
TOTAL_REVENUE	• The total amount generated by the product during the time frame (i.e 1 Year) • The product of the unit price and quantity will give the cost of the Product Item. • The sum of all the Product Items purchased by the customer will give the TOTAL_REVENUE. • It utilizes the ‘UnitPrice’ and ‘Quantity’ attributes of the dataset.
BASKETS	• The total number of times the product was taken from the online store during the time frame (i.e 1 Year) • It utilizes the ‘InvoiceNo’ attribute of the dataset.
QUANTITY_PER_ CUSTOMER	• The average amount of quantity each customer purchases during the time frame (i.e 1 Year) • It is calculated by dividing the Total Product Quantity Purchased by the Total Number of Customers • It utilizes the ‘CustomerID’ and ‘Quantity’ attributes of the dataset.

Table 4: List of Features for Product

The above features help us to roughly characterize products behavior. It could indicate the popular products as well as the products that are not doing so well. Using these features, we will be able to cluster the products into distinct clusters.

Data Cleaning / Outlier Removal:

We didn't perform data cleaning to a large extent since majority of the data seemed consistent apart from the few exceptions mentioned in the above *Data Summary - Limitations* Section. During the analysis we considered the entire dataset included the transactions (InvoiceNo), customers (CustomerID) and product items (StockCode) for our analysis.

The 'OlineRetail' dataset is then filtered to create 2 datasets namely **CustomerCluster** and **ProductCluster** based on the features selected in *Table 3 and 4*. The fileted datasets consist of 2000 records each and they are order by the highest revenue earned. The datasets are then exported as a .csv file since it would be easy to read the file from RStudio.

Note: Few records in our dataset would have a negative value. This is because for our analysis we had used all the records of 'Quantity' attribute (which consist of a lot of negative values) from the 'OnlineRetail' Dataset. We used this value to get an accurate depiction of the current situation of the customers and products behavior, so the negative value would tell if the product is doing good or bad.

Justification for Outlier Removal in CustomerCluster Dataset:

The following **Customer IDs** have certain attributes, who's values are too large compared to the rest of the records in the dataset.

Outlier	**Attributes**
Customer ID= 0	• **PRODUCT_QUANTITY** • **DISTICNCT_PRODUCT** • **TOTAL_REVENUE** • **VISITS** • **PER_VISIT_COST**
Customer ID= 1464	• **PRODUCT_QUANTITY** • **TOTAL_REVENUE**

Justification for Outlier Removal in ProductCluster Dataset:

The following **StockCode's** have certain attributes, who's values either too large or too small compared to the rest of the records in the dataset.

Outlier	**Attributes**
StockCode= 47556B	• **QUANTITY_PER_CUSTOMER**
StockCode= 23005	• **QUANTITY_PER_CUSTOMER**
StockCode= 84568	• **QUANTITY_PER_CUSTOMER**
StockCode= DOT	• **QUANTITY_PER_CUSTOMER**

A. CustomerCluster Dataset- Outlier:

We plot the records in the **CustomerCluster** dataset using ggpair (as seen in the *Figure 1*). From the graph we can clearly see 2 records are significantly differ from the other observations, these records are namely:

- **Customer ID**= 0
- **Customer ID**= 14646

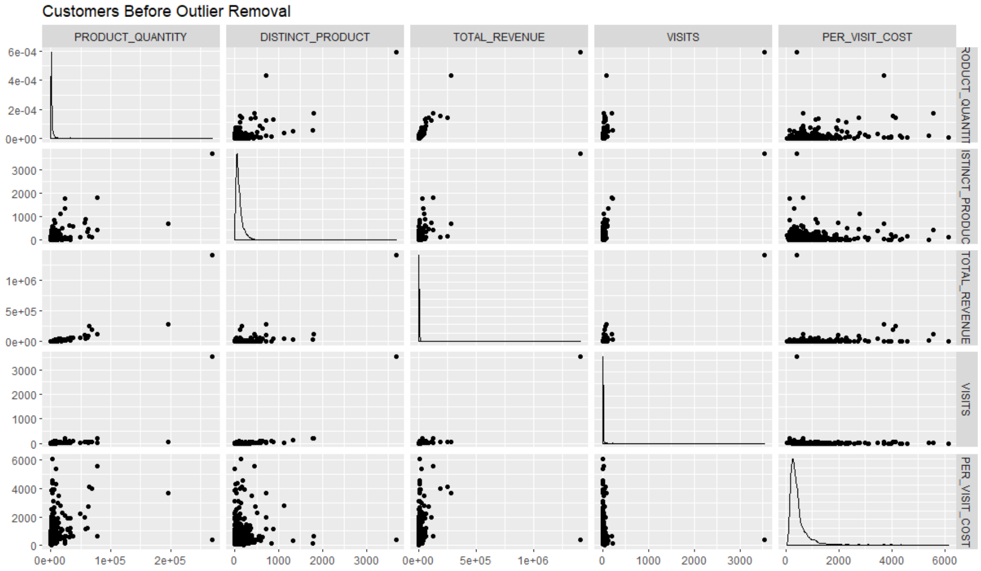


Figure 1: Customer Cluster ggpair Graph Before Removing the Outliers

On removing the outliers, we can clearly see in Figure 2 that the data is a bit more compact and all records are within a definite range.

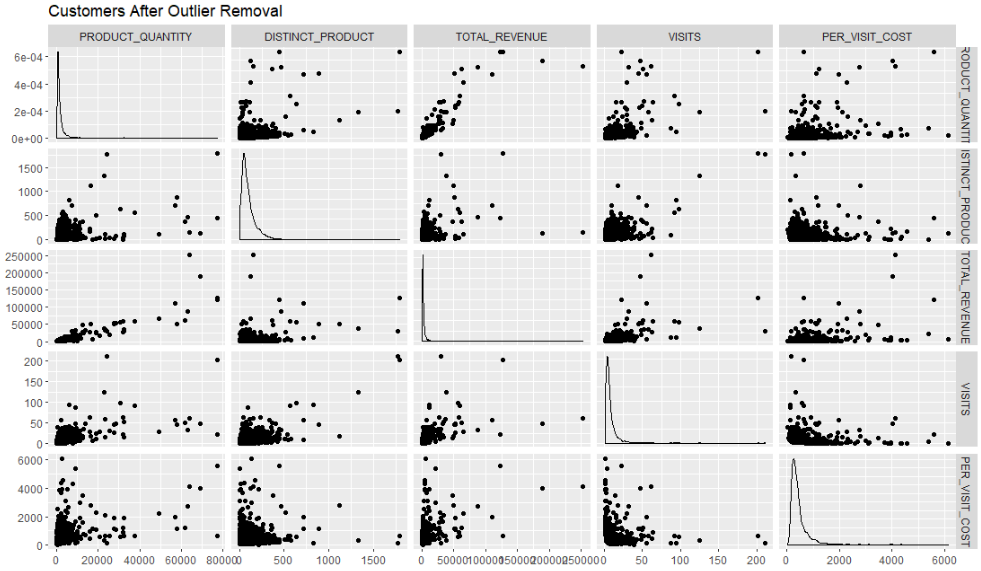


Figure 2: Customer Cluster ggpair Graph After Removing the Outliers

B. ProductCluster Dataset - Outlier:

We plot the records in the **ProductCluster** dataset using ggpair (as seen in the *Figure 3*). From the graph we can clearly see 4 records are significantly differ from the other observations, these records are namely:

- **StockCode**= 47556B   **StockCode**= 23005
- **StockCode**= 84568   **StockCode**= DOT

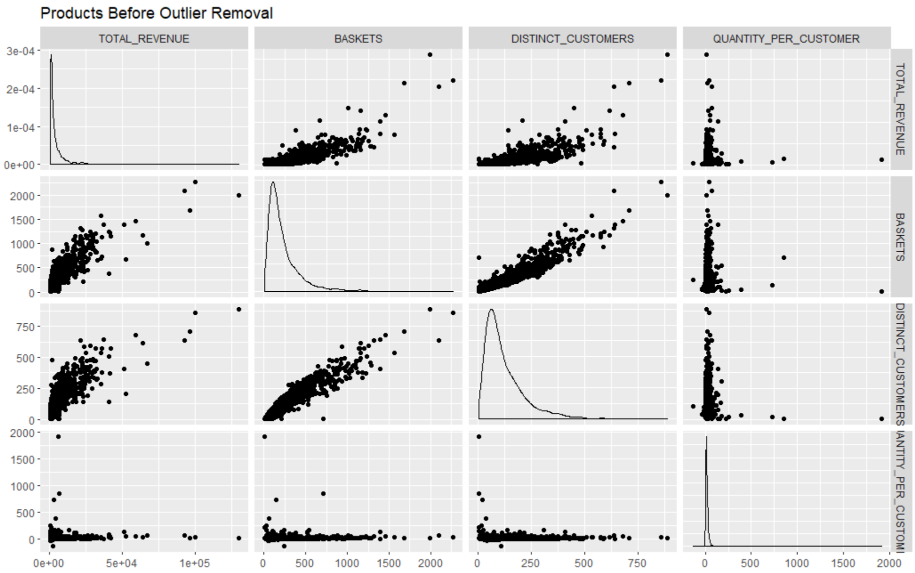


Figure 3: Product Cluster ggpair Graph Before Removing the Outliers

On removing the outliers, we can clearly see in Figure 4that the data is a bit more compact and all records are within a definite range.

