ROME BUSINESS SCHOOL
BETTER MANAGERS FOR A BETTER WORLD

ORACLE

International Master in Data Science

Rome Business School

Intake, Year: October 2023

Capstone Project by:

Laura Borri

Dario Casamassima

Sara Monteiro

Ravidu Kumara Pathirage Don Rupasingha

Data-Driven Marketing Strategies for ORA-FASHION Ltd:

K-means Clustering for Customer Classification and Comprehensive Customer Lifetime Value Analysis

Rome, 26/08/2024

Overview

---

1. Introduction **4**
1.1 Background and Overview of the Project 4
1.2 Nature of the Project and Rationale 4
1.3 Significance of the Project 5
1.4 Aims and Objectives 6
2. Executive Summary **7**
2.1 Business Problem 7
2.2 Company Vision & Mission 7
2.3 Company Business Needs & Competitive Advantage 7
2.4 Target Group 8
2.5 Technological and Strategic Opportunities 8
2.6 Risks 8
3. Literature Review **8**
3.1 Theoretical Framework & Current Knowledge 9
4. Methodology **10**
4.1 Database 11
4.2. Data Preprocessing and Curation 12
4.3 Statistical Analysis Methods 13
5. Results **16**
5.1 RFM & CLV Descriptives 16
5.2 CLV (Customer Lifetime Value) 18
5.3 Clustering for customer segmentation 20
5.4 Cluster Analysis and Radar Chart Profiling 29
6. Discussion **31**
6.1 General summary 31
6.2 Customer Group RFM 32
6.3 Customer Group Life Time Value 32
6.4 Interpretation of Segments and Strategic Recommendations 34
6.5 Deliverables for the Client 35
7. Conclusion **36**
References **37**
Appendix **38**
A) Customer Profiling Recommendation Guidelines and Tools 38
B) Interactive Dashboard 40
C) Python Script 44

Index

Index of Tables

1. **Table 1:** Overview of calculated features used in the dataset and analyses resulting from RFM and CLV steps.
2. **Table 2:** Summary statistics for measurement features based on RFM and CLV.
3. **Table 3:** PCA Component Loadings for Customer Behavior Metrics.
4. **Table 4:** Evaluation Metrics for Cluster K.
5. **Table 5:** Radar Plot Summary.
6. **Table 6:** 'Psychometrics' of Customer Behavior & Risk Management: Table for Decision Makers and a Behavioral Composite Score.
7. **Table 7:** Descriptives Summary of Behavioral Measures Grouped for Country and Cluster Membership.

Appendix:

8. **Appendix Table I:** Data-driven Customer Characterization.
9. **Appendix Table II:** Proposed Personalized Marketing Strategies Based on Customer Segments.

Index of Figures

1. **Figure 1:** Histogram plot of CLV, indicating a right skew.
2. **Figure 2:** CLV categorized by Geographical location/country.
3. **Figure 3:** Bar graph depicting outlier proportions.
4. **Figure 4:** Heat-map correlation matrix for detection of multicollinearity.
5. **Figure 5:** The cumulative explained variance plot indicates the proportion of total variance captured by each principal component.
6. **Figure 6:** Comparison of PCA dimensionality reduction and K-Means Clustering Scatterplots.
7. **Figure 7:** Elbow Method in K-Means Clustering for Determining Optimal k.
8. **Figure 8:** Evaluation of Clustering Quality Using Silhouette Scores.
9. **Figure 9:** 3D Visualization of Customer Clusters Based on Top 3 Principal Component Analysis.
10. **Figure 10:** Bar-Graph Distribution Plot for Customer Cluster Groups.
11. **Figure 11:** Radar Plots for Deep-Phenotyping of Customer Groups based on Cluster Membership.
12. **Figure 12:** Histogram Matrix for Customer Behaviors for Each Segmented Cluster Group.

Appendix:

13. **Appendix Figure I:** Pie Charts for Various Features.
14. **Appendix Figure II:** Bar Graph Matrix for Key-Demographic Features.
15. **Appendix Figure III:** RFM Curve Matrix for Various Key Features.
16. **Appendix Figure IV:** Comparison of RFM and CLV Bar Graphs.
17. **Appendix Figure V:** Heat Maps for Top 10 SKU Items (Age & Geography over Gender, Cluster).
18. **Appendix Figure VI:** Heat Maps for SKU Macrocategories (Age & Geography over Gender, Cluster).

1. Introduction

1.1 Background and Overview of the Project

In the European fashion industry, ORA-FASHION Ltd. is a relatively new retailer that has quickly gained recognition for its high-quality clothing that is made using a transparent and fully traceable "made in Italy" supply chain. ORA-FASHION had great early success and a solid strategic positioning, However, in order to maintain and accelerate future growth, it is advisable to further understand its customer base. Through better alignment with the changing needs and preferences of its diverse clientele, the company hopes to improve its customer-centric marketing campaigns.

Businesses that successfully implement the power of data analytics in the current highly competitive markets are better positioned to achieve a sizable competitive advantage. Advanced data analysis techniques enable companies like ORA-FASHION to extract actionable insights from their extensive customer data repositories. These insights can drive informed decision-making and empower the company to maintain its edge in an increasingly crowded market. Through a deeper comprehension of consumer behavior and preferences, ORA-FASHION can better customize its marketing strategies to appeal to specific target audiences, strengthening client relationships and promoting brand loyalty in the process.

1.2 Nature of the Project and Rationale

In this Capstone project, two datasets are presented containing purchasing data from the ORA-FASHION company, which was collected with Cloud database provider ORACLE. This dataset is meant to emulate a real-world scenario of working with a client for market research purposes, to provide a data analysis service including a deliverable outcome. In order to tackle this challenge, an extensive analysis plan was established, to support ORA-FASHION in its mission to develop more targeted marketing strategies and customized offers. In line with this, a comprehensive analysis of online customer purchasing data will be performed in various steps and with a multitude of insights, guiding future marketing strategies and decisions for the client.

The analysis uses segmenting customer behavior based on their buying habits and estimating their Customer Lifetime Value (CLV), based also on Recency, Frequency, and

Monetary (RFM) components. Moreover, the fashion industry is characterized by rapidly changing consumer preferences and intense competition. In such an environment, the ability to deliver personalized marketing messages that resonate with specific customer segments is crucial, as can be achieved by clustering. Well-established methods such as K-means or Principal Component Analysis (PCA) can therefore be used for clustering, to provide a more nuanced and actionable understanding of customer behavior.

The insights gained from this analysis will enable ORA-FASHION to design marketing campaigns that are not only more effective but also more efficient, by focusing on the most impactful customer segments. The rationale behind this approach is rooted in the recognition that not all customers have the same needs and preferences and seek the same value relationships with a company. By identifying the most valuable customer groups, ORA-FASHION can optimize its marketing efforts, directing resources toward the segments that are most likely to drive long-term growth and profitability.

1.3 Significance of the Project

The significance of this project lies in its potential to transform ORA-FASHION's approach to marketing and also, in general currently used marketing research techniques. Historically, marketing research has been characterized by tedious methods, such as surveying, experimenting and making observations, which is time-consuming, costly, and subject to interpretation biases due to subjectivity. By moving away from broad, one-size-fits-all marketing strategies toward more precise, data-driven segmentation, the company can achieve higher customer engagement and conversion rates. In fact, given that ORA-FASHION uses online retailing, these traditional methods cannot be applied in the same way. Online shops can produce vast and intricate amounts of data that require the application of various statistical and ML driven data analysis techniques, to derive the right observations and conclusions from it.

Additionally, the project will help ORA-FASHION optimize its marketing budget by allocating resources more effectively, ensuring that the most valuable customers receive the most attention. This targeted approach is expected to contribute significantly to the company's long-term goal of cultivating customer loyalty and driving sustainable growth. Moreover, the current analysis protocol incorporates a series of analysis techniques that enable text mining, feature engineering and various ML driven analysis techniques that enable the sourcing of both precise and layered information from the broad ‘big-dataset’ provided. We therefore with this project approach propose an innovative, efficient and objective methodology for

approaching marketing research problems with big-datasets, based on the patterns intrinsic to the datasets.

1.3.1 Personal Part

As students of data science, we are deeply motivated by the opportunity to apply analytical techniques to real-world business challenges. The decision to work with ORA-FASHION Ltd. was driven by our shared interest in exploring how data can be harnessed to shape customer experiences in the fast-paced fashion industry. The company's commitment to personalization and customer-centric marketing strategies resonated with our desire to work on a project that bridges data science and business impact. By contributing to ORA-FASHION's mission, we aim to deepen our understanding of consumer behavior and refine our skills in data-driven decision-making, which are crucial for our future careers in this field and to become key decision-makers ourselves.

1.3.2 Scientific Part

The methodology outlined in this report serves as a strategic and replicable protocol for in depth analysis of purchasing behaviors, while accounting for individual demographic characteristics. As such it is not only an applied and replicable rigorous methodology for analysts working in the field, but can also incentivize scientific endeavors and research questions alike. Segmentation studies are particularly useful in the context of an inductive 'bottom-up' approach, where patterns in the data are identified and interpreted for meaningful insights, contrary to the rather traditional scientific approach of 'confirmatory' research or prediction models. This can be of particular importance for situations in which guidance is required based on heterogenous and dynamic datasets, such as for e-commerce marketing research, to facilitate not only our understanding of behavioral groups, but to improve the positioning of companies through improved targeting of customer groups via marketing campaigns (Schlegelmilch, 2022).

1.4 Aims and Objectives

The immediate primary aim of this project is to enhance ORA-FASHION's customer-centric marketing strategies by developing a deep, data-driven understanding of customer behavior and providing in depth-expertise based on the natural patterns of the data available. The ultimate long-term goal is to maximize customer loyalty and brand binding through the achieved insights and corresponding marketing strategy adaptations. The specific objectives of the project proposed are:

1. **Estimate RFM:** To categorize customers into different segments, allowing for more efficient allocation of marketing resources.
2. **Estimate CLV:** To estimate the CLV of each customer segment, providing insights into the long-term value they bring to the company.
3. **Purchasing behavior based customer segmentation:** To apply clustering to segment customers into distinct groups based on their purchasing behavior.
4. **Create interactive dashboard for client:** Create a deliverable for the client which facilitates their understanding of customers and behavioral patterns, for in-depth improvement of current marketing campaigns and strategies.
5. **Provide expert data-drivcen guidance:** To use the insights gained from customer segmentation and RFM/CLV estimation to design targeted marketing campaigns that maximize customer engagement and conversion rates, based on customer profiling outcomes.

In summary, ORA-FASHION's objectives of improving marketing tactics and expanding client understanding are strategically matched with this project. The project intends to use advanced data analytics to produce actionable insights that will lead to more successful and efficient marketing campaigns, ultimately supporting the company's goal of long-term growth and client loyalty.

2. Executive Summary

2.1 Business Problem

ORA-FASHION's current marketing approaches are not sufficiently targeted, which risks missed opportunities and reduced customer retention. The vast amount of data generated by the online retail sector requires novel analysis techniques to uncover actionable insights. Addressing this challenge is crucial for ORA-FASHION to stay competitive and sustain growth in a dynamic market.

2.2 Company Vision & Mission

ORA-FASHION envisions itself as a premier European fashion brand, recognized not only for its dedication to "made in Italy" craftsmanship but also for its innovative marketing strategies. The company’s mission extends beyond delivering superior fashion products; it aims to foster lifelong relationships with its customers by offering personalized experiences tailored to diverse customer needs.

2.3 Company Business Needs & Competitive Advantage

To thrive in the highly competitive fashion industry, ORA-FASHION must effectively segment its customer base using RFM and CLV metrics. By tailoring marketing campaigns to specific customer segments, the company can maximize customer engagement and loyalty, thereby securing a sustainable competitive edge. The integration of advanced data analysis tools and continuous collaboration with data scientists will enable ORA-FASHION to create a dynamic and efficient marketing strategy.

2.4 Target Group

The primary target of this initiative is ORA-FASHION's online customer base across European partner countries (France, Germany, UK, Netherlands, Spain, Greece, Italy), and based on the approach, it can aid the exploration and expansion to other European and international markets.

2.5 Technological and Strategic Opportunities

ORA-FASHION is advancing its technological capabilities by investing in cloud-based databases and sophisticated data analysis techniques. These advancements will enable the company to refine its customer segmentation and deliver personalized marketing efforts with greater precision. The project's success will not only bolster ORA-FASHION's current market position in Europe but also open doors to new markets both within and outside of Europe. Potential growth strategies include expanding the product line, forming joint ventures with sustainable fashion brands, and further technological enhancements to improve customer experiences.

2.6 Risks

The primary risks involve potential data quality issues that could lead to inaccurate segmentation and ineffective campaigns. Additionally, over-segmentation could dilute marketing efforts, and external factors like economic downturns or shifting consumer preferences could impact the effectiveness of strategies. To mitigate these risks, ongoing collaboration between ORA-FASHION and the data science team is essential for continuous monitoring and refinement of customer segmentation and marketing strategies..

3. Literature Review

The fashion industry has rapidly evolved, with e-commerce playing an increasingly significant role in consumer shopping behaviors throughout the world. It is hypothesized that part of the recent massive growth in e-commerce generated revenue and capitalization was introduced due to COVID-19 related implications and restrictions, with an estimated growth of 26,7 trillion dollars for retail sales since the pandemic, according to the UN Trade and Development Organization's (UNCTAD) reports (United Nations, 2021). Furthermore, according to recent ‘Statista’ reports, e-commerce revenue has been estimated to generate $1,664 billion in Asia, $889,1 billion in the Americas and on place three globally, $533 billion in Europe (Statista, 2024). The massive trend of increasing e-commerce usage has been reported by the ‘Europe E-Commerce Report 2023’ as well for European markets, with a 3% growth since 2020 (3-year growth), and a 10% growth since 2018 (5-year growth). It is interesting to note that, when it comes to product categories, "clothes, shoes, and accessories" have the biggest purchasing impact in Europe (68%), second only to "any physical goods" (97%). According to the report's additional estimate, 87% of Europeans made purchases online in 2023 (Lone & Weltevreden, 2023). Against this background, understanding customer purchase patterns and behaviors is crucial for tailoring effective marketing strategies for long-term growth of companies, with a very big potential for e-commerce based companies in Europe for the fashion industry.

3.1 Theoretical Framework & Current Knowledge

Various analytical models for understanding intricate and complex e-commerce consumer patterns have been proposed, particularly predictive and data mining models have found popularity such as Cirqueira et al. (2019) prediction framework or Qiu, Lin & Li’s (2015) COREL model for purchasing prediction, among others. These models are useful for well-defined predictive scenarios, which are not always available when little is known and when a more dynamic analysis is required. Segmentation models are advantageous for heterogenous settings like behavioral datasets with very different customer groups, as can be expected in a European landscape.

Because of the complexity, diversity, and trend sensitivity of the European fashion e-commerce market, we here propose a data analysis pipeline that includes RFM analysis, CLV modeling, and clustering segmentation. RFM analysis allows retailers to reflect on purchasing patterns, enabling the understanding of different European consumers. CLV modeling further enhances this by identifying the long-term value of customers, guiding

efficient resource allocation and fostering loyalty in a market where customer retention is key, especially the fashion market. Clustering, with its ability to distill complex, multidimensional data into meaningful components, helps uncover hidden patterns in consumer behavior across Europe's culturally and economically diverse regions. Together, these analytical approaches applied to the fashion e-commerce businesses in Europe can disentangle the complexities of a fragmented market, staying ahead of trends and competition while fostering strong, long-term customer relationships.

3.1.1 Strengths and Potential Impact

The main advantage of our approach is its high degree of precision in condensing complex and large-scale datasets into actionable insights for the client's marketing campaign. This impacts improved customer engagement, optimizes resource allocation, and eventually boosts conversion rates. Furthermore, this approach fills a critical gap in the way financial reports are currently used in marketing strategy development. Financial reports typically provide aggregate and predictive data, offering insights at a macro level. While useful for understanding overall business performance, they often fail to capture the granular details necessary for personalized marketing.

3.1.2 Limitations

While the project offers numerous benefits, it is not without limitations. One of the primary challenges is the reliance on historical purchase data, which may not fully capture the dynamic nature of customer behavior, particularly in the fast-paced fashion industry, as it is based on non-recent and fictional data. Additionally, the input data directly affects the quality of the output data, meaning that if the data is incomplete or inconsistent, the resulting customer segments may be less accurate, affecting potentially leading to suboptimal marketing decisions. We have aimed to solve some of these problems by fine-graining our dataset with feature engineering, to create a more in-depth analysis. This project thus posed particularly useful for making the most of a dataset when relatively little information is known and present, which comes close to the 'real-world' work setting of data scientists.

4. Methodology

To conduct a comprehensive analysis of the client's purchase data and address our objectives, the following steps were undertaken using Python scripting utilizing various libraries and Tableau:

1) A data pre-processing and curation approach consisting of a) Data collection and feature engineering as a data curation step, b) data cleaning, c) data integration.

2) A data analysis pipeline consisting of a) descriptive analysis, b) RFM and CLV calculations, c) segmentation analysis, d) visualization techniques.

3) Creation of deliverables in form of informed recommendation guidelines and an interactive dashboard, based on the feature engineered big-dataset, which provides in-depth, nuanced and intuitive knowledge on the customer base.

4.1 Database

Two ORA-FASHION datasets containing historical customer purchase data provided by ORACLE will be used for the analysis. ORACLE provides Cloud databases to companies, with this dataset containing a 1-year data collection of fashion retail purchases from the year 2021.

4.1.1 Customers dataset

This dataset spreadsheet contains detailed data on customers, including key attributes such as demographics (age, gender, geographical location, customer ID), purchasing behavior, and possibly customer segmentation information. This data is essential for analyzing customer profiles, identifying target audiences, and understanding customer needs and preferences. A total of 22626 unique customer records are included in the dataset.

4.1.2 Orders dataset

This file includes order-related data, capturing details such as order dates, quantities, prices, and customer information associated with each transaction (purchase data such as transaction IDs, stock keeping units (SKU) and their overall SKU category, purchase quantity and the amount of sales per item). Analyses of this dataset enables understanding of reflects and is likely used to analyze sales trends, order volumes, and revenue generation, as well as to link customer behavior with purchasing patterns. Meta-data information presented within the dataset aids in the correct interpretation of items and variables.

