SAINT MARY'S
UNIVERSITY SINCE 1802

One University. One World. Yours.

MCDA 5580 DATA & TEXT MINING

Assignment 1

Clustering of Products & Customers

Team Members

Names	A#
Chacko, Jelson	A00446838
Kuzhippallil, Maria A.	A00442283
Muotoe, Somto J.	A00442756
Reginold, Fabian Vaniyamveetil	A00447210

Table of Contents

Executive Summary 4

Objective 5

About the Data 5

Design/Methodology/Approach 7

- Data Extraction 8
- Data Preprocessing 8
- Hyperparameter tuning 9
- Model Building 10

Feature Selection, Feature Engineering and Feature Definition 10

Data Cleaning and Outlier Removal 11

- Data cleaning and Outlier for Online Retail Data 12
- Data cleaning after aggregation (Customer) 13
- Data cleaning after aggregation (Product) 14

Cluster Analysis 15

Cluster Profiling 18

- Customer Clusters Overview 18
  - Customer Cluster Summary 18
- Product Cluster Overview 21
- Product Cluster Summary 21

Conclusion and Next Steps 23

Appendix-A [References] 24

Appendix-B [Aggregation Queries run in R] 25

- Product Aggregation 25
- Customer Aggregation 25

Appendix-C [R Code] 26

- Product Analysis 26
- Customer Analysis 28

List of Figures

Figure 1. Design methodology 7
Figure 2. Boxplots explained. 8
Figure 3. Boxplot before outlier removal for online retail data 12
Figure 4. Boxplot after outlier removal for online retail data 12
Figure 5. Boxplot before outlier removal (Customer) 13
Figure 6. Boxplot after outlier removal (Customer) 13
Figure 9. Boxplot before outlier removal (Product) 14
Figure 10. Boxplot after outlier removal (Product) 14
Figure 11. Optimal number of clusters using Elbow method (Customer) 15
Figure 12. Optimal number of clusters using Gap Stat method (Customer) 16
Figure 13. Optimal number of clusters using Elbow method (Product) 16
Figure 14. Optimal number of clusters using Gap Stat method (Product) 17
Figure 15. Customer clusters matrix 18
Figure 16. Number of Visits (Customer) Figure 17. Revenue (Customer) 18
Figure 18. Number of products (Customer) Figure 19. Number of distinct products (Customer) 18
Figure 20. Average amount spent by customer (Customer) 19
Figure 21. Product clusters matrix 21
Figure 22. Number of distinct customers (Product) Figure 23. Revenue (Product) 21
Figure 24. Total units sold (Product) Figure 25. Number of visits (Product) 21

Executive Summary

To keep pace with the need to analyze ever large volumes of data and emerging markets, every company is now focusing on understanding the customer's needs and their purchase patterns. With the help of k-means algorithm that runs through the data comprising of 500k records for an online retail store, customers and products were segmented into 4 different segments effectively. Customers and products with similar patterns and characteristics are categorized together which helps marketers to be more efficient and business leaders to glean insights to drive business decisions. Furthermore, we have given recommendations on how to optimize each of these categories to boost revenue. A summary of the respective segments is given below:

Customer Segments

Customer Segment	**# of Customers**	**Description**
Loyal (so I deserve a reward)	697	Customers who expect returns in some form due to their loyalty
Occasional consumers. (I'll buy if the deal seems right to me)	1544	Customers who purchase occasionally from the same store, taking into consideration the "offers" available
Medium Buyers (You've caught my attention, I'm listening)	740	Customers who frequently shop and are attracted by the offers provided
Extremely Loyal (has no effect on marketing influence)	370	Customers who are bound to shop from this same online retail stores over other retail stores due to their loyalty

Product Segments

Product Segment	**# of Products**	**Description**
Unsought products	1665	Products that are rarely glanced at by the customers
Branded products	295	Products that are valued high due to their brand

Less frequently purchased	775	Products rarely purchased
Frequently purchased products	513	Products purchased frequently

Objective

With the introduction of Industry 4.0, Artificial Intelligence and Machine Learning are the driving force of any retail industry. The aim of this analysis is to satisfy the business requirement of an online store, that is, to analyze systematically patterns and structures of customer's purchasing patterns and divide the products into approachable subsets and groups before going deep into the nuts-and-bolts analysis. After cleansing customer's transactional data followed by data exploration, data profiling, feature selection and unsupervised analysis, the store intends to enhance the company's marketing strategy and achieve significant revenue growth.

One of the goals to segment customers is to identify potential key customers and creating value for the targeted customers. Getting to know more about the key customers and their trends of purchasing the products online, would help in targeting them to increase the revenue of the business. On similar lines, segmentation of products that have been sold would help in determining potential marketing opportunities; improving and designing innovative products and liquidate slow-moving inventory. With the help of this strong and efficient analysis, business leaders can now make quick, informed, and accurate decisions with better intelligent course of action.

About the Data

Transactions made by customers in the store between 01/12/2010 to 09/12/2011 were recorded, organized, and formatted in a structured table. This data is evidently vital in understanding the customer's buying habits for the corresponding products they purchase. The summary of the data provided is as follows:

Number of Records	**Unique Records**	**Attributes/Fields**
536500	536500	9

The features present in the data allowing with their brief description is given as follows:

Attribute	Description
InvoiceNo	A 5-digit integral number uniquely assigned to each transaction.
StockCode	A 5-digit integral number uniquely assigned to each distinct product.
Description	Name of the Product along with certain prominent features of the product described in a word or two
Quantity	The quantities of each product (item) that is sold for in a transaction.
InvoiceDateTime	The date and time when each transaction was generated.
UnitPrice	The price of the product per unit
CustomerID	A 5-digit integral number uniquely assigned to each distinct customer.
Country	The name of the country where the customer resides.
InvoiceDate	The date on which each transaction was generated.

Design/Methodology/Approach

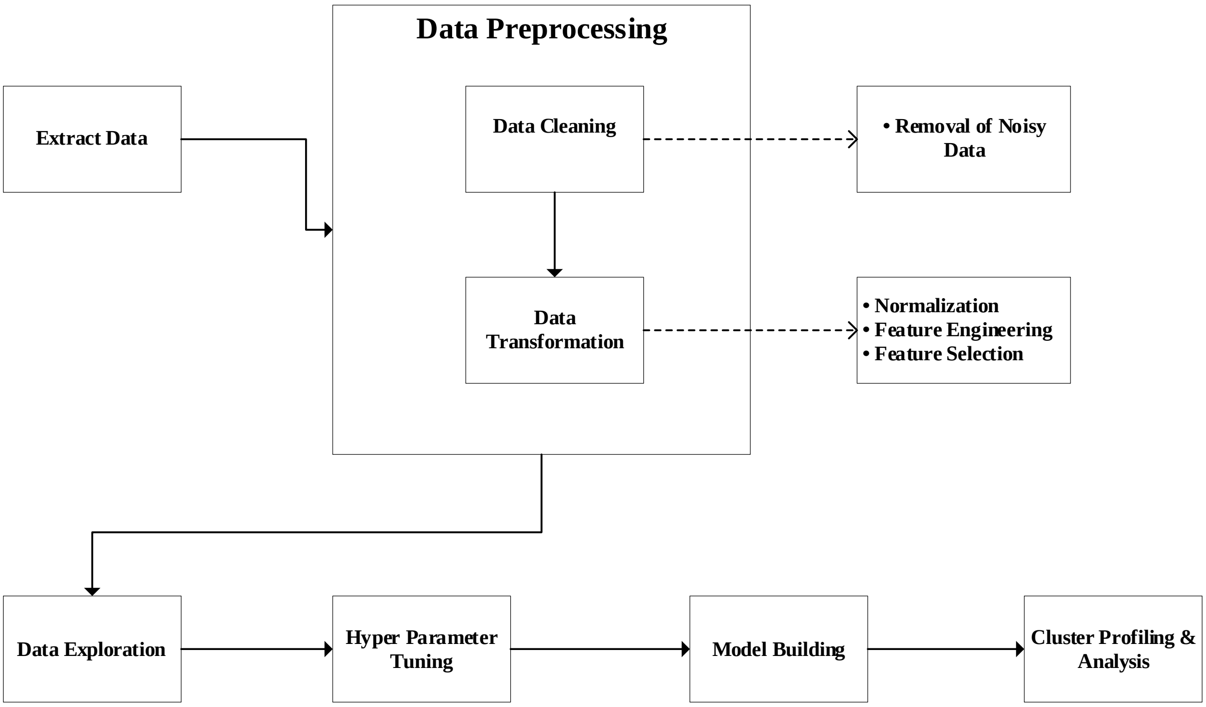


Figure 1. Design methodology

Data Extraction

For this analysis, we made use of the Online Retail dataset collected from the dev.cs.smu.ca server. We exported the entire table consisting of 536500 rows and 9 columns.

