A Minor Project report entitled On

ML-DRIVEN CUSTOMER SEGMENTATION

In partial fulfillment of the requirements for the award of

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering (Data Science)

Submitted by

BANDRA ANUSHA (21E51A6704)

ADITHYA CHILUPURI (21E516707)

THAKUR ARYAN SINGH (21E51A6760)

V RAMESH (21E51A6762)

Under the Esteemed guidance of

MR. NAVA KISHORE

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Gowdavelly (Village), Medchal, Hyderabad, Telangana, 501401

(UGC Autonomous, Affiliated to JNTUH, Accredited by NAAC (A+) and NBA)

2024-2025

HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(UGC Autonomous, Affiliated to JNTUH, Accredited by NAAC (A+) and NBA)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CERTIFICATE

This is to certify that the Minor Project entitled "**ML-DRIVEN CUSTOMER SEGMENTATION**" is being submitted by **Bandra Anusha** bearing hall ticket number **21E51A6704**, **Adithya chilupuri** bearing hall ticket number **21E51A6707**, **Thakur Aryan Singh** bearing hall ticket number **21E51A6760**, **V Ramesh** bearing hall ticket number **21E51A6762,** in partial fulfillment of the requirements for the degree **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)** by the Jawaharlal Nehru Technological University, Hyderabad, during the academic year 2024-2025. The matter contained in this document has not been submitted to any other University or institute for the award of any degree or diploma.

Under the Guidance of	**Head of the Department**
Mr.Nava Kishore	**Dr. M.V.A Naidu**
Associate Professor	**Professor & HoD**
Internal Examiner	**External Examiner**

HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(UGC Autonomous, Affiliated to JNTUH, Accredited by NAAC (A+) and NBA)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

DECLARATION

We "**Bandra Anusha, Adithya chilupuri, Thakur Aryan Singh, V Ramesh**" students of '**Bachelor of Technology in CSE (Data Science)**', session: 2024- 2025, Hyderabad Institute of Technology and Management, Gowdavelly, Hyderabad, Telangana State, hereby declare that the work presented in this Minor Project entitled '**ML-DRIVEN CUSTOMER SEGMENTATION**' is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of engineering ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

BANDRA ANUSHA (21E51A6704)

ADITHYA CHILUPURI (21E516707)

THAKUR ARYAN SINGH (21E51A6760)

V RAMESH (21E51A6762)

ACKNOWLEDGEMENT

An endeavor of a long period can be successful only with the advice of many well-wishers. We would like to thank our chairman, **SRI. ARUTLA PRASHANTH**, for providing all the facilities to carry out Project Work successfully. We would like to thank our Principal **DR. S. ARVIND**, who has inspired a lot through their speeches and providing this opportunity to carry out our Minor Project successfully. We are very thankful to our Head of the Department, **DR. M.V.A NAIDU** and B-Tech Project Coordinator **DR G. APARNA**. We would like to specially thank my internal supervisor **MR. NAVA KISHORE,** our technical guidance, constant encouragement and enormous support provided to us for carrying out our Minor Project. We wish to convey our gratitude and express sincere thanks to all **D.C (DEPARTMENTAL COMMITTEE)** and **P.R.C (PROJECT REVIEW COMMITTEE)** members, non-teaching staff for their support and Cooperation rendered for successful submission of our Minor Project work.

BANDRA ANUSHA (21E51A6704)

ADITHYA CHILUPURI (21E516707)

THAKUR ARYAN SINGH (21E51A6760)

V RAMESH (21E51A6762)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Customer segmentation has become a popular method for dividing a company's customers to enhance retention and profitability. In this study, customers from various organizations are classified based on behavioral characteristics such as spending an income. By focusing on these behavioral aspects, the classification methods become more efficient compared to others. Utilizing the clustering algorithm, customers are grouped based on their behavioral traits. These clusters enable companies to target individual customers effectively, tailoring marketing campaigns and social media content to their interests. In this project, We will perform unsupervised clustering on customer records from a grocery firm's database. Customer segmentation involves grouping customers to reflect similarities within each cluster. This segmentation optimizes the significance of each customer to the business, allowing for product modifications to meet distinct needs and behaviors. It also helps address the concerns of different types of customers, enhancing overall business strategy.

Keywords: Customer segmentation, Behavioral characteristics, Classification, Clustering, Behavioral traits, Targeted marketing, Social media content, Unsupervised clustering, Grocery firm, Product modification, Customer needs, Business strategy

CHAPTER 1

1.1 INTRODUCTION

The Business Problem of any company in retail, no matter the industry, ends up collecting, creating, and manipulating data over the course of their lifespan. These data are produced and recorded in a variety of contexts, most notably in the form of shipments, tickets, employee logs, and digital interactions. Each of these instances of data describes a small piece of how the company operates, for better or for worse. The more access to data that one has, the better the picture that the data can delineate. With a clear picture made from data, details previously unseen begin to emerge that spur new insights and innovations.

Companies that utilize proper data science and data mining practices allow themselves to dig further into their own operating strategies, which in turn allows them to optimize their commercial practices. As a result, there are increasing motivations for investigating phenomena and data that cannot be simply answered:

"*Why is product B purchased more on the first Saturday of every month compared to other weekends?, If a customer bought product B, will they like product C?, What are the defining traits of our customers? Can we predict what customers will want to buy?"*

These questions will be the broad focus of this work.

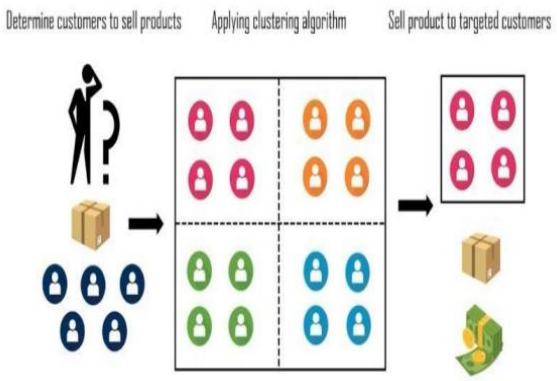


Figure 1.1: Customer Segmentation

Customer Segmentation analysis is the process of identifying and understanding the unique characteristics and traits that make up an individual customer's personality. This information can be used by companies to tailor their marketing and sales efforts to better target and serve each customer's specific needs and preferences.

1.2 PROBLEM STATEMENT

Customer Personality Analysis is a detailed analysis of a company's ideal customers. It helps a business to better understand its customers and makes it easier for them to modify products according to the specific needs, behaviors and concerns of different types of customers.

Customer personality analysis helps a business to modify its product based on its target customers from different types of customer segments. For example, instead of spending money to market a new product to every customer in the company's database, a company can analyze which customer segment is most likely to buy the product and then market the product only on that particular segment.

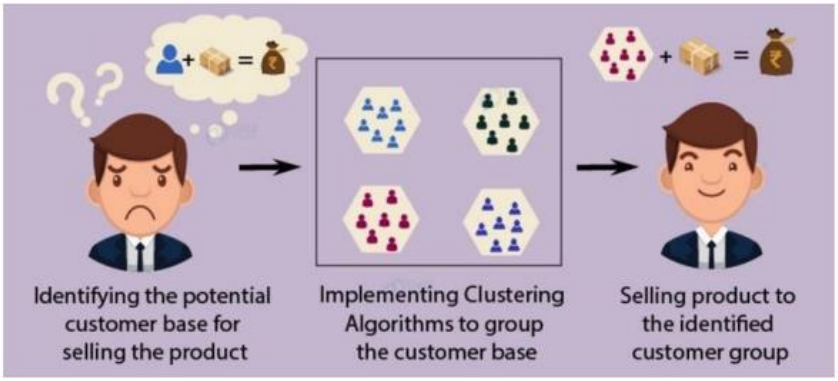


Figure 1.2: Grouping of customers

1.3 OBJECTIVES OF PROJECT

The significance of this project extends to both businesses and the field of data-driven decision-making:

1. Improved Customer Engagement: Businesses can leverage the identified customer segments to engage with their customers more effectively. Personalized marketing efforts can lead to increased customer satisfaction and loyalty.

2. Enhanced Profitability: Targeted marketing strategies for each customer segment can lead to higher conversion rates and increased sales. This, in turn, can boost profitability and revenue generation.

3. Data-Driven Decision-Making: The project highlights the power of data-driven decision-making in marketing. By harnessing the capabilities of machine learning, businesses can make more informed and strategic choices.

4. Adaptability to Change: The K-means clustering algorithm's adaptability to changing data ensures that customer segments remain relevant over time. As customer behavior evolves, so can the marketing strategies.

Key Findings:

Throughout the project, several critical findings and outcomes have emerged:

1. Effective Customer Segmentation: The implementation of the K-means clustering algorithm allowed for the effective segmentation of the Mall Customers dataset into distinct customer groups. These segments were characterized by unique behaviors and spending habits.

2. Optimal Cluster Selection: The use of the Elbow Method for selecting the optimal number of clusters (k) helped in identifying that 5 clusters were most suitable for representing the customer base, balancing granularity and practicality.

3. Multidimensional Insights: By considering multiple attributes, such as annual income and spending score, the project achieved multidimensional customer segmentation. This comprehensive approach provided a more accurate representation of customer diversity

4. Tailored Marketing Strategies: The project's results provided actionable insights for tailoring marketing strategies to each customer segment. From high-spending customers to those with specific spending constraints, businesses now have a roadmap for addressing the needs of different customer groups.