They also provide us a file with descriptions of the fields or variables present in the dataset, like a metadata document. It is crucial for interpreting the data correctly, as it explains what each field represents, the data types used, and any relevant notes on how the data should be understood or used in analysis.

4.2. Data Preprocessing and Curation

4.2.1 Data Preparation

The data preparation phase involves meticulous handling and transformation of the datasets to ensure they are suitable for analysis. This step is crucial for maintaining data quality, robustness, and minimizing bias. Two datasets containing historical customer purchase data provided by ORACLE-FASHION will be used for the analysis. ORACLE-FASHION offers a comprehensive collection of fashion retail purchases spanning one year (2021). The datasets encompass 22,626 unique customer records and 131,707 transactions from European customers, specifically from the UK, Spain, Germany, France, Greece, and the Netherlands. To optimize our analysis and gain more insightful results, we seamlessly integrated the two datasets provided by ORACLE-FASHION into a single, unified dataset, aptly named the "Merged Dataset." This consolidation allows for a comprehensive and cohesive exploration of customer data, ensuring that all relevant information is captured and analyzed together.

4.2.2 Data Collection and Feature Engineering

The data was collected from the ORA-FASHION's database and includes variables of transaction and customer details such as 'customer ID', 'Date', 'Customer_ID', 'Transaction_ID', 'SKU_Category', 'SKU', 'Quantity', 'Sales_Amount', 'GENDER', 'age', 'geography', 'description'.

4.2.3 Data Integration

In our data preparation process, we combined multiple datasets to create a unified, comprehensive dataset that integrated transaction details, product descriptions, and customer information.

Once the transaction data was enriched with product descriptions, we further enhanced the dataset by incorporating customer details. This was achieved through another inner join, this time linking the combined transaction-product dataset with customer information based on a common identifier, the 'Customer_ID'. This step allowed us to add demographic and geographical data to each transaction, providing a more holistic view of the customer base. The merging process was carefully checked to ensure that all data was consistent and relevant, with no missing or mismatched entries, making the dataset robust for further analysis.

4.2.4 Data Cleaning & Quality Control

A missing value step was performed (checked for NaNs or NULL values), as well as a duplicate removal. Additionally, an outlier detection approach using an Isolation Forest algorithm was performed. Every analytical step was controlled for logical consistency: e.g., it was controlled whether 'Sales_Amount' contains any negative values, which would not make sense in the context of sales. It was further ensured that all columns had the appropriate data types, such as dates for 'Date', integers for 'Quantity' and 'Customer_ID', and floats for 'Sales_Amount'.

4.3 Statistical Analysis Methods

Alongside general descriptive and visualization techniques, the analysis pipeline of this project was structured into three key components: 1) RFM analysis, 2) CLV analysis, 3) clustering (using PCA, K-means and various diagnostic steps like Elbow Method, Silhouette Method, Calinski Harabasz score and cluster distribution. Python libraries such as numpy, scikitlearn, seaborn, matplotlib and pandas were used to write the code and run the graphs. All analysis steps are described in detail alongside with the code in the Python Script of the project, which is outlined in the Appendix section C).

4.3.1 Descriptive Statistics

- Summary Statistics: Calculations of mean, median, mode, standard deviation, and other relevant statistics for numerical columns.
- Frequency Analysis: Counted the occurrences of categorical values such as 'SKU_Category', 'gender', and 'geography', to have a first understanding on the nature of the data.
- Layered descriptives: Additionally accompanied by descriptive tables and graphs filtered for the measurement metrics by country and cluster memberships.

4.3.2 RFM Analysis

One of the most popular methods for segmenting customers is clustering based on the Recency, Frequency, and Monetary value (RFM) of their purchasing behavior. This method is widely used because it offers a fast and straightforward framework for quantifying customer behavior, making it easy to implement (Kahan, 1998; Miglautsch, 2000) presented a method for sequential pattern mining using RFM segmentation, offering a more dynamic view of customer behavior. Further expanding on RFM, Peker et al. (2017) suggested incorporating

customer relationship length and purchase periodicity into the segmentation process, providing a more comprehensive understanding of customer dynamics.

Based on RFM calculations, customers were subsequently grouped based on their behaviors and preferences in their e-commerce behaviors. Recency, Frequency and Monetary value will be calculated as a new feature in the dataset (also collectively termed revenue). Combined with CLV analysis capturing customers lifetime value to a company, they are key aspects of the subsequent unsupervised machine learning clustering approach to identify different groups (clusters).

The RFM components were respectively: R = 'days since last purchase'; F = 'total transactions', 'total products purchased' ; M = 'total spent', 'average transaction value'. A detailed description of each step and component can be found in the corresponding Python script (see Appendix section C). In sum these new features resulted in five key KPIs for each customer. Additional measurement metrics calculated were: 'Product diversity' (unique products purchased per subject); other behavioral indicators such as 'average days between purchases', 'day of week', 'hour'. The RFM features generated from this approach will aid the understanding of behavioral metrics in relation to customer demographics, and will be part of the CLV approach.

4.3.3 CLV Analysis

1) Customer lifetime value was determined first by estimating an average customer lifespan using a commonly used formula.

$$Average\ Customer\ Lifespan\ (in\ years) = Mean\ of\ Average\ Days\ Between\ Purchases\ /\ 365$$

2) To generate new CLV features, the below formulas were applied and finally iterated for each customer (interpreted as the total time span over the lifecycle of a customer). Other components of this analysis have been generated in the RFM analysis and will not be repeated here.

$$CLV = (Average\ Transaction\ Value) \times (Total\ Transactions) \times (\ Average\ Customer\ Lifespan)$$

Results were visualized as a histogram plot for understanding of the distribution and a country-wise categorization of CLV box-plots was created.

Table 1 depicts the variables/features derived from these initial feature engineering and text mining techniques for the subsequent analysis steps, based on RFM and CLV calculations.

Table 1: Overview of calculated features used in the dataset and analyses resulting from RFM and CLV steps.

Feature Name	Description
Days_Since_Last_Purchase	Number of days since the customer's last purchase.
Total_Transactions	Total number of transactions made by the customer.
Total_Products_Purchased	Total number of products purchased by the customer across all transactions.
Total_Spend	Total amount of money spent by the customer.
Average_Transaction_Value	Average amount spent per transaction by the customer.
Unique_Products_Purchased	Number of different products purchased by the customer.
Average_Days_Between_Purchases	Average number of days between each of the customer's purchases.
Day_Of_Week	The day of the week when the transaction occurred (e.g., Monday, Tuesday).
Hour	The hour of the day when the transaction occurred (24-hour format).
Monthly_Spending_Mean	The average monthly spending by the customer.
Monthly_Spending_Std	The standard deviation of the customer's monthly spending, indicating variability.
Spending_Trend	The direction and rate of change in the customer's spending over time.
CLV	Customer Lifetime Value, total expected revenue throughout a customer's life.

4.3.4 Correlation Analysis

Correlation analyses were performed as a control for multicollinearity between features. The presence of multicollinearity could harm the interpretability of cluster outcomes, by not allowing the model to learn the actual underlying patterns in the data, as the features do not provide unique information. In such an instance, clusters would not separate well and provide less meaningful. PCA was applied for dimensionality reduction to control for instances where multicollinearity was detected. It aids in neutralizing multicollinearity by applying transformations as a new uncorrelated set of variables while preserving most of the dataset's variance.

4.3.5 Feature Scaling (Standardization)

Before proceeding with any further analyses, variables were standardized to a mean of 0 and a standard deviation of 1. Features that do not require scaling standardization were left out (e.g. Customer ID; Day of Week).

4.3.6 Dimensionality Reduction (PCA)

PCA was applied as a dimensionality reduction step due to its' excellence in capturing linear relationships (relevant due to presence of multicollinearity). This approach also aids at identifying the most important components for customer segmentation. Factor loadings were

further inspected, with higher values in the PCA loadings table indicate that a feature significantly contributes to the variation captured by the corresponding principal component.

4.3.7 Segmentation Analysis (K-Means Clustering)

K-means clustering was applied as an unsupervised machine learning algorithm for customer group segmentation. The algorithm iteratively assigns each data point to the nearest centroid, then updates the centroids by calculating the mean of all assigned points. The process repeats until convergence or a stopping criterion is reached. To determine the optimal K, both the Elbow and Silhouette methods were applied. The Elbow method identifies the optimal K by inspecting the point of least change in the curve, achieved using the Yellow Brick library. The Silhouette method evaluates cluster separation, with higher scores indicating better cluster performance. Unlike the Elbow Method, which relies on subjective interpretation of inertia, the Silhouette Method offers a more objective and comprehensive evaluation of cluster quality. It visually represents cluster consistency and can detect outliers, making it a reliable tool for determining the optimal k. Calinski Harabasz scores and Davies Bouldin score were used as additional criteria for decision making. Cluster density was also considered, and the final K was chosen based on a thorough evaluation of these results. All steps are outlined in detail in the Python Script.

4.3.8 Visualization techniques

An interactive 3D plot was generated for thorough inspection of top three principal components identified for thorough inspection of data and analysis cohesion. Additionally, bar graphs were created to visualize the proportion of customers for each cluster. Radar charts further facilitate the semantic profiling of customers for each cluster and will aid the creation of recommendation guidelines as a deliverable for the customer (see Discussion section and Appendix section A). To create radar charts, we first calculate the centroid for each cluster, which represents the mean values of all features within that cluster. These centroids are then plotted on the radar charts, providing a clear visualization of central tendencies across different clusters. This method facilitates the understanding of how clusters differ in customer attributes and behaviors, offering insights for targeted marketing strategies and informed business decisions. These will further be supported by histograms (clusters & measurement features).

5. Results

5.1 RFM & CLV Descriptives

The descriptive statistics for the customer data are summarized in *Table 2*.

Table 2: Summary statistics for measurement features based on RFM and CLV.

	count	mean	std	min	0.25	0.5	0.75	max
Days_Since_Last_Purchase	16324	148.65	115.63	0	37	123	263	363
Total_Transactions	16324	3.58	4.51	1	1	2	4	99
Total_Products_Purchased	16324	11.44	23.99	0.5	3	5	11	814.9
Total_Spend	16324	162.35	561.05	1.08	19.89	44.57	124.2	40070.49
Average_Transaction_Value	16324	43.81	160.26	0.55	11.26	19.58	39.93	13167.88
Unique_Products_Purchased	16324	6.15	6.73	1	2	4	7	176
Average_Days_Between_Purchases	16324	1.85	30.42	-266	0	0	2.63	349
Day_Of_Week	16324	2.6	1.95	0	1	2	4	6
Hour	16324	0	0	0	0	0	0	0
Monthly_Spending_Mean	16324	57.84	187.96	0.55	0	0	0	13167.88
Monthly_Spending_Std	16324	30.96	124.04	0	0	0	0	8732.25
Spending_Trend	16324	0.27	135.31	-4603.49	-1.8	0	2.25	12349.26
CLV	15,507.0	18,253.5	25,845.5	214.1	3,809.4	8,178.6	20,705.4	276,450.7
Total_Revenue	15,507.0	1,421,067.3	0.0	1,421,067.3	1,421,067.3	1,421,067.3	1,421,067.3	1,421,067.3

Days Since Last Purchase: On average, customers have not made a purchase for approximately 149 days (SD = 116), indicating a typical inactivity period of nearly five months. The range of 0 to 363 days suggests substantial variation in customer engagement.

Customer Behavior: The dataset reveals considerable variability in customer behavior. The average transaction frequency is 3.58 (SD = 4.51), with a range from 1 to 99 transactions. Customers purchase an average of 11.44 products (SD = 23.99), with some buying as many as 814.9 products.

Spending Patterns: Average total spending is $162.35 (SD = $561.05), with individual transactions averaging $43.81 (SD = $160.26). Outliers are notable, with maximum expenditures reaching $40,070.49 and transaction values up to $13,167.88.

Product Variety and Purchase Frequency: Customers buy an average of 6.15 unique products (SD = 6.73), and the average interval between purchases is 1.85 days (SD = 30.42). However, some inconsistencies in the data affect this metric. Purchases are evenly distributed across the week.

Monthly Spending: Monthly spending averages $57.84 (SD = $187.96), with significant variability. The spending trend shows a slight positive increase (Mean = 0.27, SD = 135.31).

Customer Lifetime Value (CLV): The average CLV is $18,253.5 (SD = $25,845.5), indicating that a small number of high-value customers have a disproportionate impact on total revenue. Cluster analysis suggests balanced customer segmentation. The total revenue of $1,421,067.3 likely represents an aggregate measure.

5.2 CLV (Customer Lifetime Value)

5.2.1 CLV Distribution

The histogram's right-skewed distribution indicates that fewer customers contribute to higher values, with the majority of customers having low CLVs. Since the histogram's peak is located near the right side of the CLV scale, most customers are thought to have relatively low lifetime values, meaning there are many individuals contributing less (1K-40K) revenue over their lifetimes, while there are a few people purchasing so much that they generate in between 50-250K in revenue for the company. In other words, while most customers generate modest revenue, a small number of customers may significantly increase the overall CLV. Considering this, it would be beneficial for ORA-FASHION to focus on identifying and nurturing these high-value customers to maximize profitability, which was further investigated with the clustering. However, likewise ORA-FASHION can maximize customer revenue of the low-moderate spenders by addressing them with targeted marketing.

At face value, the box plot did not show any major differences for the interquartile range up to the 75% range, however also here it reflects that it is in the very high spenders that there was a relatively visible heterogeneity among European countries, with notably more customer lifetime value in Germany, France and Italy compared to other nations. It is advisable for ORA-FASHION to target these other European nations for improving CLV performance through targeted campaigns.

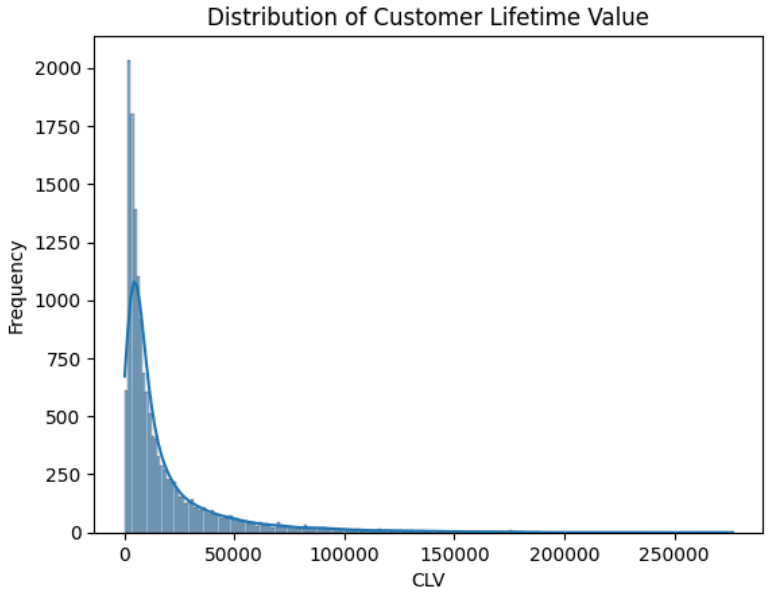

Figure 1: Histogram plot of CLV, indicating a right skew.

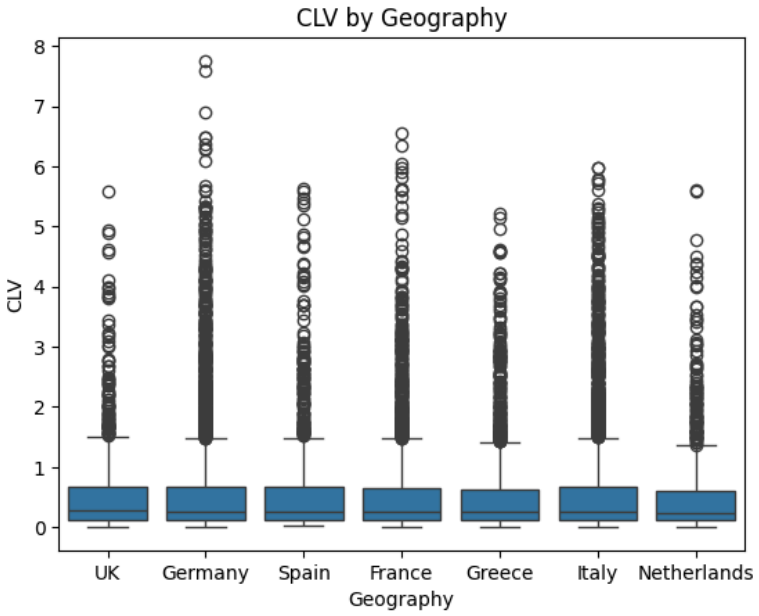

Figure 2: CLV categorized by Geographical location/country.

5.3 Clustering for customer segmentation

5.3.1 Outlier Detection

Application of the isolation forest algorithm showed that approximately 5% of the customers in our dataset were identified as outliers. This proportion is considered reasonable, as it strikes a balance between not losing a substantial amount of data and not retaining potentially noisy data points. To ensure more accurate and meaningful clustering results,

these outliers will be separated for further analysis and removed from the main dataset. This approach will refine the dataset, making it more suitable for the clustering analysis that follows. The use of the isolation forest algorithm appears to have effectively pinpointed these outliers, providing a moderate percentage that is neither too high nor too low. This step is crucial for the subsequent customer segmentation, as outliers can significantly impact the quality of the clusters formed.

It must be noted however, that the insights acquired from the CLV distribution plots indicated that particularly these heterogeneous outliers (people very high in CLV) are of particular interest for ORA-FASHION's marketing insights in separation, but nevertheless present redundancy for the next analytical steps and are thus removed.