Data Preprocessing

Data preprocessing is a crucial first step for anyone dealing with data sets and one of the main reasons as to why this is true is because it leads to a cleaner and a more manageable data set. The following techniques have been performed on the data extracted:

- Data Cleaning- Better data beats fancier algorithms. Data Cleaning is the first step in our analysis in which we remove unwanted observations from the dataset. This includes noisy, irrelevant, and redundant data.
    - Removal of Noisy Data: In the analysis performed, from the boxplot, it was observed there were significant outliers in the InvoiceNo, Quantity and UnitPrice columns of the data.

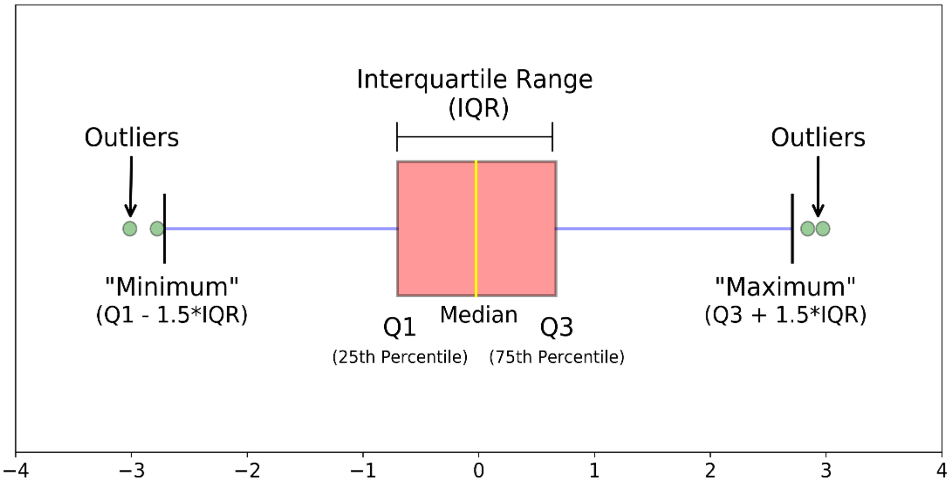


Figure 2. Boxplots explained.

From the diagram above, it is evident that the outliers are the data that fall above the "maximum" (Q3 + 1.5*IQR) or below the "minimum" (Q1 - 1.5*IQR).

To remove the outliers, a function, that checks for the outliers in the data columns was written and then this function was applied to the data frame and removed all the rows that contained an outlier value.

- Data Transformation- Every organization might have to transform the data to make it more compatible to achieve business insights. This process also facilitates the compatibility between the various data used in the analysis.
    - Feature Engineering: The raw data was transformed to generate featured variables such as Revenue, No of Visits, No of Distinct Customers and Total Units Sold. Furthermore, additional variables such as 'Average amount spent by customer' and 'Total Units Sold' were generated for clustering customers and products, respectively.

Hyperparameter tuning

In every machine learning analysis, it is essential to find the best version of the model. This is achieved by running many jobs by testing a range of hyperparameters on the dataset to get the best fit model. In the analysis done for this data, several clusters (k) have been tuned along with the maximum number of iterations to fetch the optimal number of clusters needed to segment customers and products, respectively.

Identify number of clusters

- Elbow Method
    - In cluster analysis, the elbow method is the most sought-after heuristic in determining the optimal number of clusters in a dataset. In the analysis, the optimal number of clusters were determined after running the k-means clustering algorithm for a range of values of k (1-50) and the corresponding sum of square errors were calculated. Undoubtedly, this method helped in determining the optimal number of clusters to be 4 for both customers and products, respectively.
- Gap Statistic Method
    - Besides the elbow method, yet another way to interpret the optimal number of clusters would be the Gap Statistic Method. This additional verification method confirms that the number of optimal clusters for customers and products are both 4 clusters.

Model Building

From a machine learning standpoint, it is vital to build the model by learning and generalizing from a clean dataset. In this case, k-means model is now triggered to calculate the Euclidean distances and assign a point to a cluster. In this analysis, the k-means model was fit on the pre-processed dataset with the optimal number of clusters suggested by the elbow and gap methods.

Feature Selection, Feature Engineering and Feature Definition

Customer Dataset

Feature Name	Measurement	Description
CustomerID	N/A	A 5-digit integral number uniquely assigned to each distinct customer.
Revenue	SUM	This field contains the aggregate values of the product of the quantity sold and its corresponding unit price for each distinct customer.
Number of visits	COUNT DISTINCT	The number of visits made to the store by each customer.
Number of distinct products	COUNT DISTINCT	The number of unique products bought by a customer from the store.
Number of products	SUM	The number of products bought by the customer from the store.
Average amount spent by customer	AVERAGE	This feature was engineered by calculating the average consumer expenditure. Total revenue by the number of number of transactions.

Product Dataset

Feature Name	Measurement	Description
StockCode	N/A	A 5-digit integral number uniquely assigned to each product.
Revenue	SUM	This field contains the aggregate values of the product of the quantity sold and its corresponding unit price for each distinct product.
Number of visits	COUNT DISTINCT	The number of unique visits made for each product.
Number of distinct customers	COUNT DISTINCT	The number of unique customers who bought product from the store.
Total Units Sold	SUM	The number of quantities sold for a particular product.

Data Cleaning and Outlier Removal

Interquartile Outlier detection method

For eliminating outliers, we define a decision range:

Lower Bound → Q1 - 1.5 * IQR

Upper bound → Q3 + 1.5 * IQR

Any data point lesser than the lower bound or higher than the upper bound is deleted.

Data cleaning and Outlier for Online Retail Data

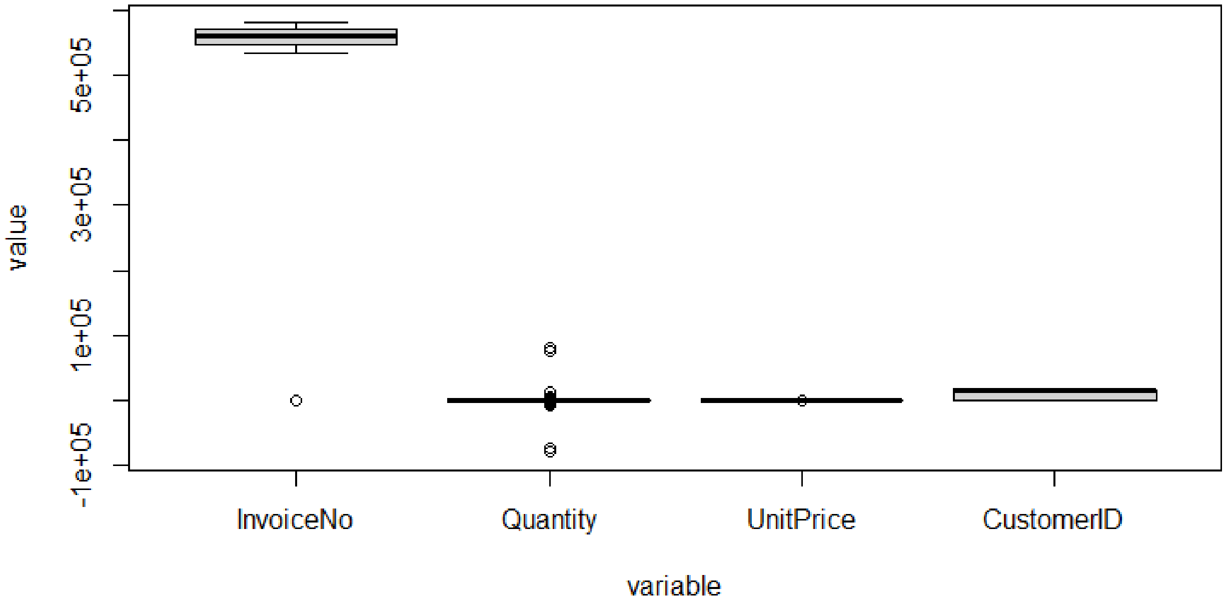


Figure 3. Boxplot before outlier removal for online retail data

From the figure above, we can see outliers in the invoice number, quantity, and unit price variables.

