Jaime Rubio Diaz

40425150

2100 words

The following illustrates the technical tasks carried out during the assignment.

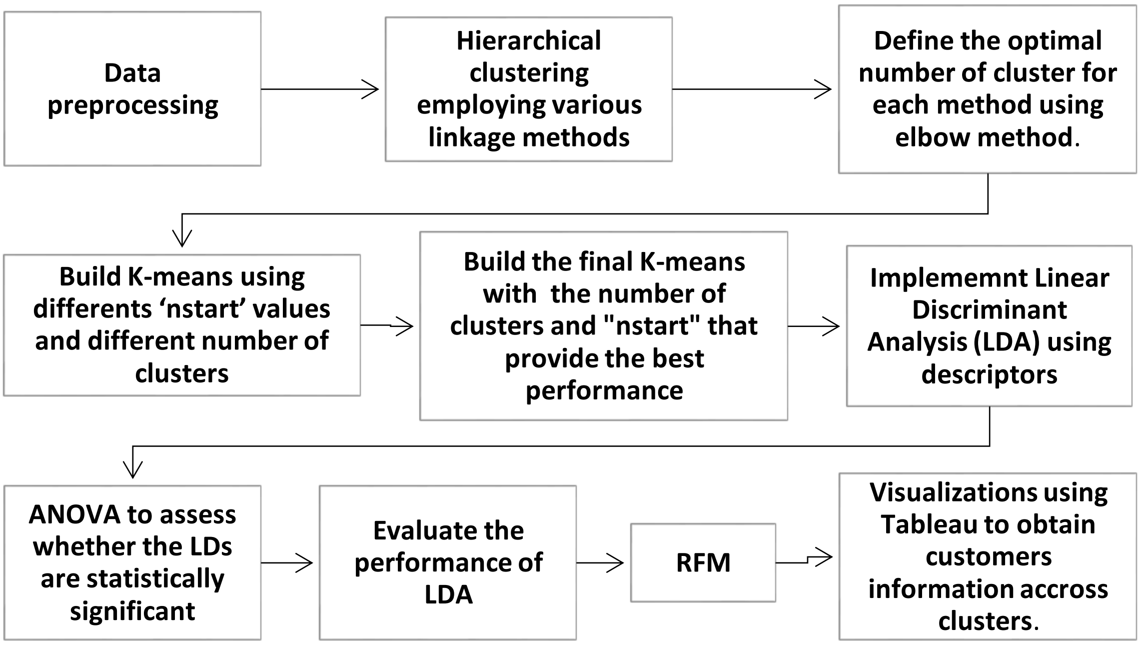


Table of content

1. Introduction

The aim of this research is to strategically segment the customers of an e-commerce platform specializing in the sale of all-occasion. Using exploratory data analysis techniques in Tableau, it will be uncovered some underlying customer patterns and profiles. Cluster analysis through R will define distinct customer segments based on transactional behaviour, complemented by LDA. The research will ultimately employ RFM analysis to further segment the customers. This approach is designed not only to effectively segment the customer base, but also to equip the e-commerce entity with data-driven insights to optimize its marketing strategies towards the most promising customers.

Currently, companies, particularly e-commerce, possess vast amounts of data about their customers. Understanding and applying customer segmentation is paramount for online retailers, as it allows them to tailor their marketing strategies and product offerings to meet the diverse needs and preferences of their customer base, ultimately fostering improving customer retention and acquisition (Chugh and Baweja, 2020). Algorithms such as K-means, allocating in the unsupervised machine learning category allows to perform this segmentation.

Tabianan, et. al., (2022) conducted similar research using k-means clustering algorithm, based on E-commerce's customer purchase behaviour data, and Tableau to present the results, being able to capture the inequalities to profitable segments and no profitable segments.

2. Methodology

Cross Industry Standard Process for Data Mining (CRISP-DM) offers a structured framework for planning and implementing the stages of a data mining project (Wirth and Hipp, 2000). This methodology has been adopted for the current project.

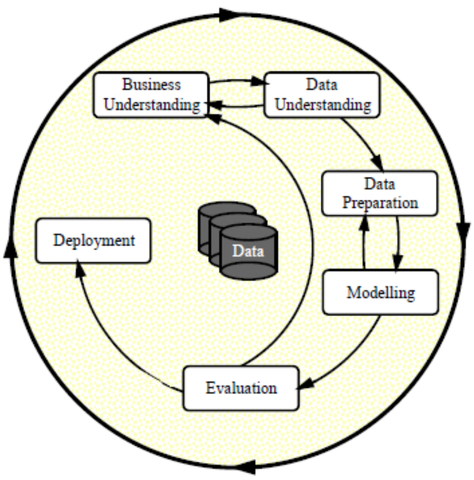


Figure 1. CRISP-DM Methodology

Business Understanding

The global economy is rapidly evolving towards digital technology-driven models that are accelerating the growth of e-commerce, significantly influencing economic structures and business sectors (Jain et. al. 2021). This company is an e-commerce entity specializing in the sale of unique gifts for all occasions, with a substantial portion of its customer base consisting of wholesalers.

Data Understanding

Descriptors			Missing Values
Married	Marital status of the customer	Categorical	0
Age	Age of the customer	Numerical	0
Income	Predicted annual income	Numerical	0
Work	Occupation or job title of the customer	Categorical	0
Education	Educational qualifications of the customer	Categorical	0
Base variables			
Quantity	Quantity of item per transaction	Numerical	0
UnitPrice	Product price per unit	Numerical	0
ReturnRate	Percentage of purchases customer returned to store	Numerical	0
Other Variables			
InvoceNo	Unique identifier to each transaction	Nominal	0
StockCode	Unique identifier to each product	Nominal	0
InvoiceDate	Date when the transaction was made	Date	0
Description	Product name	Nominal	0
CustomerID	Unique identifier for each customer	Nominal	2492
ZipCode	Zipcode for customer’s residences	Nominal	0

Table 1. Variables in the dataset

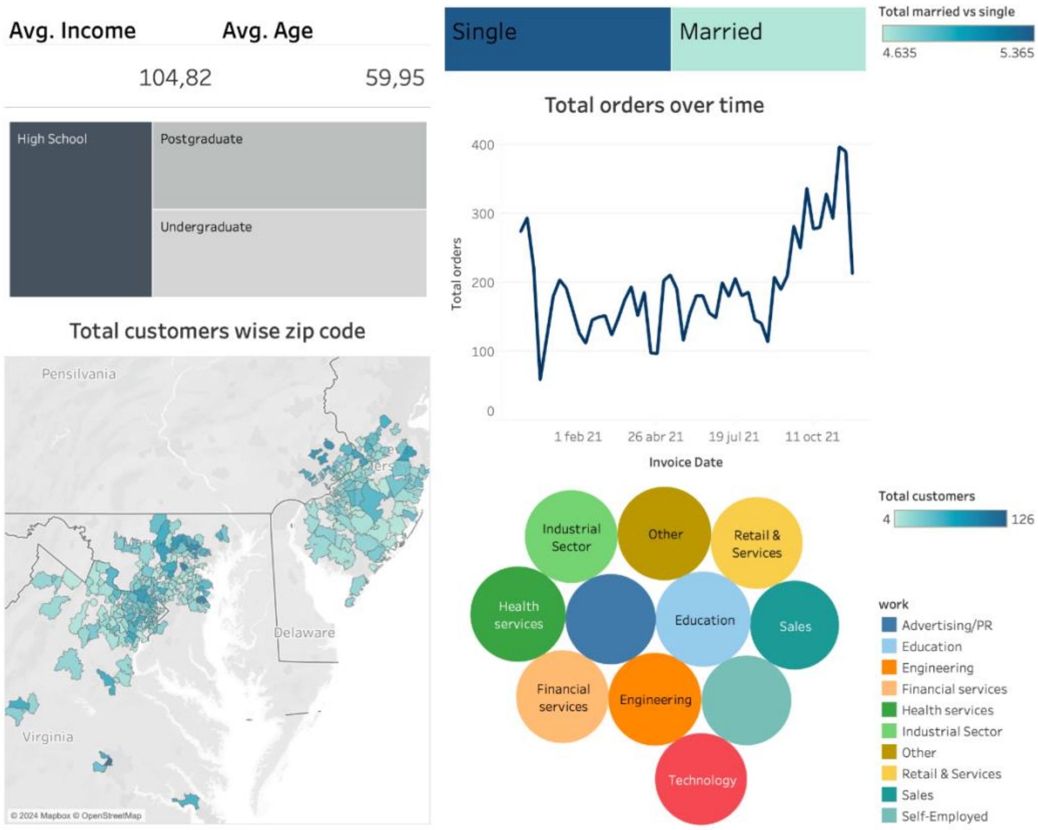


Figure 2. Initial data visualization using Tableau

Data Preparation

The data preparation for this dataset encompasses five steps: First, modifying the levels of certain variables, as specified in Table 2. Second, handling cancelled orders and inaccuracies in the record of return rates. Third, imputing missing values. Fourth, address duplicate CustomerID. Lastly, conduct feature engineering.