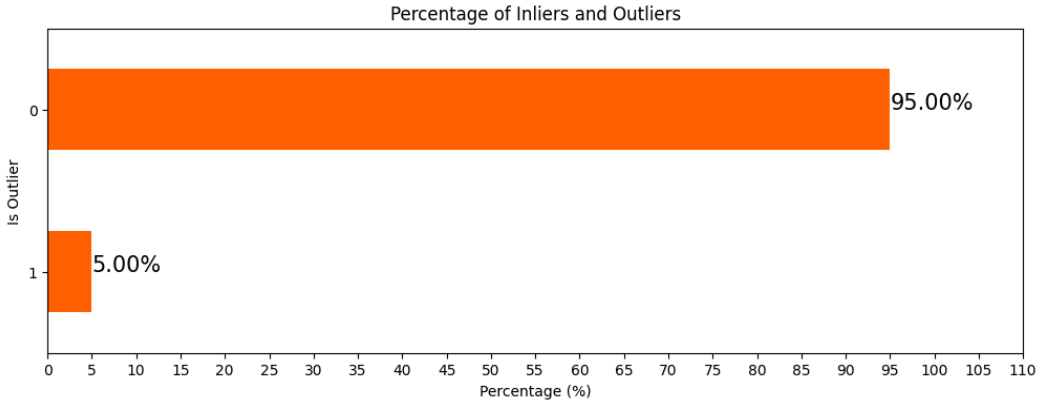

Figure 3: Bar graph depicting outlier proportions.

5.3.2 Correlation analysis

Various strengths of correlation have been identified, causing various points of consideration. Various strong correlations suggest a significant degree of multicollinearity, indicating that these variables move closely together. Multicollinearity can pose challenges in K-means clustering by potentially obscuring true data patterns and leading to less stable clusters. To address this issue, it was decided to deploy PCA as a dimensionality reduction technique prior to K-means clustering, to contribute to the formation of more distinct and stable clusters in the subsequent K-means clustering analysis.

At the same time, as a separate inspection component, the insights derived from these linear correlations indicate increased transaction frequency correlates with higher spending and product variety, while longer intervals since the last purchase are associated with decreased engagement. These insights can nevertheless be useful in isolation to inform targeted marketing strategies aimed at enhancing customer retention and maximizing lifetime value. Further analysis may be warranted to explore causative factors underlying these correlations.

Strong correlations:

The correlation matrix analysis indicates the relationships among various customer behavior *"measurement"* metrics. The findings showed that *'days since the last purchase'* exhibit a significant negative correlation with *'total transactions'* (r = -0.33), *'total products purchased'* (r = -0.25), and *'total spend'* (r = -0.21), suggesting that as the time since the last purchase increases, both the *'frequency of transactions'* and *'total spending'* tend to decrease.

Conversely, *'total transactions'* strongly correlate with *'total products purchased'* (r = 0.77) and *'total spend'* (r = 0.62), indicating that higher transaction frequency is associated with increased product purchases and spending. *'*

Moderate correlations:

Average transaction value' shows a moderate positive correlation with *'total spend'* (r = 0.54) and *'total products purchased'* (r = 0.24), suggesting that higher average transaction values contribute to overall spending. Additionally, *'unique products purchased'* positively correlate with *'total transactions'* (r = 0.70) and *'total spend'* (r = 0.16), indicating that customers who purchase a greater variety of products tend to engage in more transactions and spend more.

Monthly spending metrics indicate that the *'monthly spending mean'* correlates positively with *'total transactions'* (r = 0.36) and *'total spend'* (r = 0.29), while the *'monthly spending standard deviation'* shows a moderate correlation with *'total products purchased'* (r = 0.34). This suggests that higher average monthly spending is associated with increased transaction frequency and product variety.

Weak correlations:

'Average days between purchases' display a weak negative correlation with *'total transactions'* (r = -0.09) and *'total spend'* (r = -0.15), suggesting that longer intervals between purchases may be associated with fewer transactions and lower spending. *'The day of the week'* shows minimal correlation with other variables, indicating it does not significantly influence purchasing behavior in this dataset.

Lastly, spending trends display negligible correlations with other metrics, indicating that they do not significantly impact the variables analyzed. Overall, the correlation matrix reveals significant relationships among customer behavior metrics, with increased transaction frequency correlating with higher spending and product variety, while longer intervals since the last purchase are associated with decreased engagement. These insights can inform targeted marketing strategies aimed at enhancing customer retention and maximizing lifetime value, warranting further analysis to explore causative factors underlying these correlations.

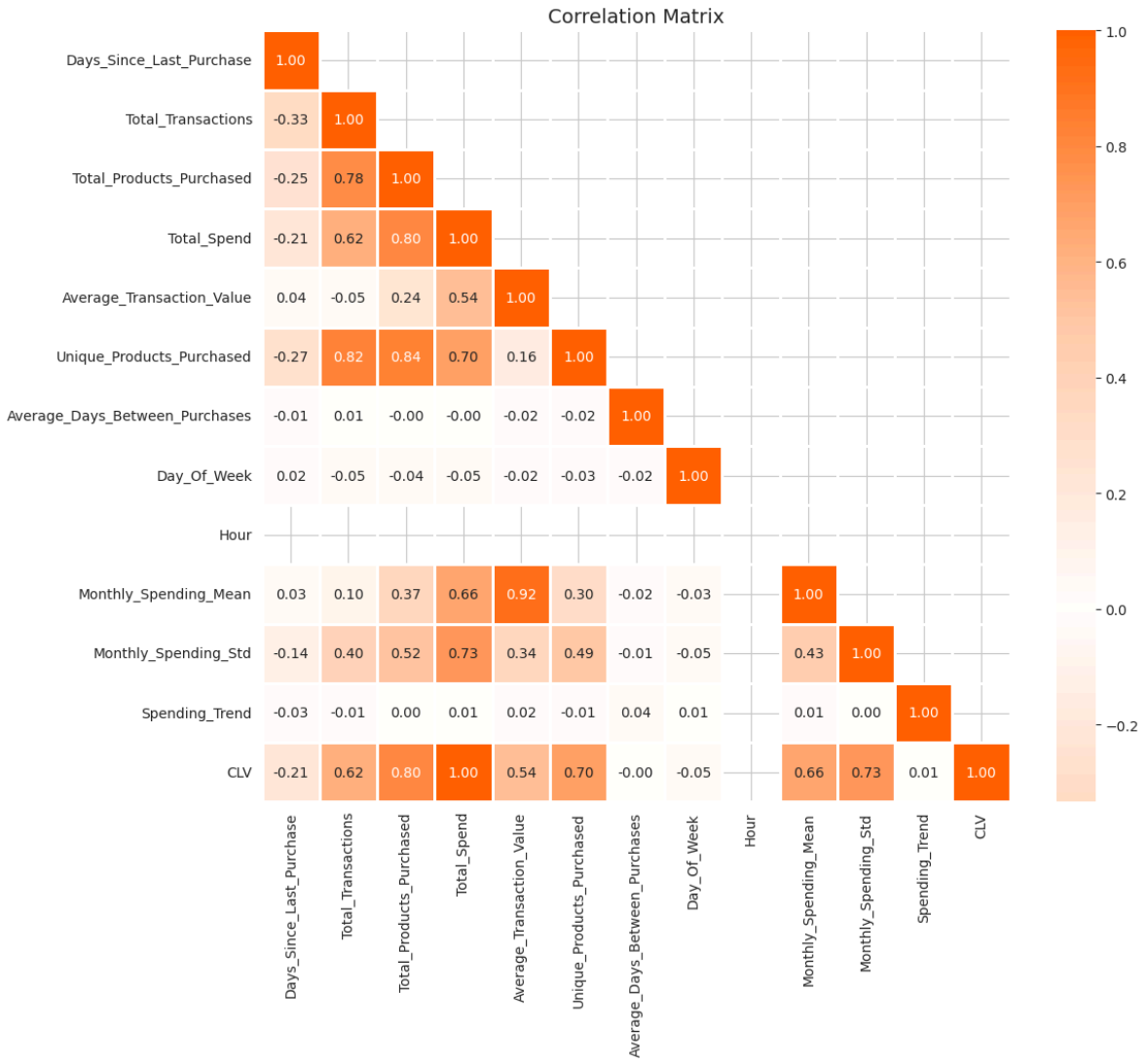

Figure 4: Heat-map correlation matrix for detection of multicollinearity.

The correlation coefficients range from -1 to 1, where values closer to 1 indicate a strong positive correlation, values closer to -1 indicate a strong negative correlation, and values around 0 suggest no correlation.

5.3.3 Feature scaling

Following standardization, PCA was used to handle multicollinearity by reducing the dataset into its most informative components. The first component explains approximately 41% of the variance, while the first two components together account for about 62%. The cumulative variance increases to 76% with the first three components and reaches 92% with the first six components. Beyond the sixth component, the incremental gain in explained variance diminishes, indicating an "elbow point." For customer segmentation, retaining the first six components strikes a balance between preserving a substantial portion of the total variance and reducing the dimensionality of the dataset, making it a practical choice for further analysis.

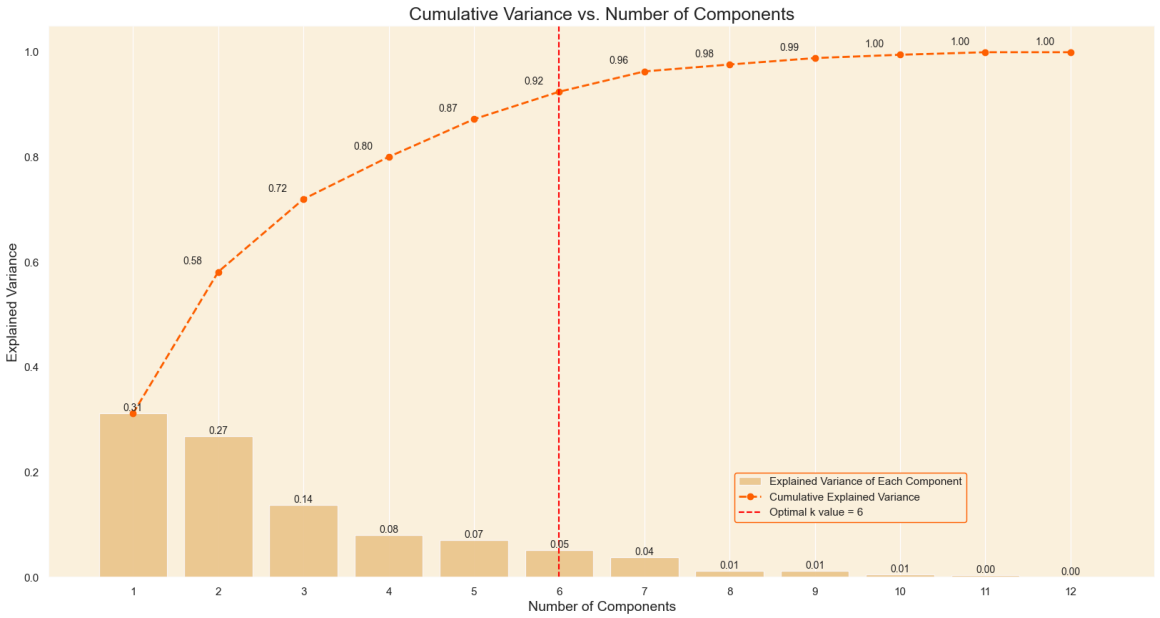

Figure 5: The cumulative explained variance plot indicates the proportion of total variance captured by each principal component.

Table 3: PCA Component Loadings for Customer Behavior Metrics.

	PC1	PC2	PC3	PC4	PC5	PC6
Days_Since_Last_Purchase	-0.12	0.31	-0.12	0.09	0.13	-0.92
Total_Transactions	0.32	-0.42	-0.02	0.01	0.01	-0.13
Total_Products_Purchased	0.38	-0.21	-0.02	-0.01	0.02	-0.14
Total_Spend	0.43	0.07	0.00	0.00	0.01	-0.02
Average_Transaction_Value	0.23	0.57	0.02	-0.02	-0.01	0.20
Unique_Products_Purchased	0.36	-0.28	-0.05	-0.02	0.02	-0.15
Average_Days_Between_Purchases	0.00	-0.02	0.70	0.14	0.70	0.01
Day_Of_Week	-0.03	0.01	-0.26	-0.86	0.43	0.01
Hour	0.00	0.00	0.00	0.00	0.00	0.00
Monthly_Spending_Mean	0.29	0.51	0.01	-0.01	0.00	0.11
Monthly_Spending_Std	0.33	0.06	0.00	0.02	0.01	-0.06
Spending_Trend	0.00	0.02	0.65	-0.47	-0.55	-0.21
CLV	0.43	0.07	0.00	0.00	0.01	-0.02

5.3.4 K-Means Clustering

K-Means is an unsupervised machine learning algorithm that clusters data into a predefined number of groups (K) by minimizing the within-cluster sum of squares (WCSS), also known as inertia. The algorithm iteratively begins by randomly initializing K centroids. Each data point is then assigned to the nearest centroid based on the shortest distance, forming clusters. The centroids are recalculated as the mean of the points within each cluster. This

process of assignment and centroid updating continues until convergence or a stopping criterion, such as a maximum number of iterations or minimal changes in centroid positions, is met. The goal is to form well-separated clusters that minimize the overall WCSS, effectively grouping similar data points.

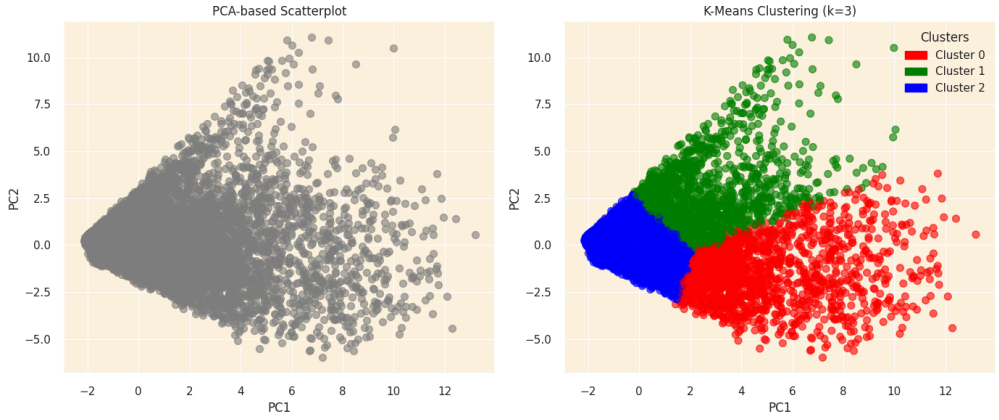

Figure 6: Comparison of PCA dimensionality reduction and K-Means Clustering Scatterplots.

5.3.5 Elbow method

The Elbow Method, applied using the YellowBrick library, was used to determine the optimal number of clusters (k) by identifying the point where the within-cluster sum-of-squares (inertia) begins to decrease more slowly. The plot suggests that k could be optimal around 5, but the elbow is not distinctly clear, with inertia continuing to decrease noticeably up to k=7. This suggests potential optimal k values between 3 and 9. To refine this range, silhouette analysis can be used to assess cluster cohesion and separation. Additionally, incorporating business insights ensures that the chosen k aligns with practical objectives and market segmentation needs.

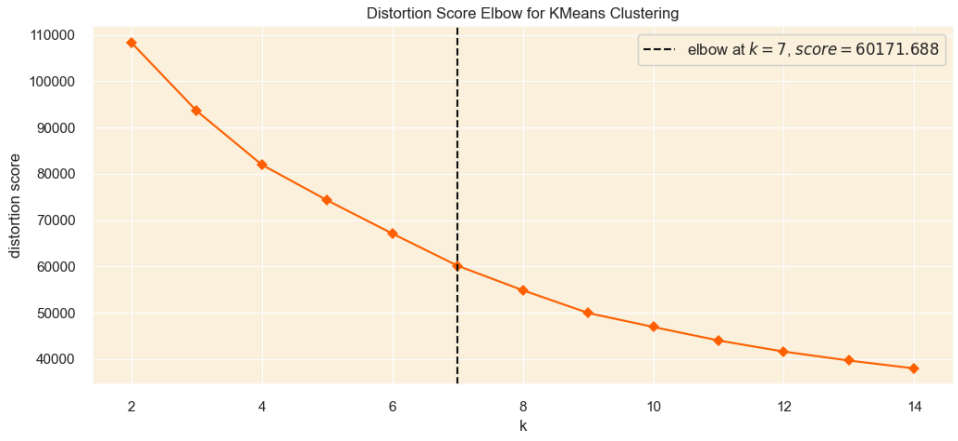

Figure 7: Elbow Method in K-Means Clustering for Determining Optimal k.

5.3.6 Silhouette method

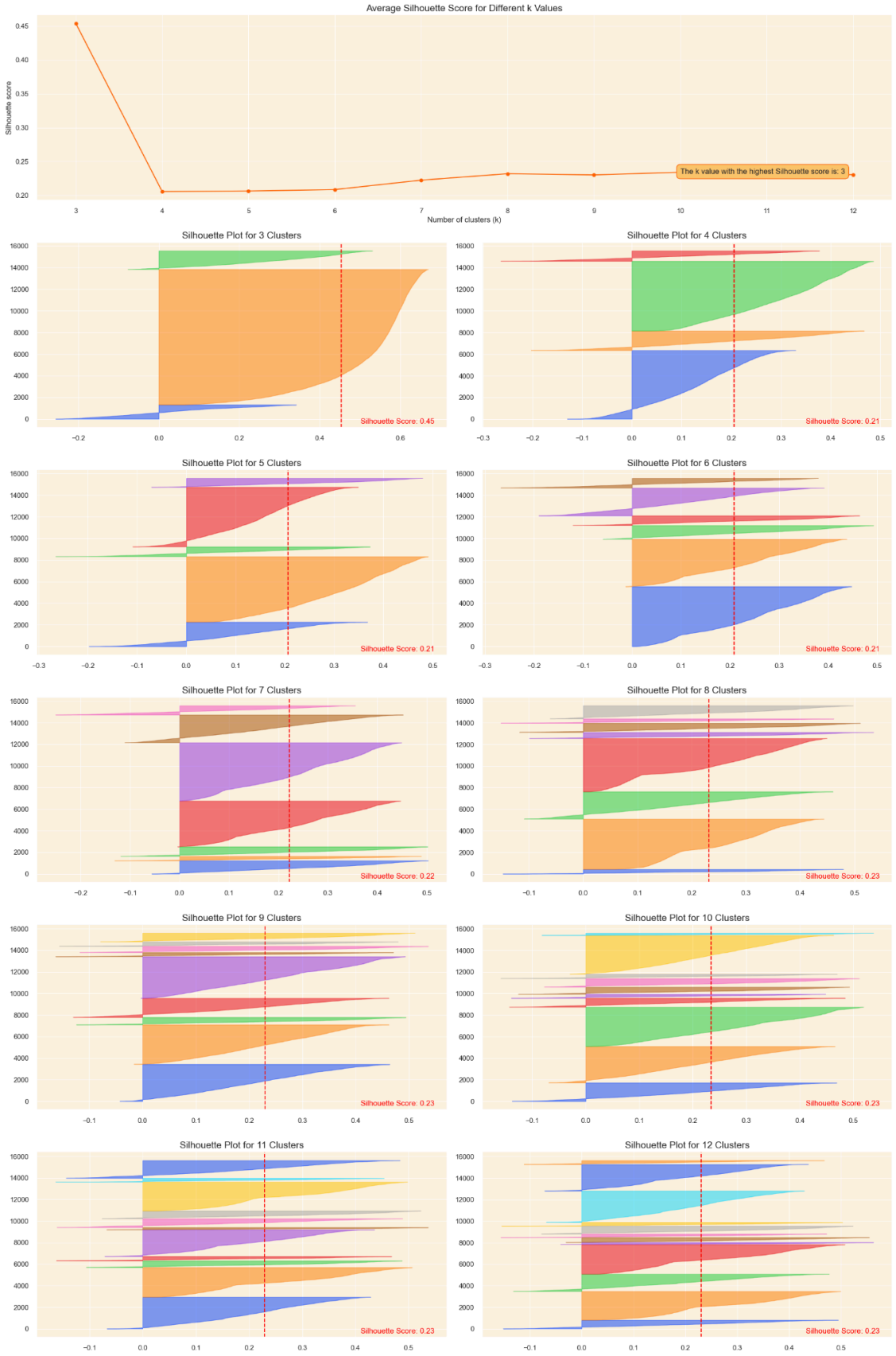

Figure 8: Evaluation of Clustering Quality Using Silhouette Scores.