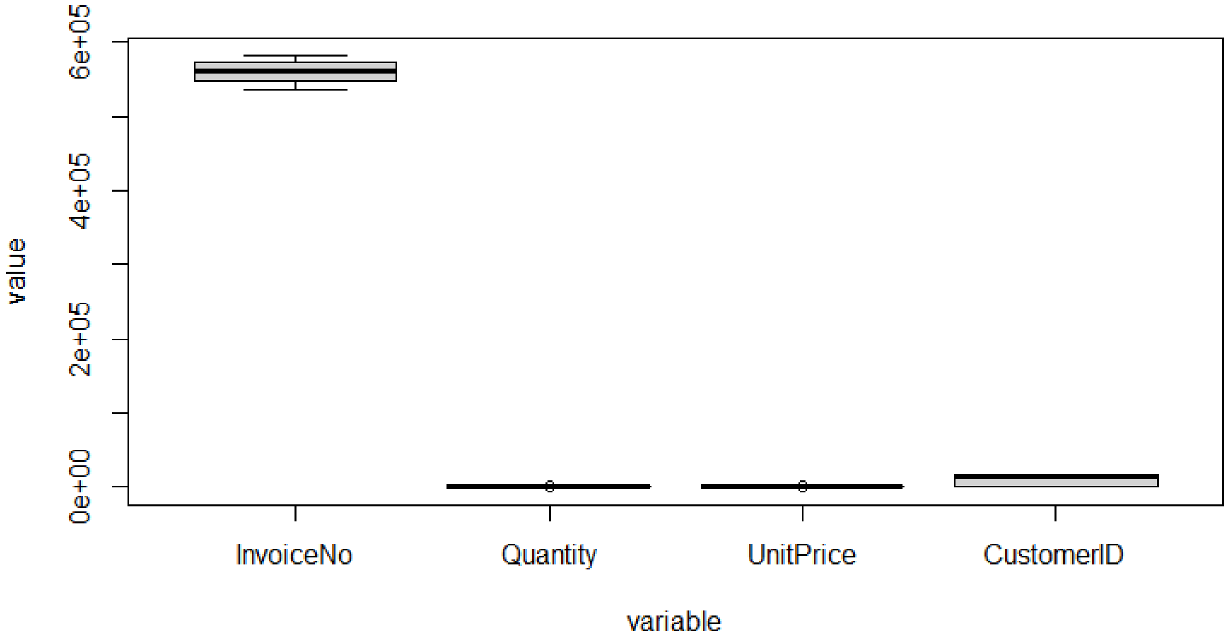


Figure 4. Boxplot after outlier removal for online retail data

After applying the function and removing the outliers, we were able to remove the outliers successfully.

Data cleaning after aggregation (Customer)

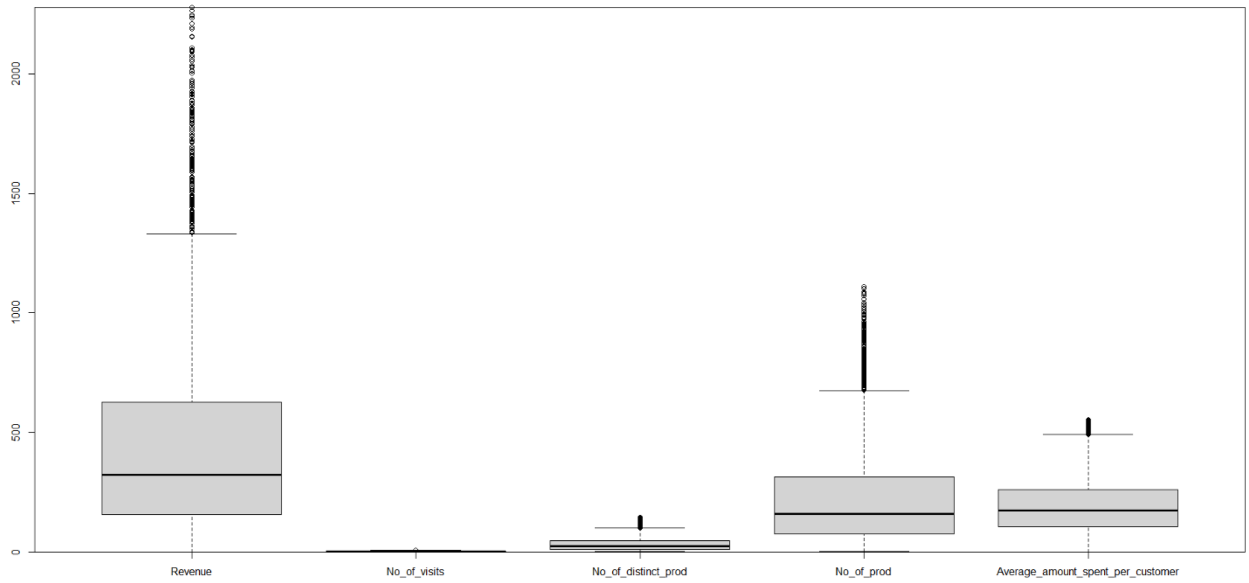


Figure 5. Boxplot before outlier removal (Customer)

From the figure above, we can observe outliers in the revenue, number of distinct products, number of products and average amount spend aggregated variables.

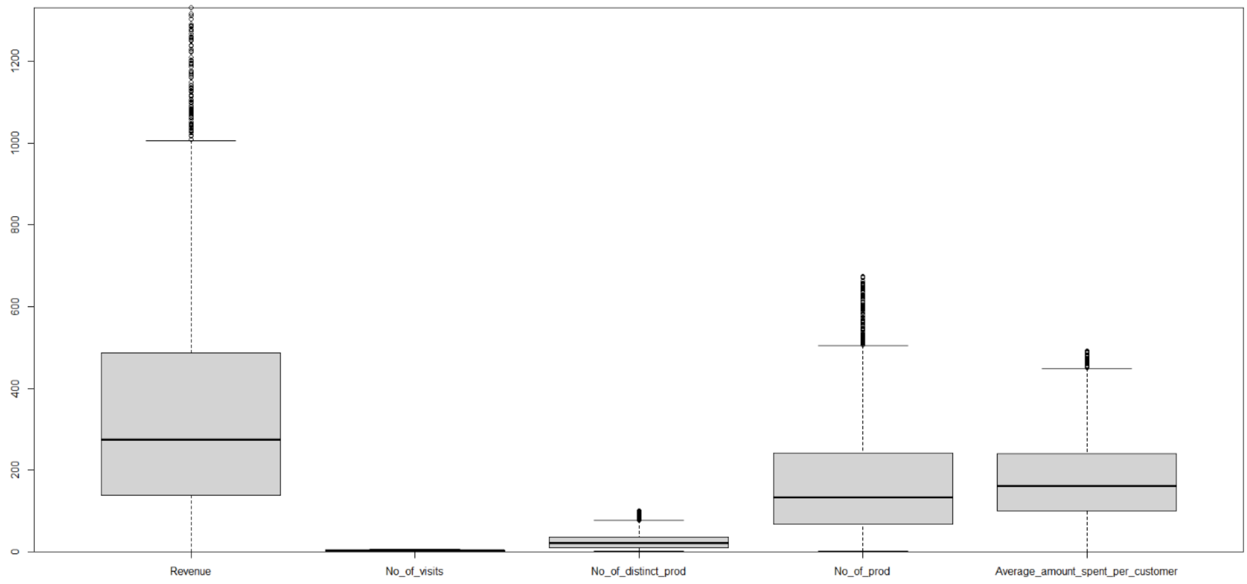


Figure 6. Boxplot after outlier removal (Customer)

After applying the function and removing the outliers, we were able to remove the outliers successfully.