Table 2. Changes in variables levels names

Variable	Original Value	New Value
education	1	High School
	2	Undergraduate
	3	Postgraduate
gender	1	Married
	2	Single
work	1	Health services
	2	Financial services
	3	Sales
	4	Advertising/PR
	5	Education
	6	Industrial Sector
	7	Engineering

	8	Technology
	9	Retail & Services
	10	Self-Employed
	11	Other

In the data dictionary, it's noted that some "InvoiceNo" codes begin with the letter "C", indicating a cancellation of the product purchase. Given that only 1.75% of the products were cancelled, these entries are removed as they do not offer any valuable insight. Regarding the "ReturnRate" variable, some values exceed 1, which is inconsistent with the variable's expected range of 0-1. Rows featuring values greater than 1 account for 1.5% of the data and are therefore eliminated.

2492 missing values are found in the “CustomerID” variable. After considering various approaches for imputation, it is decided to group these rows by "InvoiceNo," and assign the same CustomerID to rows with the identical "InvoiceNo". Given that "InvoiceNo" serves as a unique identifier for each transaction, it is assumed that for these rows the same "InvoiceNo" corresponds to purchases made by the same customer.

To resolve duplicate CustomerID entries, the dataset is grouped by CustomerID, ensuring each customer is uniquely represented. Within these groups, it is computed the average for numerical variables and the mode for categorical variables. This resulted in a new dataset containing unique customer identifiers. For instance, if there are two observations for the same CustomerID, 12347, with ages 72 and 68 respectively. In the new dataset, these observations are merged into a single row, where the CustomerID remains unchanged, but the age is replaced with the mean of the two ages, resulting in a combined age of 70. This approach is considered because merging CustomerIDs after clustering could assign different clusters to the same CustomerID. Hence, this technique ensures each CustomerID is associated with a single cluster.

Given the limited number of base variables, when grouping, the following variables are created: “Avg_Quantity”, representing the average quantity purchased by rows sharing the same CustomerID; “Total_Quantity”, which is the aggregate of quantity for those rows; “Total_Value”, the total revenue for those rows, where revenue is calculated as the product of quantity and unit price; “Avg_Unit_Price”, the mean of the unit price; and “Avg_Return_Rate”, the average return rate. Consequently, this new dataset comprises five base variables.

For the effective implementation of LDA, and to accurately extract information from it, categorical variables must be transformed into factors in R. Additionally, the "InvoiceDate" should be converted into a date-time format to facilitate proper analysis.

Modelling

Following data preparation, cluster analysis is conducted to segment the company's customers. Since the optimal number of customer segments is unspecified, hierarchical clustering is utilized to infer this number. The dataset undergoes hierarchical clustering using a variety of linkage methods such as complete, single, centroid, and average. This approach allows for a thorough investigation into the impact of different methods on the clustering outcomes, with a focus on variables that indicate customer purchasing behaviour. After hierarchical clustering, elbow method is applied to determine the appropriate number of clusters.

Once the number of clusters is determined K-means is running to allocate all the customers to the correct cluster. K-means works by iteratively assigning data points to the nearest cluster centroid and updating the centroids to minimize the total Euclidean distances within each cluster (Kansal et al., 2018). Since the algorithm is sensitive to the initial random allocation of centroid it is good practice to test multiple initial configurations. Therefore, by altering the 'nstart' values and analysing the total within-cluster sum of squares (tot.withinss) and the ratio of between-cluster sum of squares to the total sum of squares (between_SS / total_SS), the optimal 'nstart' value and the number of clusters are determined.

After assigning the optimal cluster to each customer Linear Discriminant Analysis (LDA) is employed to model the relationship between cluster assignments and customer attributes. Linear Discriminant Analysis (LDA) is a statistical method used for dimensionality reduction and classification by maximizing the separation between different classes while minimizing within-class variance (Balakrishnama and Ganapathiraju, 1998).

Utilizing the fitted LDA model, predictions are made to classify each customer into a specific cluster. The accuracy of these predictions is then evaluated through a confusion matrix, offering insights into the model's performance in assigning customers to their respective clusters.

3. Results and discussion

Hierarchical Clustering and K-means

For running hierarchical clustering in r, apart from the 'hclust' function, it should be used: cbind, to group the base variable in a matrix to run hierarchical clustering; scale, it is used to normalize data, ensuring all variables contribute equally to the analysis; dist, calculates the distances between pairs of observations in a dataset, providing a measure of similarity. After running hclust with the four linkage methods mentioned, a dendrogram and elbow plot are used to identify the optimal number of clusters. Silhouette and Elbow methods are the most used for this purpose, regarding elbow the optimal number of clusters can be determined when the graph stops descending at a rapid rate and flattens out (Shi et al., 2021).

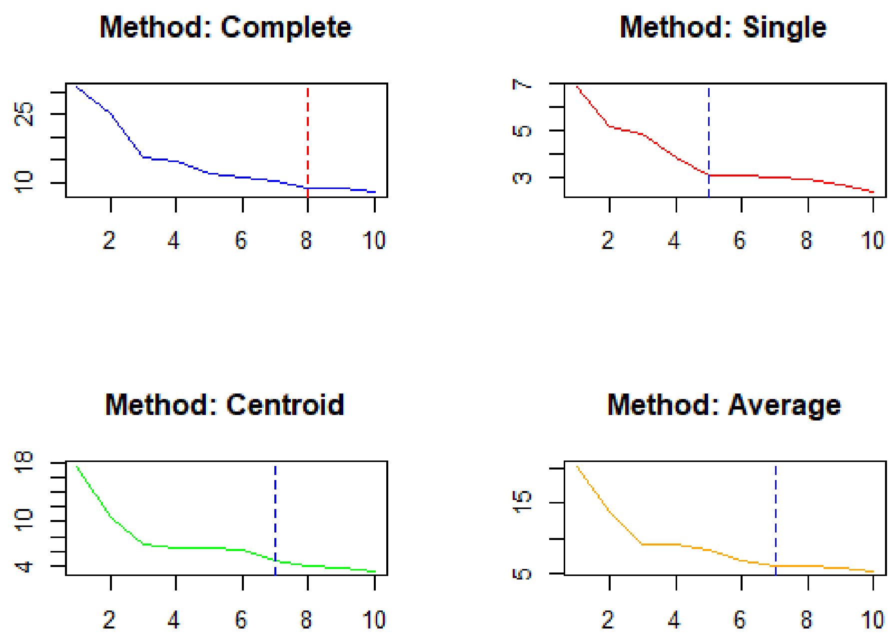


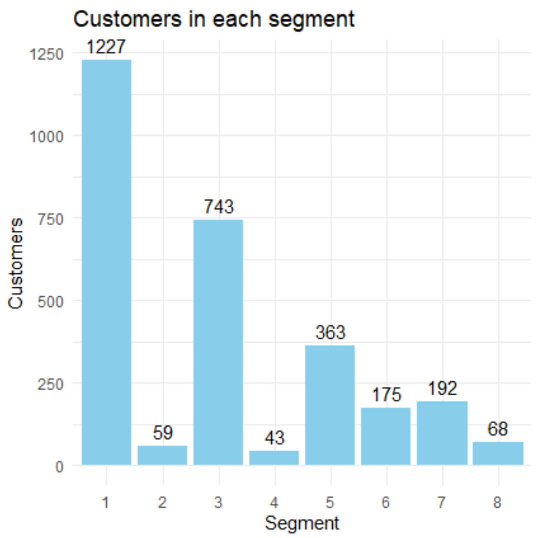
Figure 3. Elbow Plots for all linkage methods