In the analysis, k=3 emerged as the optimal number of clusters, yielding well-defined and distinct groups with strong internal cohesion. This configuration enhances the reliability of our clustering solution. Following this, the K-means algorithm was applied to segment customers based on purchasing behaviors, using the identified k=3. To ensure consistent cluster labeling across runs, a post-processing step was implemented to standardize labels by reassigning them based on cluster sample frequency. This approach ensures stability and consistency in the clustering results.

5.3.7 Clustering Evaluation & Validity metrics

The evaluation metrics indicate a moderate clustering quality. The Silhouette Score of approximately 0.454 suggests some overlap between clusters, with scores closer to 1 indicating better separation. Conversely, the high Calinski-Harabasz Score of 5041.62 reflects well-defined clusters, indicating effective structure capture. The Davies-Bouldin Score of 1.36 indicates moderate cluster similarity, suggesting reasonable separation but highlighting potential for improvement. Overall, the results demonstrate good clustering quality with opportunities for optimization through alternative clustering or dimensionality reduction techniques.

Table 4: Evaluation Metrics for Cluster K.

Metric	Value
Number of Observations	15507
Silhouette Score	0.454
Calinski Harabasz Score	5041.625
Davies Bouldin Score	1.369

-> The PCA plot reveals that most of the variance is captured in the first two principal components, which is why the K-Means clusters are well-separated in the right plot. This suggests that PCA was successful in simplifying the data while preserving meaningful structures.

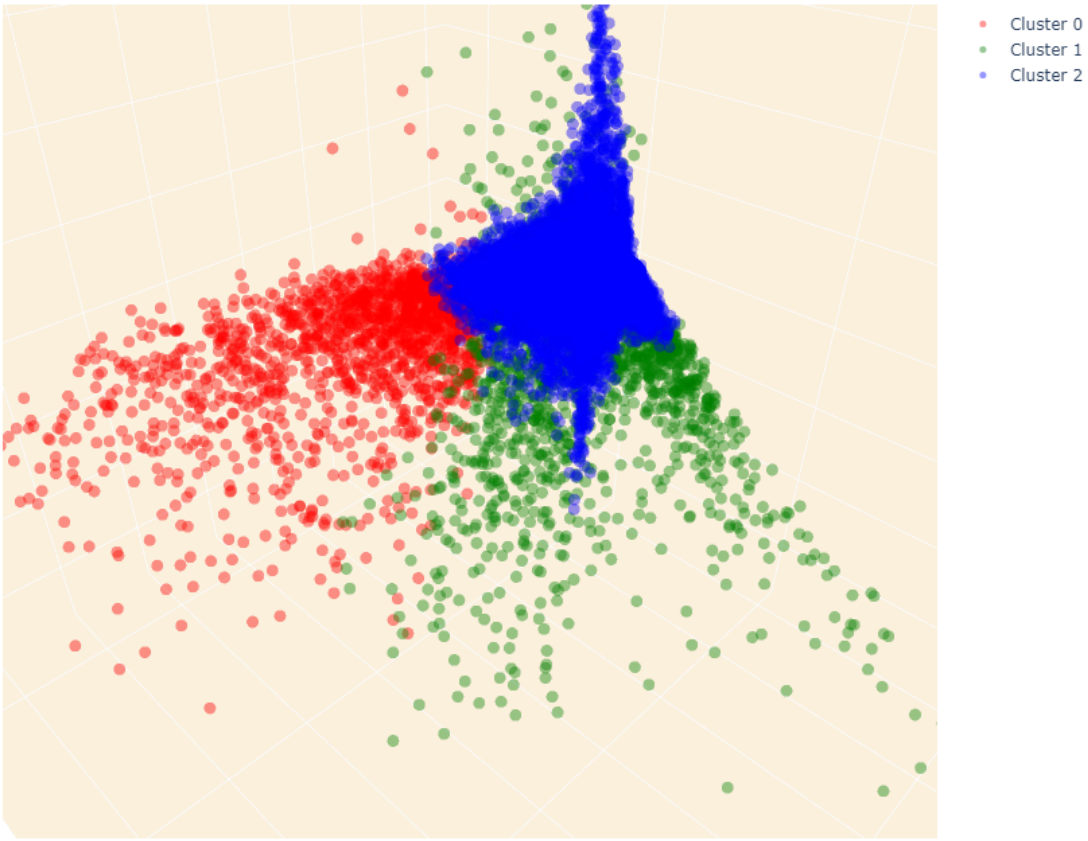

Figure 9: 3D Visualization of Customer Clusters Based on Top 3 Principal Component Analysis.

This 3D plot enables interaction in the Python environment using navigation features, enabling very precise examination of every customer observation. This enables ORA-FASHION to target specific customers through emails, newsletters and other media, to improve RFM and CLV metrics through targeted marketing campaigns.

5.3.8 Customer Cluster Proportions

The bar graph indicates an imbalanced distribution, with Cluster 2 encompassing around 81% of the customer base, while Cluster 0 contains approx. 10% and Cluster 1 approx. 9%. This distribution suggests that the clustering process has effectively captured substantial segments of the customer base, rather than simply categorizing noise or outliers. The dominant size of Cluster 2 implies that it represents a significantly big proportion of customers that share certain behavioral characteristics, while the smaller proportions indicate less prevalent but distinct groups. This is a quite important aspect of interpretation, to be combined with the specific behaviors associated for each group.

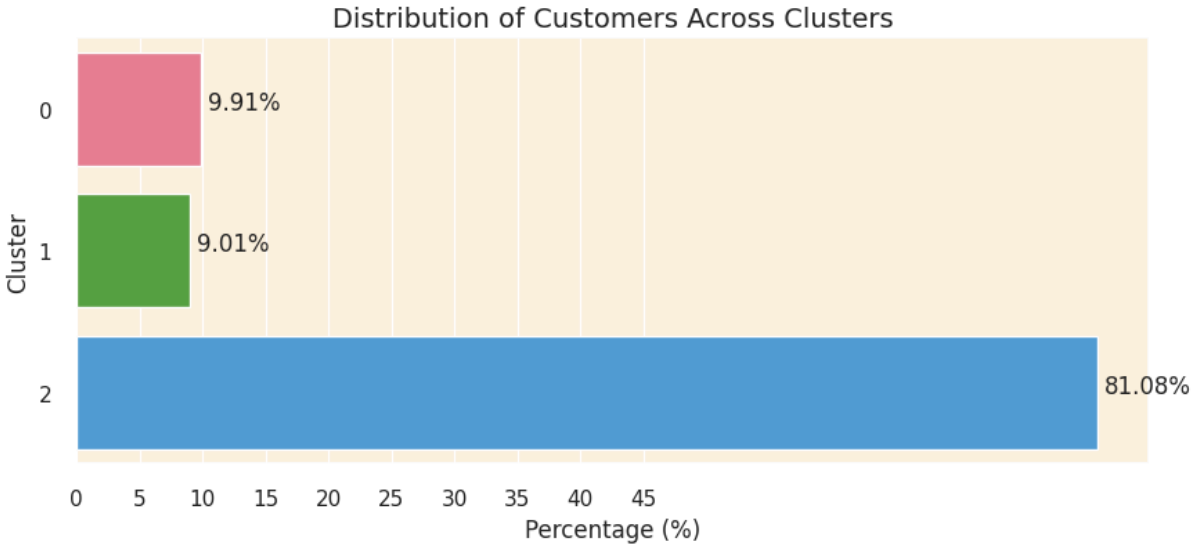

Figure 10: Bar-Graph Distribution Plot for Customer Cluster Groups.

5.4 Cluster Analysis and Radar Chart Profiling

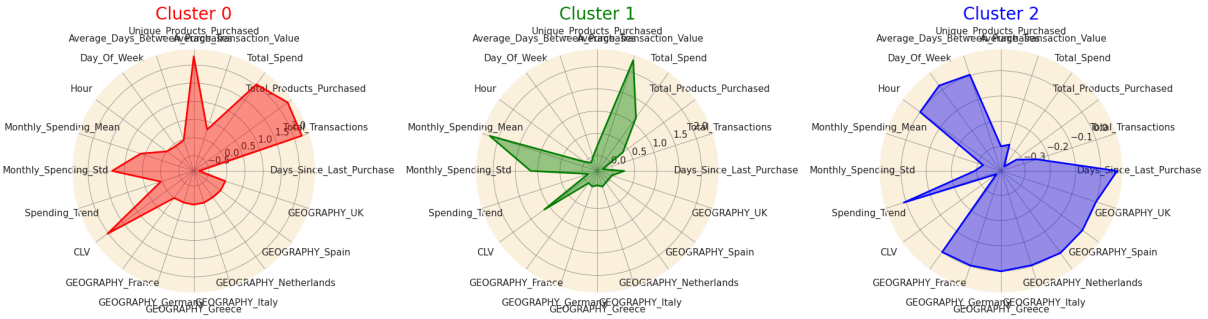

Figure 11: Radar Plots for Deep-Phenotyping of Customer Groups based on Cluster Membership.
*Note: The scales differ and thus the actual magnitude of Cluster 1 looks visually a bit smaller, but the scale is the largest.

Three radar charts were generated representing different clusters (Cluster 0, Cluster 1, and Cluster 2), each visualizing various features of the clusters. Cluster 0 (red) is characterized by many unique products purchased, high average transaction value and highest total products purchased compared throughout all groups. They display a moderate-high total spending rate, and frequent purchases with low days since the last purchase. Cluster 1 (green) displays high values for unique products purchased, moderate products purchased but also the highest spending monthly. Cluster 2 (blue), in contrast, shows low spending and transaction activity but has high variability in monthly spending and a high number of days since the last purchase, showing that they are the least engaged, while they account for the largest proportion of people.

Table 5: Radar Plot Summary.

Behavior Metric	Cluster 0	Cluster 1	Cluster 2
Unique Products Purchased	1.6	2.5	-0.3
Total Spend	1.3	1.8	-1
Total Transactions	1.6	0.3	-0.5
Total Products Purchased	1.5	0.5	-1
Days Since Last Purchase	-1	-0.2	0
Monthly Spending Mean	0	2.5	-0.3
Average Days Between Purchases	0.4	-0.2	0.2
Monthly Spending Std	1	1.5	-0.3
Spending Trend	0	0	0
Hour	-0.5	0	0
Day Of Week	-0.5	0	0.1
Geography Features (e.g., UK, France, Germany, etc.)	0	0	0

5.4.1 Psychometrics of Customer Behavior & Risk Management

Table 6: 'Psychometrics' of Customer Behavior & Risk Management: Table for Decision Makers and a Behavioral Composite Score.
This metric gives an indication of customer's high value behavior to the company based on key behavioral metrics.
Where: 'Very High' = 2, 'High' = 1, 'Moderate' = 0, 'Low' = -1, and 'Very Low' = -2 and reverse coding for recency.

Behavior Metric	Cluster 0	Cluster 1	Cluster 2
Unique Products Purchased	High	Very High	Very Low
Total Spend	High	High	Very Low
Total Transactions	High	Moderate	Low
Total Products Purchased	High	Moderate	Very Low
Days Since Last Purchase	Low	Low	Moderate
Monthly Spending Mean	Moderate	Very High	Very Low
Average Days Between Purchases	Moderate	Low	Moderate
Monthly Spending Std	High	High	Very Low
Spending Trend	Moderate	Moderate	Moderate
Hour	Low	Moderate	Moderate
Day Of Week	Low	Moderate	Moderate
Geography Features (e.g., UK, France, Germany, etc.)	Moderate	Moderate	Moderate
Behavioral Composite Score:	4	8	-11
Weigthed Purchasing Contribution:	0,44	0,68	-7,75

Table 7: Descriptives Summary of Behavioral Measures Grouped for Country and Cluster Membership.

Country	Cluster	Avera..	Avera..	CLV	Days ..	Mont..	Mont..	Sum o..	Total ..	Total ..	Total ..	Uniqu..
France	0	-27	3.195	1.535	363	3.107	1.595	0	496	1.535	-327	301
	1	45	-3.648	-4.710	805	-4.114	-3.703	12.534	-4.007	-4.710	-3.079	-3.740
	2	-18	453	3.175	-1.167	1.007	2.108	3.058	3.511	3.175	3.406	3.440
Germany	0	-27	3.195	1.535	363	3.107	1.595	0	496	1.535	-327	301
	1	45	-3.648	-4.710	805	-4.114	-3.703	12.534	-4.007	-4.710	-3.079	-3.740
	2	-18	453	3.175	-1.167	1.007	2.108	3.058	3.511	3.175	3.406	3.440
Greece	0	-27	3.195	1.535	363	3.107	1.595	0	496	1.535	-327	301
	1	45	-3.648	-4.710	805	-4.114	-3.703	12.534	-4.007	-4.710	-3.079	-3.740
	2	-18	453	3.175	-1.167	1.007	2.108	3.058	3.511	3.175	3.406	3.440
Italy	0	-27	3.195	1.535	363	3.107	1.595	0	496	1.535	-327	301
	1	45	-3.648	-4.710	805	-4.114	-3.703	12.534	-4.007	-4.710	-3.079	-3.740
	2	-18	453	3.175	-1.167	1.007	2.108	3.058	3.511	3.175	3.406	3.440
Netherlands	0	-27	3.195	1.535	363	3.107	1.595	0	496	1.535	-327	301
	1	45	-3.648	-4.710	805	-4.114	-3.703	12.534	-4.007	-4.710	-3.079	-3.740
	2	-18	453	3.175	-1.167	1.007	2.108	3.058	3.511	3.175	3.406	3.440
Spain	0	-27	3.195	1.535	363	3.107	1.595	0	496	1.535	-327	301
	1	45	-3.648	-4.710	805	-4.114	-3.703	12.534	-4.007	-4.710	-3.079	-3.740
	2	-18	453	3.175	-1.167	1.007	2.108	3.058	3.511	3.175	3.406	3.440
UK	0	-27	3.195	1.535	363	3.107	1.595	0	496	1.535	-327	301
	1	45	-3.648	-4.710	805	-4.114	-3.703	12.534	-4.007	-4.710	-3.079	-3.740
	2	-18	453	3.175	-1.167	1.007	2.108	3.058	3.511	3.175	3.406	3.440

Average Days Between Purchases, Average Transaction Value, CLV, Days Since Last Purchase, Monthly Spending Mean, Monthly Spending Std, sum of Cluster, Total Products Purchased, Total Spent, Total Transactions and Unique Products Purchased broken down by Country and Cluster.

5.4.2 Histogram Matrix for Customer Behaviors

Lastly, many insights can be derived from the histogram matrix for customer behaviors based on cluster categorization. Overall, there is a clear contrast between Cluster 0 (frequent moderate spenders) and Cluster 1 (infrequent high spenders), while Cluster 2 lags in both engagement and spending. For revenue maximization, clusters 0 and 1 should be the focus for targeted marketing strategies to maximize revenue, while strategies to re-engage Cluster 2 could help in preventing churn and improving overall business performance. Cluster 0 shows the most diverse buying behavior, indicating an openness to various product offerings, which can be leveraged in marketing campaigns. Again, this summarizes the need for managing various strategies for marketing.