Data cleaning after aggregation (Product)

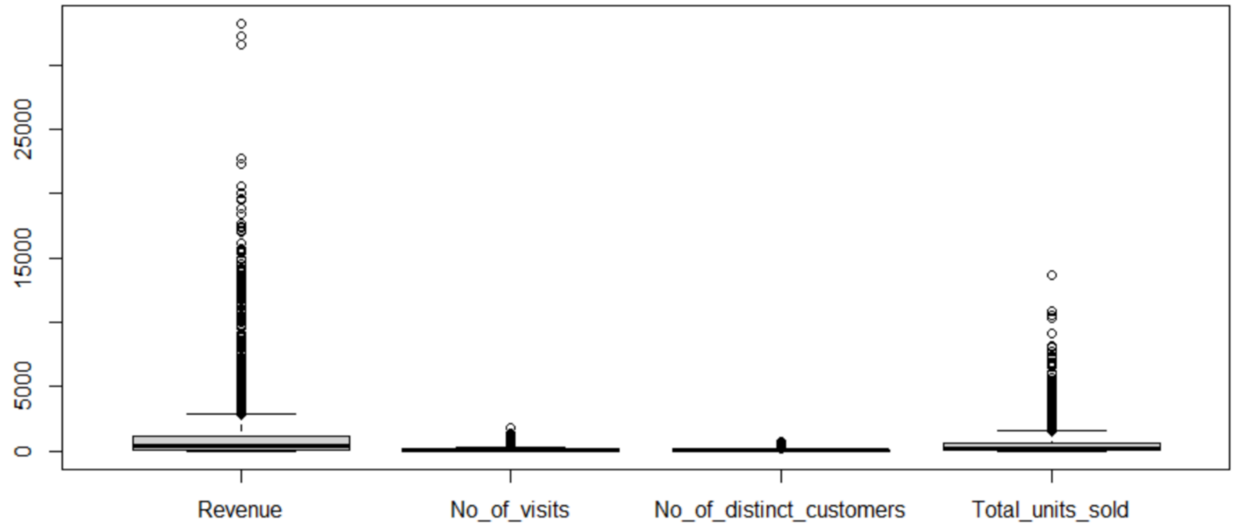


Figure 7. Boxplot before outlier removal (Product)

From the figure above, we can observe outliers in the revenue, number of visits, total units sold and number of distinct customers aggregated variables.

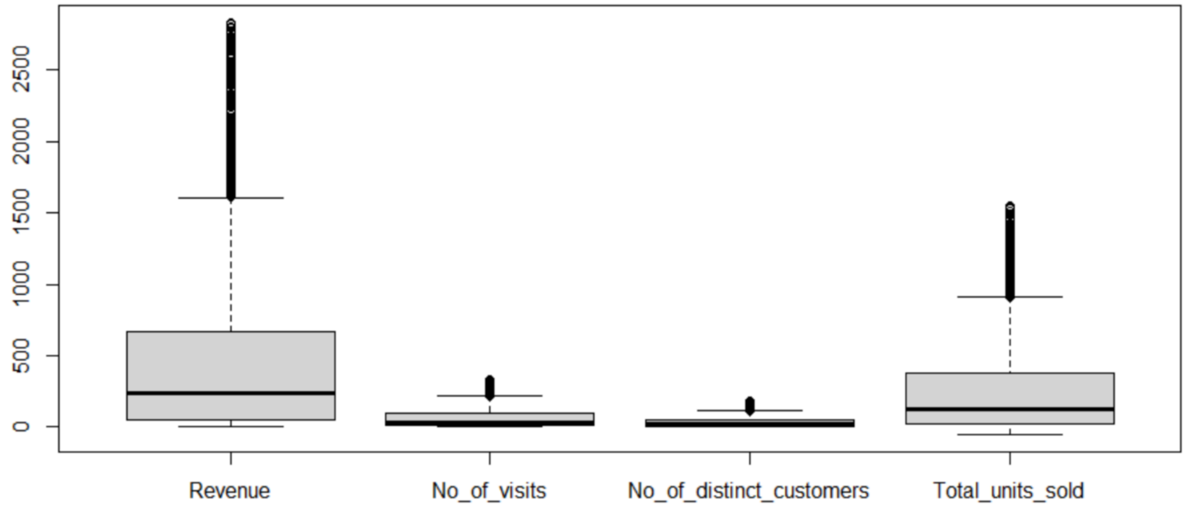


Figure 8. Boxplot after outlier removal (Product)

After applying the function and removing the outliers, we were able to remove the outliers successfully.

Cluster Analysis

For determining optimal number of clusters elbow and gap statistic methods are used.

Elbow method: Sum of squares at each number of clusters is graphed. The optimal number of clusters are identified by choosing the k value for which Within Cluster Sum of Squares (WSS) starts to diminish. Since elbow method is a naïve approach, we will cross verify with gap statistic method.

Gap statistic method: The Gap statistic calculates the total within infra-cluster variation for different values of k. The best estimated optimal number of clusters is calculated by observing the value that maximizes the curve. In our analysis, the optimal value for k=4 is chosen because it is the first peak point before which the value shrinks again.

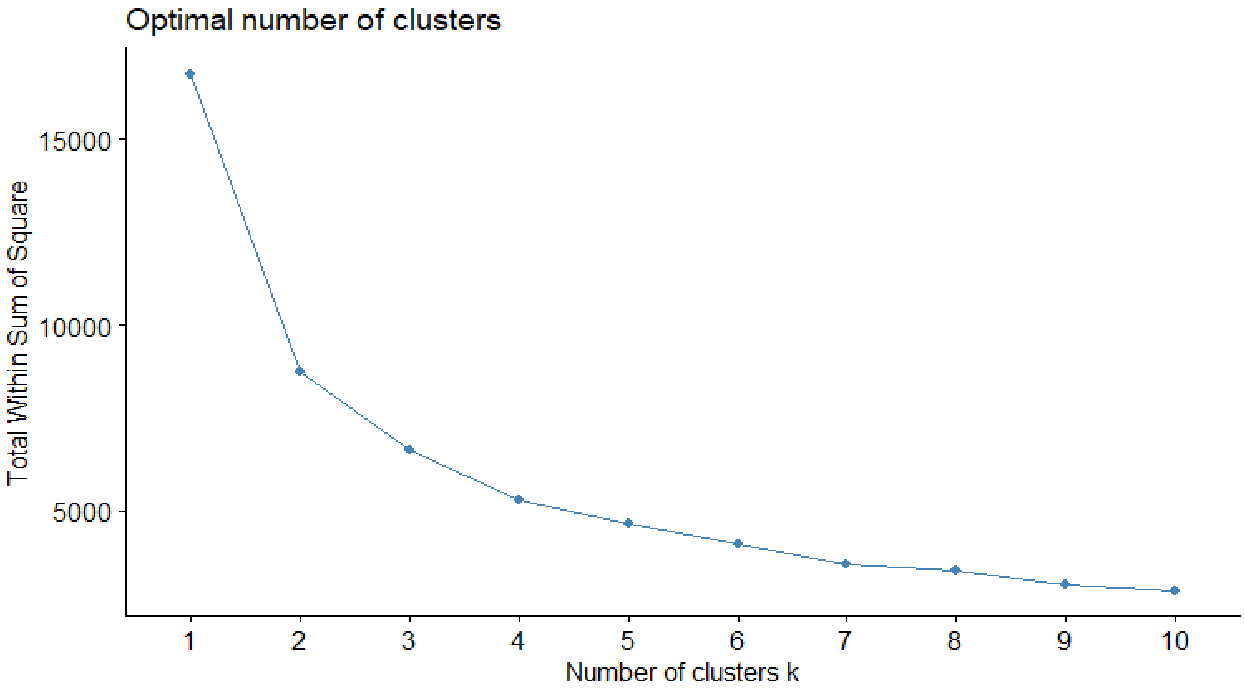


Figure 9. Optimal number of clusters using Elbow method (Customer)