CHAPTER 2

2.1 PROPOSED SOLUTION

Traditionally, customer personality analysis has been done manually by marketing and sales teams, who would use their expertise and experience to identify common patterns and trends among customers. However, with the advent of data mining and machine learning, it is now possible to automate this process using algorithms that can analyze large amounts of data and identify common patterns and traits among customers.

One type of machine learning algorithm that can be used for customer personality analysis is unsupervised learning. Unsupervised learning algorithms are trained on a large amount of data and can automatically detect patterns and similarities among customers without being explicitly told what to look for. This makes them particularly well suited for customer personality analysis, as they can uncover subtle differences and trends that might not be immediately apparent to humans.

In this Report, we will discuss the use of unsupervised learning algorithms for customer segmentation analysis, and how they can be used by companies to better understand and serve their customers.

2.2 LITERATURE SURVEY

Customer segmentation is a fundamental practice in modern marketing, aiming to divide a diverse customer base into homogeneous groups. This segmentation helps businesses better understand their customers, design targeted marketing strategies, and ultimately improve sales. Clustering, a popular data analysis technique, plays a crucial role in identifying these distinct customer segments. This literature survey explores key research papers and studies related to customer segmentation analysis using clustering for the purpose of sales improvement.

"Market Segmentation: Conceptual and Methodological Foundations" by Michel Wedel and Wagner Kamakura (2000)

This foundational work introduces the concept of market segmentation and highlights its importance in marketing strategy. It covers various segmentation techniques, including clustering, and discusses their benefits in improving sales and customer satisfaction. The paper provides a comprehensive overview of segmentation methods and their applications.

"Customer Segmentation: A Review" by V. Kumar and Rohit Aggarwal (2019)

This review paper provides an in-depth analysis of customer segmentation techniques, focusing on both traditional methods and modern data-driven approaches. It emphasizes the role of clustering in creating meaningful customer segments and discusses the impact of segmentation on

sales and business outcomes. The paper highlights the importance of considering various data sources, such as transaction data and customer behavior, in the segmentation process.

"Customer Segmentation Based on Purchasing Behavior Using K-Means Clustering" by M.A.Hossain and Mohammad Shorif Uddin (2016)

This research paper focuses on a practical application of clustering, specifically K-means clustering, for customer segmentation based on purchasing behavior. The study demonstrates the effectiveness of clustering in identifying distinct customer groups and proposes strategies to target these segments. The findings suggest that personalized marketing approaches derived from clustering analysis can significantly improve sales.

"An Empirical Analysis of Customer Segmentation Strategies Using Clustering" by Elham Fadalyet al. (2018)

This empirical study explores different customer segmentation strategies using clustering techniques and evaluates their impact on sales and customer retention. The paper compares K-means clustering with other methods and discusses the advantages and limitations of each approach. It provides valuable insights into the practical implementation of clustering for sales improvement.

"Enhancing Customer Segmentation with Machine Learning" by Abhijit J. Patil and Prashant R.Nair(2021)

This paper discusses the integration of machine learning techniques, including clustering algorithms, for customer segmentation. It emphasizes the benefits of utilizing advanced algorithms to uncover complex patterns in customer data. The study showcases real-world examples of businesses that have successfully improved sales by adopting machine learning-driven segmentation strategies.

"Customer Segmentation for E-commerce: A Comparative Study of Clustering Algorithms" by G.Santosh Kumar and P. Senthil Kumar (2018)

This comparative study evaluates the performance of various clustering algorithms for customer segmentation in the e-commerce domain. The paper assesses the effectiveness of algorithms like K- means, DBSCAN, and hierarchical clustering in identifying meaningful customer segments. It discusses the implications of segmentation on sales and provides insights into algorithm selection for different scenarios.

CHAPTER 3

3.1 REQUIREMENTS AND INSTALLATION

3.1.1 SOFTWARE REQUIREMENTS

A significant application of machine learning is in the detection of water quality, it mainly defines essential water quality attributes in order to ensure the reliability of drinking water. So executing this sort of system necessitates a few software requirements, which ranges from data handling and preprocessing to assessment and execution of machine learning models. So these are the following precise software requirements for enlarging a water quality detection system that operates using a Grid Search algorithm.

1.The Operating System

Windows 10- It provides good assistance for a wide range of machine learning libraries and development tools, and in the view of its robustness, reliability and integration with a significant range of machine learning frameworks, Linux (Ubuntu 20.04 or later) is recommended.

2. Programming Language

Python 3.7 or later- Based on its variety of libraries for the data analysis, visualization and machine learning python is considered as the predominant language.

3. Integrated Development Environment (IDE)

Jupyter Notebook - its an open-source web software which permits the users to create and share documents consisting of active code, equations, visualizations and narrative text. PyCharm- PyCharm, it’s an IDE(integrated development environment) which provides dynamic capabilities like reviewing of code, graphical debugging and integration with Jupyter Notebooks for the evolution of python.

Libraries

Seaborn

Seaborn is one of Python's most amazing graphical statistical visualization libraries. Seaborn provides several color palettes and delightful styles by default to make it a lot enticing to form several applied math charts in Python.The main aim is to visualize the central part of data understanding and exploration in a lot of enticing ways. This is often supported by the core of matplotlib; it's a library and also provides a record-oriented API. This library is integrated with Panda's knowledge structures, thus you'll be able to simply switch between totally different visual representations of a specific variable to better understand the dataset to be used for analysis.

Numpy

NumPy implies Numeric Python, a Python bundle for figuring and interacting multi-layered and one-layered cluster parts. Travis Oliphant made the NumPy bundle in 2005 also due to the utility of the past Numeric module in another Numarray module. This is a Python expansion module composed principally in C. Different capacities adjust rapid mathematical computations. NumPy gives a spread of elite execution information structures that carry out complex exhibits and frameworks. These DS region units are utilized for ideal estimations on arithmetic

Pandas

Pandas is characterized in Python as an open-source library that has strong data control. The name of this library comes from the term board data, which suggests financial science made out of two-layered data. Utilized for data examination in Python and created by Wes McKinney in 2008. data investigation needs a store of cycles like z, python, panda, etc. Be that as it may, I like pandas because they are quicker, simpler, and much more informative than various apparatuses. Pandas are made on the NumPy bundle. All in all, NumPy is required for the panda to figure. Before pandas were presented, Python was ready for data anyway had confined help for data investigation. That is any place pandas came in, expanding the potential for data examination. regardless of the inventory of the data, you'll play out the 5 key advances expected to technique and dissect the data. NS. Stacking, activity, arrangement, displaying, and examination

Sklearn

Scikit-learn, also referred to as sklearn, was called scikit. Learning can be a free AI programming framework library for the Python fake language. SVM, irregular woods, inclination supporting, k means, DBSCAN, and quite a bit of other grouping, relapse, and pack calculations are intended to work with the mathematical and logical Python libraries NumPy and SciPy. Increment. Scikit-learn can be an undertaking financed by NumFOCUS

3.1.2 HARDWARE REQUIREMENTS

Hardware requirements are the components which are physical such as RAM, CPU, and other storage devices needed for the execution of software applications or running other tasks.

1.CPU(Central Processing Unit)

A basic small-core processor (CPU)would be enough for basic Ml tasks and smaller datasets.For larger and complex datasets, we can use a processor which is of high performance. We used Intel Core i5. The CPU is basically the brain of the computer and used for the execution of all tasks.

2.RAM(Random Access Memory)

For running basic Ml tasks and smaller datasets, we can use 8GB RAM which is more than enough. However, since our dataset is large and complex, we used 16GB RAM. RAM of large size ensures large data to be executed efficiently without any issues.

3. Storage and Capacity

SSD(Solid State Drive) - We used an SSD which plays a critical role in handling large datasets. They help in reading and writing the data at a faster pace.Based on the size of the dataset, the capacity keeps changing. We are using 256GB storage. Usually, it goes between 256 GB-512GB. This helps us have enough space for our models and datasets.

4. OS(Operating System)

An operating system is the most important for hardware management. It acts as an intermediary between the users and hardware , by providing an application for running the tasks. Ml frameworks can be run with Windows, Mac, and Linux operating systems; they are compatible with all three. We used a Windows operating system for our project.

INSTALL DEPENDENCIES

```
Pandas:           $ sudo pip install pandas
numpy:            $ sudo pip install numpy
scipy:            $ sudo pip install scipy
scikit-learn:     $ sudo pip install -U scikit-learn
matplotlib:
                  $ sudo apt-get install libfreetype6-dev libpng-dev
                  $ sudo pip install matplotlib
seaborn:          $ sudo pip install seaborn
jupyter notebook: $ sudo apt-get -y install ipython ipython-notebook
                  $ sudo -H pip install jupyter
nltk:              $ sudo pip install nltk
wordcloud:         $ sudo pip install wordcloud
```

Figure 3.1: Required dependencies

CHAPTER 4

4.1 BLOCK DIAGRAM

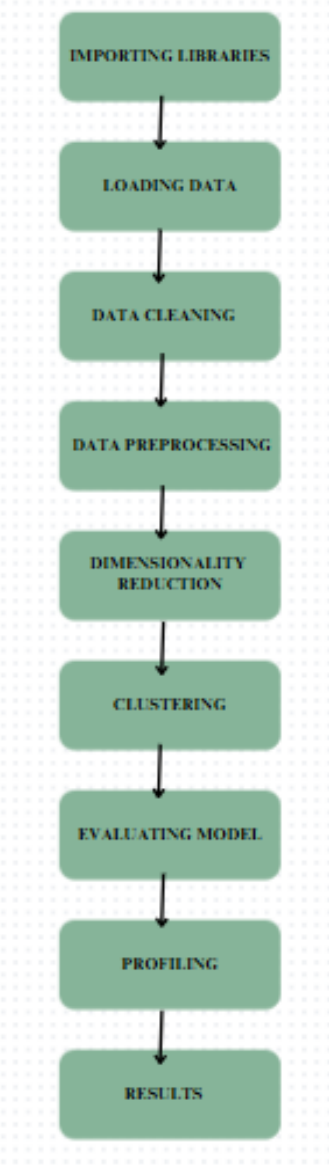


Figure 4.1: Block Diagram

CHAPTER 5

5.1 METHODOLOGY

In this project, we will be performing an unsupervised clustering of data on the customer's records from a groceries firm's database. Customer segmentation is the practice of separating customers into groups that reflect similarities among customers in each cluster. we will divide customers into segments to optimize the significance of each customer to the business. To modify products according to distinct needs and behaviors of the customers. It also helps the business to cater to the concerns of different types of customers.