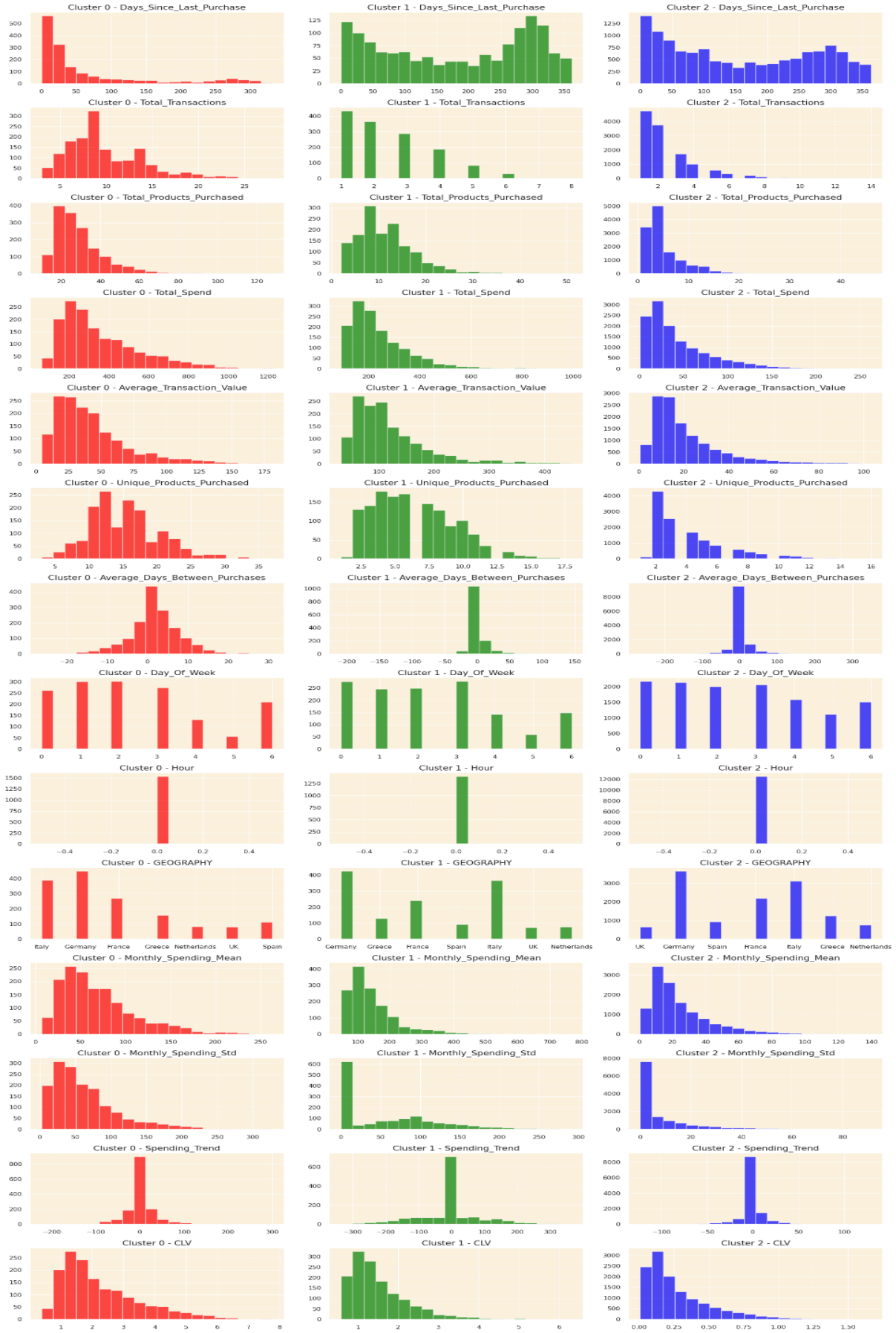

Figure 12: Histogram Matrix for Customer Behaviors for Each Segmented Cluster Group.

6. Discussion

6.1 General summary

In summary, the application of data science methodologies has contributed to significantly improving ORA-FASHION's understanding of its customer base and it’s chances to further succeed in the European fashion market. A data-driven approach to future strategies and the management of innovative campaigns has made it possible to deeply profile customer groups for a detailed understanding of their e-commerce purchasing behaviors through RFM, CLV, and clustering analyses. By implementing the recommended strategies, ORA-FASHION can strengthen its market presence, drive customer loyalty, and secure a competitive advantage within the European fashion industry. Moreover, this project positions ORA-FASHION for long-term success in a market that is becoming more and more data-centric while providing value to its clients as an example of how data-driven decision-making can result in measurable business outcomes.

The analysis of customer demographics and sales trends reveals essential insights into the target market and consumer behavior patterns on a Eurpean scale. Primarily, the large customer base consists of individuals aged 30 to 50, with a nearly even male-to-female distribution. This demographic characteristic indicates a diverse audience that likely seeks different product attributes and preferences, positioning the brand to cater to a wide range of needs. Furthermore, the predominance of customers from the UK, alongside significant representation from France and Greece, outlines the geographic focus for marketing and sales strategies, however other countries might be adding greater revenue through occasional high-spenders such as in Germany. Additionally, geographical variances in purchasing patterns reveal that UK customers tend to exhibit higher purchase frequencies than their counterparts in other regions, signaling a potential need for tailored engagement strategies specific to geographical segments.

Diverse purchasing behaviors among demographic groups were also noted. Younger customers display a preference for trendy and affordable products, contrasting with older customers, who are more inclined towards premium offerings. Such differences underline the importance of segmenting marketing strategies to resonate with the distinct preferences of these groups.

Seasonal and promotional trends further confirm that strategic timing is paramount, with significant sales spikes observed during holidays and select sales events. This finding

validates the effectiveness of focused promotional campaigns during these critical periods and highlights the potential for optimizing marketing efforts to capitalize on these trends.

Lastly, but as the most significant part of this analysis pipeline, it was identified that a small yet impactful segment of high-value customers - the top 20% of spenders - who significantly contribute to overall sales figures. These individuals represent prime candidates for targeted loyalty programs and personalized marketing initiatives designed to enhance retention and foster long-term engagement. However, these main drivers of revenue can be further split into two groups, those that prefer quantity (Cluster 0) and those that prefer quality (Cluster 1). Understanding how to further engage them and how to transfer more of the less engaged individuals (Cluster 2) is crucial to drive further growth and expansion.

6.2 Customer Group RFM

The Recency, Frequency, and Monetary (RFM) analysis can aid improving revenue generation and in understanding specific customer behaviors with respect to their demographic features. The RFM analysis offers a strong framework for comprehending customer value and engagement by quantifying what customers spend (Monetary), how frequently they purchase (Frequency), and how recently they have purchased (Recency).

Cluster 0 individuals are the most engaged customers, while their monetary contributions do not account for the largest proportion of revenue (they buy more but spend less). Upselling with upgrades and abonnements, or cross-selling techniques for product recommendations might be a clever way to adress this issue. They behave differently compared to cluster 1 which are the rather infrequent buyers, that nevertheless spend much more than other people (which will also later explain the CLV outcomes). Lastly, cluster 2 comprises the majority of customers, characterized by both low frequency and low monetary value. This indicates a lack of engagement with the brand, and these customers are at risk of churning, but the same is true for cluster 1 which might necessitate a similar technique for maximization of revenue.

6.3 Customer Group Life Time Value

Considering CLV is of utmost importance for ORA-FASHION, considering the right skew of the presence of almost 10% high spenders and a big portion of disengaged individuals. For ORA-FASHION’s future success, ongoing monitoring of these segments to track the effectiveness of implemented strategies is crucial. Regular updates to the CLV and

segmentation analysis are recommended and will allow ORA-FASHION to remain agile, adapting to changing customer behaviors and market conditions in real time.

A small number of high-value clients contribute disproportionately to the total revenue, as is common in many industries, as evidenced by the clearly right-skewed distribution that the CLV analysis presented. This indicates that ORA-FASHION's business success heavily relies on identifying, retaining, and nurturing these high-value customers, but also in understanding and measuring in ongoing long-term analysis whether this performance metric can be improved for the future. Regional variations in CLV were also identified by the analysis, with a particular emphasis on the greater proportion of high-value clients in Germany, France, and Italy relative to other European countries. This suggests that ORA-FASHION's brand presence or market maturity is stronger in these regions, offering a strategic opportunity for focused, high-end marketing campaigns.

Various strategic recommendations can be interpreted based on the CLV of customer groups. For cluster 0 which - includes frequent buyers, moderate spenders - ORA-FASHION should focus on strategies that encourage higher spending per transaction, which would scale the CLV per customer in the long run towards higher company revenue. Appendix Table II indicates various marketing strategies that would aid the scaling of this metric.

Cluster 1, consisting of infrequent buyers that are high spenders, probably account for the skew of the distribution. It is important to retain these customers and eventually target the frequency of purchases. The geographical analysis of CLV highlighted that customers in Germany, France, and Italy exhibit higher spending behaviors, particularly among the top 25% of customers in these regions. This finding suggests that these markets are ripe for tailored marketing strategies that cater to high-value customers. ORA-FASHION could consider implementing region-specific campaigns that leverage cultural nuances and consumer preferences unique to these countries. For instance, luxury-focused campaigns or collaborations with local influencers could resonate well with these audiences, further enhancing customer loyalty and increasing CLV.

Lastly, the largest group of individuals is present in cluster 2, which are the most disengaged customers. They can be disengaged for various reasons, and eventually for a portion of these individuals, their “true” membership might be in fact another one. This can be further investigated after a marketing campaign and long-term sampling of these data. Due to 80% of the sample consisting of these individuals, the right skew might also be influenced by the “major weight” of the size of this group. Again, future studies would be interesting, to identify

whether there are other underlying customer groups present in this chunk of customers, and whether they can be “reactivated” through targeted marketing.

6.4 Interpretation of Segments and Strategic Recommendations

Based on the findings, several strategic recommendations emerge that can fortify the client’s market position and enhance overall business performance:

6.4.1 Cluster 0 - "Frequent Buyers with Moderate Spending”

Contains 9,91% of customers, which have a very high engagement level, characterized by a diverse range of unique products purchased and frequent shopping behavior, leading to significant total spending. This behavioral group shows the greatest dynamism in their purchasing patterns and they tend to prefer quantity and frequency over quality. They might be more prone to various types of engagements and more open to advertisements, offers, etc. Marketing campaigns targeting this group should make them feel as part of the branding and perhaps offer them special bonding through premium services. It is essential to develop personalized marketing campaigns for high-value customers, leveraging data analytics to tailor promotional efforts based on demographic segments such as age and geography. By doing so, the brand can ensure its messaging resonates more effectively with various customer groups, potentially increasing conversion rates. These people are also very essential for social media advertisements and any type of campaign that requires frequent interaction and person centered-ads, that can be maximized using e.g. browser behaviors metrics and cookies to further adapt ads and customer insights.

6.4.2 Cluster 1 - “Infrequent Buyers with High Spending”

With 9,01% of customers being in this group, displays a more selective and moderate purchasing style, with longer intervals between purchases and a focus on value. This group although purchasing less frequently compared to Cluster 0, consists of high-spenders with greater selectivity, that probably seek high-quality prestige products as they spend the most on a monthly basis and have the strongest behavioral metrics for revenue. These are also the people who potentially are more inclined to turn to lifetime customers once targeting has succeeded. Enhancing or introducing loyalty programs aimed at high-value customers could encourage repeat purchases and solidify brand loyalty. They probably also like products that indicate their quality indicators, which can be further developed by the company. By offering personalized incentives and reinforcement incentives like bonus programs and credit collections, the client can cultivate a sense of appreciation and connection with this vital customer segment and be more inclined to engage further.

6.4.3 Cluster 2 - “Infrequent Buyers with Low Spending”

Consists of 81,01% of the dataset and contains the vast majority of people, which tend to be less engaged, therefore there is a lot of room for improvement (and potential) for targeting these individuals. They are predicted to be more conservative and show a very infrequent shopping propensity with the current marketing management campaigns, resulting in lower key performance indicators as defined by RFM and CLV. For these individuals, general broad marketing campaigns, sales, regular newsletters and general advertisement presence might be influential. This group is probably overrepresented, with the real cluster groups being more balanced. To succeed with harmonizing the groups and transferring less engaged customers to engaged ones, understanding the demographic features of this group is essential and regular follow-up analyses to compare the performance of marketing campaigns.

In conclusion, this analysis provides actionable insights that can enhance the client’s ability to meet customer needs, optimize sales strategies, and improve overall business performance. Implementing these strategic recommendations will not only drive immediate sales growth but also foster long-term customer loyalty and brand advocacy.

6.5 Deliverables for the Client

Based on the acquired insights, various output services were created for the client:

6.5.1 Customer Profiling & Personalized Marketing Strategies

This section includes two deliverables for the client: a worksheet with detailed customer profile characteristics and guidelines for recommending customized marketing strategies based on insights obtained. The worksheet helps the client interpret the customer groups.

To effectively target different customer segments, personalized marketing strategies should be tailored to their unique purchasing behaviors and preferences. For Cluster 0,loyalty programs such as tiered rewards and exclusive discounts can incentivize customers to increase their spending or frequency of purchases. Personalized recommendations, including product bundling and AI-driven suggestions, can enhance their shopping experience by highlighting items that align with their past purchases. Additionally, targeted email campaigns with frequent purchase reminders and time-sensitive promotions, like flash sales, can keep them engaged and encourage more consistent buying behavior.

For Cluster 1,high-value loyalty perks such as VIP programs and cashback offers can reinforce their premium shopping experience and prompt them to make significant purchases

more often. Personalized communication that emphasizes luxury, like premium packaging and special discounts on purchase anniversaries, can create a sense of exclusivity. Marketing exclusive product launches and providing early access to limited editions cater to their preference for unique, high-end items. Cross-selling opportunities through curated collections and luxury bundle offers can further increase their overall spending by introducing complementary products that enhance their previous purchases.

Cluster 2, requires strategies focused on boosting engagement and increasing their transaction frequency. Engagement incentives, such as welcome back discounts and gamified rewards, can reignite their interest and encourage repeat purchases. Implementing basic loyalty programs with simple point systems and offering first-time buyer promotions can help build a habit of shopping. Reactivation campaigns, like abandoned cart reminders and exit-intent popups, can convert browsing interest into actual sales. Offering low-cost sampling and trial discounts can also introduce these customers to new products, encouraging them to explore more of your catalog without a significant upfront commitment.

6.5.2 Interactive Dashboard

The dashboard further enhances the customer’s exploration of the granular customer specific insights. This way, they can analyse customers one by one, which will aid them in directing their communication efforts (especially Cluster 1, as they are very few but very high ROI clientele.

6.5.3 Python Script for Replicable Data-Analysis Pipeline

Lastly, the Python code is provided as it further enables the monitoring of campaign effects on future e-commerce datasets of the same company, but can be tailored to other contexts as well.

7. Conclusion

This project aimed to enhance ORA-FASHION Ltd's customer relationship management through data-driven marketing strategies, specifically employing K-means clustering with RFM and CLV analyses. The study revealed distinct customer segments characterized by unique purchasing behaviors, highlighting opportunities for personalized marketing initiatives and tailored strategies to meet the specific needs of each segment. The segmentation analysis successfully identified high-value customers, empowering ORA-FASHION to customize loyalty programs and marketing efforts aimed at fostering stronger relationships

and enhancing customer retention. The CLV analysis further emphasized the importance of maximizing long-term customer value through sustained engagement, guiding the allocation of resources toward the most profitable segments and as a key metric to compare future performance improvement for the less engaged customer groups. In summary, the application of data science methodologies has significantly improved ORA-FASHION's understanding of its customer base. By implementing the recommended strategies, ORA-FASHION can strengthen its market presence, drive customer loyalty, and secure a competitive advantage within the European fashion industry.

Implications and added value: In addition to enabling deep phenotyping of customer groups, this analytical framework facilitated the creation of actionable recommendation guidelines and an interactive dashboard, along with a replicable Python script for future analyses. This added value not only optimizes ORA-FASHION's marketing strategies but also equips the company with tools for ongoing customer insights, allowing it to remain agile in a dynamic market. From the customer perspective, these initiatives enhance the shopping experience by delivering tailored communications and offers that resonate with their specific preferences. Personalization fosters a sense of recognition and appreciation, which can lead to increased satisfaction and loyalty. Customers are more likely to engage with a brand that understands their unique preferences and provides relevant recommendations, creating a win-win scenario for both ORA-FASHION and its customer base.

References

Books and Book Chapters:

- Kahan, R. (1998). *A review of the application of RFM model*.
- Miglautsch, J. (2000). *A review of the application of RFM model*.
- Lopez, S. (2020). *Data-driven Buyer Personas: What They Are and How to Build them*.
- Schlegelmilch, B. B. (2022). Segmenting, targeting, and positioning in global markets. In *Global Marketing Strategy: An Executive Digest* (pp. 129-159). Cham: Springer International Publishing.
- Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In *Text mining: applications and theory* (pp. 1-20).

Journal Articles:

- Peker, S., Ada, S., & Taşkin, H. (2017). LRFMP model for customer segmentation in the grocery retail industry: a case study. *Journal of Retailing and Consumer Services*.
- Liu, D., & Guo, X. (2020). Data-driven segmentation of customer bases in the era of big data: Current practices and future challenges. *Business Horizons, 63*(6), 705-717.
- Cirqueira, D., Hofer, M., Nedbal, D., Helfert, M., & Bezbradica, M. (2019). Customer purchase behavior prediction in e-commerce: A conceptual framework and research agenda. In *International workshop on new frontiers in mining complex patterns* (pp. 119-136). Cham: Springer International Publishing.
- Qiu, J., Lin, Z., & Li, Y. (2015). Predicting customer purchase behavior in the e-commerce context. *Electronic Commerce Research, 15*(4), 427-452.

Online Sources:

- United Nations. (2021, May 17). *World e-commerce and the global economy*. Retrieved from https://news.un.org/en/story/2021/05/1091182
- Statista. (2021). *Worldwide e-commerce revenue by region*. Retrieved from https://www.statista.com/forecasts/1117851/worldwide-e-commerce-revenue-by-region
- Ecommerce Europe. (2023). *European Ecommerce Report 2023 - Light Version*. Retrieved from https://ecommerce-europe.eu/wp-content/uploads/2023/11/European-Ecommerce-Report-2023-Light-Version.pdf

Datasets:

- ORA-FASHION. (2021). Anonymous purchase data from 22,626 customers [Data set]. ORACLE Database, ORACLE Cloud Services.

Appendix

A) Customer Profiling Recommendation Guidelines and Tools

Table 1: Aspect matrix for different customer features based on the segmentation patterns.

Cluster 0: focuses on encouraging moderate spenders to increase their spending and frequency through loyalty programs and personalized recommendations.

Cluster 1: targets infrequent high spenders with VIP treatment, exclusive product access, and high-value perks.

Cluster 2: aims to re-engage low-spending, low-engagement customers with incentives, basic loyalty programs, and reactivation campaigns.

I. Customer Profiling for Marketing Strategies

Table I: Data-driven Customer Characterization.