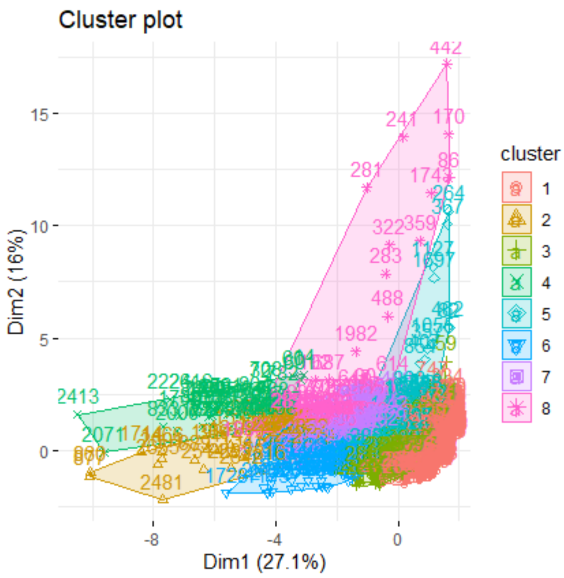
Table 3. Comparative of k-means performance

Method	Tried "nstart"	tot.withinss *	between_SS / total_SS *
Complete	10	1189097	85.7 %
	50	1189097	85.7 %
	100	1189097	85.7 %
Single	10	1475788	82.2 %
	50	1477150	82.2 %
	100	1477150	82.2 %
Centroid	10	1694280	79.6 %
	50	1693887	79 %
	100	1694280	79.6 %
Average	10	1933954	76.7 %
	50	1933954	76.7 %
	100	1933954	76 %

* tot.withniss indicates total sum of squared distance with the points of one cluster and the centroid (the lower, the better). between_SS / total_SS, indicates total variability of the data that is due to the variability between clusters (the higher, the better)

8 cluster is selected as they demonstrate the best performance, and the distribution of customers across these clusters is as follows:





Linear Discriminant Analysis (LDA)

LDA is built with the segments of customers as the target variable and work, age, income, education and married as predictors.

```
Call:
lda(segment ~ Married + Age + Income + Edcation + Work, data = segmentation
)

Prior probabilities of groups:
         1          2          3          4          5          6 
0.42752613 0.02055749 0.25888502 0.01498258 0.12648084 0.06097561 
         7          8 
0.06689895 0.02369338 

Group means:
  MarriedSingle      Age   Income EdcationPostgraduate
1     0.5338223 59.38225 106.5182            0.3227384
2     0.6610169 60.33301 108.4237            0.3220339
3     0.5625841 60.78028 103.3910            0.3297443
4     0.5813953 61.16634 106.9108            0.3023256
5     0.5454545 58.76811 105.1400            0.3305785
6     0.5885714 60.36736 108.2735            0.3428571
7     0.5520833 60.39155 105.0618            0.3802083
8     0.5588235 62.56442 106.7828            0.2794118
  EdcationUndergraduate WorkEducation WorkEngineering
1             0.3325183    0.09861451      0.08475958
2             0.2881356    0.15254237      0.08474576
3             0.3310902    0.08209960      0.09555855
4             0.3720930    0.11627907      0.18604651
5             0.3305785    0.12672176      0.07438017
6             0.2857143    0.08000000      0.07428571
7             0.3593750    0.07812500      0.10937500
8             0.3382353    0.05882353      0.08823529
  WorkFinancial services WorkHealth services WorkIndustrial Sector
1             0.08638957          0.09046455            0.09779951
2             0.06779661          0.06779661            0.06779661
3             0.09421265          0.09825034            0.09421265
4             0.02325581          0.00000000            0.06976744
5             0.08264463          0.09917355            0.09641873
6             0.12000000          0.11428571            0.13142857
7             0.05729167          0.15104167            0.08854167
8             0.10294118          0.08823529            0.08823529
   WorkOther WorkRetail & Services  WorkSales WorkSelf-Employed
1 0.09372453            0.09127954 0.09209454        0.08720456
2 0.10169492            0.06779661 0.13559322        0.16949153
3 0.08748318            0.10363392 0.08613728        0.08613728
4 0.11627907            0.13953488 0.11627907        0.02325581
5 0.09090909            0.06336088 0.09366391        0.07988981
6 0.08000000            0.06857143 0.08000000        0.05142857
7 0.06250000            0.08333333 0.08854167        0.10937500
8 0.17647059            0.07352941 0.02941176        0.13235294
  WorkTechnology
1     0.09209454
2     0.03389831
3     0.08479139
4     0.09302326
5     0.10468320
6     0.11428571
7     0.08333333
8     0.04411765

Coefficients of linear discriminants:
                               LD1          LD2           LD3
MarriedSingle           0.19577979  0.118250877  0.1355260773
Age                     0.02364732 -0.003888674  0.0147668418
Income                 -0.00681084  0.007333072  0.0003063207
EdcationPostgraduate    0.15789402 -0.947659652 -0.8509043156
EdcationUndergraduate   0.43756049 -0.494743329 -0.8570570072
WorkEducation          -1.32394580  1.229794400 -1.6126381162
WorkEngineering         0.93925756  0.151330794 -0.9143173163
WorkFinancial services -0.77886123 -0.410481390  1.2484840638
WorkHealth services    -0.27087955 -1.990948520 -0.7943173447
WorkIndustrial Sector  -0.99544319 -0.510246922  0.2951197536
```

```
WorkOther                 0.17050447  1.388530682  0.5985844978
WorkRetail & Services     0.80041037  0.168580910 -0.2640025957
WorkSales                -0.70337113  0.538510675 -1.6482981610
WorkSelf-Employed         0.91199291  0.416362778 -1.2484187256
WorkTechnology           -1.62979272 -0.519493252 -0.3117164496
                                  LD4           LD5          LD6
MarriedSingle            -0.320441871 -1.3434414301 -0.394907757
Age                      -0.002740894 -0.0141837174 -0.008865224
Income                   -0.003394226  0.0001523116 -0.022689543
EdcationPostgraduate      0.123711834  0.0698946419 -0.395342814
EdcationUndergraduate     0.363768138  0.7811531923 -0.193806908
WorkEducation            -0.732356935 -1.1371278497  0.951137046
WorkEngineering           0.773420398 -1.3034807619 -0.488173064
WorkFinancial services -0.970571314 -1.3510583978  1.649360368
WorkHealth services      -1.504333902 -0.6688776336  0.147002535
WorkIndustrial Sector    -0.449710555 -1.0935167836  0.169803858
WorkOther                -0.911536406  0.2307937207  0.006417425
WorkRetail & Services     0.848275097 -1.1820259114  1.577572194
WorkSales                -0.186239288 -1.9177221648  0.890521688
WorkSelf-Employed        -2.612917711 -0.5270667630  1.080350013
WorkTechnology            0.387519393 -0.7822424094  0.447956978
                                  LD7
MarriedSingle             0.600554420
Age                       0.007058898
Income                   -0.017175920
EdcationPostgraduate      0.086897512
EdcationUndergraduate     0.135573148
WorkEducation             0.199161914
WorkEngineering          -0.726172600
WorkFinancial services -1.243668519
WorkHealth services      -0.811242942
WorkIndustrial Sector    -1.408569166
WorkOther                -0.314521100
WorkRetail & Services    -2.485765663
WorkSales                -1.766976009
WorkSelf-Employed        -1.692924360
WorkTechnology           -0.610271234

Proportion of trace:
   LD1    LD2    LD3    LD4    LD5    LD6    LD7
0.2352 0.2225 0.1885 0.1707 0.0865 0.0578 0.0388
```

The "prior probabilities of groups" refer to the prior probabilities of each group or segment within the data set. For this data set, there is a higher prior probability that a new observation belongs to Group 1 rather than to any other group. This can be attributed to the larger number of customers within cluster.

Table 4. Prior probabilities LDA

Segment	Prior Probability
Segment 1	42.75%
Segment 2	2.06%
Segment 3	25.89%
Segment 4	1.50%
Segment 5	12.65%
Segment 6	6.10%
Segment 7	6.69%
Segment 8	2.37%

The “Group means” section of the LDA analysis output provides the average of each predictor variable for each segment. The numbers 1 to 8 correspond to the segments and, for instance, in segment 1, the average age of customers is 59 and the proportion of singles individuals is 53.38%. These characteristics across the segments will be further analysed with visualizations in Tableau.