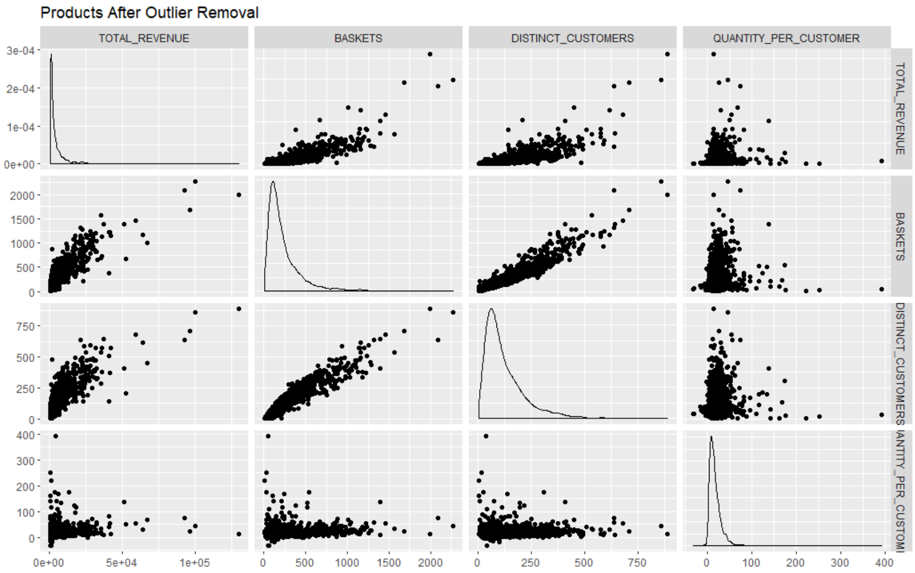


Figure 4: Product Cluster ggpair Graph After Removing the Outliers

Cluster Analysis

To identify how many clusters to use for the respective datasets we utilized the following methods:

1. **Elbow Method:** It validates and interprets the consistency within a cluster such that the total within cluster sum of square (WSS) is minimized. The number of clusters can be considered as a function of WSS. A cluster should then be chosen in such a way that adding another cluster doesn't improve the total WSS. (Kassambara, 2019)
2. **Gap Statistics:** This method calculates the distances between the clusters and plots a line graph. The minimum intercept of a point on y axis is considered as the best cluster which suggests that, the overlaps between the clusters are minimum. (Tibshirani, Walther and Hastie, 2001)

We combine and compare the results of the above method, to get a better estimate of the optimal number of clusters for the respective datasets.

1. Clusters for Customer:

A. Optimal Number of Clusters:

Elbow Method:

On using the *Elbow method* on the **CustomerCluster** Dataset, we get the optimal number of clusters as **6** as shown in the following Figure:

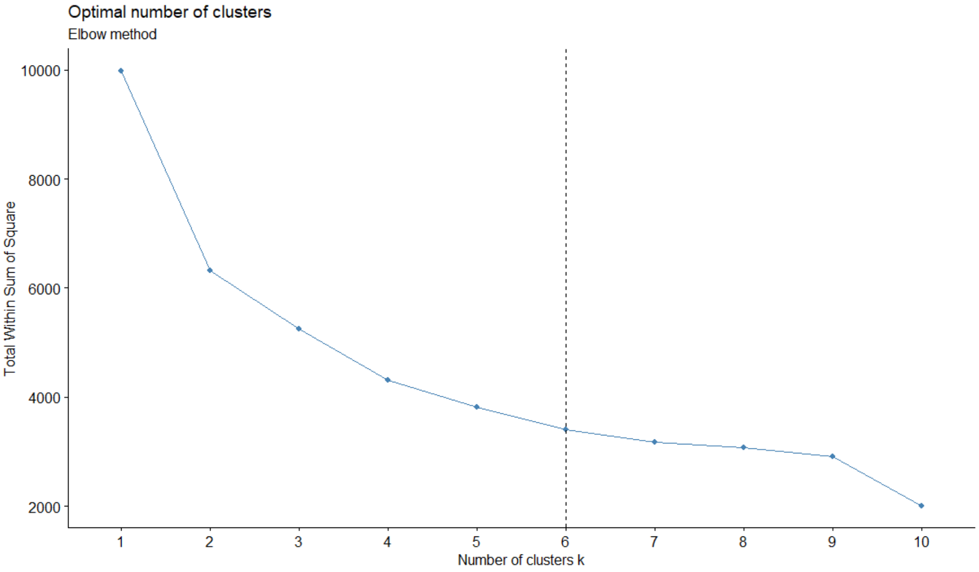


Figure 5: Elbow Method for CustomerCluster Dataset

Gap Statistics:

On using *Gap Statistics* on the **CustomerCluster** Dataset, we get the optimal number of clusters as **6** as shown in the following Figure:

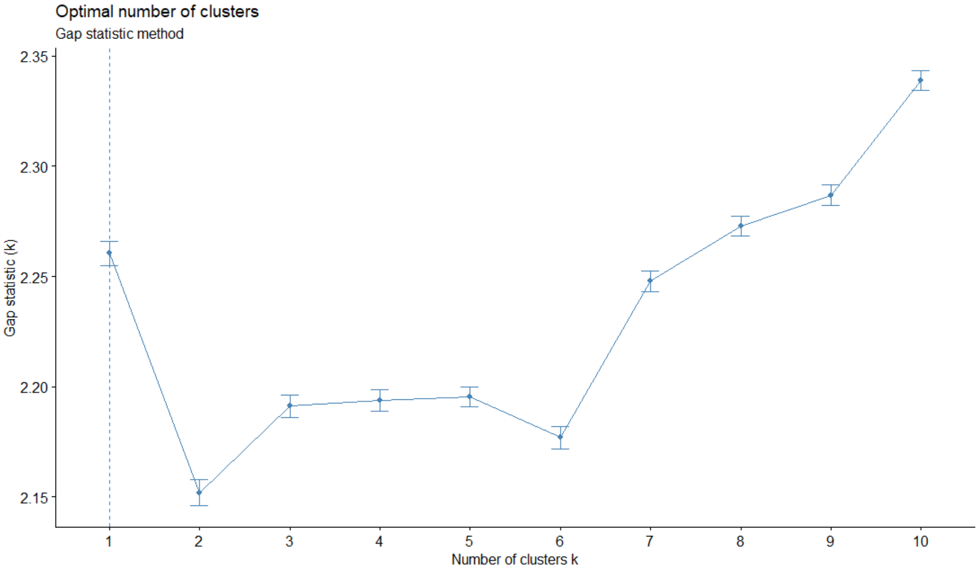


Figure 6: Gap Statistics Method for CustomerCluster Dataset

Combination:

On combining the results from the Elbow Method and Gap Statistics we can estimate that the optimal number of clusters to use for the **CustomerCluster** Dataset is **6**.

	Elbow Method	Gap Method
Optimal Number of Clusters	6	6

Table 5: Combination of Elbow Method and Gap Statistics for CustomerCluster

B. Plotting the Clusters:

The clusters are segregated according to the mentioned parameters\attributes given to the k means algorithm and the overlap is minimal for 6 clusters as observed in Figure 7. We plotted the clusters using 'fviz_cluster' utility in R (Kassambara, 2019), and we observe minimal overlap between the groups when the cluster count is 6, which was concluded using gap-statistic method.

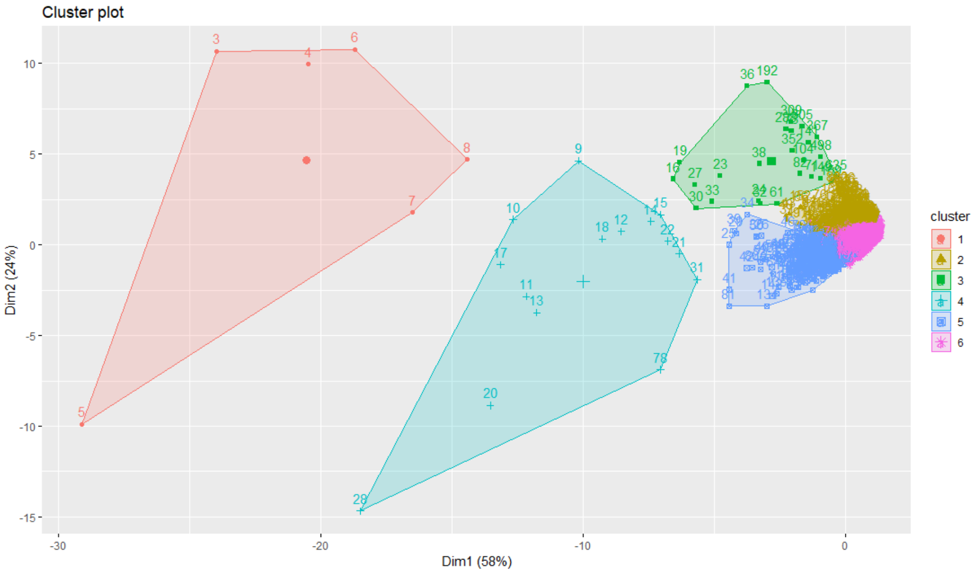


Figure 7: CustomerCluster Dataset-Cluster Plot using fviz_cluster Library

The scatter plot graph (Figure 8) is implemented on the **CustomerCluster** dataset for different attributes/ parameters, which resulted into a 5X5 matrix, depicting the relationships with respect to 2 individual parameters in each plot. We can observer districts colors for the different clusters.

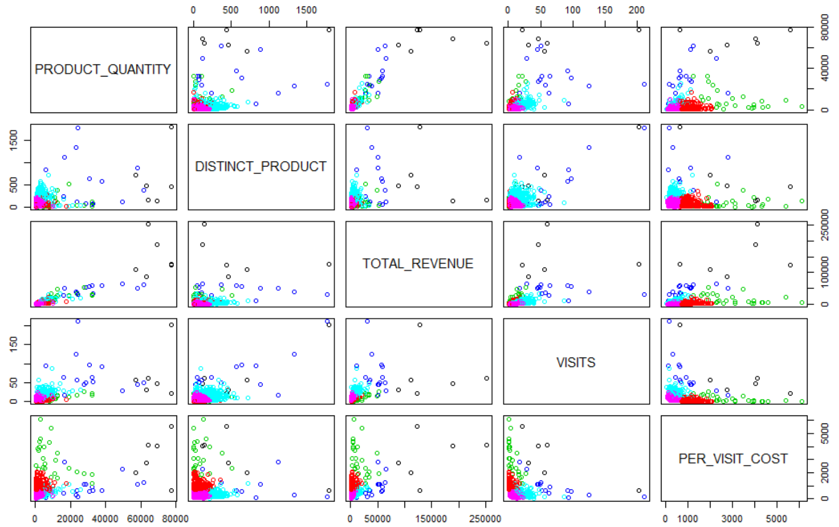


Figure 8: CustomerCluster Dataset Cluster Plot using ggpair Library

2. Clusters for Products:

A. Optimal Number of Clusters:

Elbow Method:

On using the *Elbow method* on the **ProductCluster** Dataset, we get the optimal number of clusters as **5** as shown in the following Figure:

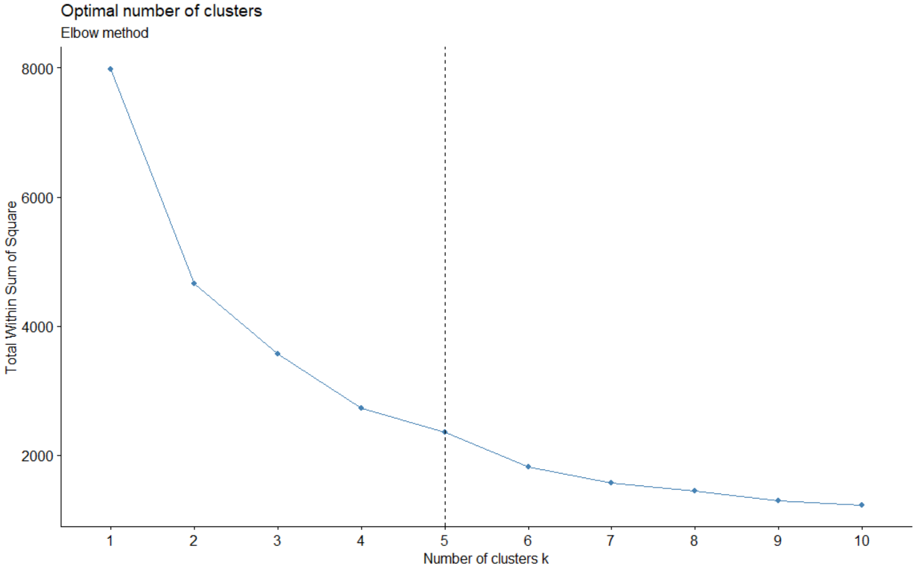


Figure 9: Elbow Method for ProductrCluster Dataset

Gap Statistics:

On using *Gap Statistics* on the **ProductCluster** Dataset, we get the optimal number of clusters as **5** as shown in the following Figure:

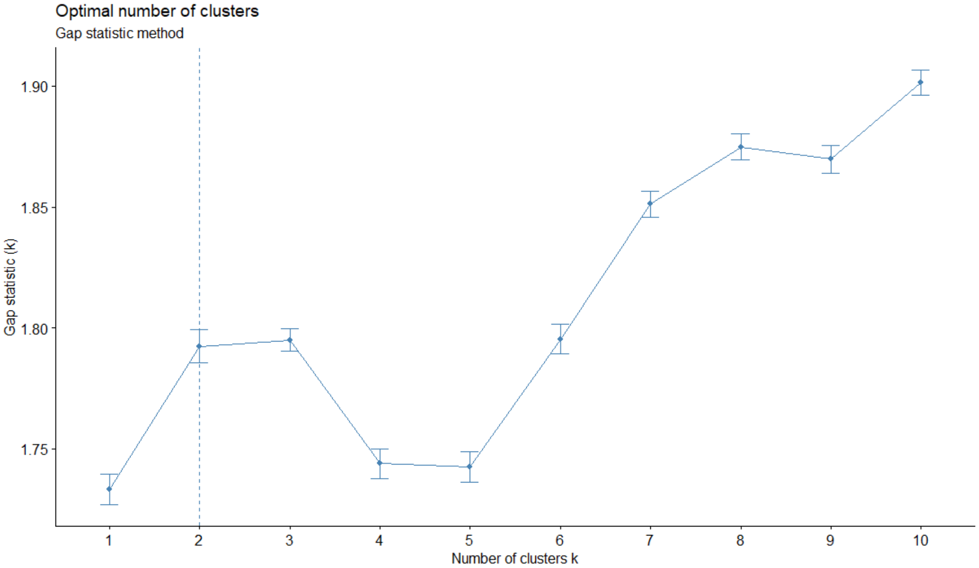


Figure 10: Gap Statistics Method for ProductrCluster Dataset

Combination:

On combining the results from the Elbow Method and Gap Statistics we can estimate that the optimal number of clusters to use for the **ProductCluster** Dataset is **5**.

	Elbow Method	Gap Method
Optimal Number of Clusters	5	5

Table 6: Combination of Elbow Method and Gap Statistics for ProductCluster

B. Plotting the Clusters:

The clusters are segregated according to the mentioned parameters\attributes given to the k means algorithm and the overlap is minimal for 5 clusters as observed in Figure 11.

We plotted the clusters using 'fviz_cluster' utility in R (Kassambara, 2019), and we observe minimal overlap between the groups when the cluster count is 5, which was concluded using gap- statistic method.

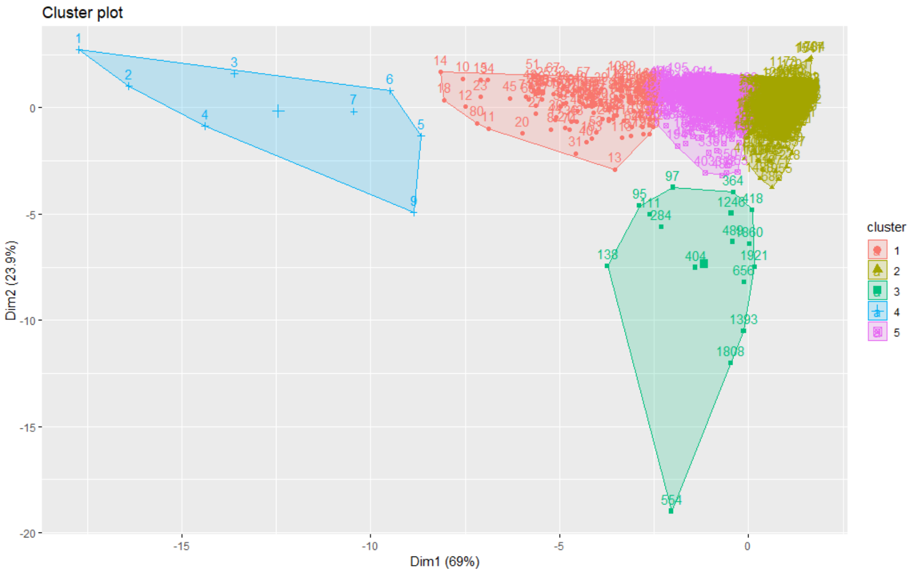


Figure 11: ProductCluster Dataset-Cluster Plot using fviz_cluster Library

The scatter plot graph (Figure 12) is implemented on the **ProductCluster** dataset for the different attributes, which resulted into a 4X4 matrix, depicting the relationships with respect to 2 individual parameters in each plot. We can observer districts colors for the different clusters.

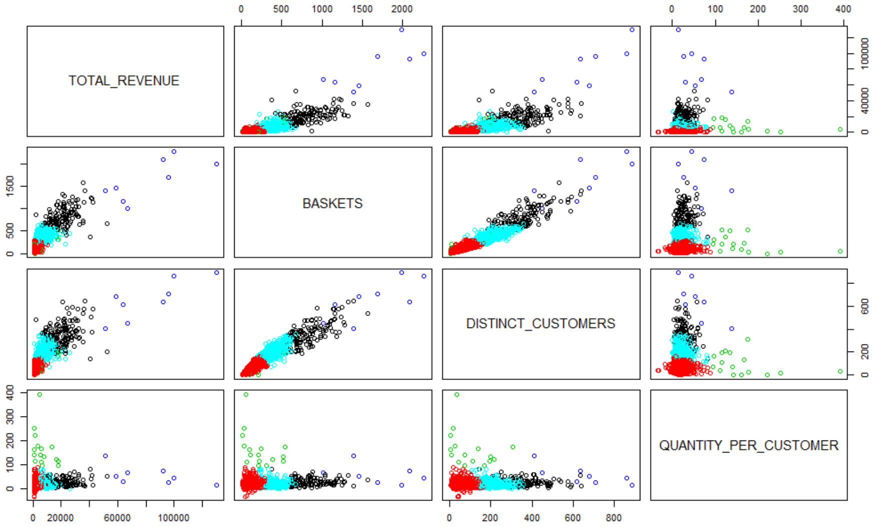


Figure 12: ProductCluster Dataset using ggpair Library

Cluster Profiling

1. Customer Clusters:

We begin by getting the count of the number of customers in each cluster. The following Figure shows the density of customers in each cluster. It shows that clusters 6, 5 and 2 has the highest count of customers.

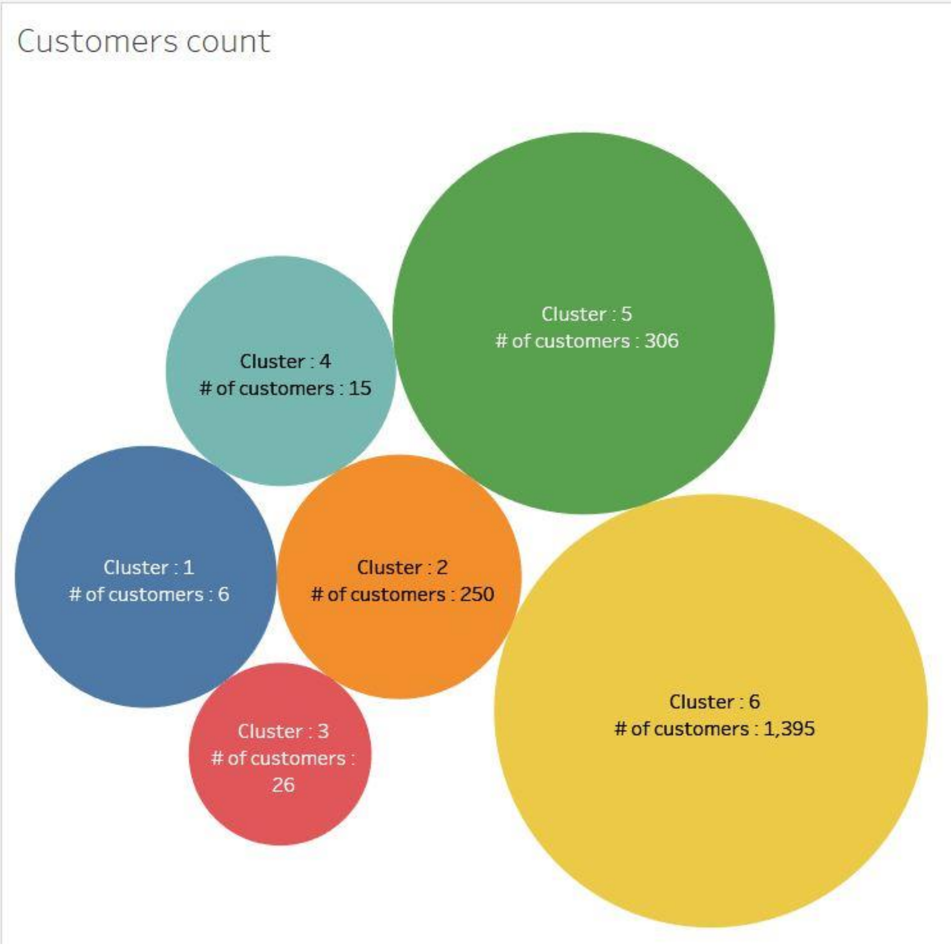


Figure 13: Count of Customers in the specific Customer Cluster

Then we checked the monthly visits of the customer in their respective clusters. The following Figure shows the line graph for the customers visits in each cluster. It shows that there is a steady growth for clusters 2,5 and 6.