	Cluster 0: "Frequent Buyers with Moderate Spending"	Cluster 1: "Infrequent High-Spenders"	Cluster 2: "Low Engagement Low-Spenders"
1	• **Days Since Last Purchase:** This cluster has customers who made recent purchases, with a significant drop-off after around 50 days. • **Total Transactions:** Customers tend to have a moderate number of transactions, mostly between 5 to 15. • **Total Products Purchased:** This cluster shows customers purchasing a moderate amount of products, typically between 20 to 40. • **Total Spend:** Spending is moderate, mostly ranging between $200 to $400. • **Average Transaction Value:** Generally low to moderate, between $25 to $75. • **Unique Products Purchased:** Customers in this cluster tend to buy between 5 to 15 unique products. • **Average Days Between Purchases:** Customers purchase relatively frequently, with an average interval of around 10 days. • **Day of Week:** No significant preference for a specific day. • **Hour:** Concentrated around a single hour, suggesting a peak purchasing time.	2	• **Days Since Last Purchase:** Wide range of days since the last purchase, peaking around the one-year mark. • **Total Transactions:** These customers have fewer transactions, typically between 1 to 4. • **Total Products Purchased:** Purchase fewer products overall, between 10-30. • **Total Spend:** Despite fewer transactions, their total spend is relatively high, mostly ranging from $200 to $600. • **Average Transaction Value:** High, generally between $50 to $100. • **Unique Products Purchased:** Tend to buy between 2.5 to 7.5 unique products. • **Average Days Between Purchases:** Long intervals between purchases, averaging around 25 days. • **Day of Week:** Varied, with no strong preference. • **Hour:** Concentrated around a single hour, indicating a peak time for purchases.	3	• **Days Since Last Purchase:** This cluster has customers who generally made recent purchases, but with a gradual decrease in frequency. • **Total Transactions:** Customers tend to have very few transactions, typically between 1 to 3. • **Total Products Purchased:** They purchase a low number of products, mostly between 5 to 15. • **Total Spend:** Spending is low, mostly under $100. • **Average Transaction Value:** Generally low, around $10 to $30. • **Unique Products Purchased:** Customers in this cluster typically buy between 2 to 5 unique products. • **Average Days Between Purchases:** Very short intervals, suggesting more recent and frequent engagement, averaging around 10 days. • **Day of Week:** Varied, with purchases spread throughout the week. • **Hour:** Concentrated around a single hour, suggesting a peak purchasing time.

II. Personalized Marketing Strategies

Table II: Proposed Personalized Marketing Strategies based on Customer Segments.

	Cluster 0: "Frequent Buyers with Moderate Spending"	**Cluster 1: "Infrequent High-Spenders"**	**Cluster 2: "Low Engagement Low-Spenders"**
Profile Summary	Regular purchases with moderate spending.	Less frequent purchases with high spending.	Infrequent, low-value purchases.
Loyalty Programs	Tiered Rewards, Exclusive Discounts	VIP Programs, Cashback Offers, VIP Tiers, Special Membership Cards	Simple Point Systems, First-Time Buyer Promotions
Personalized Recommendations	Product Bundling, Tailored Suggestions	Curated Collections, Luxury Bundles	Low-Cost Sampling, Trial Discounts
Targeted Email Campaigns	Frequent Purchase Reminders, Special Offers for Loyal Customers	Luxury Packaging, Anniversary Discounts	Abandoned Cart Reminders, Exit-Intent Popups
Time-Sensitive Promotions	Flash Sales, Limited-Time Offers	Limited Editions, Pre-Sale Access	Welcome Back Discounts, Gamified Rewards
Exclusive Product Launches	Countdown timers	Market exclusive or limited-edition products directly to this group	Free goodies or trial for product use
Cross-Sell & Up-Sell Opportunities	Upgrades, Bundles, Package deals	Curated Collections, Bundle Offers	Upgrades, Bundles, Package deals
Engagement Incentives	Flash Sales, Limited-Time Offers	Limited Editions, Pre-Sale Access	Welcome Back Discounts, Gamified Rewards
Reactivation Campaigns	Flash-Sales, Limited Time offers	Personalized communication, scarcity products	Abandoned Cart Reminders, Exit-Intent Popups, Customer Testimonials, Influencer Endorsements

B) Interactive Dashboard

Access-URL for Interactive Dashboard via Tableau Public found here. The final dataframe resulting from the Python script was eventually merged back to the original dataframe (which had already merged two datasets) and utilized for Tableau analytics and dashboards.

Static preview of dashboard

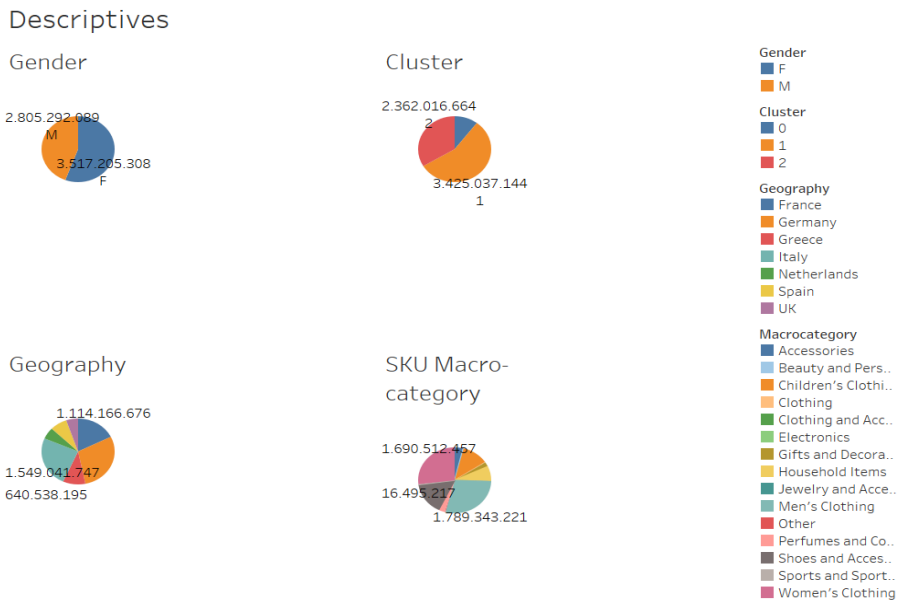

Figure I: Pie Charts for Various Features.

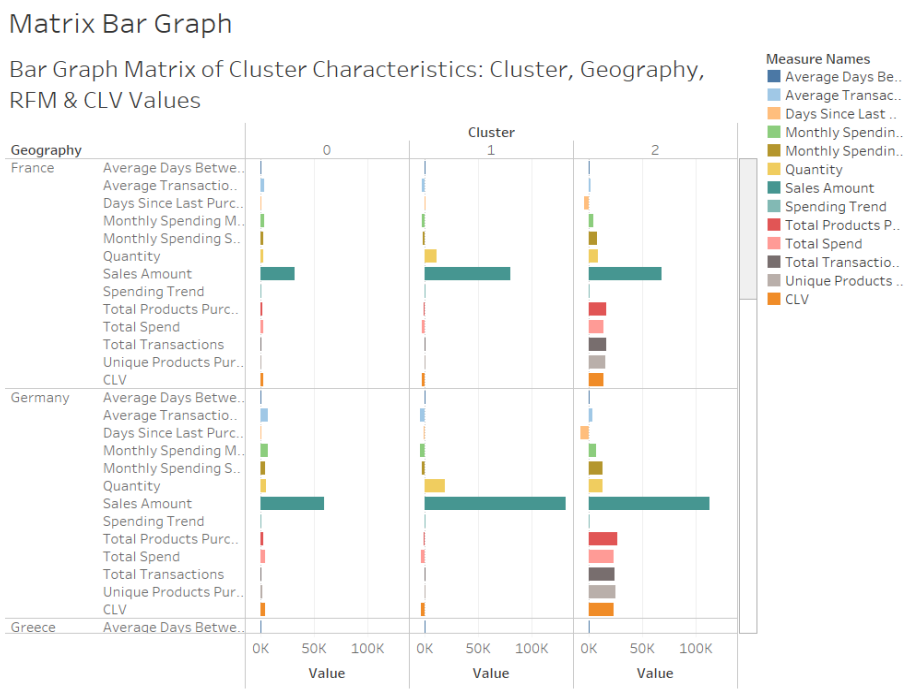

Figure II: Bar Graph Matrix for Key-Demographic Features.

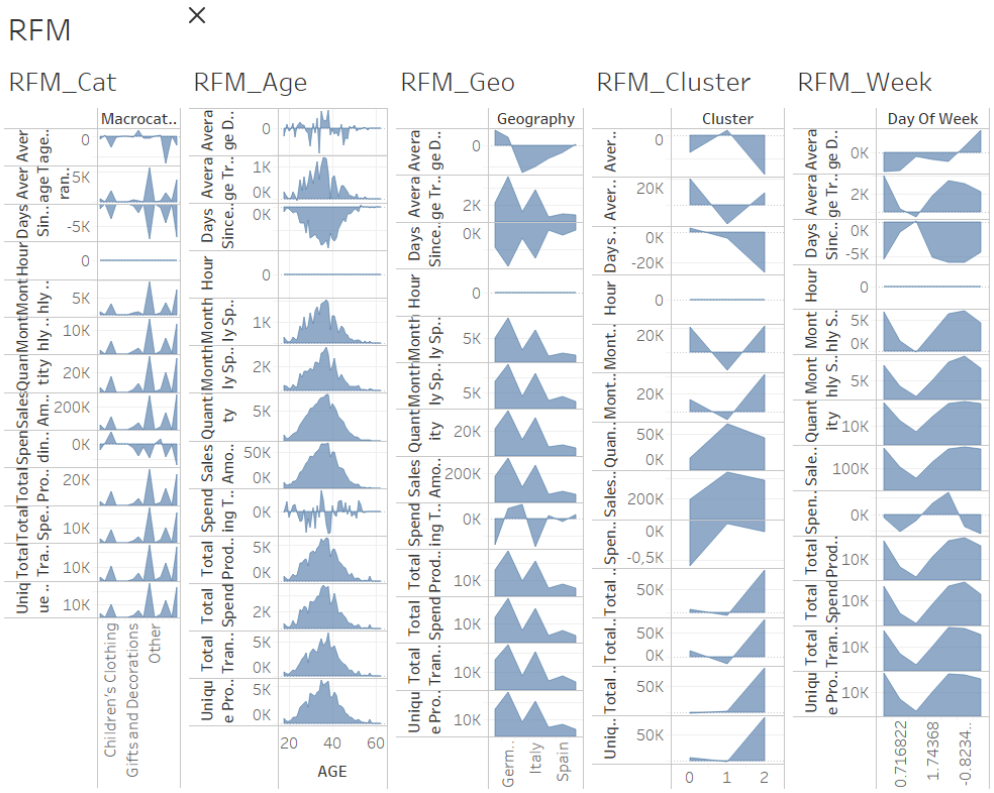

Figure III: RFM Curve Matrix for various Key Features.

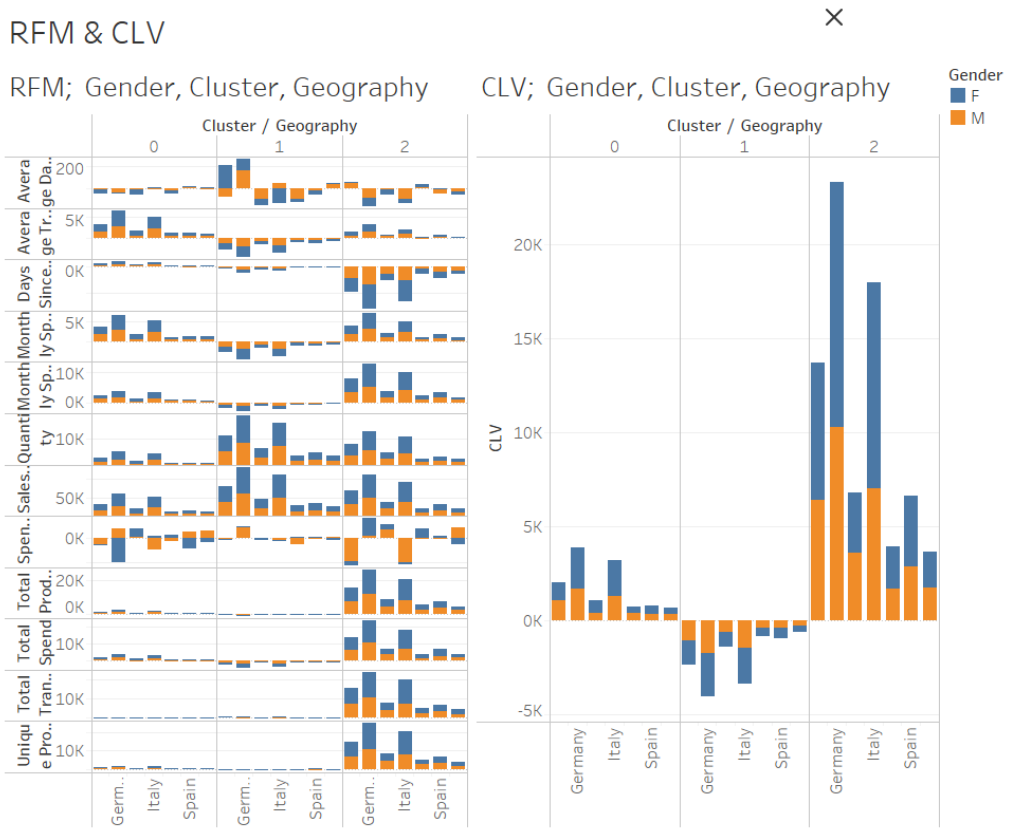

Figure IV: Comparison of RFM and CLV Bar Graphs.

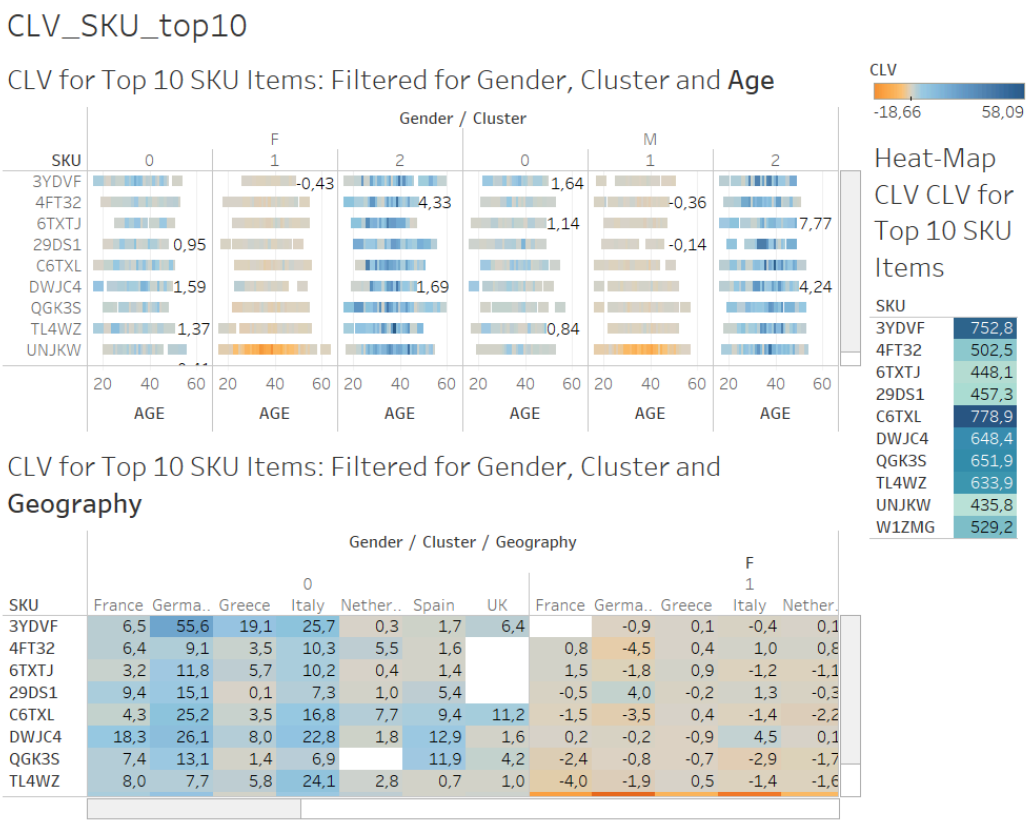

Figure V: Heat Maps for Top 10 SKU Items (Age & Geography over Gender, Cluster).

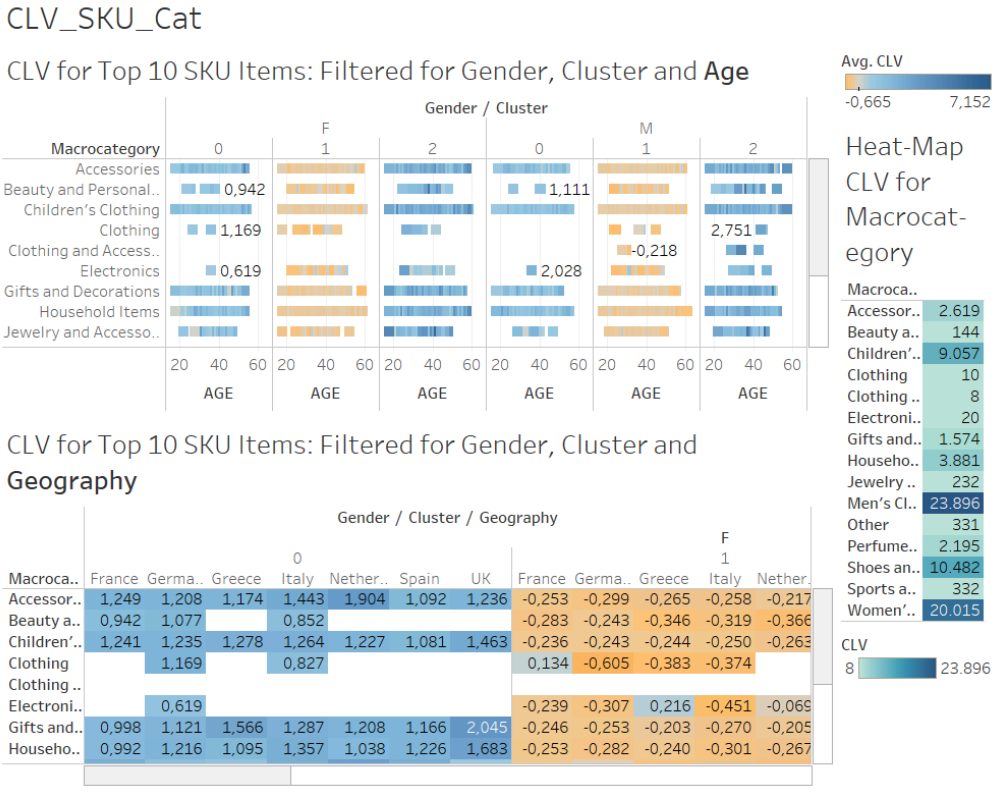

Figure VI: Heat Maps for SKU Macrocategories (Age & Geography over Gender, Cluster).