To achieve that goal, we will go through under steps:

1. Importing libraries.
2. Loading data.
3. Data cleaning.
4. Data preprocessing.
5. Dimensionality reduction.
6. Clustering.
7. Evaluating model.
8. Profiling.
9. Results.

5.1.1 IMPORTING LIBRARIES

For Data Preprocessing, certain libraries are very important. They needed to be imported. They are:

Pandas: For data manipulation and handling.

NumPy: For numerical operations.

Matplotlib and **Seaborn**: For data visualization, especially to plot the Elbow curve.

Scikit-Learn: To access clustering algorithms, including Agglomerative Clustering, and other preprocessing tools.

Scipy: Required for hierarchical clustering dendrograms and to support Agglomerative Clustering.

```
#Importing the Libraries
import numpy as np
import pandas as pd
import datetime
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import colors
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt, numpy as np
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import AgglomerativeClustering
from matplotlib.colors import ListedColormap
from sklearn import metrics
import warnings
import sys
```

5.1.2 LOADING DATA

Importing Dataset: We have selected a dataset from Kaggle and have read it with the help of a CSV file

```
[ ]  #Loading the dataset
     data = pd.read_csv("/content/drive/MyDrive/marketing_campaign.csv", sep="\t")
     print("Number of datapoints:", len(data))
     data.head()
```

Dataset Description

The dataset for this project is a public dataset from Kaggle. This dataset has 2,240 rows of observations and 28 columns of variables. Among the variables, there are 5-character variables and 23 numerical variables.

The dataset used for customer segmentation in this project contains information about customers, their purchasing habits, and interactions with promotional campaigns. The data is organized into four main categories: **People**, **Products**, **Promotions**, and **Place**.

Attributes

People

- ID: Customer's unique identifier
- Year_Birth: Customer's birth year
- Education: Customer's education level
- Marital_Status: Customer's marital status

- Income: Customer's yearly household income
- Kidhome: Number of children in customer's household
- Teenhome: Number of teenagers in customer's household
- Dt_Customer: Date of customer's enrollment with the company
- Recency: Number of days since customer's last purchase
- Complain: 1 if the customer complained in the last 2 years, 0 otherwise

Products

- MntWines: Amount spent on wine in last 2 years
- MntFruits: Amount spent on fruits in last 2 years
- MntMeatProducts: Amount spent on meat in last 2 years
- MntFishProducts: Amount spent on fish in last 2 years
- MntSweetProducts:Amount spent on sweets in last 2 years
- MntGoldProds: Amount spent on gold in last 2 years

Promotion

- NumDealsPurchases: Number of purchases made with a discount
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- Response: 1 if customer accepted the offer in the last campaign, 0 otherwise

Place

- NumWebPurchases: Number of purchases made through the company’s website
- NumCatalogPurchases: Number of purchases made using a catalogue
- NumStorePurchases: Number of purchases made directly in stores
- NumWebVisitsMonth: Number of visits to company’s website in the last month

5.1.3 DATA CLEANING

Data cleaning refers to identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Data cleansing can be done in batches using scripting or a data quality firewall, or it can be done interactively with data wrangling tools. Data cleaning is an essential step in the customer segmentation process, as it helps to ensure that the data used for analysis is accurate and reliable. To clean data for customer segmentation, you will need to identify and address any errors, inconsistencies, and missing values in your dataset. This can be done by manually reviewing the data, using visualizations, or applying statistical tests. The dataset was first inspected for missing values and incorrect data types. The following steps were undertaken for cleaning and feature engineering:

1. **Handling Missing Values**:
    - The feature Income had missing values in 1.07% of the rows, so rows with missing values were dropped, reducing the dataset from 2240 to 2216 records.

```
#To remove the NA values
data = data.dropna()
print("The total number of data-points after removing the rows with missing values are:", len(data))
```

```
The total number of data-points after removing the rows with missing values are: 2216
```

2. **Datetime Parsing**:
    - The Dt_Customer feature, which contains the date a customer joined, was converted from object type to a proper datetime format. The number of days since the customer's registration (Customer_For) was calculated relative to the most recent customer enrollment date, which was 2014-12-06. The oldest date in the dataset was 2012-01-08.
3. **Exploration of Categorical Features**:
    - The Marital_Status and Education features were explored. Unusual categories in Marital_Status such as "Absurd" and "YOLO" were found and addressed during feature engineering.
4. **Feature Engineering**: Several new features were created for better analysis and segmentation:
    - **Age**: Derived from the Year_Birth to reflect the customer's age as of 2023.
    - **Spent**: The total spending was calculated by summing up all monetary spend categories (Wines, Fruits, Meat, Fish, Sweets, Gold).
    - **Living_With**: A feature created from Marital_Status to group customers into categories such as "Partner" or "Alone".
    - **Children**: Calculated as the sum of Kidhome and Teenhome, indicating the total children in a household.
    - **Family_Size**: Derived as the sum of household members (Living_With and Children).
    - **Is_Parent**: A binary indicator for parenthood, based on whether the household has children.
    - **Simplified Education Levels**: The Education feature was re-categorized into three simplified groups: "Undergraduate," "Graduate," and "Postgraduate."
5. **Dropping Redundant Features**:
    - Features such as Marital_Status, Dt_Customer, Z_CostContact, Z_Revenue, Year_Birth, and ID were dropped as they were either redundant or had been replaced with more useful features.
6. **Outlier Removal**:
    - Outliers were identified in the Age and Income columns. Records with Age greater than 90 and Income greater than 600,000 were removed. The total number of records was reduced to 2212 after outlier removal.

```
[ ]  #Dropping the outliers by setting a cap on Age and income.
     data = data[(data["Age"]<90)]
     data = data[(data["Income"]<600000)]
     print("The total number of data-points after removing the outliers are:", len(data))
```

```
The total number of data-points after removing the outliers are: 2212
```

7. **Correlation Analysis**:
   - A correlation matrix was generated to analyze relationships between numerical features. Categorical attributes were excluded from this analysis.This cleaning process ensured a more refined dataset, ready for further analysis and modeling.

```
# Compute the correlation matrix
corrmat = numerical_data.corr()

# Plot the heatmap
plt.figure(figsize=(20, 20))
sns.heatmap(corrmat, annot=True, cmap='coolwarm', center=0)
plt.show()
```

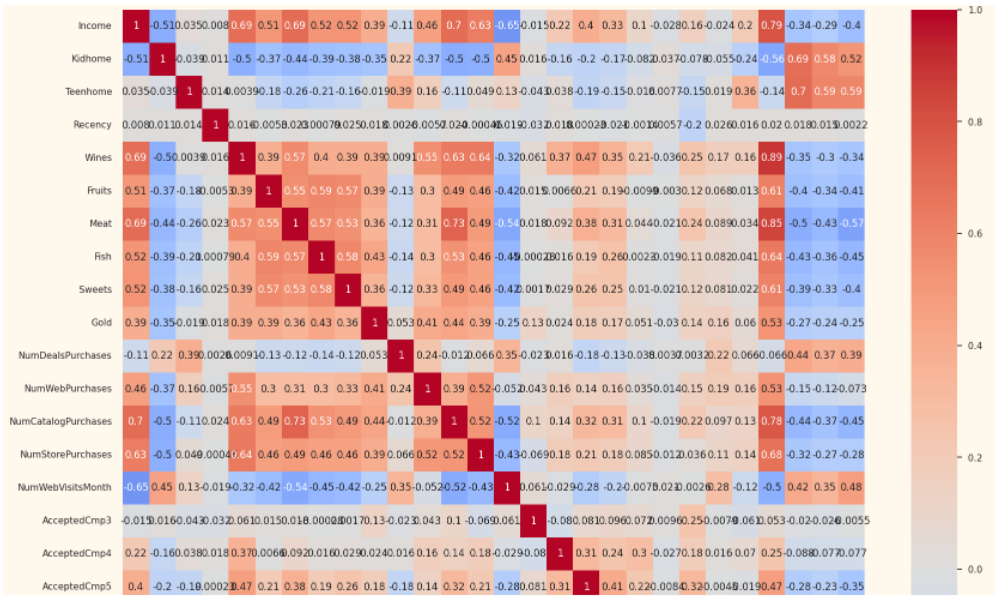


Figure 5.1: Correlation Matrix

5.1.4 DATA PREPROCESSING

In this section, several preprocessing steps are applied to the dataset to prepare it for clustering operations:

Label Encoding Categorical Features:
The categorical variables in the dataset, Education and Living_With, are identified and transformed into numerical values using Label Encoding. This step ensures that all features are

numerical, which is necessary for machine learning algorithms.