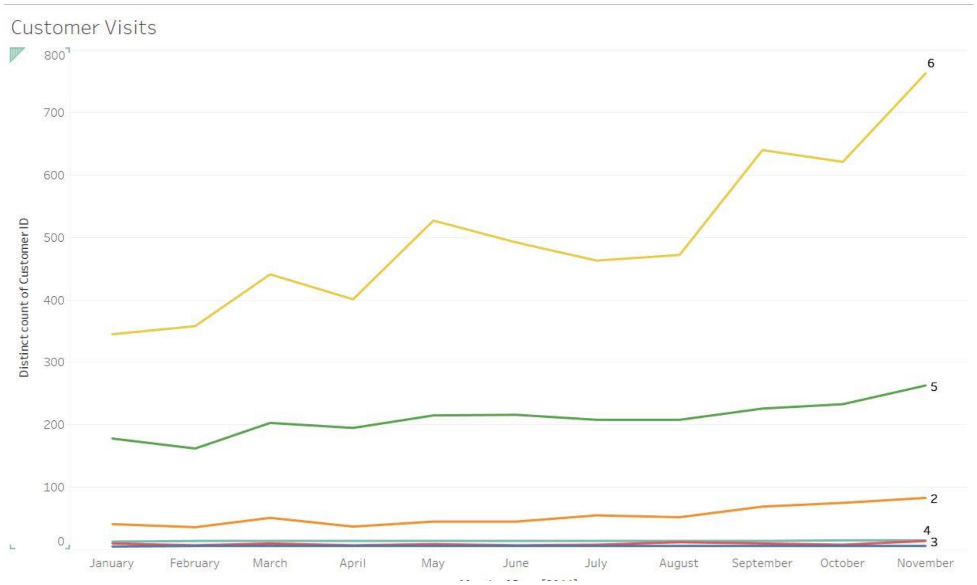


Figure 14: Count of Customers visits in the specific Customer Cluster

The below graph (Figure 15) depicts a more focused observation for cluster 1,3,4 while it shows, 6 customers from the cluster 1 were visiting the store consistently across the timeline. Cluster 4 has 14 customers visiting the store consistently across the timeline. Cluster 3 pattern is random, and the customer count varies negligibly.

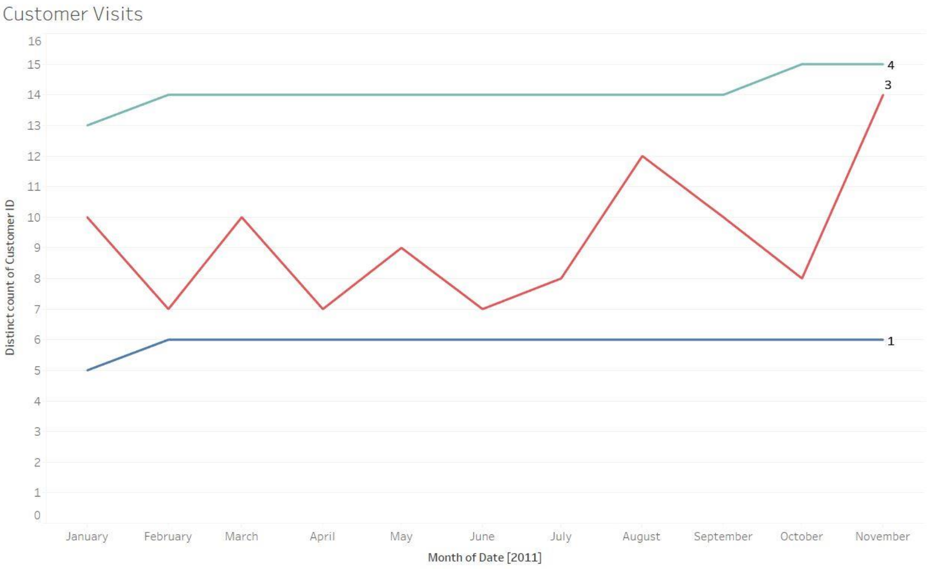


Figure 15: Count of Customers visits in Customer Cluster 1,3 and 4

Then we check the Revenue earned form products bought by each customer cluster. The below scatter plot is made with the X -axis as the count of the distinct products and Y-axis as the total revenue generated by the customer clusters. It is evident that the clusters 6 and 5 stood out when compares to the others in terms of the distinct products and the total revenue. One interesting insight was that the customers in the cluster 1 had contributed to a fair amount of revenue despite of the limited number of the distinct products they bought.

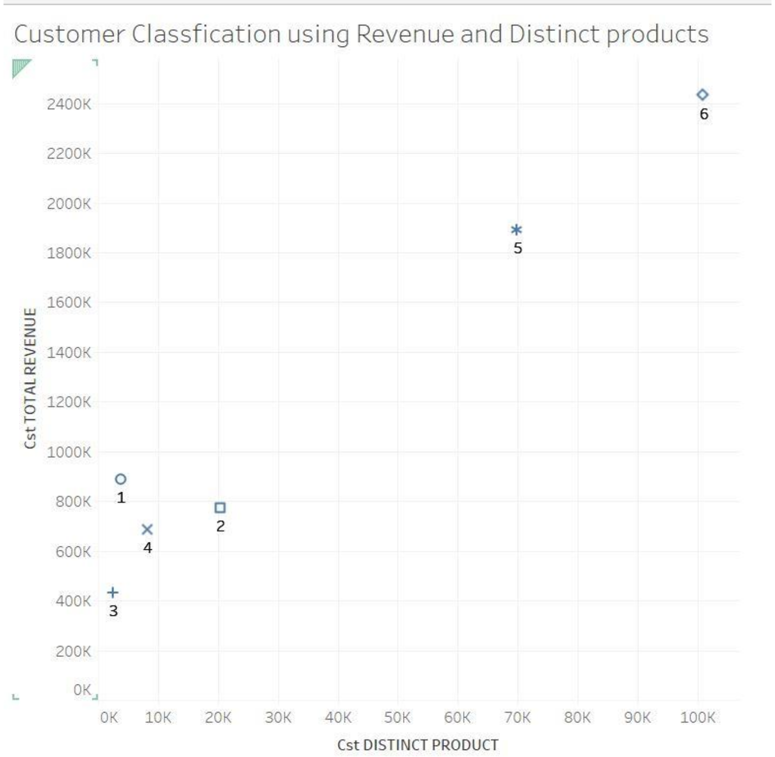


Figure 16: Revenue Earned from the specific Customer Cluster

Then we group the records of each cluster by attribute. By doing so we can get information pertaining to the number of distinct customers, product quantity, total revenue, total visits, average cost per visit, average visits per customer for each cluster.

Cluster	CUSTOMERS	TOTAL_REVENUE	PRODUCT_QUANTITY	DISTINCT_PRODUCT	VISITS	PER_VISIT_COST
1	6	889,291	407,610	3,699	420	3,180.03
2	250	775,011	479,564	20,315	809	985.61
3	26	432,649	253,199	2,407	193	3,067.59
4	15	687,468	467,643	8,202	1,095	937.41
5	306	1,892,377	1,116,877	69,837	4,994	393.25
6	1395	2,435,476	1,478,285	100,875	8,189	320.47
Total	1998	7,112272	4,203,178	205,335	15,700	8,884.35

Table 7: Customer Profile Based on Attributes

The following Table depicts same information as that in Table 7 but in in percentage format.

The records highlighted in green are the best 2 records of the attribute.

The records highlighted in orange are the intermediate 2 records of the attribute.

The records highlighted in red are the worst 2 records of the attribute.

PERCENTAGE						
Cluster	CUSTOMERS	TOTAL_REVENUE	PRODUCT_QUANTITY	DISTINCT_PRODUCT	VISITS	PER_VISIT_COST
Total	100.00	100.00	100.00	100.00	100.00	100.00
6	69.82	34.24	35.17	49.13	52.16	48.24
5	15.32	26.61	26.57	34.01	31.81	12.99
1	0.30	12.50	9.70	1.80	2.68	2.06
2	12.51	10.90	11.41	9.89	5.15	26.59
4	0.75	9.67	11.13	3.99	6.97	1.52
3	1.30	6.08	6.02	1.17	1.23	8.61

Table 8: Percentage of the Customer Profile Based on Attributes

We have performed further verification by plotting the bar charts of the sum and average of the various attribute in the Customer Cluster, which can be seen in Appendix – C

Analyzing the customer clusters, we have categorized them according to RFM model standardized by the industry and the insights describe how the society is responding to this business.

Cluster	Description	Recommendation
Cluster 6 – FMCG (Champions)	Highest revenue cluster, loyal customers. FMCG (fast moving consumer goods).	Monitor customers if the trend is increasing. Keep the stock ready.
Cluster 1 - Local Business/Vendors -> Heavy Purchasers & Buy in big quantities (Loyal Customers)	Increasing trend in visits. Retailers/small business owners. Average purchase cost per visit is high . Highest visits Sept, Oct, Nov.	Monitor this cluster items and make the inventory always available in huge quantities. Especially in Sept, Oct , Nov. Premium consideration since they are the 3rd highest contributors in terms of Revenue. 21733, 22469, 22470, 22492, 85123A are the "StockCode" items which are supposed to be in the inventory. 14911 has been a primary contributor to this cluster.
Cluster 5 – Related to FMCG - Wealthy customers purchasing household items. (Potential Loyalist)	Second highest revenue cluster, loyal customers, FMCG (fast moving consumer goods), household items. Per visit cost is slightly higher than cluster 6. Wealthy customers.	Monitor customers if the trend/volume is increasing.
Cluster 4 - Household items. (Promising)	Consistent increasing trend in visits. Consistent visitors, able to grab new customers. Every month activity is observed.	Monitor, attention required since they are bargaining customers. Offer them discounts and connect the items so that we can grab the customer to purchase more items.
Cluster 3 - Local business/ Vendors -> Needs attention (At Risk)	Poorly increasing trend in visits. Retailers/ small business owners. This cluster has high volumes of buying with less visits.	This cluster group resembles the primitive features of cluster 1. This group needs attention , because they have the potential to bring revenues as cluster 1.
Cluster 2 - Home Appliances -> Furniture (Can't Lose Them)	Increase Tread in visits. Home users, purchase things like furniture, which are single time purchase for home use. The average expenditure per visit is high.	Monitor this cluster items so that the first choice for the customer to check would be our store. Not required to have high stock in the inventory. Maintaining minimum stock would help in running capital to invest in other areas of business.