C) Python Script

Final Python script and in-depth explanation here and below as a transcript (code only).

Dataframes used in the analysis: Customer set; Orders set.

Python transcript

```
"""Copy of Final_Capstone_RBS-IMDS_PRJ_GROUP 5_code only.ipynb

Original (code only) file is located at:

https://colab.research.google.com/drive/1MliTJ2RqQVe0orAdpWa2BOfC8uBXY48m


# Import

"""


#IF jupyter notebook or VS


!pip install scikit-learn tabulate plotly pandas seaborn matplotlib numpy scipy


# Data Manipulation

import pandas as pd

import numpy as np


# Data Visualization

import matplotlib.pyplot as plt

import seaborn as sns

import matplotlib.patches as mpatches

from matplotlib.colors import LinearSegmentedColormap

import plotly.graph_objects as go


# Machine Learning and Preprocessing
```

```
from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

from sklearn.cluster import KMeans

from sklearn.ensemble import IsolationForest

from sklearn.metrics import silhouette_score, calinski_harabasz_score,
davies_bouldin_score

from scipy.stats import linregress


# Clustering Evaluation and Visualization

from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer


# Utilities

from collections import Counter

from tabulate import tabulate

from google.colab import files


"""# Data-Preprocessing: Import, Merge and Manage Dataframes"""


# Load the Orders Excel file

orders_file = pd.ExcelFile('/content/Orders2.0.xlsx')


# Display sheet names to understand what we're working with

sheet_names = orders_file.sheet_names

print(sheet_names)


# Load the two sheets into separate DataFrames

scanner_data_df = pd.read_excel(orders_file, sheet_name='scanner_data')

product_description_df = pd.read_excel(orders_file, sheet_name='product
description ')
```

```
# Display the first few rows of each dataframe to understand their
structure

scanner_data_df.head(), product_description_df.head()


# Perform an inner join on the two DataFrames using 'SKU' and
'SKU_Category' as keys

merged_df = pd.merge(scanner_data_df, product_description_df,
on=['SKU', 'SKU_Category'], how='inner')



merged_df.head()


# Load the Customers data

customers_df = pd.read_excel('/content/Customers.xlsx')

customers_df.head()


"""Inner join"""


# Perform an inner join on the merged DataFrame and the Customer data
using 'Customer_ID' as the key

final_merged_df = pd.merge(merged_df, customers_df, on='Customer_ID',
how='inner')


# Display the first few rows of the final merged dataframe

final_merged_df.head()


"""# **Data Cleaning**
```

```
"""

final_merged_df.info()

final_merged_df.describe()

duplicate_rows = final_merged_df.duplicated()

# Remove duplicates
final_merged_df_cleaned = final_merged_df[~duplicate_rows]

# Count of removed duplicates
duplicates_removed = duplicate_rows.sum()

duplicates_removed, final_merged_df_cleaned.shape

negative_sales =
final_merged_df_cleaned[final_merged_df_cleaned['Sales_Amount'] < 0]

negative_sales_count = negative_sales.shape[0]
negative_sales_count

df1 = final_merged_df_cleaned

print(df1.head())

df1.info()

# Summary statistics for numerical variables
```

```
df1.describe().T


# Summary statistics for categorical variables

df1.describe(include='object').T


"""# Unique stock codes"""


# Finding the number of unique stock codes

unique_stock_codes = df1['SKU'].nunique()


# Printing the number of unique stock codes

print(f"The number of unique stock codes in the dataset is:
{unique_stock_codes}")


# Finding the top 10 most frequent stock codes

top_10_stock_codes = df1['SKU'].value_counts(normalize=True).head(10) *
100


# Plotting the top 10 most frequent stock codes

plt.figure(figsize=(12, 5))

top_10_stock_codes.plot(kind='barh', color='#ff6200')


# Adding the percentage frequency on the bars

for index, value in enumerate(top_10_stock_codes):

    plt.text(value, index+0.25, f'{value:.2f}%', fontsize=10)


plt.title('Top 10 Most Frequent Stock Codes')

plt.xlabel('Percentage Frequency (%)')

plt.ylabel('SKU')
```

```
plt.gca().invert_yaxis()

plt.show()


"""# **RFM Analysis**


# Recency (R)

"""


# Convert Date to datetime type

df1['Date'] = pd.to_datetime(df1['Date'])


# Convert Date to datetime and extract only the date

df1['Day'] = df1['Date'].dt.date


# Find the most recent purchase date for each customer

customer_data = df1.groupby('Customer_ID')['Day'].max().reset_index()


# Find the most recent date in the entire dataset

most_recent_date = df1['Day'].max()


# Convert Day to datetime type before subtraction

customer_data['Day'] = pd.to_datetime(customer_data['Day'])

most_recent_date = pd.to_datetime(most_recent_date)


# Calculate the number of days since the last purchase for each
customer

customer_data['Days_Since_Last_Purchase'] = (most_recent_date -
customer_data['Day']).dt.days
```

```
# Remove the Day column
customer_data.drop(columns=['Day'], inplace=True)

"""Now, customer_data dataframe contains the Days_Since_Last_Purchase
feature:"""

customer_data.head()

"""Note:
I've named the customer-centric dataframe as customer_data, which will
eventually contain all the customer-based features we plan to create.

# Frequency (F)
"""

# Calculate the total number of transactions made by each customer
total_transactions =
df1.groupby('Customer_ID')['Transaction_ID'].nunique().reset_index()
total_transactions.rename(columns={'Transaction_ID':
'Total_Transactions'}, inplace=True)

# Calculate the total number of products purchased by each customer
total_products_purchased =
df1.groupby('Customer_ID')['Quantity'].sum().reset_index()
total_products_purchased.rename(columns={'Quantity':
'Total_Products_Purchased'}, inplace=True)

# Merge the new features into the customer_data dataframe
customer_data = pd.merge(customer_data, total_transactions,
on='Customer_ID')
```

```
customer_data = pd.merge(customer_data, total_products_purchased,
on='Customer_ID')


# Display the first few rows of the customer_data dataframe

customer_data.head()


"""# Monetary (M)"""


# Calculate the total spend by each customer

df1['Total_Spend'] = df1['Sales_Amount'] * df1['Quantity']

total_spend =
df1.groupby('Customer_ID')['Total_Spend'].sum().reset_index()


# Calculate the average transaction value for each customer

average_transaction_value = total_spend.merge(total_transactions,
on='Customer_ID')

average_transaction_value['Average_Transaction_Value'] =
average_transaction_value['Total_Spend'] /
average_transaction_value['Total_Transactions']


# Merge the new features into the customer_data dataframe

customer_data = pd.merge(customer_data, total_spend, on='Customer_ID')

customer_data = pd.merge(customer_data,
average_transaction_value[['Customer_ID',
'Average_Transaction_Value']], on='Customer_ID')


# Display the first few rows of the customer_data dataframe

customer_data.head()


"""# Product Diversity"""
```

```
# Calculate the number of unique products purchased by each customer

unique_products_purchased =
df1.groupby('Customer_ID')['SKU'].nunique().reset_index()

unique_products_purchased.rename(columns={'SKU':
'Unique_Products_Purchased'}, inplace=True)


# Merge the new feature into the customer_data dataframe

customer_data = pd.merge(customer_data, unique_products_purchased,
on='Customer_ID')


# Display the first few rows of the customer_data dataframe

customer_data.head()


"""# Behavioral Features"""


# Extract day of week and hour from Date

df1['Day_Of_Week'] = df1['Date'].dt.dayofweek

df1['Hour'] = df1['Date'].dt.hour


# Calculate the average number of days between consecutive purchases

days_between_purchases = df1.groupby('Customer_ID')['Day'].apply(lambda
x: (x.diff().dropna()).apply(lambda y: y.days))

average_days_between_purchases =
days_between_purchases.groupby('Customer_ID').mean().reset_index()

average_days_between_purchases.rename(columns={'Day':
'Average_Days_Between_Purchases'}, inplace=True)


# Find the favorite shopping day of the week

favorite_shopping_day = df1.groupby(['Customer_ID',
'Day_Of_Week']).size().reset_index(name='Count')
```

```
favorite_shopping_day =
favorite_shopping_day.loc[favorite_shopping_day.groupby('Customer_ID')[
'Count'].idxmax()][['Customer_ID', 'Day_Of_Week']]


# Find the favorite shopping hour of the day

favorite_shopping_hour = df1.groupby(['Customer_ID',
'Hour']).size().reset_index(name='Count')

favorite_shopping_hour =
favorite_shopping_hour.loc[favorite_shopping_hour.groupby('Customer_ID'
)['Count'].idxmax()][['Customer_ID', 'Hour']]


# Merge the new features into the customer_data dataframe

customer_data = pd.merge(customer_data, average_days_between_purchases,
on='Customer_ID')

customer_data = pd.merge(customer_data, favorite_shopping_day,
on='Customer_ID')

customer_data = pd.merge(customer_data, favorite_shopping_hour,
on='Customer_ID')


# Display the first few rows of the customer_data dataframe

customer_data.head()


"""# Geographic Features"""


df1['GEOGRAPHY'].value_counts(normalize=True).head()


# Group by Customer_ID and Country to get the number of transactions
per country for each customer

customer_country = df1.groupby(['Customer_ID',
'GEOGRAPHY']).size().reset_index(name='Number_of_Transactions')
```

```
# Get the country with the maximum number of transactions for each
customer (in case a customer has transactions from multiple countries)

customer_main_country =
customer_country.sort_values('Number_of_Transactions',
ascending=False).drop_duplicates('Customer_ID')


# Merge this data with our customer_data dataframe

customer_data = pd.merge(customer_data,
customer_main_country[['Customer_ID','GEOGRAPHY']], on='Customer_ID',
how='left')


# Display the first few rows of the customer_data dataframe

customer_data.head()


"""# Seasonality & Trends"""


# Extract month and year from InvoiceDate

df1['Year'] = df1['Date'].dt.year

df1['Month'] = df1['Date'].dt.month


# Calculate monthly spending for each customer

monthly_spending = df1.groupby(['Customer_ID', 'Year',
'Month'])['Total_Spend'].sum().reset_index()


# Calculate Seasonal Buying Patterns: We are using monthly frequency as
a proxy for seasonal buying patterns

seasonal_buying_patterns =
monthly_spending.groupby('Customer_ID')['Total_Spend'].agg(['mean',
'std']).reset_index()

seasonal_buying_patterns.rename(columns={'mean':
'Monthly_Spending_Mean', 'std': 'Monthly_Spending_Std'}, inplace=True)
```

```
# Replace NaN values in Monthly_Spending_Std with 0, implying no
variability for customers with single transaction month

seasonal_buying_patterns['Monthly_Spending_Std'].fillna(0,
inplace=True)


# Calculate Trends in Spending

# We are using the slope of the linear trend line fitted to the
customer's spending over time as an indicator of spending trends

def calculate_trend(spend_data):

    # If there are more than one data points, we calculate the trend
using linear regression

    if len(spend_data) > 1:

        x = np.arange(len(spend_data))

        slope, _, _, _, _ = linregress(x, spend_data)

        return slope

    # If there is only one data point, no trend can be calculated,
hence we return 0

    else:

        return 0


# Apply the calculate_trend function to find the spending trend for
each customer

spending_trends =
monthly_spending.groupby('Customer_ID')['Total_Spend'].apply(calculate_
trend).reset_index()

spending_trends.rename(columns={'Total_Spend': 'Spending_Trend'},
inplace=True)


# Merge the new features into the customer_data dataframe

customer_data = pd.merge(customer_data, seasonal_buying_patterns,
on='Customer_ID')
```

```
customer_data = pd.merge(customer_data, spending_trends,
on='Customer_ID')


# Display the first few rows of the customer_data dataframe

customer_data.head()


# Changing the data type of 'CustomerID' to string as it is a unique
identifier and not used in mathematical operations

customer_data['Customer_ID'] = customer_data['Customer_ID'].astype(str)


# Convert data types of columns to optimal types

customer_data = customer_data.convert_dtypes()


customer_data.head(10)


customer_data.info()


"""#Outlier Detection and Treatment"""


# Selecting only numerical columns for fitting the model

numerical_columns =
customer_data.select_dtypes(include=[np.number]).columns


# Initializing the IsolationForest model with a contamination parameter
of 0.05

model = IsolationForest(contamination=0.05, random_state=0)


# Fitting the model on our dataset (using only numerical columns)

customer_data['Outlier_Scores'] =
model.fit_predict(customer_data[numerical_columns])
```

```
# Creating a new column to identify outliers (1 for inliers and -1 for
outliers)
customer_data['Is_Outlier'] = [1 if x == -1 else 0 for x in
customer_data['Outlier_Scores']]


# Display the first few rows of the customer_data dataframe
customer_data.head()


# Calculate the percentage of inliers and outliers
outlier_percentage =
customer_data['Is_Outlier'].value_counts(normalize=True) * 100


# Plotting the percentage of inliers and outliers
plt.figure(figsize=(12, 4))
outlier_percentage.plot(kind='barh', color='#ff6200')


# Adding the percentage labels on the bars
for index, value in enumerate(outlier_percentage):
    plt.text(value, index, f'{value:.2f}%', fontsize=15)


plt.title('Percentage of Inliers and Outliers')
plt.xticks(ticks=np.arange(0, 115, 5))
plt.xlabel('Percentage (%)')
plt.ylabel('Is Outlier')
plt.gca().invert_yaxis()
plt.show()


# Separate the outliers for analysis
outliers_data = customer_data[customer_data['Is_Outlier'] == 1]
```

```
# Remove the outliers from the main dataset
customer_data_cleaned = customer_data[customer_data['Is_Outlier'] == 0]


# Drop the 'Outlier_Scores' and 'Is_Outlier' columns
customer_data_cleaned =
customer_data_cleaned.drop(columns=['Outlier_Scores', 'Is_Outlier'])


# Reset the index of the cleaned data
customer_data_cleaned.reset_index(drop=True, inplace=True)


# Getting the number of rows in the cleaned customer dataset
customer_data_cleaned.shape[0]


"""# **CLV analysis**



"""


customer_data_cleaned.head()


# Calculate average customer lifespan in years, assuming daily
transactions
avg_customer_lifespan =
customer_data_cleaned['Average_Days_Between_Purchases'].mean() / 365


def calculate_clv(row, avg_customer_lifespan):
    # CLV Formula: (Average Transaction Value) * (Total Transactions) *
(Customer Lifespan)
```

```
    return row['Average_Transaction_Value'] * row['Total_Transactions']
* avg_customer_lifespan


customer_data_cleaned['CLV'] = customer_data_cleaned.apply(lambda row:
calculate_clv(row, avg_customer_lifespan), axis=1)


# Distribution of CLV

sns.histplot(customer_data_cleaned['CLV'], kde=True)

plt.title('Distribution of Customer Lifetime Value')

plt.xlabel('CLV')

plt.ylabel('Frequency')

plt.show()


# CLV by Geography

sns.boxplot(x='GEOGRAPHY', y='CLV', data=customer_data_cleaned)

plt.title('CLV by Geography')

plt.xlabel('Geography')

plt.ylabel('CLV')

plt.show()


customer_data_cleaned.columns #quick check


"""# **Clustering Preprocessing Steps**


# Correlation Analysis

"""


# Reset background style

sns.set_style('whitegrid')
```

```
# Select only numerical columns for the correlation matrix
numerical_data =
customer_data_cleaned.select_dtypes(include=[np.number])


# Calculate the correlation matrix
corr = numerical_data.corr()


# Define a custom colormap
colors = ['#ff6200', '#ffcaa8', 'white', '#ffcaa8', '#ff6200']
my_cmap = LinearSegmentedColormap.from_list('custom_map', colors,
N=256)


# Create a mask to only show the lower triangle of the matrix (since
it's mirrored around its top-left to bottom-right diagonal)
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask, k=1)] = True


# Plot the heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(corr, mask=mask, cmap=my_cmap, annot=True, center=0,
fmt='.2f', linewidths=2)
plt.title('Correlation Matrix', fontsize=14)
plt.show()


"""# Feature Scaling"""


# Initialize the StandardScaler
scaler = StandardScaler()
```

```
# List of columns that don't need to be scaled
columns_to_exclude = ['CustomerID','Day_Of_Week']

# Ensure only numeric columns are selected for scaling
numeric_columns =
customer_data_cleaned.select_dtypes(include=[np.number]).columns

# List of columns that need to be scaled
columns_to_scale = numeric_columns.difference(columns_to_exclude)

# Copy the cleaned dataset
customer_data_scaled = customer_data_cleaned.copy()

# Applying the scaler to the necessary columns in the dataset
customer_data_scaled[columns_to_scale] =
scaler.fit_transform(customer_data_scaled[columns_to_scale])

# Display the first few rows of the scaled data
print(customer_data_scaled.head())

"""# Dimensionality Reduction (PCA)"""

# Selecting only numerical columns for PCA
numerical_columns =
customer_data_scaled.select_dtypes(include=[np.number])

# Apply PCA
pca = PCA().fit(numerical_columns)

# Calculate the Cumulative Sum of the Explained Variance
```

```
explained_variance_ratio = pca.explained_variance_ratio_

cumulative_explained_variance = np.cumsum(explained_variance_ratio)


# Set the optimal k value (based on our analysis, we can choose 6)

optimal_k = 6


# Set seaborn plot style

sns.set(rc={'axes.facecolor': '#fcf0dc'}, style='darkgrid')


# Plot the cumulative explained variance against the number of
components

plt.figure(figsize=(20, 10))


# Bar chart for the explained variance of each component

barplot = sns.barplot(x=list(range(1,