Given the nature and objectives of Linear Discriminant Analysis (LDA), the number of linear discriminants is n−1 dimensions, where n is the number of groups. In this case, with 8 groups, LD=7 because 7 linear boundaries are sufficient to fully separate the 8 groups. The coefficients of the linear discriminants indicate the contribution of each predictor to every linear discriminant function, showing the importance of each predictor in distinguishing between the groups.

Proportion of trance indicates the proportion of the total between-group variability that each discriminant function (LD) captures, LD1 and LD2 are the most critical for differentiating between groups, together capturing almost half of the total variability.

Table 5. ANOVA for LD

ANOVA		
LD	**F value**	**P - value**
1	3.67	0.000657
2	3.44	0.0011
3	2.91	0.0048
4	2.63	0.01
5	1.33	0.22
6	0.89	0.51
7	0.59	0.75

P-value shows how likely is possible to obtain the results only by chance, while F value is a measure of the ratio of the variance explained by the model to the variance unexplained within the model (IBM, 2024). From LD1 to LD4, each discriminant function significantly differentiates between groups, being statistically significant with P-values well below the common alpha level of 0.05. This indicates strong evidence against the null hypothesis, suggesting that these functions are effective at distinguishing between the groups in the model.

Table 6. Confusion Matrix LDA

Confusion Matrix								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1223	0	4	0	0	0	0	0
2	59	0	0	0	0	0	0	0
3	743	0	0	0	0	0	0	0
4	43	0	0	0	0	0	0	0
5	363	0	0	0	0	0	0	0

6	175	0	0	0	0	0	0	0
7	192	0	0	0	0	0	0	0
8	68	0	0	0	0	0	0	0

On the one hand Xie and Qiu (2007), concluded that results of LDA are negatively affected by unbalanced datasets, by using four methods to rebalance data set and comparing performances of LDA. On the other hand, Xue and Titterington (2007), claimed that in their study the improvement in the AUC is not enough to confirm that the nature of the data set significantly affects the performance.

In the case of this data set, it can be clearly seen in the confusion matrix that the model is predicting all values to segment 1, except from 4. So, the performance of the model is highly poor, with a resulting accuracy of 0.42. Therefore, the unbalanced presence in segments generated with k-means may conclude poor performance. Taking into account that the only 4 observations that are not predicted to segment 1 are predicted to segment 3 (the second with higher number of customers).

Furthering this reason, descriptors lack robust discriminatory power across the clusters, as it will be seen in the descriptive analysis.

Recency, Frequency and Monetary Value (RFM)

After building RFM table by calculating recency, frequency, monetary and RFM score for all the company customers, the following segments are extracted from it.

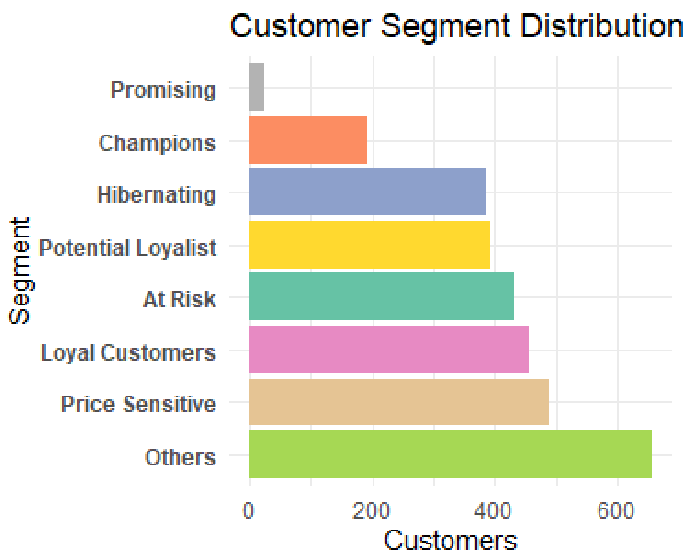


Figure 4. Customer segmentation based on RFM

Table 7. RFM Customer segmentation

RFM Customer Segmentation				
Segments	**Recency**	**Frequency**	**Monetary**	**Description**
Champions	≤ 2	≥ 4	≥ 4	The most valuable segment
Loyal Customers	≤ 3	≥ 3	≥ 3	Shoppers who purchase frequently and spend a lot, but may not have made a purchase very recently

Potential Loyalist	≤ 2	≤ 3	≤ 3	Newer customers with average frequency and monetary values who have the potential to become more valuable over time
At Risk	≥ 4	≥ 3	≥ 3	Customers who spent well and shopped often in the past but haven't purchased recently
Promising	≤ 3	= 2	= 2	Recent customers with few transactions but who have spent a moderate amount
Hibernating	≥ 4	≤ 2	≥ 2	Long-time customers who haven't made recent purchases and are infrequent shoppers but have spent a moderate amount in the past
Price Sensitive		≥ 4	≤ 2	Customers who purchase frequently but spend less

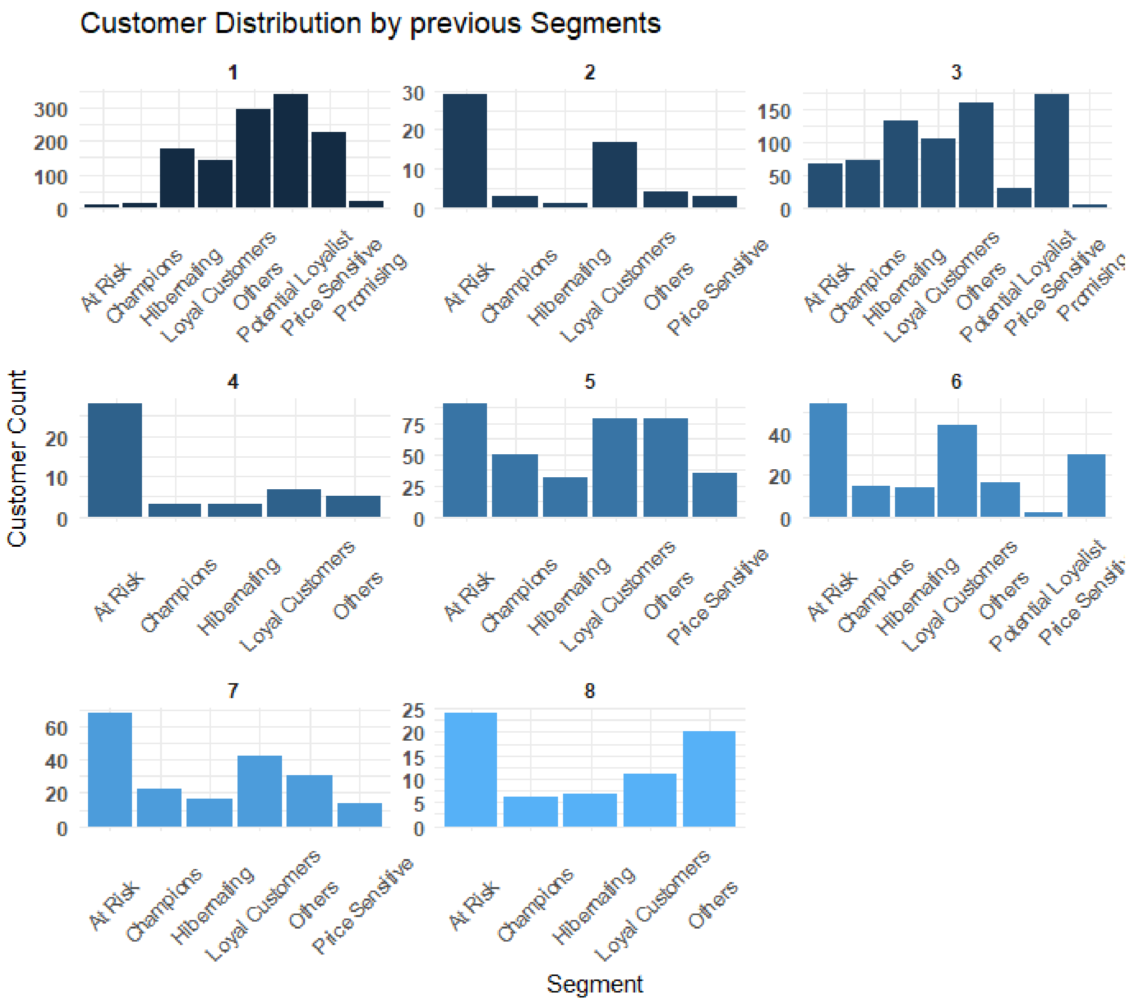
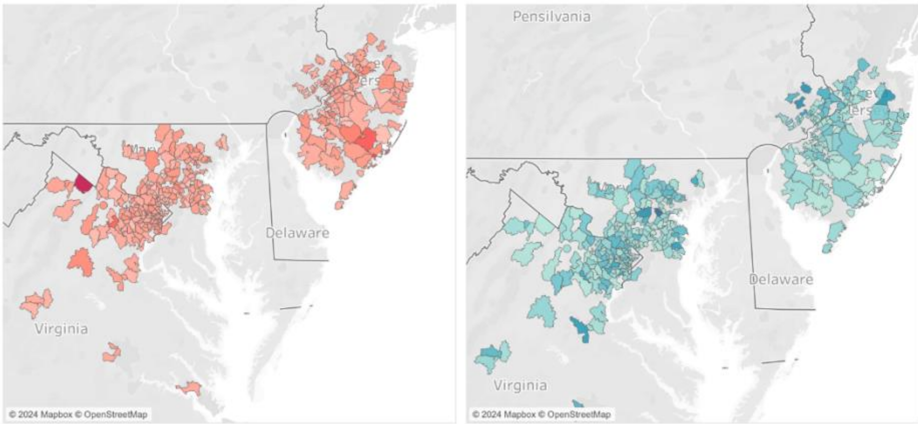