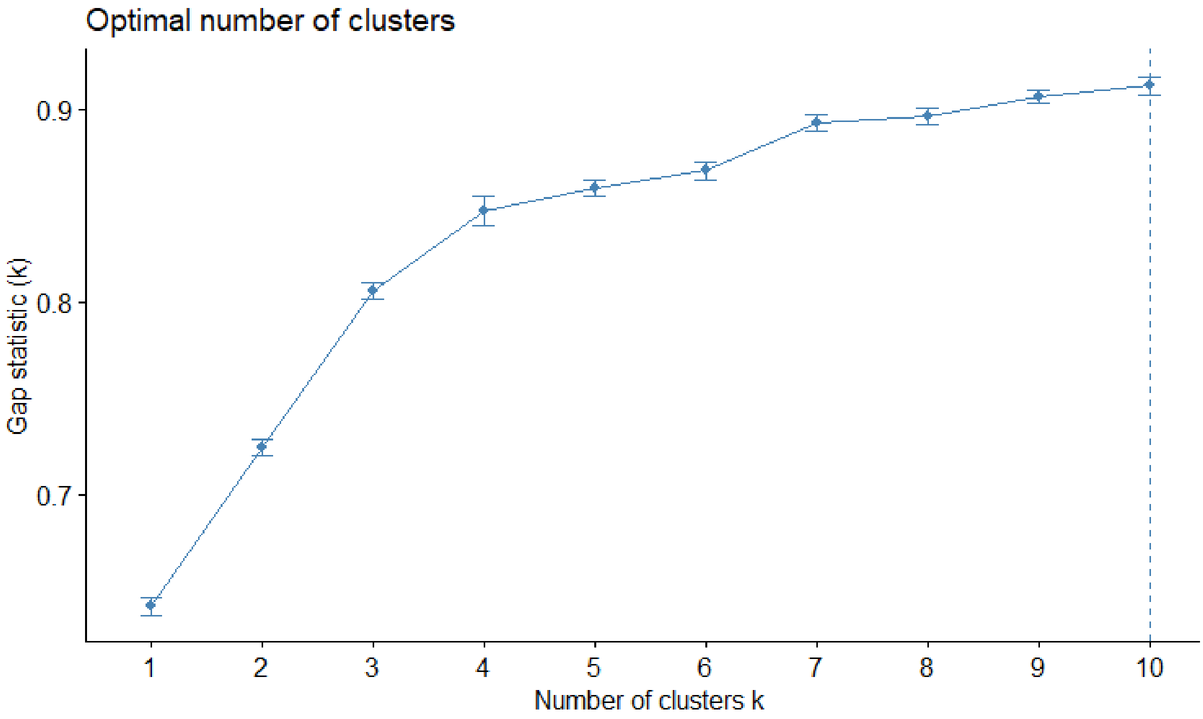


Figure 10. Optimal number of clusters using Gap Stat method (Customer)

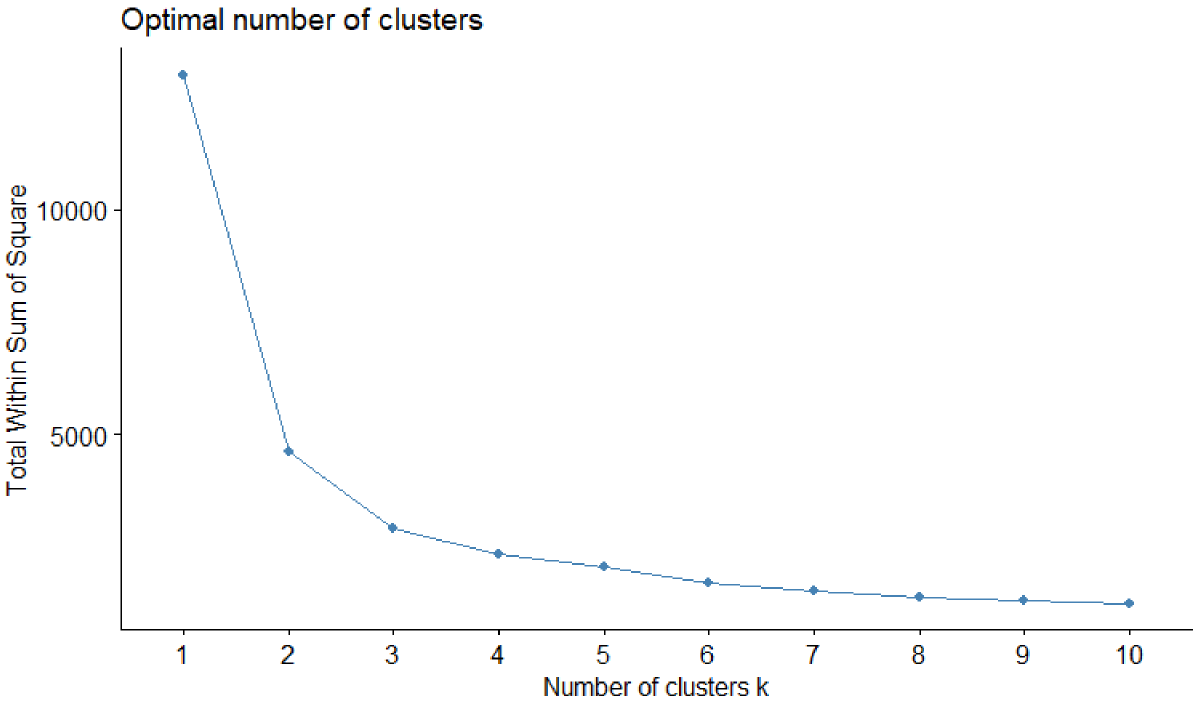


Figure 11. Optimal number of clusters using Elbow method (Product)

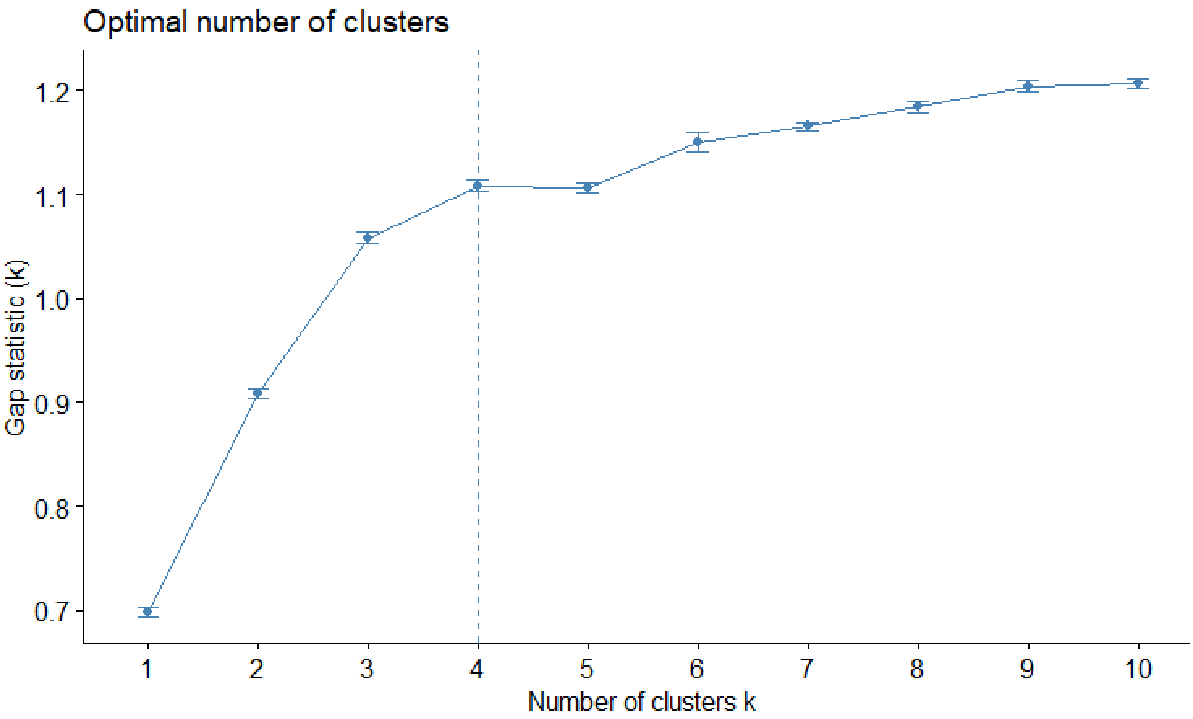


Figure 12. Optimal number of clusters using Gap Stat method (Product)

From the diagrams above, we used the elbow method as well as the gap method to find the optimal number of clusters for the customers and product aggregation. Both methods gave an estimate of 4 clusters for both aggregated data.