len(cumulative_explained_variance) + 1)),

                      y=explained_variance_ratio,

                      color='#fcc36d',

                      alpha=0.8)


# Line plot for the cumulative explained variance

lineplot, = plt.plot(range(0, len(cumulative_explained_variance)),
cumulative_explained_variance,

                     marker='o', linestyle='--', color='#ff6200',
linewidth=2)


# Plot optimal k value line

optimal_k_line = plt.axvline(optimal_k - 1, color='red',
linestyle='--', label=f'Optimal k value = {optimal_k}')
```

```
# Set labels and title
plt.xlabel('Number of Components', fontsize=14)
plt.ylabel('Explained Variance', fontsize=14)
plt.title('Cumulative Variance vs. Number of Components', fontsize=18)

# Customize ticks and legend
plt.xticks(range(0, len(cumulative_explained_variance)))
plt.legend(handles=[barplot.patches[0], lineplot, optimal_k_line],
           labels=['Explained Variance of Each Component', 'Cumulative
Explained Variance', f'Optimal k value = {optimal_k}'],
           loc=(0.62, 0.1),
           frameon=True,
           framealpha=1.0,
           edgecolor='#ff6200')

# Display the variance values for both graphs on the plots
x_offset = -0.3
y_offset = 0.01
for i, (ev_ratio, cum_ev_ratio) in
enumerate(zip(explained_variance_ratio,
cumulative_explained_variance)):
    plt.text(i, ev_ratio, f"{ev_ratio:.2f}", ha="center", va="bottom",
fontsize=10)
    if i > 0:
        plt.text(i + x_offset, cum_ev_ratio + y_offset,
f"{cum_ev_ratio:.2f}", ha="center", va="bottom", fontsize=10)

plt.grid(axis='both')
plt.show()
```

```
# Selecting only numerical columns for PCA
numerical_columns =
customer_data_scaled.select_dtypes(include=[np.number])


# Creating a PCA object with 6 components
pca = PCA(n_components=6)


# Initializing the StandardScaler
scaler = StandardScaler()


# Fit and transform the scaler on the numerical data
scaled_numerical_columns = scaler.fit_transform(numerical_columns)


# Fitting and transforming the original data to the new PCA dataframe
customer_data_pca = pca.fit_transform(scaled_numerical_columns)


# Creating a new dataframe from the PCA dataframe, with columns labeled
PC1, PC2, etc.
customer_data_pca = pd.DataFrame(customer_data_pca,
columns=['PC'+str(i+1) for i in range(pca.n_components_)])


# Adding the CustomerID index back to the new PCA dataframe
customer_data_pca.index = customer_data_scaled.index


# Display the first few rows of the PCA-transformed data
print(customer_data_pca.head())


"""# Factor Loadings"""


# Identify and exclude non-numeric columns (example exclusion)
```

```
numeric_columns =
customer_data_scaled.select_dtypes(include=[np.number])


# Initialize PCA and StandardScaler

pca = PCA(n_components=6)

scaler = StandardScaler()


# Scale and fit_transform the numeric data

scaled_data = scaler.fit_transform(numeric_columns)


# Fit PCA on the scaled data

pca.fit(scaled_data)


# Create PCA components DataFrame

pc_df = pd.DataFrame(pca.components_.T, columns=['PC{}'.format(i+1) for
i in range(pca.n_components_)],

                     index=numeric_columns.columns)


# Function to highlight top 3 absolute values in each column

def highlight_top3(column):

    top3 = column.abs().nlargest(3).index

    return ['background-color: #ffeacc' if i in top3 else '' for i in
column.index]


# Apply highlighting function using .style.apply

styled_pc_df = pc_df.style.apply(highlight_top3, axis=0)


# Display styled DataFrame

styled_pc_df
```

```
"""#**K-Means Clustering**

# Determining Optimal K

# Elbow method
"""

# Set plot style, and background color
sns.set(style='darkgrid', rc={'axes.facecolor': '#fcf0dc'})

# Set the color palette for the plot
sns.set_palette(['#ff6200'])

# Instantiate the clustering model with the specified parameters
km = KMeans(init='k-means++', n_init=10, max_iter=100, random_state=0)

# Create a figure and axis with the desired size
fig, ax = plt.subplots(figsize=(12, 5))

# Instantiate the KElbowVisualizer with the model and range of k
values, and disable the timing plot
visualizer = KElbowVisualizer(km, k=(2, 15), timings=False, ax=ax)

# Fit the data to the visualizer
visualizer.fit(customer_data_pca)

# Finalize and render the figure
visualizer.show();
```

```
"""
# Silhouette Method"""

def silhouette_analysis(df, start_k, stop_k, figsize=(15, 16)):
    """
    Perform Silhouette analysis for a range of k values and visualize
the results.
    """

    # Set the size of the figure
    plt.figure(figsize=figsize)

    # Create a grid with (stop_k - start_k + 1) rows and 2 columns
    grid = gridspec.GridSpec(stop_k - start_k + 1, 2)

    # Assign the first plot to the first row and both columns
    first_plot = plt.subplot(grid[0, :])

    # First plot: Silhouette scores for different k values
    sns.set_palette(['darkorange'])

    silhouette_scores = []

    # Iterate through the range of k values
    for k in range(start_k, stop_k + 1):
        km = KMeans(n_clusters=k, init='k-means++', n_init=10,
max_iter=100, random_state=0)
        km.fit(df)
        labels = km.predict(df)
```

```
        score = silhouette_score(df, labels)
        silhouette_scores.append(score)

    best_k = start_k + silhouette_scores.index(max(silhouette_scores))

    plt.plot(range(start_k, stop_k + 1), silhouette_scores, marker='o')
    plt.xticks(range(start_k, stop_k + 1))
    plt.xlabel('Number of clusters (k)')
    plt.ylabel('Silhouette score')
    plt.title('Average Silhouette Score for Different k Values',
fontsize=15)

    # Add the optimal k value text to the plot
    optimal_k_text = f'The k value with the highest Silhouette score
is: {best_k}'
    plt.text(10, 0.23, optimal_k_text, fontsize=12,
verticalalignment='bottom',
             horizontalalignment='left', bbox=dict(facecolor='#fcc36d',
edgecolor='#ff6200', boxstyle='round, pad=0.5'))

    # Second plot (subplot): Silhouette plots for each k value
    colors = sns.color_palette("bright")

    for i in range(start_k, stop_k + 1):
        km = KMeans(n_clusters=i, init='k-means++', n_init=10,
max_iter=100, random_state=0)
        row_idx, col_idx = divmod(i - start_k, 2)

        # Assign the plots to the second, third, and fourth rows
```

```
        ax = plt.subplot(grid[row_idx + 1, col_idx])

        visualizer = SilhouetteVisualizer(km, colors=colors, ax=ax)
        visualizer.fit(df)

        # Add the Silhouette score text to the plot
        score = silhouette_score(df, km.labels_)
        ax.text(0.97, 0.02, f'Silhouette Score: {score:.2f}',
fontsize=12, \
                ha='right', transform=ax.transAxes, color='red')

        ax.set_title(f'Silhouette Plot for {i} Clusters', fontsize=15)

    plt.tight_layout()
    plt.show()

silhouette_analysis(customer_data_pca, 3, 12, figsize=(20, 50))

"""# Unsupervised Machine Learning Clustering Model - K-means"""

# Apply KMeans clustering using the optimal k
kmeans = KMeans(n_clusters=3, init='k-means++', n_init=10,
max_iter=100, random_state=0)
kmeans.fit(customer_data_pca)

# Get the frequency of each cluster
cluster_frequencies = Counter(kmeans.labels_)

# Create a mapping from old labels to new labels based on frequency
```

```
label_mapping = {label: new_label for new_label, (label, _) in
                 enumerate(cluster_frequencies.most_common())}

# Reverse the mapping to assign labels as per your criteria
label_mapping = {v: k for k, v in {2: 1, 1: 0, 0: 2}.items()}

# Apply the mapping to get the new labels
new_labels = np.array([label_mapping[label] for label in
kmeans.labels_])

# Append the new cluster labels back to the original dataset
customer_data_cleaned['cluster'] = new_labels

# Append the new cluster labels to the PCA version of the dataset
customer_data_pca['cluster'] = new_labels

# Display the first few rows of the original dataframe
customer_data_cleaned.head()

# Setting up the color scheme for the clusters (RGB order)
colors = ['#e8000b', '#1ac938', '#023eff']

"""# PCA and K-Means Scatterplots"""

# Define the custom color palette
colors = {0: 'red', 1: 'green', 2: 'blue'}

# Create a figure with two subplots side by side
fig, ax = plt.subplots(1, 2, figsize=(14, 6))
```

```
# Scatterplot 1: PCA-based (original PCA components)
ax[0].scatter(customer_data_pca['PC1'], customer_data_pca['PC2'],
c='gray', s=50, alpha=0.6)
ax[0].set_title('PCA-based Scatterplot')
ax[0].set_xlabel('PC1')
ax[0].set_ylabel('PC2')

# Map new_labels to corresponding colors
mapped_colors = np.array([colors[label] for label in new_labels])

# Scatterplot 2: K-Means Clustering based on PCA components
scatter = ax[1].scatter(customer_data_pca['PC1'],
customer_data_pca['PC2'], c=mapped_colors, s=50, alpha=0.6)
ax[1].set_title('K-Means Clustering (k=3)')
ax[1].set_xlabel('PC1')
ax[1].set_ylabel('PC2')

# Custom legend for the clusters
import matplotlib.patches as mpatches
legend_labels = [mpatches.Patch(color=colors[i], label=f'Cluster {i}')
for i in range(3)]
ax[1].legend(handles=legend_labels, title="Clusters")

# Adjust layout and display the plots
plt.tight_layout()
plt.show()

"""# 3D Visualization of Top Principal Components"""
```

```
# Create separate data frames for each cluster
cluster_0 = customer_data_pca[customer_data_pca['cluster'] == 0]
cluster_1 = customer_data_pca[customer_data_pca['cluster'] == 1]
cluster_2 = customer_data_pca[customer_data_pca['cluster'] == 2]

# Create a 3D scatter plot
fig = go.Figure()

# Add data points for each cluster separately and specify the color
fig.add_trace(go.Scatter3d(x=cluster_0['PC1'], y=cluster_0['PC2'],
z=cluster_0['PC3'],
                           mode='markers', marker=dict(color=colors[0],
size=5, opacity=0.4), name='Cluster 0'))
fig.add_trace(go.Scatter3d(x=cluster_1['PC1'], y=cluster_1['PC2'],
z=cluster_1['PC3'],
                           mode='markers', marker=dict(color=colors[1],
size=5, opacity=0.4), name='Cluster 1'))
fig.add_trace(go.Scatter3d(x=cluster_2['PC1'], y=cluster_2['PC2'],
z=cluster_2['PC3'],
                           mode='markers', marker=dict(color=colors[2],
size=5, opacity=0.4), name='Cluster 2'))

# Set the title and layout details
fig.update_layout(
    title=dict(text='3D Visualization of Customer Clusters in PCA
Space', x=0.5),
    scene=dict(
        xaxis=dict(backgroundcolor="#fcf0dc", gridcolor='white',
title='PC1'),
        yaxis=dict(backgroundcolor="#fcf0dc", gridcolor='white',
title='PC2'),
```

```
        zaxis=dict(backgroundcolor="#fcf0dc", gridcolor='white',
title='PC3'),
    ),
    width=900,
    height=800
)

# Show the plot
fig.show()

"""# Cluster Distribution Visualization"""

# Calculate the percentage of customers in each cluster
cluster_percentage =
(customer_data_pca['cluster'].value_counts(normalize=True) *
100).reset_index()
cluster_percentage.columns = ['Cluster', 'Percentage']
cluster_percentage.sort_values(by='Cluster', inplace=True)

# Define a color palette
colors = sns.color_palette("husl",
len(cluster_percentage['Cluster'].unique()))

# Create a horizontal bar plot
plt.figure(figsize=(10, 4))
sns.barplot(x='Percentage', y='Cluster', data=cluster_percentage,
orient='h', hue='Cluster', palette=colors, dodge=False, legend=False)

# Adding percentages on the bars
for index, value in enumerate(cluster_percentage['Percentage']):
```

```
    plt.text(value+0.5, index, f'{value:.2f}%')

plt.title('Distribution of Customers Across Clusters', fontsize=14)
plt.xticks(ticks=np.arange(0, 50, 5))
plt.xlabel('Percentage (%)')

# Show the plot
plt.show()

"""# Evaluation Metrics"""

# Compute number of customers
num_observations = len(customer_data_pca)

# Separate the features and the cluster labels
X = customer_data_pca.drop('cluster', axis=1)
clusters = customer_data_pca['cluster']

# Compute the metrics
sil_score = silhouette_score(X, clusters)
calinski_score = calinski_harabasz_score(X, clusters)
davies_score = davies_bouldin_score(X, clusters)

# Create a table to display the metrics and the number of observations
table_data = [
    ["Number of Observations", num_observations],
    ["Silhouette Score", sil_score],
    ["Calinski Harabasz Score", calinski_score],
    ["Davies Bouldin Score", davies_score]
```

```
]


# Print the table
print(tabulate(table_data, headers=["Metric", "Value"],
tablefmt='pretty'))


"""# Radar Chart Comparisons"""


# Setting 'CustomerID' column as index and assigning it to a new
dataframe
df_customer = customer_data_cleaned.set_index('Customer_ID')


# Separate the features and the cluster column
features = df_customer.drop(columns=['cluster'])
clusters = df_customer['cluster']


# One-hot encode categorical variables
features_encoded = pd.get_dummies(features)


# Standardize the data
scaler = StandardScaler()
features_standardized = scaler.fit_transform(features_encoded)


# Create a new dataframe with standardized values and add the cluster
column back
df_customer_standardized = pd.DataFrame(features_standardized,
columns=features_encoded.columns, index=df_customer.index)
df_customer_standardized['cluster'] = clusters


# Calculate the centroids of each cluster
```

```
cluster_centroids = df_customer_standardized.groupby('cluster').mean()


# Function to create a radar chart
def create_radar_chart(ax, angles, data, color, cluster):
    # Plot the data and fill the area
    ax.fill(angles, data, color=color, alpha=0.4)
    ax.plot(angles, data, color=color, linewidth=2, linestyle='solid')

    # Add a title
    ax.set_title(f'Cluster {cluster}', size=20, color=color, y=1.1)

# Set data
labels = np.array(cluster_centroids.columns)
num_vars = len(labels)

# Compute angle of each axis
angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()

# The plot is circular, so we need to "complete the loop" and append
the start to the end
labels = np.concatenate((labels, [labels[0]]))
angles += angles[:1]

# Define colors for each cluster
colors = ['red', 'green', 'blue']

# Initialize the figure
fig, ax = plt.subplots(figsize=(20, 10), subplot_kw=dict(polar=True),
nrows=1, ncols=3)
```

```
# Create radar chart for each cluster
for i, color in enumerate(colors):
    data = cluster_centroids.loc[i].tolist()
    data += data[:1]  # Complete the loop
    create_radar_chart(ax[i], angles, data, color, i)

# Set x-ticks and labels for all subplots
for i in range(3):
    ax[i].set_xticks(angles[:-1])
    ax[i].set_xticklabels(labels[:-1])

# Add a grid
for i in range(3):
    ax[i].grid(color='grey', linewidth=0.5)

# Display the plot
plt.tight_layout()
plt.show()

"""# Histogram Chart Comparisons"""

# Plot histograms for each feature segmented by the clusters
features = customer_data_cleaned.columns[1:-1]
clusters = customer_data_cleaned['cluster'].unique()
clusters.sort()

# Setting up the subplots
n_rows = len(features)
```

```
n_cols = len(clusters)

fig, axes = plt.subplots(n_rows, n_cols, figsize=(20, 3*n_rows))


# Plotting histograms

for i, feature in enumerate(features):

    for j, cluster in enumerate(clusters):

        data = customer_data_cleaned[customer_data_cleaned['cluster']
== cluster][feature]

        axes[i, j].hist(data, bins=20, color=colors[j], edgecolor='w',
alpha=0.7)

        axes[i, j].set_title(f'Cluster {cluster} - {feature}',
fontsize=15)

        axes[i, j].set_xlabel('')

        axes[i, j].set_ylabel('')


# Adjusting layout to prevent overlapping

plt.tight_layout()

plt.show()


"""# **Summary statistics for the entire dataset**"""


# Summary statistics for the entire dataset

overall_summary = customer_data_cleaned.describe()

print(overall_summary)


# Group by GEOGRAPHY and calculate summary statistics

geo_summary = customer_data_cleaned.groupby('GEOGRAPHY').describe()

print(geo_summary)


"""# Cluster descriptives summary"""
```

```
# Group by clusters and calculate summary statistics
cluster_summary = customer_data_cleaned.groupby('cluster').describe()
print(cluster_summary)

"""# Revenue"""

# Calculate total spend if it's not already present
customer_data_cleaned['Total_Spend'] =
customer_data_cleaned['Average_Transaction_Value'] *
customer_data_cleaned['Total_Transactions']

# Now calculate the total company revenue
total_revenue = customer_data_cleaned['Total_Spend'].sum()
print(f"Total Company Revenue: ${total_revenue}")

customer_data_cleaned['Total_Spend'] =
customer_data_cleaned['Average_Transaction_Value'] *
customer_data_cleaned['Total_Transactions']
total_revenue = customer_data_cleaned['Total_Spend'].sum()
customer_data_cleaned['Total_Revenue'] = total_revenue

revenue_per_country =
customer_data_cleaned.groupby('GEOGRAPHY')['Total_Spend'].sum().reset_i
ndex()
print(revenue_per_country)

# Assuming customer_data_cleaned is your DataFrame
revenue_per_cluster =
customer_data_cleaned.groupby('cluster')['Total_Spend'].sum().reset_ind
ex()
```

```
print(revenue_per_cluster)


"""# **Follow-up Tableau Analysis Set-Up**"""


df_customer_standardized
# check dataset to be exported for subsequent analysis


# simplify the file name:


Final_df_capstone = df_customer_standardized


# Save the modified DataFrame as an Excel and CSV file
Final_df_capstone.to_excel('Final_df_capstone.xlsx', index=True)
Final_df_capstone.to_csv('Final_df_capstone.csv', index=True)


# Download the Excel file
files.download('Final_df_capstone.xlsx')


# Download the CSV file
files.download('Final_df_capstone.csv')


print("Downloaded as 'Final_df_capstone.xlsx' and
'Final_df_capstone.csv'.")
```