Table 9: Cluster Profiling for Customer Cluster Dataset

2. Products Clusters:

We begin by getting the count of the distinct number of products in each cluster. The following Figure shows the density of products in each cluster. It shows that clusters 2, 5 and 1 has the highest count of products.

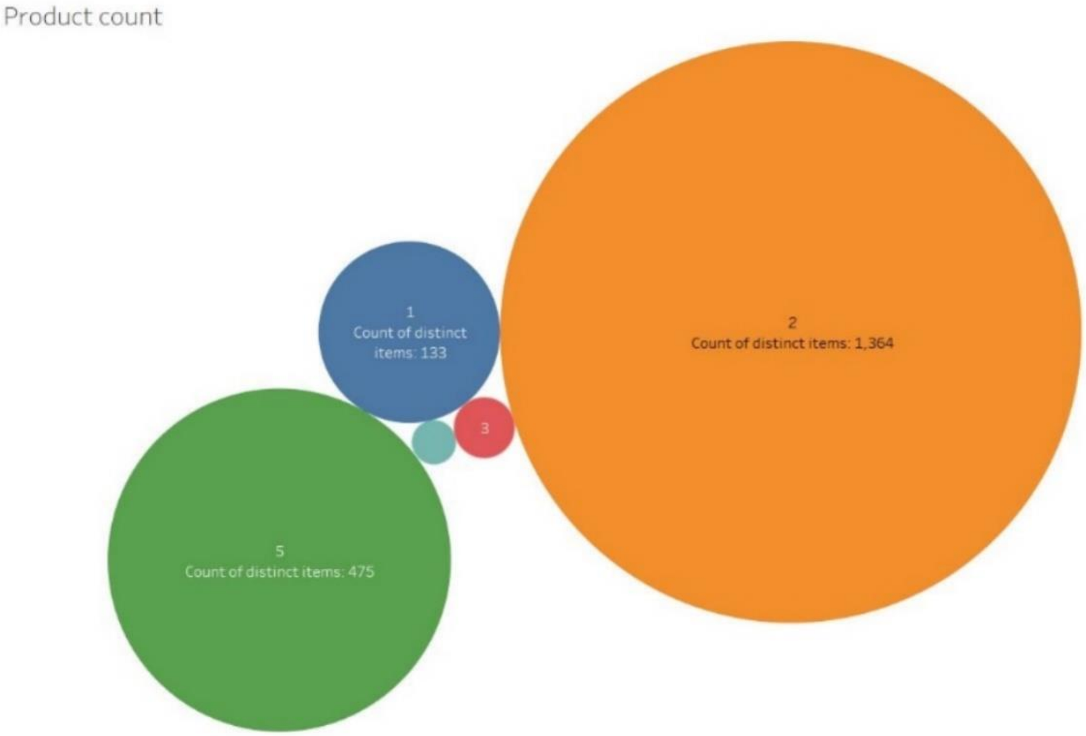


Figure 17: Count of Products in the specific Product Cluster

Then we checked the quantity of products brought per month for the respective clusters. The line graph of the various clusters of the products shows us the trend of the quantities sold over the months. It is evident from the graph that the products in the clusters 5,2 and 1 were in an increasing trend when compared to the rest (i.e. clusters 3,4) .

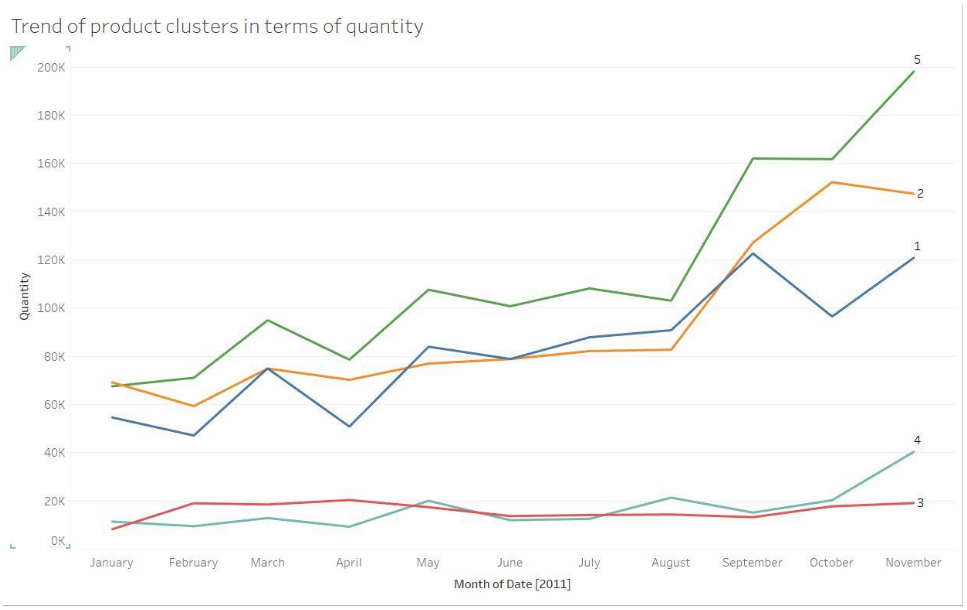


Figure 18: Count of Quantity of Products brought for the specific Product Cluster

From Figure 19 we can see that the amount of quantity being sold by the cluster 3 is fluctuating over the period. It shows trend/seasonality as it crossed the average in the first half of the year, but the data given (1 year) is not enough to conclude seasonality. Overall, the performance of the products in cluster 3 is poor as many products in the cluster were returned.

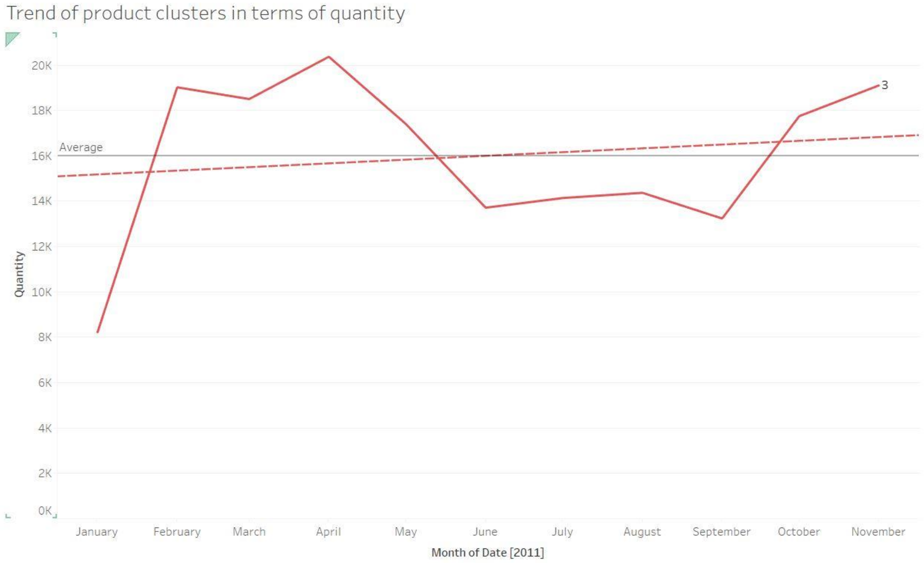


Figure 19: Count of Quantity of Products brought for the Product Cluster 3

Then we check the Revenue earned by the quantity of products sold from each product cluster. The below scatter plot is made with the X -axis as the count of the quantity of products and Y-axis as the total revenue generated by the product clusters. It clearly illustrates that the products present in the cluster 5 contributed the most to the revenue and had the huge quantity sold when compared to the other clusters. Even though cluster 3 and 4 were sold in the equal number of amounts, the revenue generated by the cluster 4 is more than double that of the cluster 3.

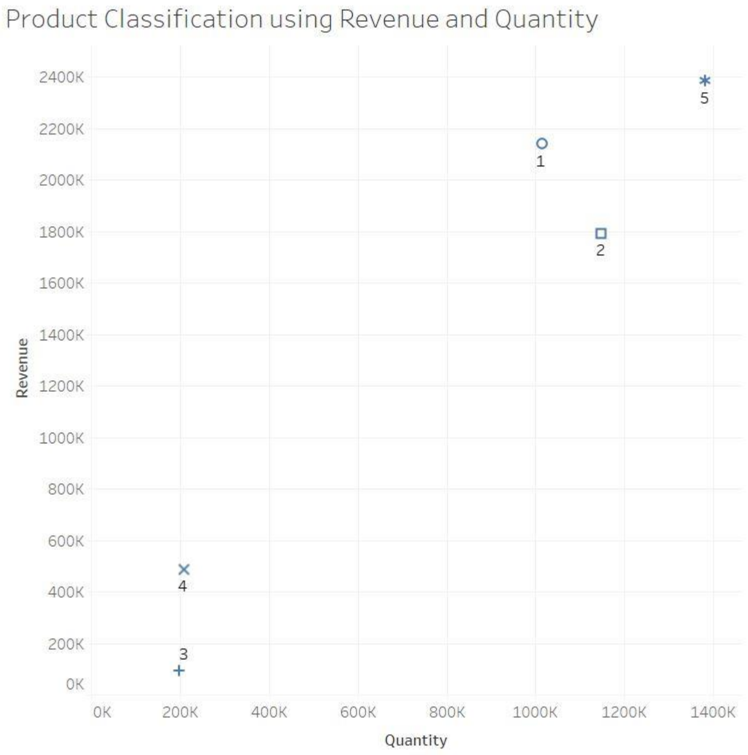


Figure 20: Revenue Earned from the specific Product Cluster

Then we group the records of each cluster by attribute. By doing so we can get information pertaining to the number of distinct products, total revenue, total quantity, distinct customer, quantity per customer for each cluster.