Cluster Profiling

Customer Clusters Overview

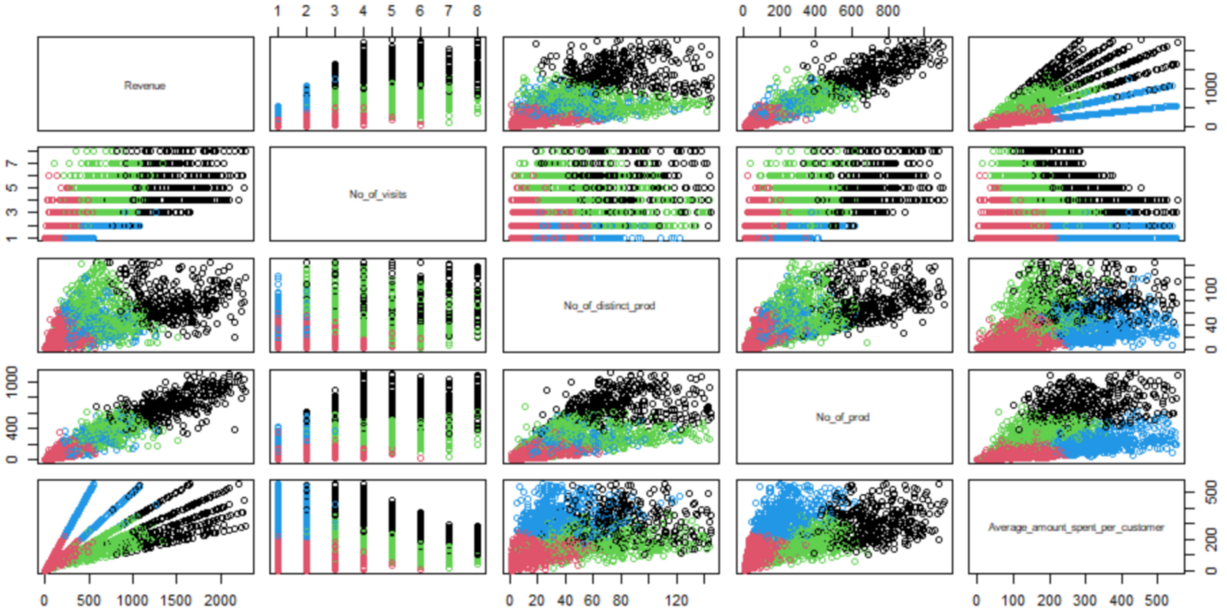


Figure 13. Customer clusters matrix

Customer Cluster Summary

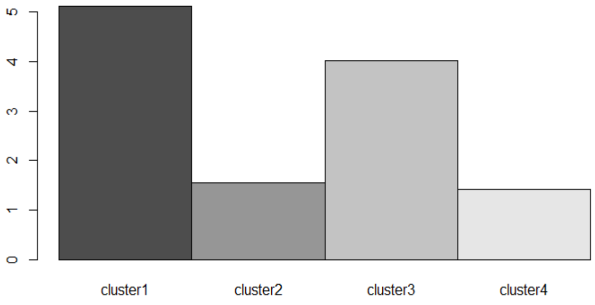


Figure 14. Number of Visits (Customer)

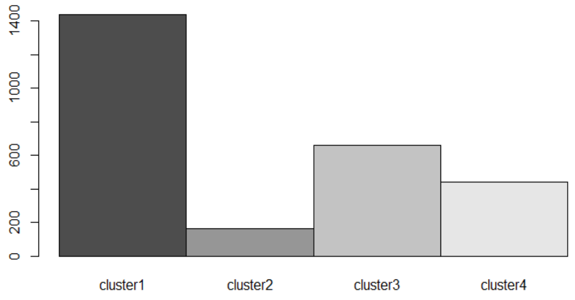


Figure 15. Revenue (Customer)

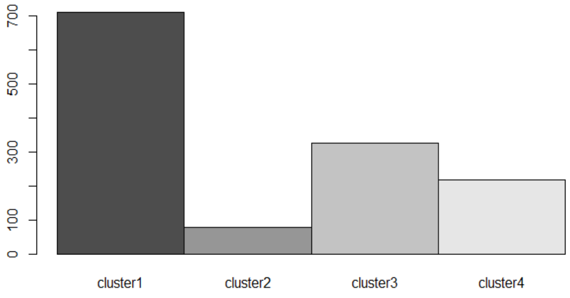


Figure 16. Number of products (Customer)

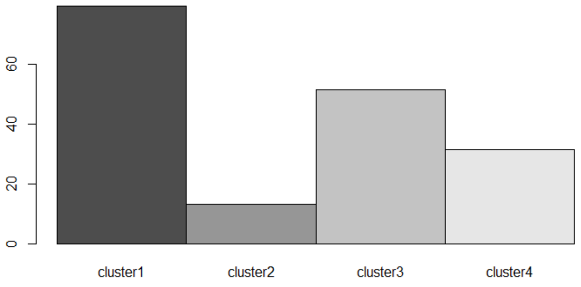


Figure 17. Number of distinct products (Customer)

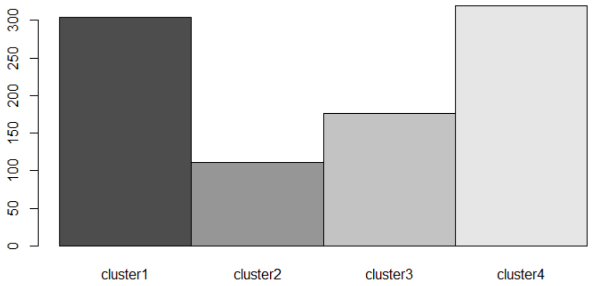


Figure 18. Average amount spent by customer (Customer)

Customer Segment	Description	Recommendation
Cluster 3 Loyal, (so I deserve a reward)	• Medium Revenue • Medium number of visits • High average amount spent • Medium number of products purchased	• Provide customers with bundled offers and discounts which helps them look at one single source that offers several intuitive solutions. • Introduce Referral Programs as a marketing tactic that encourages existing customers to recommend the company's brand to their connections.
Cluster 2 Occasional consumers (I'll buy if the deal seems right to me)	• Low Revenue • Low number of visits • Low average amount spent • Low number of products purchased	• Soliciting customer feedback on "what could be made better?" and understanding the issues they faced. • Understand the customer's pain points and needs through the customer satisfaction LOB which opens better opportunities for successful targeted marketing. • Furthermore, target these customers with email

		marketing campaigns to thank your customers and offer tips which encourages them to return frequently to the store.
Cluster 4 Medium Buyers (You've caught my attention, I'm listening)	• Low Revenue • Low number of visits • High average amount spent • Medium number of products purchased	• Provide redeem coupons to increase the visits and create a sense of urgency by providing limited time or quantity. • Reduce Customer Churn and retain them by introducing point reward system such as redeeming reward points after each purchase.
Cluster 1 Extremely Loyal (has no effect on marketing influence)	• High Revenue • High number of visits • High average amount spent • High number of products purchased	• Now that we gained their loyalty, we need to maintain this ongoing relationship. Establish CRM in the store to manage customer loyalty effectively. • As we know from Pareto's principle, 80% of the store's revenue is generated from 20% of the customers. Hence, we need to be grateful to these customers by going the extra mile in anticipating their challenges and providing outstanding customer service.