Figure 5. RFM customer segmentation wise K-means customer segmentation

Descriptive Analysis

	1	2	3	4	5	6	7	8
Average Age	59	60	61	60	58	61	62	60
Average Income	106,41	110,47	103,15	104,52	104,80	104,54	106,43	108,02



Most frequent Job wise segment

1	Education	121
2	Education	9
	Self-Employed	9
3	Retail & Services	77
4	Engineering	21
	Self-Employed	21
5	Education	46
6	Engineering	8
7	Other	12
8	Technology	20

Most frequent Education wise segment

1	High School	423
2	High School	21
3	High School	252
4	Postgraduate	72
5	High School	123
6	Undergraduate	17
7	High School	26
8	High School	65

Figure 6. Tableu dashboard after segmentation

In the dashboard, it's evident that the average incomes and ages are comparable across all segments. Notably, high school education emerges as the predominant level in segments 1, 2, 3, 5, 7, and 8, while postgraduate and undergraduate levels are more prevalent in segments 4.

	Segment							
	1	2	3	4	5	6	7	8
Prom. Return Rate	14,54%	13,11%	14,50%	14,90%	15,55%	17,48%	17,70%	15,58%
Prom. Quantity	4	33	9	12	7	30	11	26
Total Quantity	6.575	6.280	12.586	9.039	8.349	5.489	2.823	9.702
Total Revenue	14.060	6.578	22.500	20.028	23.346	10.685	11.995	8.140
Prom. Unit Price	3	2	3	4	6	3	20	1

Figure 7. Tableu base variables wise segmentation

The average quantity of product purchases per customer notably stands out in clusters 2, 6, and 8, indicating higher purchasing propensity within these segments. Particularly noteworthy is the considerable total quantity bought in segments 3, 4, and 8 compared to others. Despite segments 4 and 8 having relatively fewer customers (43 and 68, respectively), their tendency to place orders with larger product quantities or higher frequency suggests a notable engagement with the ecommerce platform.

Total revenue is predictably high in cluster 1 due to its size, yet it surpasses expectations in clusters 3, 4, and 5. While cluster 3's substantial size contributes to its revenue, clusters 4 and 5, despite being less populous, exhibit remarkable revenue generation potential, indicating them as prime targets for marketing efforts.

4. Conclusion and Limitations

In the previous study, k-means were running to obtain the optimal number of segments across customers of an E-commerce, having obtained 8 as the optimal number, and clusters 4 and 5 as the optimal target to maximize the revenue. Moreover, LDA was performance to build an accurate model to make predictions about future customers. However, the analysis underscores the challenge of working with unbalanced clusters and the lack of discriminatory power of descriptors across the segments.

Another limitation encountered in this project stemmed from issues related to data integrity, specifically the mishandling of 'CustomerID' and 'InvoiceNo'. This inadequacy in data collection led to an extensive need for data preparation, significantly impacting the project's outcomes by producing unrealistic results. A pivotal recommendation to mitigate these issues is to improve the data collection practice, whereby the customer can only modify descriptive data in their account profile and not each time a transaction is made. This approach would prevent the misinterpretation of demographic information as fluctuating data or the erroneous assignment of customer IDs, thereby enhancing the dataset's quality and reliability.

It may be considered to balance the segments with techniques such as SMOTE and observe the performance of LDA. Authors such as Anitha and Gustriansyah (2022) or Suhandiet. al. (2020) have conducted k-means clustering after performing RFM, using recency, frequency and monetary as base variables, obtaining good performance clusters. The code for doing this is providing in the Appendix having obtained balance results, as it can be seen in the following graph.

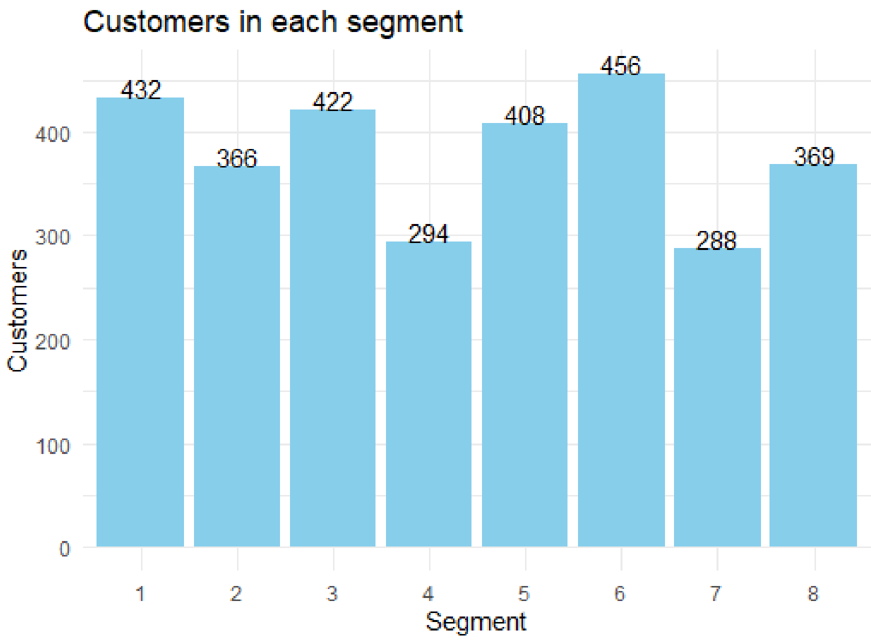


Figure 8. Customer segments when running K-means with RFM results

References

Chugh, S. and Baweja, V. (2020) ‘Data mining application in segmenting customers with clustering’, International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) [Online]. doi:10.1109/ic-etite47903.2020.259. Available at: https://ieeexplore.ieee.org/abstract/document/9077834 (Accessed: 18 February 2024)

Kansal, T. et al. (2018) ‘Customer segmentation using K-means clustering’, 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS) [Preprint]. doi:10.1109/ctems.2018.8769171. Available at: https://ieeexplore.ieee.org/abstract/document/8769171?casa_token=yvwdyWR9MKcAAAAA:CPEyK5GgX5yyUR8FfSev8MSRVy4hFMshoP8HTb2n9-KIr_IhEkoC8QzUzO9uz-bijxUE_DMROg (Accessed: 27 February 2024)

Shi, C. et al. (2021) ‘A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm’, EURASIP Journal on Wireless Communications and Networking, 2021(1). doi:10.1186/s13638-021-01910-w. Available at: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-021-01910-w (Accessed: 01 March 2024)#

IBM (2024) F value. Available at: https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=terms-f-value (Accessed: 02 March 2024).