Scaling Features:
To standardize the dataset, features are scaled using the StandardScaler. Scaling transforms all features to have a mean of 0 and a standard deviation of 1, ensuring uniformity in data ranges and facilitating better performance of distance-based algorithms such as clustering.

Subset Dataframe Creation:
A subset of the original data frame is created by excluding columns related to deals accepted and promotions, specifically: AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, AcceptedCmp1, AcceptedCmp2, Complain, and Response. These features are excluded to focus on customer behavior and demographic characteristics without direct influence from promotional campaigns.

Steps taken for preprocessing:

1. The income column of the data frame has 24 missing values. The missing numbers are being dropped because there are fewer.
2. We are creating a feature that displays the length of time a customer has been in the company's database. This feature will be based on the "Dt Customer" data.
3. In order to improve the clarity and consistency of the data, I added a new column called "Living With" which has the same meaning as an existing column with a different name. The same approach was used for the "Education" data as well.
4. The "Is parent" feature has been implemented and it returns a value of 1 for parents and 0 for non-parents.
5. Add a new feature called "Spent" that displays the customer's overall spending across all categories over a two-year period.
6. Dropping some of the redundant features.
7. After plotting for “Income” and “Age”, outliers are present which will be deleted
8. Doing label encoding for categorical features i.e., ‘Education’ and ‘Living With’.
9. Dropping the columns of deals accepted and promotions, then scaling the remaining features using “Standard scaler”.

The data is quite clean, and the new features have been included.

5.1.5 DIMENSIONALITY REDUCTION

In this section, I will address the challenge of working with high-dimensional data by reducing the number of features through dimensionality reduction techniques. Having too many features can complicate model building, especially when many of the features are correlated, leading to redundancy. By reducing dimensionality, we aim to enhance the interpretability of the dataset while retaining as much information as possible.

One effective method for dimensionality reduction is **Principal Component Analysis (PCA)**. PCA transforms the original set of features into a new set of orthogonal components, ordered by the amount of variance they capture. The main advantage of this technique is that it allows us to minimize information loss while reducing the number of features.

Principal component analysis (PCA)

Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss.

The final classification in this challenge will be based on a wide range of variables. These aspects or characteristics are essentially these factors. Working with it becomes more challenging the more features it has. The correlation between several of these features makes them unnecessary. For this reason, before running the features through a classifier, I shall reduce their dimensionality.

We are reducing the dimensions to 3. After this we will be plotting these data frames.

```
[ ]  #A 3D Projection Of Data In The Reduced Dimension
     x =PCA_ds["col1"]
     y =PCA_ds["col2"]
     z =PCA_ds["col3"]
     #To plot
     fig = plt.figure(figsize=(10,8))
     ax = fig.add_subplot(111, projection="3d")
     ax.scatter(x,y,z, c="maroon", marker="o" )
     ax.set_title("A 3D Projection Of Data In The Reduced Dimension")
     plt.show()
```

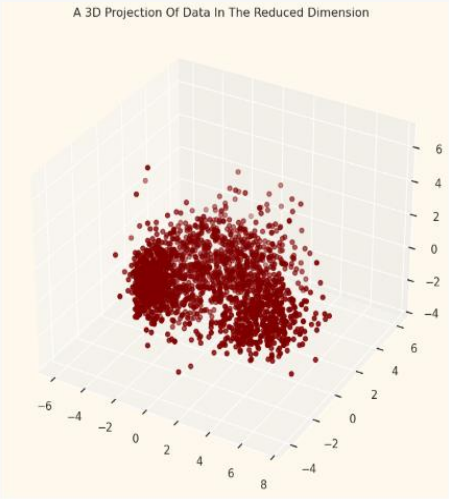


Figure 5.2: 3D plot of data after PCA

The plot above shows the data in three dimensions after it has been processed using principal component analysis (PCA). This projection allows us to visualize the data in a way that is easier to understand and interpret.

5.1.6 CLUSTERING

Hierarchical Clustering

We will be using hierarchical clustering in this project. The most typical hierarchical clustering method used to put objects in clusters based on their similarity is called agglomerative clustering.

Another name for it is **AGNES** (Agglomerative Nesting). Each object is first treated as a singleton cluster by the algorithm. Once all clusters have been merged into a single large cluster containing all items, pairs of clusters are gradually combined. The outcome is a dendrogram, which is a tree-based representation of the objects. This would be useful for identifying distinct customer personality types and for targeted marketing and communication efforts to specific groups of customers.

To perform clustering on the data, I will use a method called Agglomerative clustering on the dataset that has been reduced to three dimensions. Agglomerative clustering is a type of hierarchical clustering that involves successively merging individual examples into clusters until the desired number of clusters is reached.

The process of clustering the data includes the following steps:

1. Using the Elbow Method to determine the optimal number of clusters to form.
2. Applying Agglomerative Clustering to the data to create the clusters.
3. Visualizing the resulting clusters using a scatter plot.

Elbow Method

The Elbow Method is a technique used to determine the optimal number of clusters to form by plotting the sum of squared distances for each possible number of clusters and identifying the point of inflection, or "elbow," on the curve. This helps to identify the optimal number of clusters by choosing the number of clusters that minimizes the sum of squared distances. Once the optimal number of clusters has been determined, Agglomerative Clustering is applied to the data to create the clusters. Finally, the resulting clusters can be examined using a scatter plot to visualize the distribution of the data within each cluster.

Elbow Method Visualization

Based on the results of the Elbow Method, it appears that four clusters will be the optimal number for this dataset. Now, we will fit the Agglomerative Clustering model to the data to obtain the final clusters. This involves applying the Agglomerative Clustering algorithm to the data using the determined number of clusters. The resulting clusters will represent groups of data points that are similar to one another in some way.

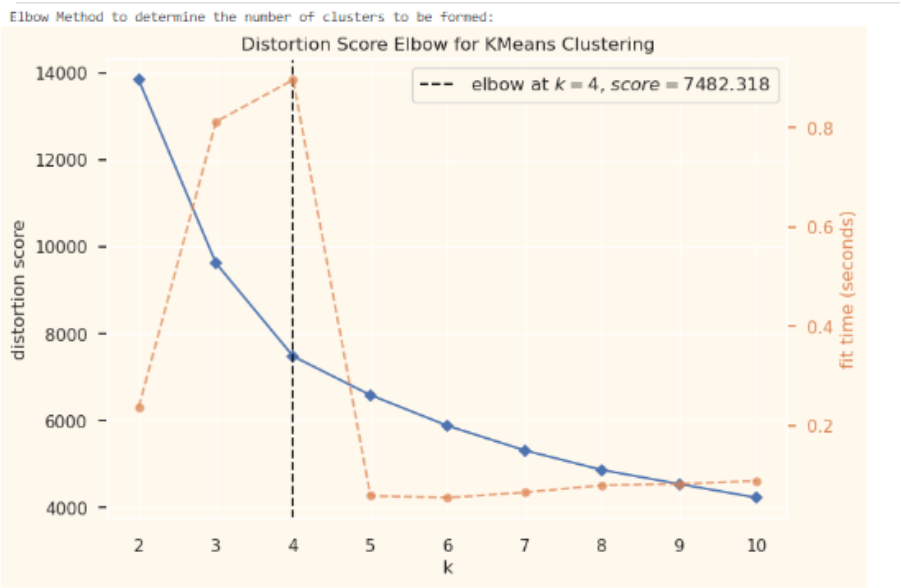


Figure 5.3: Elbow method visualization

After reducing the attributes to three dimensions, I will apply Agglomerative clustering to perform the clustering. A hierarchical clustering technique is agglomerative clustering. Up until the appropriate number of clusters is reached, examples are merged. First, we will be using the elbow method to determine the number of clusters. Once we have got the clusters then we will apply hierarchical clustering based on the PCA dataset that we had created earlier.

5.1.7 EVALUATING MODEL

Since this is an unsupervised clustering model, there is no tagged feature available to evaluate or score it directly. Therefore, the purpose of this evaluation is to analyze the patterns formed by the clusters through exploratory data analysis (EDA) and derive insights into the nature of these clusters.

To start, we assess the distribution of the clusters within the dataset.

Cluster Distribution: The distribution of the clusters shows that they are fairly balanced. A count plot was used to visualize the number of customers in each cluster, ensuring an equitable spread across all clusters.

Income vs. Spending Patterns: A scatter plot of income vs. spending, segmented by clusters, reveals distinct customer profiles:

- Cluster 0: High spending with average income.
- Cluster 1: High spending with high income.
- Cluster 2: Low spending with low income.
- Cluster 3: High spending with low income.

This indicates that each cluster represents a unique group based on income and spending behavior.