Product Cluster Overview

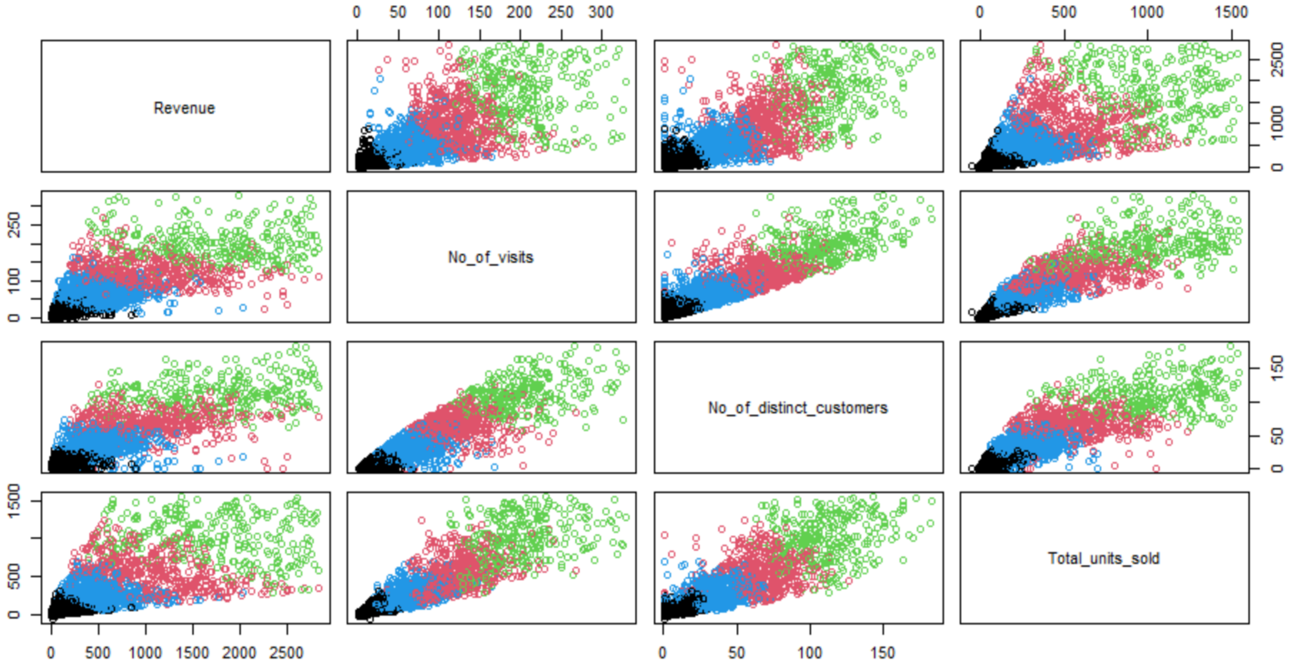
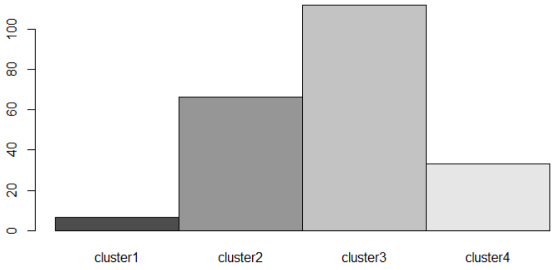


Figure 19. Product clusters matrix

Product Cluster Summary



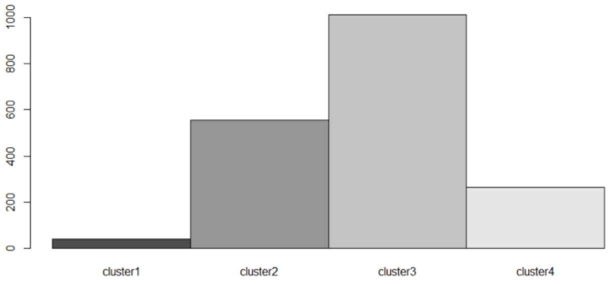
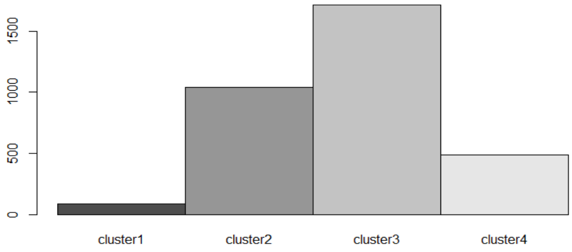


Figure 20. Number of distinct customers (Product)

Figure 21. Revenue (Product)



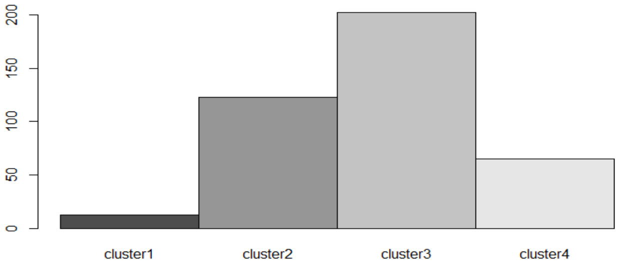


Figure 22. Total units sold (Product)

Figure 23. Number of visits (Product)

Product Segment	Description	Recommendation
Cluster 1 Unsought (little interest)	• Low revenue generated. • Low number of visits • Less number of distinct customers purchased this product. • Less total units sold	• Recommend the manufacturer to consider re-innovating the product according to the current market demand. • Optimize the prize for these products with a new strategy to boost sales. • Change the existing sale's strategy to expose the unique value proposition of the product. People buy feelings and not products.
Cluster 2 Convenience (Frequent purchased)	• High revenue generated. • High number of visits • High number of distinct customers purchased this product. • Moderate total units sold.	• Make sure that these products are adequately stocked in the inventory especially during peak hours. • Implement cross-sales ecommerce recommendation with the help of collaborative filtering and apriori recommendation engines. "Customers who bought this product also bought.." algorithm can escalate crowd wisdom, sales and revenue.
Cluster 3 Specialty (Strong brand and loyalty preference)	• Extremely High revenue generated. • High number of visits	• Now that you've made your customers fallen in love with these products, it

	• High number of distinct customers purchased this product • High total units sold	is vital to maintain a positive brand equity. • These products should now build an international brand and achieve ultimate brand leadership.
Cluster 4 Shopping (Less Frequently purchased)	• Moderate Revenue generated • Moderate number of visits • Moderate number of distinct customers purchased this product. • Moderate total units sold	• Implement bundle pricing strategy (both pure and impure) and offer products as a single package available at a single price.

Conclusion and Next Steps

In a nutshell, K - means algorithm has found widespread usage in lots of fields ranging from unsupervised learning, neural networks, pattern recognition, classification analysis, artificial intelligence, image processing, computer vision and many more. Although clustering was hard to assess, it is quite useful in the practical environment. After extracting the data and pre-processing it, the optimal number of clusters were easily estimated which helped in building the best fit machine learning unsupervised model for further analysis and recommendations.

By offering meaningful recommendations such as effective inventory management, product bundle strategy, soliciting customer feedbacks, offering discounts and coupons, the management can now make data-driven decisions to tackle business challenges and to drive their business forward.

Our next steps would be to use Isolation forest to remove outliers effectively and conveniently, and venture into other unsupervised machine learning algorithms to best fit our data.

Appendix-A [References]

Charrad, M. N. (2014). *Determining The Optimal Number Of Clusters: 3 Must Know Methods*. Retrieved from https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/

Galarnyk, M. (2018, September 12). *Understanding Boxplots*. Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51#:~:text=A%20boxplot%20is%20a%20standardized,and%20what%20their%20values%20are.

Lesonsky, R. (2015, August 12). *6 Great Ways to Move Slow Selling Merchandise*. Retrieved from https://smallbiztrends.com/2015/08/move-slow-selling-products.html

Mahendru, K. (2019, June 17). *How to Determine the Optimal K for K-Means?* Retrieved from https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb

O, M. (2019, January 27). *10 Tips for Choosing the Optimal Number of Clusters*. Retrieved from https://towardsdatascience.com/10-tips-for-choosing-the-optimal-number-of-clusters-277e93d72d92

What is data transformation: definition, benefits, and uses. (n.d.). Retrieved from https://www.stitchdata.com/resources/data-transformation/

Yulia Gavrilova, O. B. (2020, September 23). *What Is Data Preprocessing in ML?* Retrieved from https://serokell.io/blog/data-preprocessing

Appendix-B [Aggregation Queries run in R]

Product Aggregation

```
# product aggregation
product.agg <-
    data_outliers.removed %>%
    group_by(StockCode) %>%
    mutate(Revenue = sum(Quantity*UnitPrice)) %>%
    mutate(No_of_visits = length(unique(InvoiceNo))) %>%
    mutate(No_of_distinct_customers = length(unique(CustomerID))) %>%
    mutate(Total_units_sold = sum(Quantity)) %>%
    select(Revenue,No_of_visits,No_of_distinct_customers,Total_units_sold) %>%
    distinct()
```

Customer Aggregation

```
# customer aggregation
customer.agg <-
    data_outliers.removed %>%
    group_by(CustomerID) %>%
    mutate(Revenue = sum(Quantity*UnitPrice)) %>%
    mutate(No_of_visits = length(unique(InvoiceNo))) %>%
    mutate(No_of_distinct_prod = length(unique(StockCode))) %>%
    mutate(No_of_prod = sum(Quantity)) %>%
    mutate(Average_amount_spent_per_customer =
sum(Quantity*UnitPrice)/length(unique(InvoiceNo))) %>%

select(CustomerID,Revenue,No_of_visits,No_of_distinct_prod,No_of_prod,Average_amount_spent_
per_customer) %>%
    distinct()
```

Appendix-C [R Code]