Xie, J. and Qiu, Z. (2007) ‘The effect of imbalanced data sets on LDA: A theoretical and empirical analysis’, Pattern Recognition, 40(2), pp. 557–562. doi:10.1016/j.patcog.2006.01.009. Available at: https://www.sciencedirect.com/science/article/pii/S0031320306000136 (Accessed: 02 March 2024)

Xue, J.-H. and Titterington, D.M. (2008) ‘Do unbalanced data have a negative effect on LDA?’, Pattern Recognition, 41(5), pp. 1558–1571. doi:10.1016/j.patcog.2007.11.008. Available at: https://www.sciencedirect.com/science/article/pii/S0031320307005006 (Accessed: 02 March 2024)

Wirth, R. and Hipp, J. (2000) CRISP-DM: Towards a Standard Process Model for Data Mining [Online]. Available at: https://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf (Accessed: 03 March 2024).

JAIN, V., ARYA, S. and MALVIYA, B. (2021) ‘An overview of Electronic Commerce (e-commerce)’, Journal of Contemporary Issues in Business and Government, 27(3). doi:10.47750/cibg.2021.27.03.090.

Gustriansyah, R., Suhandi, N. and Antony, F. (2020) ‘Clustering optimization in RFM analysis based on K-means’, Indonesian Journal of Electrical Engineering and Computer Science, 18(1), p. 470. doi:10.11591/ijeecs.v18.i1.pp470-477. Available at: https://d1wqtxts1xzle7.cloudfront.net/63983488/20264-40355-1-PB20200721-118680-1taddep-libre.pdf?1595349331=&response-content-disposition=inline%3B+filename%3DClustering_optimization_in_RFM_analysis.pdf&Expires=1709741865&Signature=GaNeroA6wvZgVQWw93ZEkMxpeubVuUc5svKZcDiALMHbcQvP9bHIGFiZXHWSSDQey2ntD5hP1nGoFfs2KduUqKNoUdZ3-J4FJ~BGh8kA9IV~PvuV3GRAbQgtGj~NgvskcIwhJRkvEGZaTA8p7H5a5xSMVnwt1CVARbJrPCn8~zdeTa9VYYi5l~bXUIjcJMpShOvJv6ADS1g25-

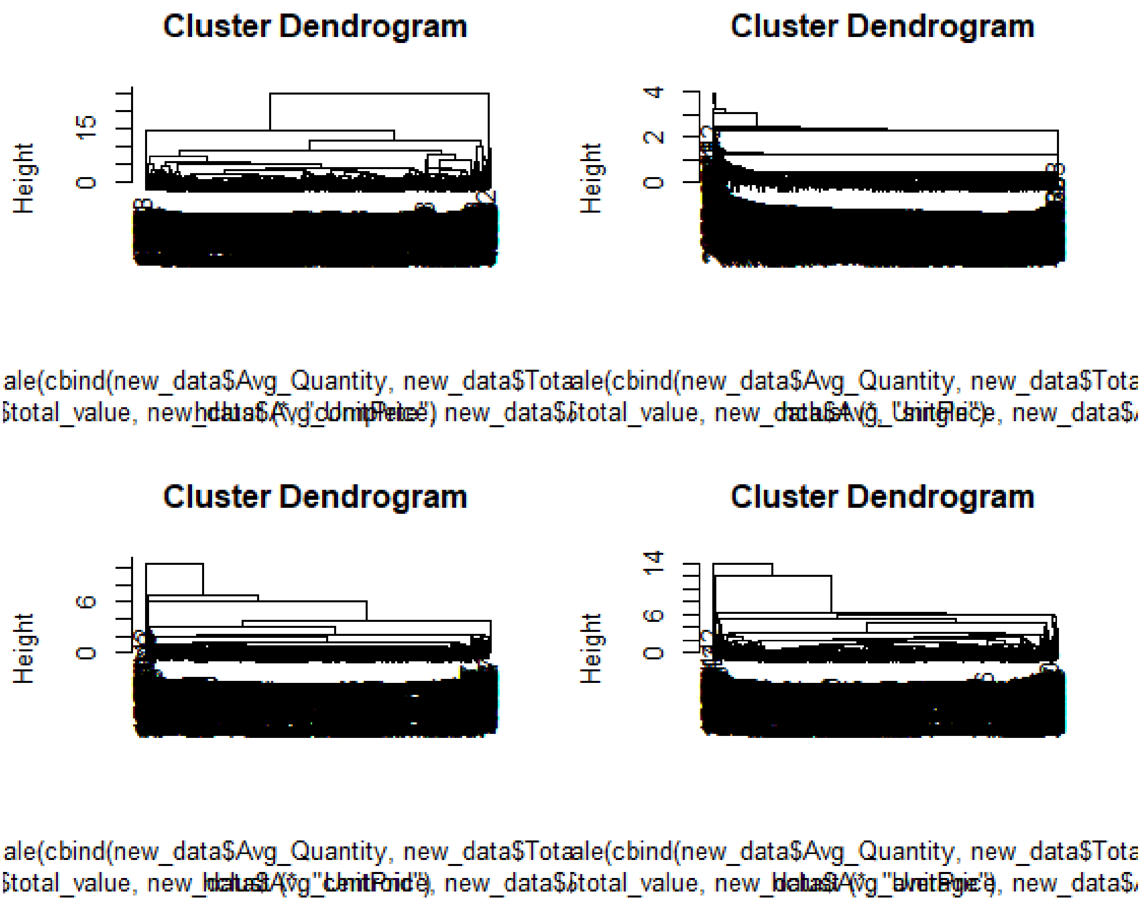
eAcYGcsKzgvh7xVdIR~jkFPaNOCS6QHPoPDjD4xotfEMuPIx86AizzHWx7XvAjPhbaXU26CiQ5OV-meDkd~R-RFMSyRgI4UPWIxdC5jZsNk80ySGcs-5PEvsLeVBtvP1wJPSboow__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA Accessed: 04 March 2023

Anitha, P. and Patil, M.M. (2022) 'RFM model for customer purchase behavior using K-means algorithm', Journal of King Saud University - Computer and Information Sciences, 34(5), pp. 1785–1792. doi:10.1016/j.jksuci.2019.12.011. Available at: https://www.sciencedirect.com/science/article/pii/S1319157819309802 Accessed: 05 March 2023.

Balakrishnama, S. and Ganapathiraju, A. (1998) 'LINEAR DISCRIMINANT ANALYSIS - A BRIEF TUTORIAL', INSTITUTE FOR SIGNAL AND INFORMATION PROCESSING, pp. 1–8. Available at: https://datajobs.com/data-science-repo/LDA-Primer-[Balakrishnama-and-Ganapathiraju].pdf (Accessed: 04 March 2024).

Tabianan, K., Velu, S. and Ravi, V. (2022) 'K-means clustering approach for intelligent customer segmentation using Customer Purchase Behavior Data', Sustainability, 14(12), p. 7243. doi:10.3390/su14127243. Available at: https://www.mdpi.com/2071-1050/14/12/7243 (Accessed: 27 February 2024).