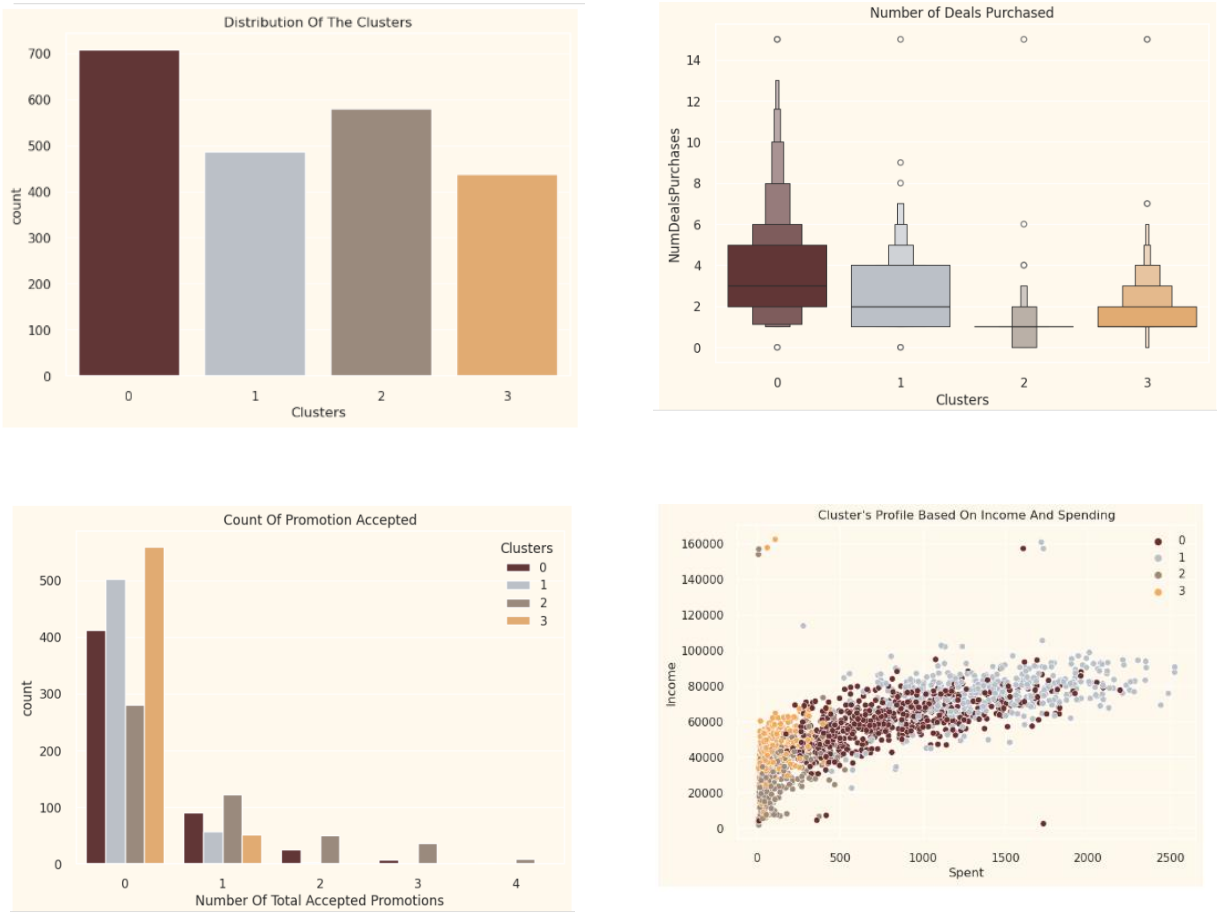


Figure 5.4: Distribution of Clusters

Product-Based Spending Distribution: A boxen plot and swarm plot of spending across different product categories (Wines, Fruits, Meat, Fish, Sweets, and Gold) demonstrate that Cluster 1 is the largest group of customers, followed by Cluster 0. This provides insights into what each cluster spends on, helping to refine marketing strategies.

Campaign Participation: The analysis of past marketing campaigns shows that the response has been generally low. A feature for the total number of accepted promotions was created to assess the performance of these campaigns. The count plot of promotion acceptance indicates that no customer participated in all five campaigns. This suggests the need for more targeted and better-planned campaigns to increase engagement and boost sales.

Deals Purchased: Unlike the campaigns, the deals offered to customers performed better, particularly for Cluster 0 and Cluster 3. However, Cluster 1, the star customers, showed little

interest in deals, while Cluster 2 had no overwhelming preferences.These insights reveal varying preferences across clusters, allowing for more focused marketing efforts based on each cluster's unique behavior and characteristics.

5.1.8 PROFILING

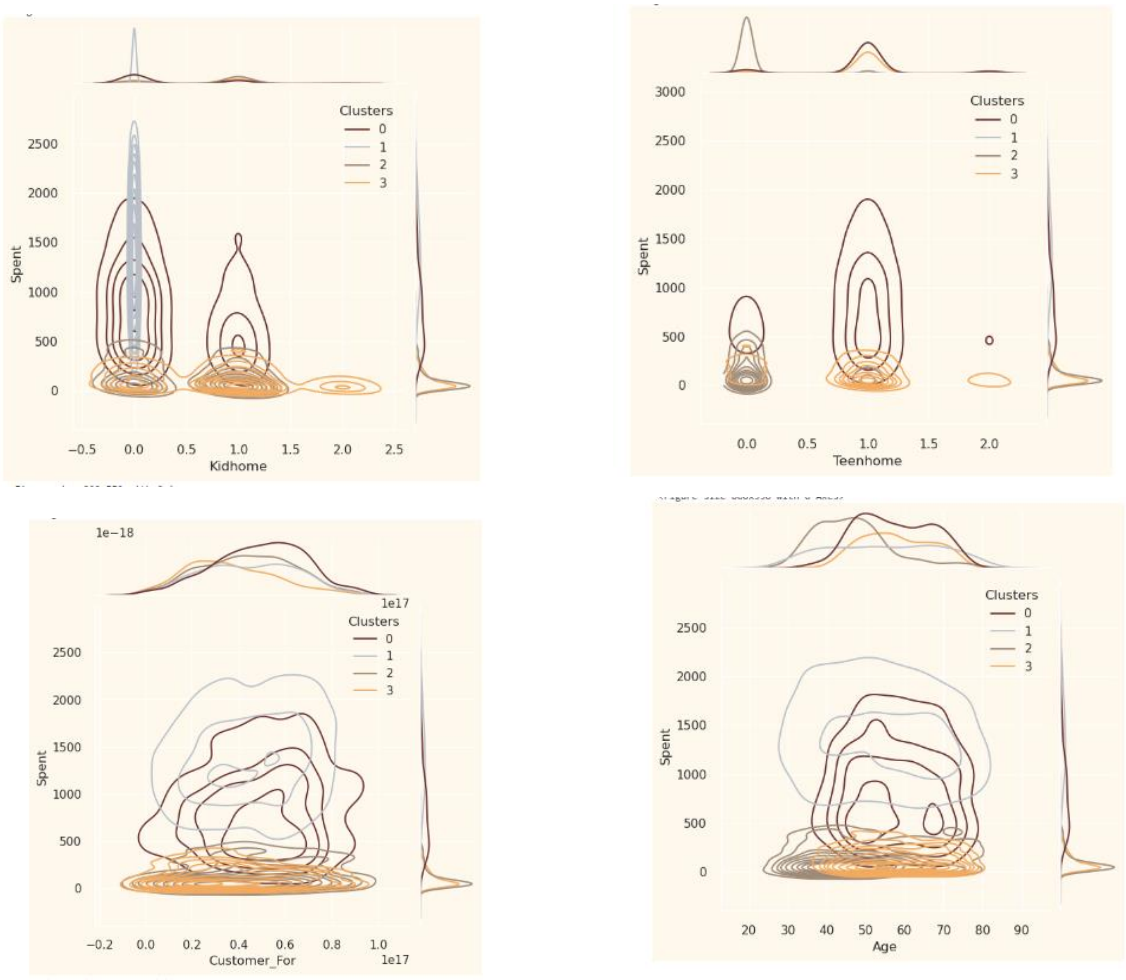


Figure 5.5: Profiling of Clusters

After clustering the customers and examining their purchasing behavior, it's essential to understand the types of customers within each cluster. Profiling allows us to identify who our key customers are and which groups may need more focus from the retail store's marketing team.

To achieve this, we examined several features that indicate personal traits of the customers and

analyzed them in the context of their respective clusters.

The personal features analyzed were: Kidhome, Teenhome, Customer_For, Age, Children, Family_Size, Is_Parent, Education, and Living_With.

For visual analysis, we plotted these personal features against the amount spent by customers across different clusters using joint KDE plots to observe the distribution and correlation.

Based on this analysis, we have identified the following characteristics of customers in each cluster:

Cluster 0:

- Mostly parents, including some single parents.
- Family size ranges from 2 to 4 members.
- Majority have teenagers at home.
- Customers in this cluster tend to be relatively older.

Cluster 1:

- Non-parents, with a maximum family size of 2 members.
- Couples form a significant portion, with fewer single individuals.
- Includes customers from all age groups.
- This is a high-income group.

Cluster 2:

- Predominantly parents with a family size of up to 3 members.
- Most have young children (not teenagers).
- This group consists of relatively younger customers.

Cluster 3:

- All are parents, with family sizes ranging from 2 to 5 members.
- Majority have teenagers at home.
- Customers in this cluster are relatively older and belong to a lower-income group.

Through this profiling, we gain insights into who our star customers are (high-income, non-parents in Cluster 1) and which groups might require more targeted marketing efforts (e.g., lower-income, larger families in Cluster 3).