Product Analysis

```
library(DMwR)
library(dplyr)
library(factoextra)
library(reshape)

# read data
online_retail_df <- read.csv("OnlineRetail_2.csv")

# data summary
summary(online_retail_df)

set.seed(5580)

# boxplot
meltData <- melt(online_retail_df)
boxplot(data = meltData, value ~ variable)

# The boxplot suggests that we need to remove outliers

remove_outliers <- function(x, na.rm = TRUE, ...) {
    qnt <- quantile(x, probs = c(.25, .75), na.rm = na.rm, ...)
    H <- 1.5 * IQR(x, na.rm = na.rm)
    y <- x
    y[x < (qnt[1] - H)] <- NA
    y[x > (qnt[2] + H)] <- NA
    return(y)
}

# outlier columns
data_outliers <- online_retail_df[c("InvoiceNo", "Quantity", "UnitPrice")]

# apply remove_outlier function to the data
data_outliers.removed <- lapply(data_outliers,function(x) { remove_outliers(x)})

# convert list to dataframe and remove all NA rows
data_outliers.removed <- na.omit(as.data.frame(data_outliers.removed))

# assigning an index column so we can merge both data
data_outliers.removed$index <- as.numeric(rownames(data_outliers.removed))
online_retail_df$index <-  as.numeric(rownames(online_retail_df))

data_outliers.removed = merge(x = data_outliers.removed,
                              y = online_retail_df[c("StockCode","Description",
                                                     "InvoiceDate","CustomerID",
                                                     "Country","InvoiceDateTime",
                                                     "index")],
                              by = "index")

# end of data cleaning part 1

# boxplot after removing outliers
meltData <- melt(data_outliers.removed[-1])
boxplot(data = meltData, value ~ variable)


# end of data cleansing part 1


# boxplots after removing outliers
meltData <- melt(data_outliers.removed[-1])
boxplot(data = meltData, value ~ variable)

# aggregation
product.agg <-
    data_outliers.removed %>%
    group_by(StockCode) %>%
    mutate(Revenue = sum(Quantity*UnitPrice)) %>%
    mutate(No_of_visits = length(unique(InvoiceNo))) %>%
    mutate(No_of_distinct_customers = length(unique(CustomerID))) %>%
    mutate(Total_units_sold = sum(Quantity)) %>%
    select(Revenue,No_of_visits,No_of_distinct_customers,Total_units_sold) %>%
    distinct()
```

```
boxplot(product.agg[-1])

# Removing outliers

product.agg.remove_outlier <- as.data.frame(lapply(product.agg[-1],function(x) {
remove_outliers(x)}))

product.agg.remove_outlier <- na.omit(product.agg.remove_outlier)

product.agg$index <- as.numeric(rownames(product.agg))
product.agg.remove_outlier$index <-  as.numeric(rownames(product.agg.remove_outlier))

product.agg.clean = merge(x = product.agg[c("StockCode", "index")], y =
product.agg.remove_outlier, by = "index")

# product boxplot after removing outliers
boxplot(product.agg.remove_outlier)

# scaling data
product.agg.scaled <- scale(product.agg.remove_outlier[-5])


withinSSrange <- function(data,low,high,maxIter)
{
    withinss = array(0, dim = c(high - low + 1));
    for (i in low:high)
    {
        withinss[i - low + 1] <- kmeans(data, i, maxIter)$tot.withinss
    }
    withinss
}


# elbow method
fviz_nbclust(product.agg.scaled, kmeans, method = "wss")
plot(withinSSrange(product.agg.scaled, 1 , 50, 150))

# gap method
fviz_nbclust(product.agg.scaled, kmeans, method = "gap_stat")

# perform kmeans
product.kmeans = kmeans(product.agg.scaled, 4, 150)

# centroids
realCenters = unscale(product.kmeans$centers, product.agg.scaled)

# table with customer clusters
clusteredProd = cbind(product.agg.remove_outlier, product.kmeans$cluster)

# rename cluster column
colnames(clusteredProd)[6] <- "cluster"

# cluster matrix
plot(clusteredProd[,1:4], col = product.kmeans$cluster)

product.agg.final = merge(x = product.agg.clean[c("StockCode", "index")], y =
clusteredProd, by = "index")
product.agg.final = merge(x = data_outliers.removed[c("StockCode", "Description")], y =
product.agg.final, by = "StockCode")
final <- product.agg.final[!duplicated(product.agg.final$StockCode), ]

final %>% group_by(cluster) %>% count()>% count()
```

Customer Analysis

```
library(DMwR)
library(dplyr)
library(factoextra)
library(reshape)

# read data
online_retail_df <- read.csv("OnlineRetail_2.csv")

# data summary
summary(online_retail_df)

set.seed(5580)

# boxplot
meltData <- melt(online_retail_df)
boxplot(data = meltData, value ~ variable)

# The boxplot suggests that we need to remove outliers

remove_outliers <- function(x, na.rm = TRUE, ...) {
    qnt <- quantile(x, probs = c(.25, .75), na.rm = na.rm, ...)
    H <- 1.5 * IQR(x, na.rm = na.rm)
    y <- x
    y[x < (qnt[1] - H)] <- NA
    y[x > (qnt[2] + H)] <- NA
    return(y)
}

# outlier columns
data_outliers <- online_retail_df[c("InvoiceNo", "Quantity", "UnitPrice")]

# apply remove_outlier function to the data
data_outliers.removed <- lapply(data_outliers,function(x) { remove_outliers(x)})

# convert list to dataframe and remove all NA rows
data_outliers.removed <- na.omit(as.data.frame(data_outliers.removed))

# assigning an index column so we can merge both data
data_outliers.removed$index <- as.numeric(rownames(data_outliers.removed))
online_retail_df$index <-  as.numeric(rownames(online_retail_df))

data_outliers.removed = merge(x = data_outliers.removed,
                              y = online_retail_df[c("StockCode","Description",
                                                     "InvoiceDate","CustomerID",
                                                     "Country","InvoiceDateTime",
                                                     "index")],
                              by = "index")

# end of data cleaning part 1

# boxplot after removing outliers
meltData <- melt(data_outliers.removed[-1])
boxplot(data = meltData, value ~ variable)

# aggregation
customer.agg <-
    data_outliers.removed %>%
    group_by(CustomerID) %>%
    mutate(Revenue = sum(Quantity*UnitPrice)) %>%
    mutate(No_of_visits = length(unique(InvoiceNo))) %>%
    mutate(No_of_distinct_prod = length(unique(StockCode))) %>%
    mutate(No_of_prod = sum(Quantity)) %>%
    mutate(Average_amount_spent_per_customer =
sum(Quantity*UnitPrice)/length(unique(InvoiceNo))) %>%

select(CustomerID,Revenue,No_of_visits,No_of_distinct_prod,No_of_prod,Average_amount_spent_
per_customer) %>%
    distinct()

# customer boxplots
boxplot(customer.agg)
boxplot(customer.agg$CustomerID)

# As you can see, customer with id 0 is the outlier, so we remove it
```

```
customer.agg <- customer.agg[customer.agg$CustomerID != 0,]
customer.agg <- as.data.frame(lapply(customer.agg,function(x) { remove_outliers(x)}))
customer.agg <- na.omit(customer.agg)

# customer boxplot after removing outliers
boxplot(customer.agg)

# scaling data
customer.agg.scaled <- scale(customer.agg[-1])


withinSSrange <- function(data,low,high,maxIter)
{
    withinss = array(0, dim = c(high - low + 1));
    for (i in low:high)
    {
        withinss[i - low + 1] <- kmeans(data, i, maxIter)$tot.withinss
    }
    withinss
}


# elbow method
plot(withinSSrange(customer.agg.scaled, 1 , 50, 150))

# Verifying with Silhouette method

fviz_nbclust(customer.agg.scaled, kmeans, method = "silhouette")

# perform kmeans
customer.kmeans = kmeans(customer.agg.scaled, 4, 150)

# centroids
realCenters = unscale(customer.kmeans$centers, customer.agg.scaled)

# table with customer clusters
clustered_cust = cbind(customer.agg[-1], customer.kmeans$cluster)

# rename cluster column
colnames(clustered_cust)[6] <- "cluster"

# cluster matrix
plot(clustered_cust[,1:5], col = customer.kmeans$cluster)

clustered_cust$CustomerId = customer.agg$CustomerID

clustered_cust %>% group_by(cluster) %>% count()
```