Appendix 1


Cluster Dendrogram
Height
0
15
Cluster Dendrogram
Height
0
2
4
Cluster Dendrogram
Height
0
6
Cluster Dendrogram
Height
0
6
14

Appendix 2

```
 library(dplyr)

data <- read.csv(file.choose())

summary(data)

##Data preparation -----

#remove observations which purchase was cancelled
data <- subset(data, !startsWith(InvoiceNo, "C"))

#check how many values are higher that 1
filtered_data <- data[data$ReturnRate > 1, ]

# Calculate the percentage of the filtered data set compared to the original dataset
percentage_higher_than_1 <- (nrow(filtered_data) / nrow(data)) * 100

# Print the percentage
print(percentage_higher_than_1)

#remove rows where returnrate is higher than 1
data <- data[data$ReturnRate <= 1, ]

#convert invocedate into the correct format
data$InvoiceDate <- as.POSIXct(data$InvoiceDate, format = "%Y-%m-%dT%H:%M")

#mutate work levels
data <- data %>%
  mutate(Work = case_when(
    Work == 1 ~ "Health services",
    Work == 2 ~ "Financial services",
    Work == 3 ~ "Sales",
    Work == 4 ~ "Advertising/PR",
    Work == 5 ~ "Education",
    Work == 6 ~ "Industrial Sector",
    Work == 7 ~ "Engineering",
    Work == 8 ~ "Technology",
    Work == 9 ~ "Retail & Services",
    Work == 10 ~ "Self-Employed",
    Work == 11 ~ "Other"
  ))

#mutate Education levels

data <- data %>%
```

```
  mutate(Edcation = case_when(
    Edcation == 1 ~ "High School",
    Edcation == 2 ~ "Undergraduate",
    Edcation == 3 ~ "Postgraduate"
  ))

### Marriage

data <- data %>%
  mutate(Married = case_when(
    Married == 1 ~ "Married",
    Married == 0 ~ "Single"
  ))

##convert categorical variables into factors

data$Work <- as.factor(data$Work)
data$Edcation <- as.factor(data$Edcation)
data$Married <- as.factor(data$Married)
data$ZipCode <- as.factor(data$ZipCode)

## imputation of missing values in customer ID

#create a data set with all missing values
data_na <- data %>%
  filter(is.na(CustomerID))

#filter the original data for non missing values
data <- data %>%
  filter(!is.na(CustomerID))

##same InvoceNo same customer
data_na <- data_na %>%
  group_by(InvoiceNo) %>%
  mutate(CustomerID = cur_group_id())

# join both data sets
data <- bind_rows(data, data_na)


#mode for categorical variables
get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

#group by customer id w mean for numerical and mode for categorical
```

```
new_data <- data %>%
  group_by(CustomerID) %>%
  summarise(Age = mean(Age),
          Work = get_mode(Work),
          Avg_Quantity = mean(Quantity),
          Total_Quantity = sum(Quantity),
          total_value = sum(Quantity * UnitPrice),
          Avg_UnitPrice = mean(UnitPrice),
          Married = get_mode(Married),
          total_invoice = n_distinct(InvoiceNo),
          Avg_ReturnRate = mean(ReturnRate),
          Income = mean(Income),
          Edcation = get_mode(Edcation),
          zipcode = get_mode(ZipCode))%>% filter(
            total_value >= quantile(total_value, 0.025),
            total_value <= quantile(total_value, 0.975),
            Total_Quantity >= quantile(Total_Quantity, 0.025),
            Total_Quantity <= quantile(Total_Quantity, 0.975))

hist(data$ReturnRate)

#set seed
set.seed(40425150)

#hierarchical clustering whit 4 linkage methods -----
hclust<- hclust(dist(scale(cbind(new_data$Avg_Quantity, new_data$Total_Quantity,
new_data$total_value, new_data$Avg_UnitPrice, new_data$Avg_ReturnRate))),
method = "complete")
hclust1<- hclust(dist(scale(cbind(new_data$Avg_Quantity, new_data$Total_Quantity,
new_data$total_value, new_data$Avg_UnitPrice, new_data$Avg_ReturnRate))),
method = "single")
hclust2<- hclust(dist(scale(cbind(new_data$Avg_Quantity, new_data$Total_Quantity,
new_data$total_value, new_data$Avg_UnitPrice, new_data$Avg_ReturnRate))),
method = "centroid")
hclust3<- hclust(dist(scale(cbind(new_data$Avg_Quantity, new_data$Total_Quantity,
new_data$total_value, new_data$Avg_UnitPrice, new_data$Avg_ReturnRate))),
method = "average")

#different nstart values
nstart_values <- c(10, 50, 100)

x <- c(1:10)

#for complete method---
plot(hclust)
y <- sort(hclust$height, decreasing = TRUE)[1:10]
plot(x,y); lines(x,y, col= "blue")
```

```
results <- vector("list", length = 3)
for (i in 1:length(nstart_values)) {
  seg_kmeans <- kmeans(x = data.frame(new_data$Avg_Quantity,
new_data$Total_Quantity, new_data$total_value, new_data$Avg_UnitPrice,
new_data$Avg_ReturnRate), centers = 8, nstart = nstart_values[i])
  results[[i]] <- seg_kmeans
}

# Comparing results
for (i in 1:length(results)) {
  cat("Results for nstart =", nstart_values[i], ":\n")
  print(results[[i]])
  cat("\n")
}


#for single method
plot(hclust1)
y <- sort(hclust1$height, decreasing = TRUE)[1:10]
plot(x,y); lines(x,y, col= "blue")

results1 <- vector("list", length = 3)
for (i in 1:length(nstart_values)) {
  seg_kmeans1 <- kmeans(x = data.frame(new_data$Avg_Quantity,
new_data$Total_Quantity, new_data$total_value, new_data$Avg_UnitPrice,
new_data$Avg_ReturnRate), centers = 6, nstart = nstart_values[i])
  results1[[i]] <- seg_kmeans1
}

# Comparing results
for (i in 1:length(results1)) {
  cat("Results for nstart =", nstart_values[i], ":\n")
  print(results1[[i]])
  cat("\n")
  cat("Results for nstart =", nstart_values[i], ":\n")
  cat("tot.withinss:", results1[[i]]$tot.withinss, "\n\n")
}


## for centroid method

plot(hclust2)
y <- sort(hclust2$height, decreasing = TRUE)[1:10]
plot(x,y); lines(x,y, col= "blue")

results2 <- vector("list", length = 3)
```

```
for (i in 1:length(nstart_values)) {
  seg_kmeans2 <- kmeans(x = data.frame(new_data$Avg_Quantity,
new_data$Total_Quantity, new_data$total_value, new_data$Avg_UnitPrice,
new_data$Avg_ReturnRate), centers = 5, nstart = nstart_values[i])
  results2[[i]] <- seg_kmeans2
}

# Comparing results
for (i in 1:length(results2)) {
  cat("Results for nstart =", nstart_values[i], ":\n")
  print(results2[[i]])
  cat("\n")
  cat("Results for nstart =", nstart_values[i], ":\n")
  cat("tot.withinss:", results2[[i]]$tot.withinss, "\n\n")
}


## for average method

plot(hclust3)
y3 <- sort(hclust3$height, decreasing = TRUE)[1:10]
plot(x,y3); lines(x,y3, col= "blue")

results3 <- vector("list", length = 3)
for (i in 1:length(nstart_values)) {
  seg_kmeans3 <- kmeans(x = data.frame(new_data$Avg_Quantity,
new_data$Total_Quantity, new_data$total_value, new_data$Avg_UnitPrice,
new_data$Avg_ReturnRate), centers = 4, nstart = nstart_values[i])
  results3[[i]] <- seg_kmeans3
}

seg_kmeans3$tot.withinss
# Comparing results
for (i in 1:length(results3)) {
  cat("Results for nstart =", nstart_values[i], ":\n")
  print(results3[[i]])
  cat("\n")
  cat("Results for nstart =", nstart_values[i], ":\n")
  cat("tot.withinss:", results3[[i]]$tot.withinss, "\n\n")
}


optimal_clusters_complete <- 8
optimal_clusters_single <- 6
optimal_clusters_centroid <- 5
optimal_clusters_average <- 4
```

```
# Create elbow plots
par(mfrow=c(2,2))

# elbow plot for "complete"
y <- sort(hclust$height, decreasing = TRUE)[1:10]
plot(x, y, type = "l", col = "blue", main = "Method: Complete", xlab = "", ylab = "")
abline(v = optimal_clusters_complete, col = "red", lty = 2)

# elbow plot for "single"
y1 <- sort(hclust1$height, decreasing = TRUE)[1:10]
plot(x, y1, type = "l", col = "red", main = "Method: Single", xlab = "", ylab = "")
abline(v = optimal_clusters_single, col = "blue", lty = 2)