Cluster	ITEMS	TOTAL_REVENUE	BASKETS	DISTINCT_CUSTOMERS	QUANTITY_PER_CUSTOMER
1	133	2,820,651.2	109,524	479,18	3,628.00
2	1,364	2,472,623.59	175,639	97,773	20,247.41
3	16	110,433.88	3,509	1,598	2,621.97
4	8	658,601.75	13,066	5,248	450.20
5	475	3,141,217.34	172,827	92,813	9,000.14
Total	1,996	9,203,527.76	474,565	245,350	35,947.72

Table 10: Product Profile Based on Attributes

The following Table depicts same information as that in Table 10 but in in percentage format.

The records highlighted in green are the best 2 records of the attribute.

The records highlighted in orange are the intermediate 2 records of the attribute.

The records highlighted in red are the worst 2 records of the attribute.

PERSENTAGE					
Cluster	Items	TOTAL_REVENUE	BASKETS	DISTINCT_CUSTOMERS	QUANTITY_PER_CUSTOMER
Total	100	100	100	100	100
5	23.80	34.13	36.42	37.83	25.04
1	6.66	30.65	23.08	19.53	10.09
2	68.34	26.87	37.01	39.85	56.32
4	0.40	7.16	2.75	2.14	1.25
3	0.80	1.20	0.74	0.65	7.29

Table 11: Product Profile Based on Attributes

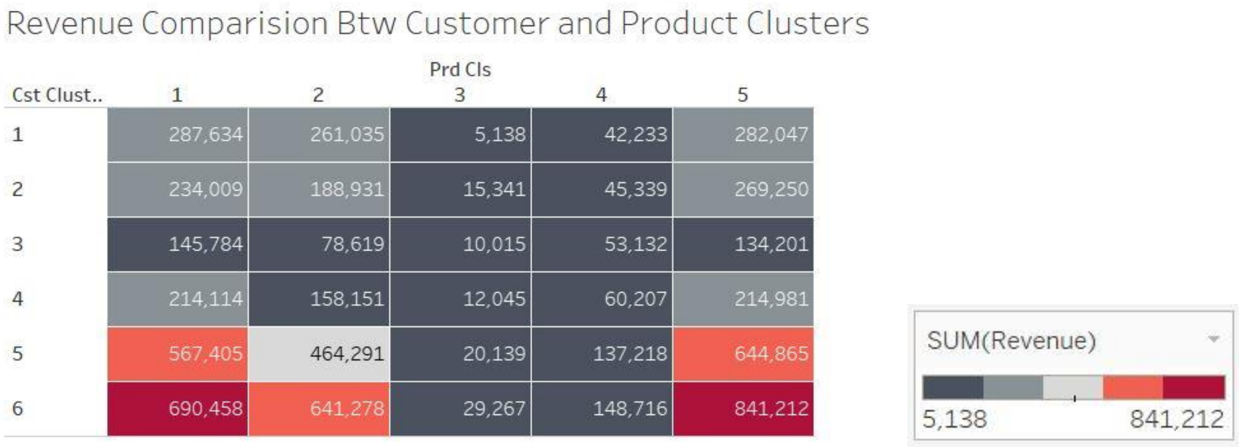
We have performed further verification by plotting the bar charts of the percentage of the average values of the various attribute in the Product Cluster, which can be seen in Appendix – D

Analyzing the Product clusters, we have categorized them, and the insights describe how the society is responding to the items provided.

Product	Description	Recommendations
Cluster 1 - Slim Cluster	Second highest revenue, 3rd highest quantity and these products contribute majority to the revenue	Monitor the availability of the inventory.
Cluster 5 - Champions	Highest revenue, Highest quantity, the products are more popular and are bringing a lot of revenue.	Best performing cluster, monitor and maintain the inventory timely.
Cluster 2 - Bulky Cluster	Second highest quantity, 3rd highest revenue. The items are bought regularly.	Monitor the availability of the inventory in large quantities.
Cluster 4 - Costly Items	These items have the potential to get new revenue into the business. The average unit price is the highest among all the clusters. These products are more popular. The items in this cluster are costlier. These items Generates more than double the revenue than cluster 3 but quantity is similar to cluster 3.	Attention needed as the trend for this item is increasing. Inventory must be increased as we observe increase in sale.
Cluster 3 - Poor Performing Items	Least revenue, least quantity, poorly performing cluster. These are the items which have negative quantities , which indicate that the items are returned frequently.	Attention needed, as the products in this cluster are not being sold consistently. Preferred limited or less inventory. Not gaining much profit to the business.

Table 12: Cluster Profiling for ProductCluster Dataset

Figure 21 concludes the revenue generated by the distinct customers according to their categories with respective to product clusters. The involvement of every customer in different product category defines which products are producers of revenue. The heat map mentioned demonstrates aptly the highs and lows with respective to the revenue.



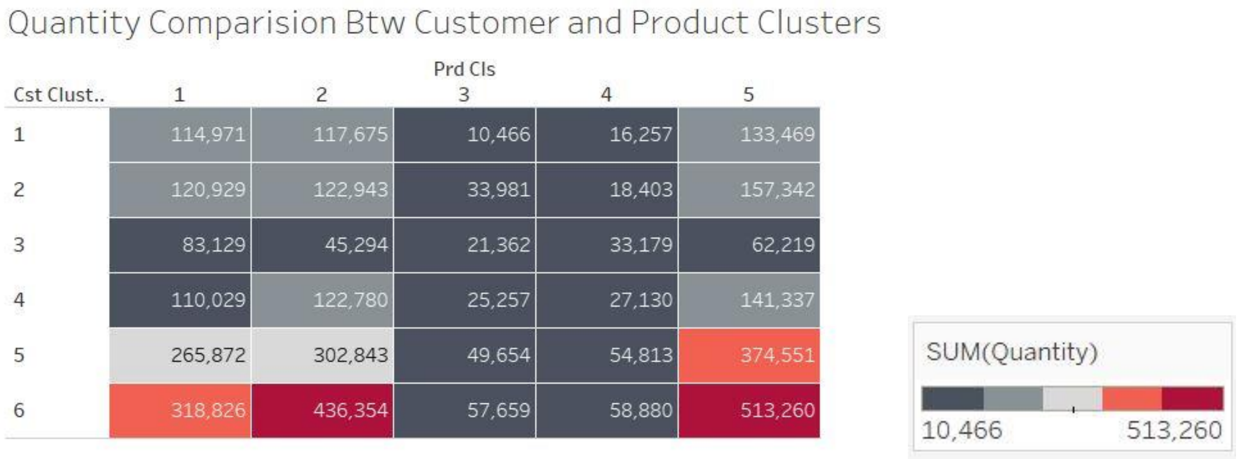
Revenue Comparision Btw Customer and Product Clusters

	Prd Cls				
Cst Clust..	1	2	3	4	5
1	287,634	261,035	5,138	42,233	282,047
2	234,009	188,931	15,341	45,339	269,250
3	145,784	78,619	10,015	53,132	134,201
4	214,114	158,151	12,045	60,207	214,981
5	567,405	464,291	20,139	137,218	644,865
6	690,458	641,278	29,267	148,716	841,212

SUM(Revenue)
5,138
841,212

Figure 21: Revenue Comparison Between Customer and Product Cluster

Figure 22 defines the quantity comparison for each cluster of the product with respect to each cluster of the customer purchases. An individual cell defines the popularity of the products where it is laid out in the format of a heat map.



Quantity Comparision Btw Customer and Product Clusters

	Prd Cls				
Cst Clust..	1	2	3	4	5
1	114,971	117,675	10,466	16,257	133,469
2	120,929	122,943	33,981	18,403	157,342
3	83,129	45,294	21,362	33,179	62,219
4	110,029	122,780	25,257	27,130	141,337
5	265,872	302,843	49,654	54,813	374,551
6	318,826	436,354	57,659	58,880	513,260

SUM(Quantity)
10,466
513,260

Figure 22: Quantity Comparison Between Customer and Product Cluster

Conclusion

We analyzed different clusters by considering from the perspective of items and customers. This allowed us to segregate different types of customers and identify them to RFM model. The same procedure is applied on items. We identified the best of the customers and items which are driving the business. The recommendations and conclusions for each cluster are provided to improve the business and further plan for profitable business strategy.

References:

Brownlee, J. (2019). An Introduction to Feature Selection. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/an-introduction-to-feature-selection/ [Accessed 25 May 2019].

Garbade, M. (2019). Understanding K-means Clustering in Machine Learning. [online] Towards Data Science. Available at: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1 [Accessed 24 May 2019].

Kassambara, A. (2019). Determining The Optimal Number Of Clusters: 3 Must Know Methods - Datanovia. [online] Datanovia. Available at: https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/ [Accessed 24 May 2019].

Nair, A. (2019). RFM Analysis For Successful Customer Segmentation - Putler. [online] Putler. Available at: https://www.putler.com/rfm-analysis/ [Accessed 25 May 2019].

Tibshirani, R., Walther, G. and Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), pp.411-423.

Appendix:

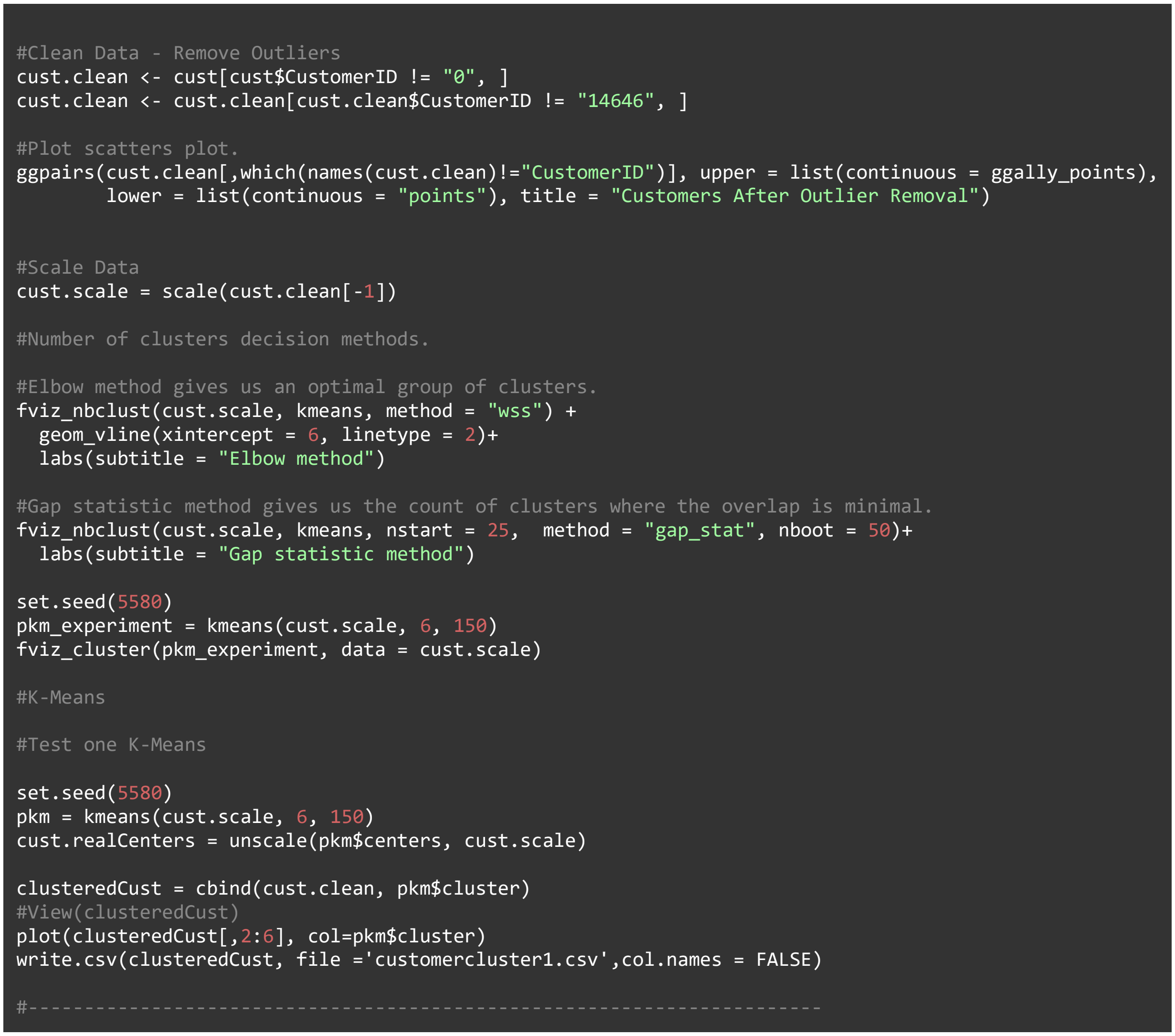
Appendix – A:

```sql
-- PRODUCT_QUANTITY is the Sum of Quantities, it will adjust the aggregation of the amount of product
brought and returned by the customer.
-- DISTINCT_PRODUCT is the count of unique product brought by the customer
-- TOTAL_REVENUE is the Total amount spent by the customer, it’s the product of the UnitPrice and Quantity
-- InvoiceNo is the count of unique invoices, it can give the number of visits made by the Customer
-- PER_VISIT_COST is to remove the customer whose purchase quantity is either negative or zero

SELECT
`CustomerID`,
SUM(`Quantity`) AS PRODUCT_QUANTITY,
COUNT(distinct `StockCode`) AS DISTINCT_PRODUCT,
SUM(`UnitPrice` * `Quantity`) as TOTAL_REVENUE,
COUNT(distinct `InvoiceNo`) as VISITS,
SUM(`UnitPrice` * `Quantity`) / COUNT(distinct `InvoiceNo`) as PER_VISIT_COST
FROM `OnlineRetail`
GROUP BY CustomerID
ORDER BY TOTAL_REVENUE DESC
LIMIT 2000
```

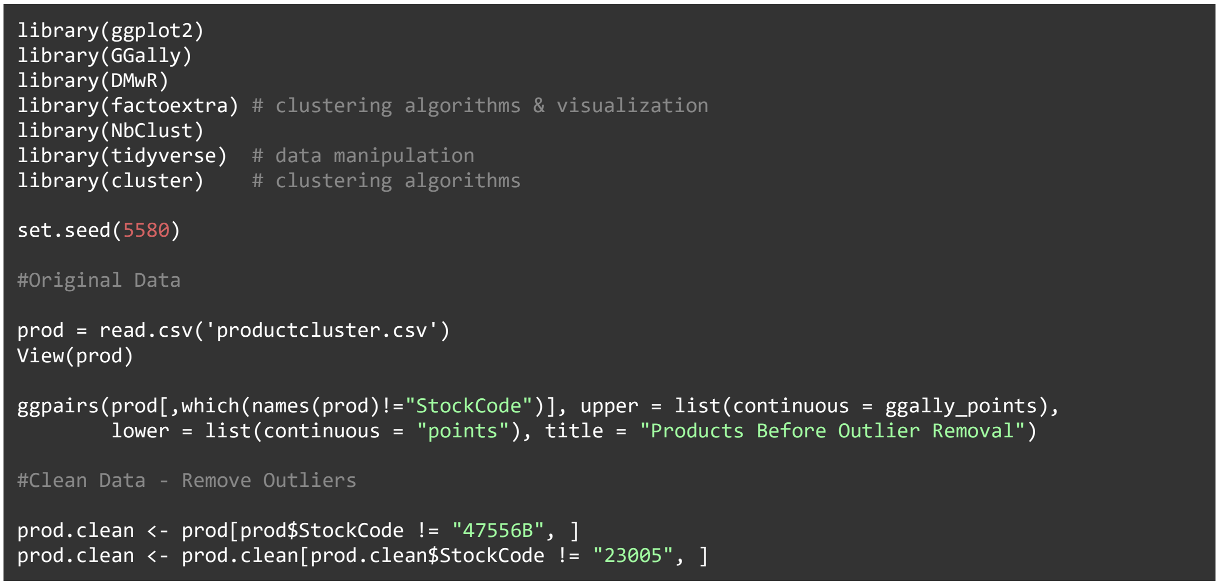
```sql
-- The Stock code is the product item’s unique number.
-- TOTAL_REVENUE is the revenue an item has generated.
-- BASKETS is the number of times the items are purchased.
-- DISTINCT_CUSTOMERS identifies the unique customers who purchased that particular item.
-- QUANTITY_PER_CUSTOMER identifies the average quantity considered by a customer.

SELECT
`StockCode`,
SUM(`UnitPrice` * `Quantity`) as TOTAL_REVENUE,
COUNT(distinct `InvoiceNo`) as BASKETS,
COUNT(distinct `CustomerID`) as DISTINCT_CUSTOMERS,
ROUND(SUM(Quantity)/COUNT(distinct `CustomerID`), 2) as QUANTITY_PER_CUSTOMER
FROM `OnlineRetail`
GROUP BY StockCode
ORDER BY TOTAL_REVENUE DESC
LIMIT 2000
```

Appendix – B:

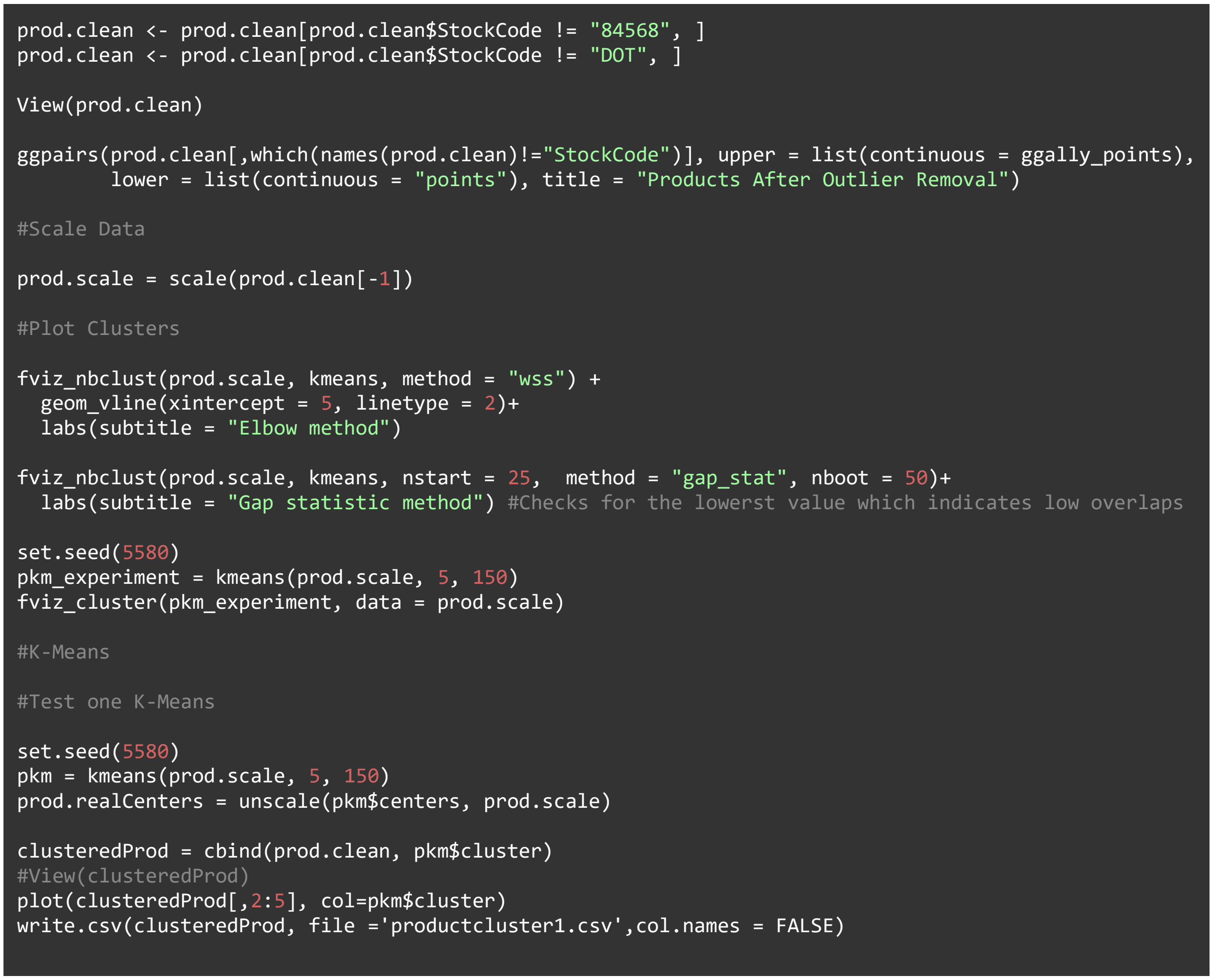
```r
library(ggplot2)
library(GGally)
library(DMwR)
library(factoextra) # clustering algorithms & visualization
library(NbClust)
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms

set.seed(5580)

#Original Data

cust = read.csv('customercluster.csv')
View(cust)

#Plot scatters plot.
ggpairs(cust[,which(names(cust)!="CustomerID")], upper = list(continuous = ggally_points),
        lower = list(continuous = "points"), title = "Customers Before Outlier Removal")
```



```
#Clean Data - Remove Outliers
cust.clean <- cust[cust$CustomerID != "0", ]
cust.clean <- cust.clean[cust.clean$CustomerID != "14646", ]

#Plot scatters plot.
ggpairs(cust.clean[,which(names(cust.clean)!="CustomerID")], upper = list(continuous = ggally_points),
        lower = list(continuous = "points"), title = "Customers After Outlier Removal")


#Scale Data
cust.scale = scale(cust.clean[-1])

#Number of clusters decision methods.

#Elbow method gives us an optimal group of clusters.
fviz_nbclust(cust.scale, kmeans, method = "wss") +
  geom_vline(xintercept = 6, linetype = 2)+
  labs(subtitle = "Elbow method")

#Gap statistic method gives us the count of clusters where the overlap is minimal.
fviz_nbclust(cust.scale, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")

set.seed(5580)
pkm_experiment = kmeans(cust.scale, 6, 150)
fviz_cluster(pkm_experiment, data = cust.scale)

#K-Means

#Test one K-Means

set.seed(5580)
pkm = kmeans(cust.scale, 6, 150)
cust.realCenters = unscale(pkm$centers, cust.scale)

clusteredCust = cbind(cust.clean, pkm$cluster)
#View(clusteredCust)
plot(clusteredCust[,2:6], col=pkm$cluster)
write.csv(clusteredCust, file ='customercluster1.csv',col.names = FALSE)

#------------------------------------------------------------------------
```