Dashboard

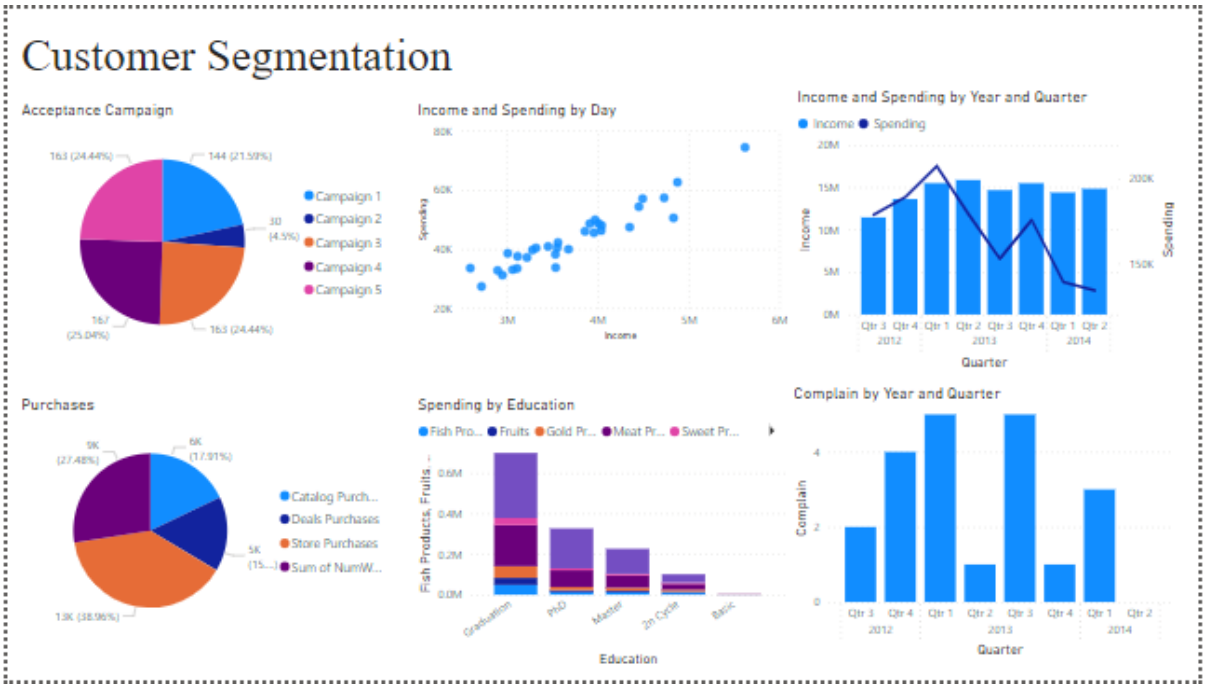


Figure 5.6: Power BI Dashboard

5.1.9 RESULTS

The customer profiling analysis provides valuable insights into the distinct characteristics of each cluster, facilitating the development of targeted marketing strategies:

Below is the Analysis of Cluster Numbers 0-3

About Cluster Number: 0

As shown in the chart, it can be concluded that the group being discussed is a group of parents who are relatively old and have a family with a maximum of four members and at least two members. Most of the parents in this group have a teenager at home, and single parents are a subset of this group. However, please note that this is only a conclusion based on the information provided, and it may not be accurate in all cases. It is important to verify and confirm any information before making conclusions or decisions based on it.

About Cluster Number: 1

According to the data in the charts, it can be concluded that the group being discussed is a group of non-parents who have a maximum of two members in their families. The majority of this group consists of couples, rather than single people. The members of this group span a wide range of ages and are part of a high-income group. However, please note that this is only a conclusion based

on the information provided, and it may not be accurate in all cases. It is important to verify and confirm any information before making conclusions or decisions based on it.

About Cluster Number: 2

The charts indicate, it can be concluded that the group being discussed is a group of parents who are relatively younger and have families with a maximum of three members. Most of these parents have one child, who is typically not a teenager. However, please note that this is only a conclusion based on the information provided, and it may not be accurate in all cases. It is important to verify and confirm any information before making conclusions or decisions based on it.

About Cluster Number: 3

The data in the charts suggests, it can be concluded that the group being discussed is a group of parents who are relatively older and have a family with a maximum of five members and at least two members. Most of these parents have a teenager at home, and they are part of a lower-income group. However, please note that this is only a conclusion based on the information provided, and it may not be accurate in all cases. It is important to verify and confirm any information before making conclusions or decisions based on it.

By leveraging these insights, the marketing team can implement tailored strategies that resonate with each customer segment, ultimately enhancing engagement, increasing customer satisfaction, and driving overall sales growth.

Insights:

- **Star Customer:** High-income, non-parents in Cluster 1
- **Focus Needed:** Lower-income families in Cluster 3

CHAPTER 6

6.1 CONCLUSION

The project on customer segmentation and personality analysis using unsupervised learning has successfully illustrated the potential of machine learning to unlock valuable insights into customer behavior, preferences, and spending patterns. By leveraging clustering techniques and dimensionality reduction, the project was able to identify meaningful customer segments and personality traits that offer a deeper understanding of diverse customer needs. These insights provide a strong foundation for developing personalized strategies, allowing businesses to optimize marketing and sales efforts effectively. Tailoring customer interactions based on behavioral trends and preferences leads to better resource allocation, enhanced customer experience, and overall business growth.

Despite the promising outcomes, challenges remain in accurately predicting customer personality traits due to data quality and availability issues. The success of customer analysis models is highly dependent on high-quality, complete datasets free from biases, as limited or incomplete data can reduce the model's predictive accuracy. As technology advances, the future of customer personality analysis will likely involve the development of increasingly sophisticated algorithms capable of uncovering hidden patterns in customer data. By incorporating these insights into virtual assistants, chatbots, and broader business operations, companies can improve customer satisfaction and loyalty, making unsupervised learning a valuable tool for driving customer-centered business success.

6.2 REFERENCES

Julien Ah-Pine (2018). *An Efficient and Effective Generic Agglomerative Hierarchical Clustering Approach*, 19(42):1–43, 2018. URL: https://www.jmlr.org/papers/volume19/18-117/18-117.pdf

Margareta Ackerman and Shai Ben-David (2016). *A characterization of linkage-based hierarchical clustering. Journal of Machine Learning Research*, 17:1–17, 2016. URL: https://www.jmlr.org/papers/volume17/11-198/11-198.pdf

D T Pham, S S Dimov, and C D Nguyen (2004). *Selection of K in K-means clustering.* URL: https://www.ee.columbia.edu/~dpwe/papers/PhamDN05-kmeans.pdf

Other References:

- Python 3: https://www.python.org/
- PCA: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- Pandas: https://pandas.pydata.org/
- Numpy: https://numpy.org/
- Seaborn: https://seaborn.pydata.org/
- Matplotlib: https://matplotlib.org/
- Elbow: https://www.scikit-yb.org/en/latest/
- Label Encoding: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html
- Standard Scaler: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- K-Means Clustering: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- Hierarchical Clustering: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html
- Data Set: https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis?datasetId=1546318&sortBy=voteCount

6.3 APPENDICES

1. Introduction

In this project, I will be performing an unsupervised clustering of data on the customer's records from a groceries firm's database. Customer segmentation is the practice of separating customers into groups that reflect similarities among customers in each cluster. I will divide customers into segments to optimize the significance of each customer to the business. To modify products according to distinct needs and behaviours of the customers. It also helps the business to cater to the concerns of different types of customers.

I got this dataset on Kaggle.

2. Import libraries

```
# Importing the Libraries
import numpy as np
import pandas as pd
import datetime
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import colors
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt, numpy as np
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import AgglomerativeClustering
from matplotlib.colors import ListedColormap
from sklearn import metrics
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
np.random.seed(42)
```

3. Load data

```
# Loading the dataset
data =
pd.read_csv("../input/customer-personality-analysis/marketing_campaign
.csv", sep="\t")
print("Number of datapoints:", len(data))
data.head()
```

```
Number of datapoints: 2240
```

```
     ID  Year_Birth   Education Marital_Status   Income  Kidhome
Teenhome  \
0  5524        1957  Graduation          Single  58138.0        0
0
1  2174        1954  Graduation          Single  46344.0        1
1
2  4141        1965  Graduation        Together  71613.0        0
0
3  6182        1984  Graduation        Together  26646.0        1
0
4  5324        1981         PhD         Married  58293.0        1
0

  Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  AcceptedCmp3
\
0  04-09-2012       58       635  ...                  7             0

1  08-03-2014       38        11  ...                  5             0

2  21-08-2013       26       426  ...                  4             0

3  10-02-2014       26        11  ...                  6             0

4  19-01-2014       94       173  ...                  5             0


   AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
0             0             0             0             0         0
1             0             0             0             0         0
2             0             0             0             0         0
3             0             0             0             0         0
4             0             0             0             0         0

   Z_CostContact  Z_Revenue  Response
0              3         11         1
1              3         11         0
2              3         11         0
3              3         11         0
4              3         11         0
```

```
[5 rows x 29 columns]
```

Attributes

People

- ID: Customer's unique identifier.
- Year_Birth: Customer's birth year.
- Education: Customer's education level.
- Marital_Status: Customer's marital status.
- Income: Customer's yearly household income.
- Kidhome: Number of children in customer's household.
- Teenhome: Number of teenagers in customer's household.
- Dt_Customer: Date of customer's enrollment with the company.
- Recency: Number of days since customer's last purchase.
- Complain: 1 if the customer complained in the last 2 years, 0 otherwise.