# elbow plot for "centroid"
y2 <- sort(hclust2$height, decreasing = TRUE)[1:10]
plot(x, y2, type = "l", col = "green", main = "Method: Centroid", xlab = "", ylab = "")
abline(v = optimal_clusters_centroid, col = "blue", lty = 2)

# elbow plot for "average"
y3 <- sort(hclust3$height, decreasing = TRUE)[1:10]
plot(x, y3, type = "l", col = "orange", main = "Method: Average", xlab = "", ylab = "")
abline(v = optimal_clusters_average, col = "blue", lty = 2)

#endogram
plot(hclust)
plot(hclust1)
plot(hclust2)
plot(hclust3)


##final k mean selection ---


seg_kmeans_final <- kmeans(x = data.frame(new_data$Avg_Quantity,
new_data$Total_Quantity, new_data$total_value, new_data$Avg_UnitPrice,
new_data$Avg_ReturnRate), centers = 8, nstart = 50)
seg_kmeans_final$tot.withinss

segment <- seg_kmeans_final$cluster
segmentation <- cbind(new_data, segment)
table(segmentation$segment)

#visualize the segments
segment_counts <- table(segmentation$segment)
segment_data <- as.data.frame(segment_counts)

names(segment_data) <- c("Segment", "Count")
```

```
# Create bar plot
ggplot(segment_data, aes(x = Segment, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = Count), vjust = -0.5) +
  labs(title = "Customers in each segment", x = "Segment", y = "Customers") +
  theme_minimal()

library(cluster)
library(factoextra)

new_data_numeric <- new_data[sapply(new_data, is.numeric)]

# clusters visualization
fviz_cluster(seg_kmeans_final, new_data_numeric,
         ggtheme = theme_minimal())

##LDA ----

##duplicate the data set to group the

library(MASS)


segmentation$Work <- as.factor(segmentation$Work)
segmentation$segment <- as.factor(segmentation$segment)

fit <- lda(segment ~ Married + Age + Income + Edcation + Work , data = segmentation)
plot(fit)

ldapred <- predict(fit, segmentation)

ld <- ldapred$x

ld

anova(lm(ld[,1]~segmentation$segment))

anova(lm(ld[,2]~segmentation$segment))

anova(lm(ld[,3]~segmentation$segment))

anova(lm(ld[,4]~segmentation$segment))

anova(lm(ld[,5]~segmentation$segment))

anova(lm(ld[,6]~segmentation$segment))
```

```
anova(lm(ld[,7]~segmentation$segment))


pred.seg <- predict(fit)$class


cf<- table(segmentation$segment, ldapred$class)
cf

#overal accuracy of the predicting model
sum(diag(cf))/nrow(segmentation)



##rfm analysis ----

data <- data %>%
  mutate(revenue = Quantity * UnitPrice)

rfm <- data

data2 <- data %>%
  filter(!is.na(CustomerID))

rfm <- data %>%
  group_by(CustomerID) %>%
  summarise(
    revenue = sum(revenue),
    number_of_orders = n_distinct(InvoiceNo),
    recency_days = round(as.numeric(difftime(as.POSIXct("2021-11-24 17:06:00 UTC",
format = "%Y-%m-%d %H:%M:%S", tz = "UTC"), max(InvoiceDate), units = "days"))),
    purchase = 1,
    zip_code = get_mode(ZipCode))

groups <- 5

## 5.3 Run RFM Analysis with Independent Sort
rfm$recency_score_indep <- ntile(rfm$recency_days*-1, groups)
rfm$frequency_score_indep <- ntile(rfm$number_of_orders, groups)
rfm$monetary_score_indep <- ntile(rfm$revenue, groups)
rfm$rfm_score_indep <- paste(rfm$recency_score_indep*100 +
rfm$frequency_score_indep * 10 + rfm$monetary_score_indep)
rfm$recency_score_seq <- ntile(rfm$recency_days*-1, groups)
r_groups <- NULL; rf_groups <- NULL; temp <- NULL ## Initialize empty matrices

for (r in 1:groups) {
```

```
  r_groups[[r]] <- filter(rfm, rfm$recency_score_seq == r)
  r_groups[[r]]$frequency_score_seq <- ntile(r_groups[[r]]$number_of_orders, groups)
  for (m in 1:groups) {
    rf_groups[[m]] <- filter(r_groups[[r]], r_groups[[r]]$frequency_score_seq == m)
    rf_groups[[m]]$monetary_score_seq <- ntile(rf_groups[[m]]$revenue, groups)
    temp <- bind_rows(temp, rf_groups[[m]])
  }
}

rfm_result <- temp[order(temp$CustomerID),]
View(rfm_result)
rfm_result$rfm_score_seq <- paste(rfm_result$recency_score_seq*100 +
rfm_result$frequency_score_seq * 10 + rfm_result$monetary_score_seq)

## Export RFM Results with Independent and Sequential Sort
write.csv(rfm_result, "Q:/Marketing Analytics/rfm_results.csv", row.names = FALSE) ##
Name file rfm_result.csv


rfm_result <- data.frame(rfm_result)


##customer segmentation for rfm results
rfm_result <- rfm_result %>%
  mutate(
    Segment2 = case_when(
      recency_score_seq <= 2 & frequency_score_seq >= 4 & monetary_score_seq >= 4 ~
"Champions",
      recency_score_seq <= 3 & frequency_score_seq >= 3 & monetary_score_seq >= 3 ~
"Loyal Customers",
      recency_score_seq <= 2 & frequency_score_seq <= 3 & monetary_score_seq <= 3 ~
"Potential Loyalist",
      recency_score_seq >= 4 & frequency_score_seq >= 3 & monetary_score_seq >= 3 ~
"At Risk",
      recency_score_seq == 1 & frequency_score_seq <= 2 & monetary_score_seq <= 2 ~
"New Customers",
      recency_score_seq <= 3 & frequency_score_seq == 2 & monetary_score_seq == 2 ~
"Promising",
      recency_score_seq >= 4 & frequency_score_seq <= 2 & monetary_score_seq >= 2 ~
"Hibernating",
      frequency_score_seq >= 4 & monetary_score_seq <= 2 ~ "Price Sensitive",
      TRUE ~ "Others"
    )
  )

#join rfm table and segmentation table
join <- inner_join(rfm_result, segmentation, by = "CustomerID")
```

```
segment_counts <- join %>%
  group_by(Segment2) %>%
  summarise(Count = n())


print(segment_counts)



library(ggplot2)

#this is created because when running the second time seed was not included and
segments numbers changed
join <- join %>%
  mutate(segment = case_when(
    segment == 1 ~ 2,
    segment == 2 ~ 7,
    segment == 4 ~ 6,
    segment == 5 ~ 4,
    segment == 6 ~ 5,
    segment == 7 ~ 1,
    TRUE ~ segment
  ))


join$segment <- as.numeric(join$segment)


#bar plot of customer segmentation with RFM
ggplot(segment_counts, aes(x = reorder(Segment2, -Count), y = Count, fill =
Segment2)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  theme_minimal() +
  labs(x = "Segment", y = "Customers", title = "Customer Segment Distribution") +
  coord_flip() +
  scale_fill_brewer(palette = "Set2") +
  theme(
    axis.text.y = element_text(face = "bold")
  )

#bar plot of customer combining both types of segmentation done
ggplot(join, aes(x = Segment2, fill = segment)) +
  geom_bar(show.legend = FALSE) +
  theme_minimal() +
```

```
  labs(x = "Segment", y = "Customer Count", title = "Customer Distribution by previous Segments") +
  facet_wrap(~ segment, scales = "free") +
  theme(
    strip.text = element_text(face = "bold"),
    axis.text.y = element_text(face = "bold"),
    axis.text.x = element_text(angle = 45, vjust = 0.5)
  )


##k-means for rfm results ----


hclust4 <- hclust(dist(scale(cbind(rfm_result$recency_score_seq, rfm_result$frequency_score_seq, rfm_result$monetary_score_seq))), method = "complete")

y <- sort(hclust$height, decreasing = TRUE)[1:10]
plot(x,y); lines(x,y, col= "blue")

kmeans_rfm <- kmeans(x = data.frame(rfm_result$recency_score_seq, rfm_result$frequency_score_seq, rfm_result$monetary_score_seq), centers = 8, nstart = 50)


segmentrfm <- kmeans_rfm$cluster
segmentationrfm <- cbind(rfm_result, segmentrfm)
segment_countsrfm <- table(segmentationrfm$segmentrfm)

segment_datarfm <- as.data.frame(segment_countsrfm)

names(segment_datarfm) <- c("Segment", "Count")


ggplot(segment_datarfm, aes(x = Segment, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = Count), vjust = 0) +
  labs(title = "Customers in each segment", x = "Segment", y = "Customers") +
  theme_minimal()
```