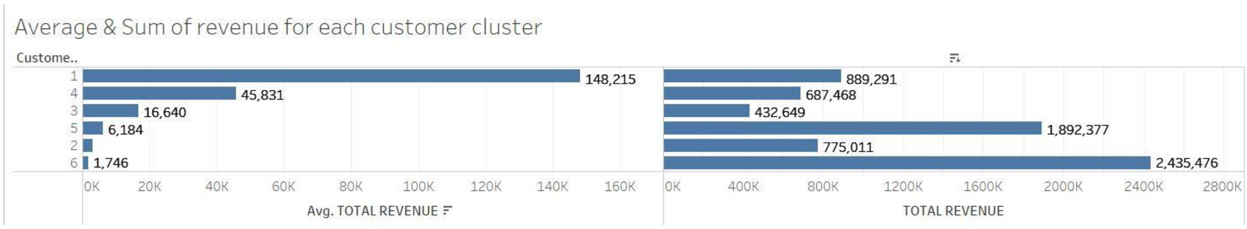


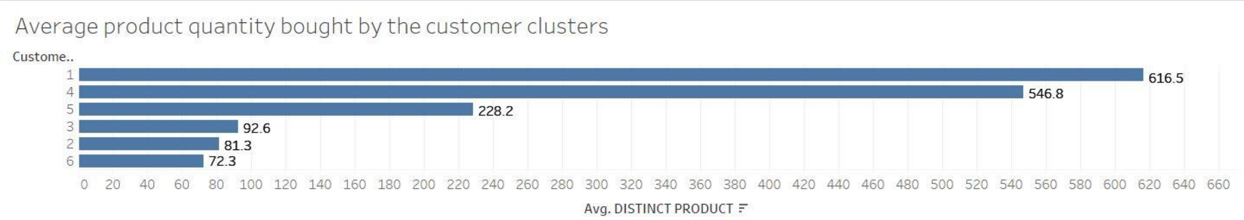
```
library(ggplot2)
library(GGally)
library(DMwR)
library(factoextra) # clustering algorithms & visualization
library(NbClust)
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms

set.seed(5580)

#Original Data

prod = read.csv('productcluster.csv')
View(prod)

ggpairs(prod[,which(names(prod)!="StockCode")], upper = list(continuous = ggally_points),
        lower = list(continuous = "points"), title = "Products Before Outlier Removal")

#Clean Data - Remove Outliers

prod.clean <- prod[prod$StockCode != "47556B", ]
prod.clean <- prod.clean[prod.clean$StockCode != "23005", ]
```

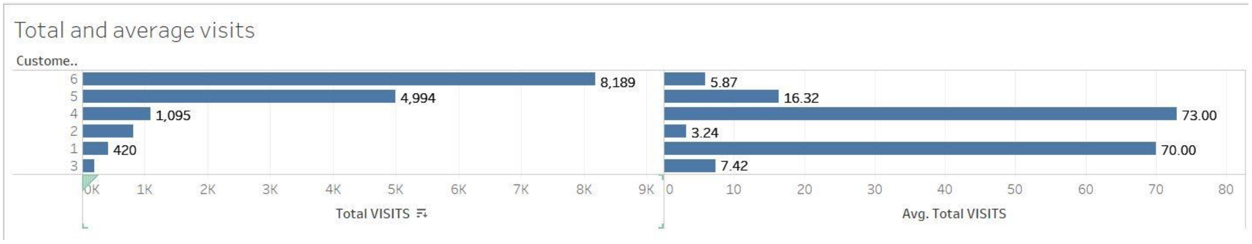


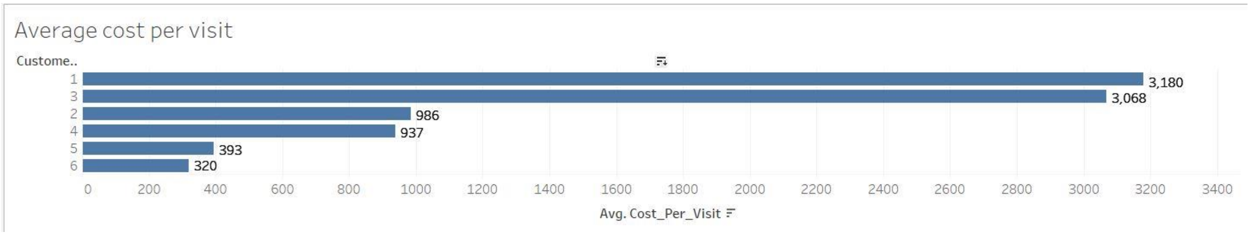
```
prod.clean <- prod.clean[prod.clean$StockCode != "84568", ]
prod.clean <- prod.clean[prod.clean$StockCode != "DOT", ]

View(prod.clean)

ggpairs(prod.clean[,which(names(prod.clean)!="StockCode")], upper = list(continuous = ggally_points),
        lower = list(continuous = "points"), title = "Products After Outlier Removal")

#Scale Data

prod.scale = scale(prod.clean[-1])

#Plot Clusters

fviz_nbclust(prod.scale, kmeans, method = "wss") +
  geom_vline(xintercept = 5, linetype = 2)+
  labs(subtitle = "Elbow method")

fviz_nbclust(prod.scale, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method") #Checks for the lowerst value which indicates low overlaps

set.seed(5580)
pkm_experiment = kmeans(prod.scale, 5, 150)
fviz_cluster(pkm_experiment, data = prod.scale)

#K-Means

#Test one K-Means

set.seed(5580)
pkm = kmeans(prod.scale, 5, 150)
prod.realCenters = unscale(pkm$centers, prod.scale)

clusteredProd = cbind(prod.clean, pkm$cluster)
#View(clusteredProd)
plot(clusteredProd[,2:5], col=pkm$cluster)
write.csv(clusteredProd, file ='productcluster1.csv',col.names = FALSE)
```

Appendix – C:

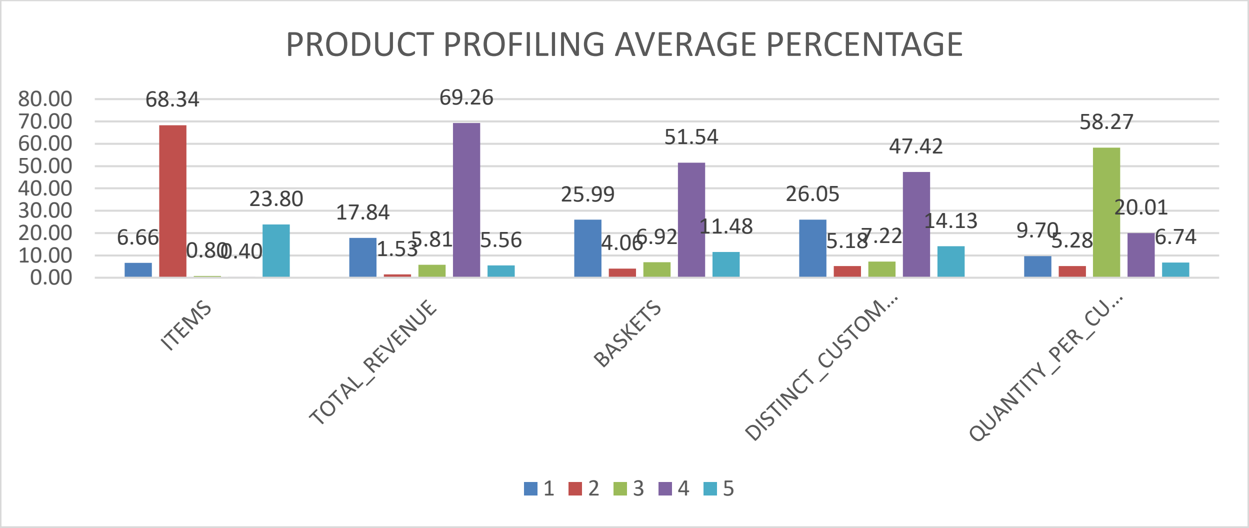





Total and average visits
Custome..
6
5
4
2
1
3
8,189
4,994
1,095
420
0K
1K
2K
3K
4K
5K
6K
7K
8K
9K
Total VISITS
5.87
16.32
73.00
3.24
70.00
7.42
0
10
20
30
40
50
60
70
80
Avg. Total VISITS


Average cost per visit
Custome..
1
3
2
4
5
6
3,180
3,068
986
937
393
320
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
3000
3200
3400
Avg. Cost_Per_Visit

Appendix – D:


PRODUCT PROFILING AVERAGE PERCENTAGE
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
6.66
68.34
0.80
0.40
23.80
17.84
1.53
5.81
69.26
5.56
25.99
4.06
6.92
51.54
11.48
26.05
5.18
7.22
47.42
14.13
9.70
5.28
58.27
20.01
6.74
ITEMS
TOTAL_REVENUE
BASKETS
DISTINCT_CUSTOM...
QUANTITY_PER_CU...
1
2
3
4
5