Products

- MntWines: Amount spent on wine in last 2 years.
- MntFruits: Amount spent on fruits in last 2 years.
- MntMeatProducts: Amount spent on meat in last 2 years.
- MntFishProducts: Amount spent on fish in last 2 years.
- MntSweetProducts: Amount spent on sweets in last 2 years.
- MntGoldProds: Amount spent on gold in last 2 years.

Promotion

- NumDealsPurchases: Number of purchases made with a discount.
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise.
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise.
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise.
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise.
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise.
- Response: 1 if customer accepted the offer in the last campaign, 0 otherwise.

Place

- NumWebPurchases: Number of purchases made through the company’s website.
- NumCatalogPurchases: Number of purchases made using a catalogue.
- NumStorePurchases: Number of purchases made directly in stores.
- NumWebVisitsMonth: Number of visits to company’s website in the last month.

4. Clean data

```
# Information on features
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   int64  
 2   Education            2240 non-null   object 
 3   Marital_Status       2240 non-null   object 
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64  
 6   Teenhome             2240 non-null   int64  
 7   Dt_Customer          2240 non-null   object 
 8   Recency              2240 non-null   int64  
 9   MntWines             2240 non-null   int64  
 10  MntFruits            2240 non-null   int64  
 11  MntMeatProducts      2240 non-null   int64  
 12  MntFishProducts      2240 non-null   int64  
 13  MntSweetProducts     2240 non-null   int64  
 14  MntGoldProds         2240 non-null   int64  
 15  NumDealsPurchases    2240 non-null   int64  
 16  NumWebPurchases      2240 non-null   int64  
 17  NumCatalogPurchases  2240 non-null   int64  
 18  NumStorePurchases    2240 non-null   int64  
 19  NumWebVisitsMonth    2240 non-null   int64  
 20  AcceptedCmp3         2240 non-null   int64  
 21  AcceptedCmp4         2240 non-null   int64  
 22  AcceptedCmp5         2240 non-null   int64  
 23  AcceptedCmp1         2240 non-null   int64  
 24  AcceptedCmp2         2240 non-null   int64  
 25  Complain             2240 non-null   int64  
 26  Z_CostContact        2240 non-null   int64  
 27  Z_Revenue            2240 non-null   int64  
 28  Response             2240 non-null   int64  
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

From the above output, we can conclude and note that:

- There are missing values in income.
- Dt_Customer that indicates the date a customer joined the database is not parsed as DateTime.
- There are some categorical features in our data frame; as there are some features in dtype: object). So we will need to encode them into numeric forms later.

First of all, for the missing values, I am simply going to drop the rows that have missing income values because it's only 1.07% missing.

```
# To remove the NA values
data = data.dropna()
print("The total number of data-points after removing the rows with
missing values are:", len(data))
```

```
The total number of data-points after removing the rows with missing
values are: 2216
```

In the next step, I am going to create a feature out of `Dt_Customer` that indicates the number of days a customer is registered in the firm's database. However, in order to keep it simple, I am taking this value relative to the most recent customer in the record.

Thus to get the values I must check the newest and oldest recorded dates.

```
data["Dt_Customer"] = pd.to_datetime(data["Dt_Customer"])
dates = []
for i in data["Dt_Customer"]:
    i = i.date()
    dates.append(i)
# Dates of the newest and oldest recorded customer
print("The newest customer's enrolment date in
therecords:",max(dates))
print("The oldest customer's enrolment date in the
records:",min(dates))
```

```
The newest customer's enrolment date in therecords: 2014-12-06
The oldest customer's enrolment date in the records: 2012-01-08
```

Creating a feature `Customer_For` of the number of days the customers started to shop in the store relative to the last recorded date.

```
# Created a feature "Customer_For"
days = []
d1 = max(dates) # taking it to be the newest customer
for i in dates:
    delta = d1 - i
    days.append(delta)
data["Customer_For"] = days
data["Customer_For"] = pd.to_numeric(data["Customer_For"],
errors="coerce")
```

Now we will be exploring the values in the categorical features to get a clear idea of the data.

```
print("Total categories in the feature Marital_Status:\n",
data["Marital_Status"].value_counts(), "\n")
```

```
print("Total categories in the feature Education:\n",
data["Education"].value_counts())
```

```
Total categories in the feature Marital_Status:
 Married     857
Together    573
Single      471
Divorced    232
Widow        76
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64

Total categories in the feature Education:
 Graduation    1116
PhD             481
Master          365
2n Cycle        200
Basic            54
Name: Education, dtype: int64
```

In the next bit, I will be performing the following steps to engineer some new features:

- Extract the `Age` of a customer by the `Year_Birth` indicating the birth year of the respective person.
- Create another feature `Spent` indicating the total amount spent by the customer in various categories over the span of two years.
- Create another feature `Living_With` out of `Marital_Status` to extract the living situation of couples.
- Create a feature `Children` to indicate total children in a household that is, kids and teenagers.
- To get further clarity of household, Creating feature indicating `Family_Size`
- Create a feature `Is_Parent` to indicate parenthood status.
- Lastly, I will create three categories in the `Education` by simplifying its value counts.
- Dropping some of the redundant features.

```
# Feature Engineering
# Age of customer today
data["Age"] = 2023 - data["Year_Birth"]

# Total spendings on various items
data["Spent"] = data["MntWines"] + data["MntFruits"] +
data["MntMeatProducts"] + data["MntFishProducts"] +
data["MntSweetProducts"] + data["MntGoldProds"]

# Deriving living situation by marital status"Alone"
data["Living_With"] =
data["Marital_Status"].replace({"Married":"Partner",
```

```
"Together":"Partner", "Absurd":"Alone", "Widow":"Alone",
"YOLO":"Alone", "Divorced":"Alone", "Single":"Alone",})

# Feature indicating total children living in the household
data["Children"] = data["Kidhome"] + data["Teenhome"]

# Feature for total members in the householde
data["Family_Size"] = data["Living_With"].replace({"Alone": 1,
"Partner":2}) + data["Children"]

# Feature pertaining parenthood
data["Is_Parent"] = np.where(data.Children > 0, 1, 0)

# Segmenting education levels in three groups
data["Education"] =
data["Education"].replace({"Basic":"Undergraduate","2n
Cycle":"Undergraduate", "Graduation":"Graduate",
"Master":"Postgraduate", "PhD":"Postgraduate"})

# For clarity
data = data.rename(columns={"MntWines":
"Wines","MntFruits":"Fruits","MntMeatProducts":"Meat","MntFishProducts
":"Fish","MntSweetProducts":"Sweets","MntGoldProds":"Gold"})

# Dropping some of the redundant features
to_drop = ["Marital_Status", "Dt_Customer", "Z_CostContact",
"Z_Revenue", "Year_Birth", "ID"]
data = data.drop(to_drop, axis=1)

data.describe()
```

```
                Income      Kidhome     Teenhome      Recency
Wines  \
count     2216.000000  2216.000000  2216.000000  2216.000000
2216.000000
mean     52247.251354     0.441787     0.505415    49.012635
305.091606
std      25173.076661     0.536896     0.544181    28.948352
337.327920
min       1730.000000     0.000000     0.000000     0.000000
0.000000
25%      35303.000000     0.000000     0.000000    24.000000
24.000000
50%      51381.500000     0.000000     0.000000    49.000000
174.500000
75%      68522.000000     1.000000     1.000000    74.000000
505.000000
max     666666.000000     2.000000     2.000000    99.000000
1493.000000
```

```
             Fruits         Meat         Fish       Sweets          Gold
...  \
count  2216.000000  2216.000000  2216.000000  2216.000000  2216.000000
...
mean     26.356047   166.995939    37.637635    27.028881    43.965253
...
std      39.793917   224.283273    54.752082    41.072046    51.815414
...
min       0.000000     0.000000     0.000000     0.000000     0.000000
...
25%       2.000000    16.000000     3.000000     1.000000     9.000000
...
50%       8.000000    68.000000    12.000000     8.000000    24.500000
...
75%      33.000000   232.250000    50.000000    33.000000    56.000000
...
max     199.000000  1725.000000   259.000000   262.000000   321.000000
...

       AcceptedCmp1  AcceptedCmp2     Complain     Response
Customer_For  \
count   2216.000000   2216.000000  2216.000000  2216.000000
2.216000e+03
mean       0.064079      0.013538     0.009477     0.150271
4.423735e+16
std        0.244950      0.115588     0.096907     0.357417
2.008532e+16
min        0.000000      0.000000     0.000000     0.000000
0.000000e+00
25%        0.000000      0.000000     0.000000     0.000000
2.937600e+16
50%        0.000000      0.000000     0.000000     0.000000
4.432320e+16
75%        0.000000      0.000000     0.000000     0.000000
5.927040e+16
max        1.000000      1.000000     1.000000     1.000000
9.184320e+16

               Age        Spent     Children  Family_Size    Is_Parent

count  2216.000000  2216.000000  2216.000000  2216.000000  2216.000000

mean     54.179603   607.075361     0.947202     2.592509     0.714350

std      11.985554   602.900476     0.749062     0.905722     0.451825

min      27.000000     5.000000     0.000000     1.000000     0.000000

25%      46.000000    69.000000     0.000000     2.000000     0.000000
```

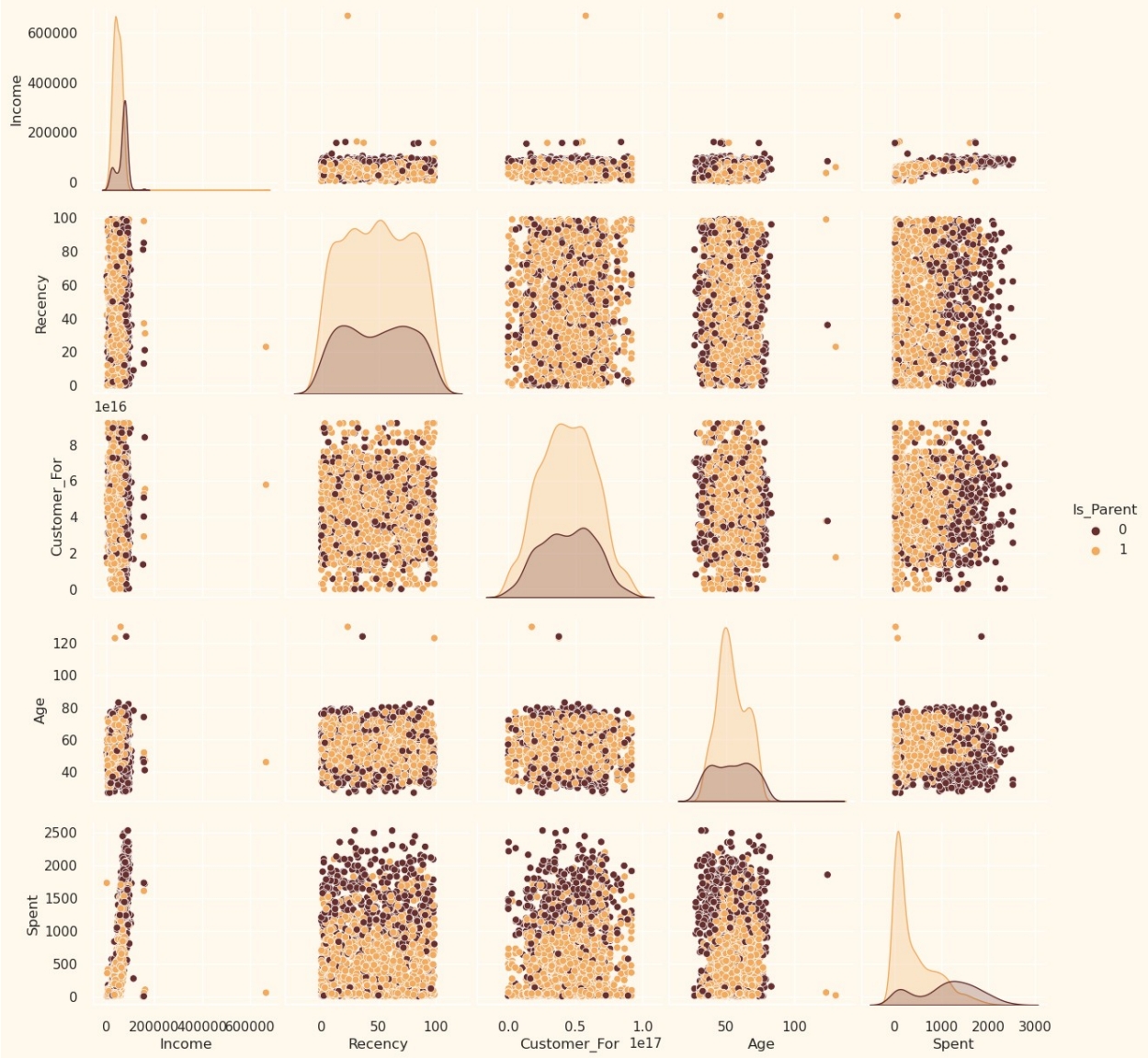
```
50%      53.000000   396.500000     1.000000     3.000000     1.000000
75%      64.000000  1048.000000     1.000000     3.000000     1.000000
max     130.000000  2525.000000     3.000000     5.000000     1.000000

[8 rows x 28 columns]
```

Do note that max-age is 130 years, As I calculated the age that would be today (i.e. 2023) and the data is old.

I must take a look at the broader view of the data. I will plot some of the selected features.

```
# To plot some selected features
# Setting up colors prefrences
sns.set(rc={"axes.facecolor":"#FFF9ED","figure.facecolor":"#FFF9ED"})
pallet = ["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78",
"#F3AB60"]
cmap = colors.ListedColormap(["#682F2F", "#9E726F", "#D6B2B1",
"#B9C0C9", "#9F8A78", "#F3AB60"])

# Plotting following features
To_Plot = [ "Income", "Recency", "Customer_For", "Age", "Spent",
"Is_Parent"]
print("Reletive Plot Of Some Selected Features: A Data Subset")
plt.figure()
sns.pairplot(data[To_Plot], hue= "Is_Parent",palette=
(["#682F2F","#F3AB60"]))

plt.show()
```

```
Reletive Plot Of Some Selected Features: A Data Subset

<Figure size 800x550 with 0 Axes>
```



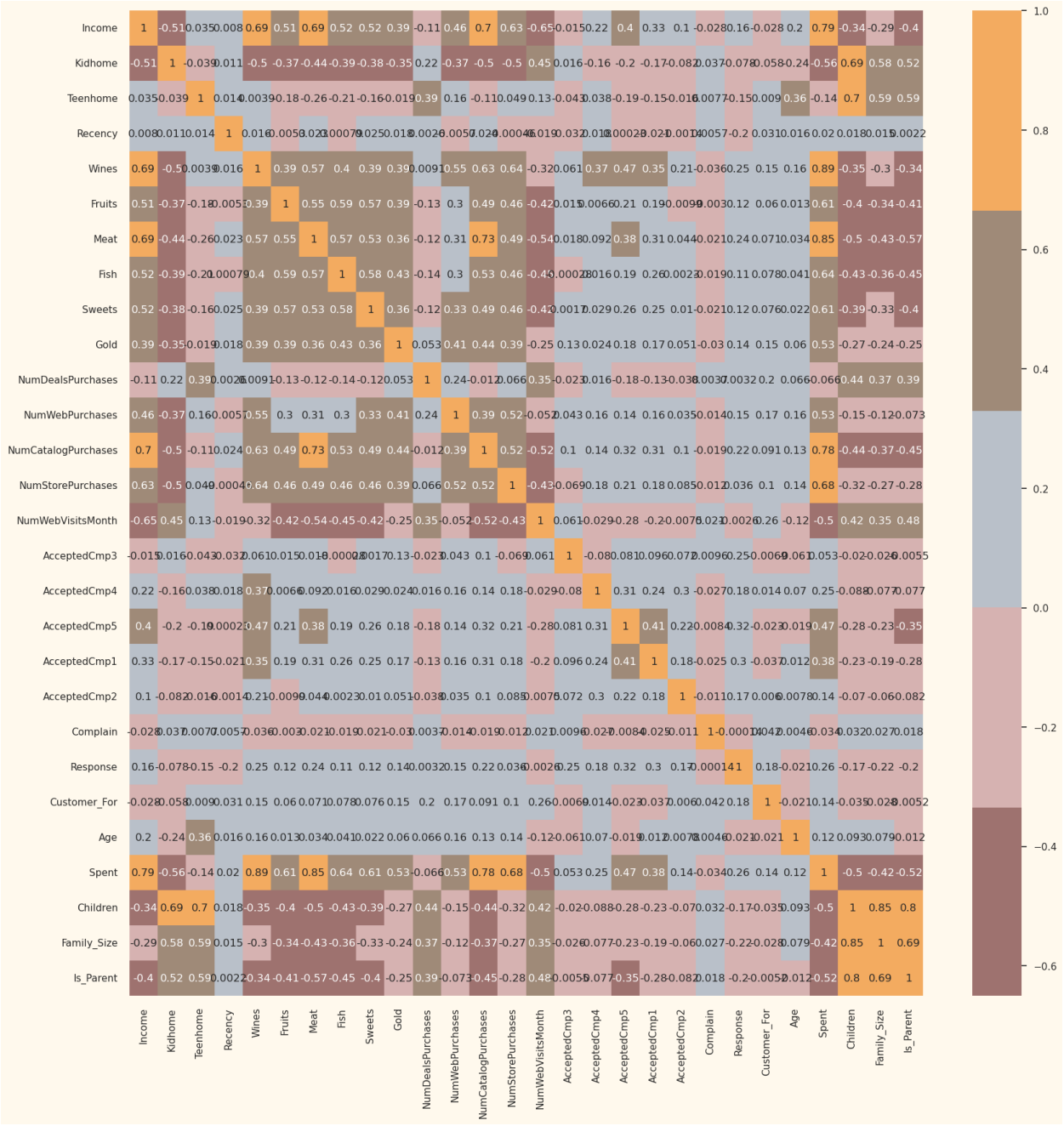
Clearly, there are a few outliers in the `Income` and `Age` features. I will be deleting the outliers in the data.

```
# Dropping the outliers by setting a cap on Age and income.
data = data[(data["Age"] < 90)]
data = data[(data["Income"] < 600000)]
print("The total number of data-points after removing the outliers are:", len(data))
```

```
The total number of data-points after removing the outliers are: 2212
```

Next, let us look at the correlation amongst the features. (Excluding the categorical attributes at this point)

```
# Correlation matrix
corrmat = data.corr()
plt.figure(figsize=(20,20))
sns.heatmap(corrmat,annot=True, cmap=cmap, center=0)
plt.show()
```



5. Preprocessing data

In this section, I will be preprocessing the data to perform clustering operations.

The following steps are applied to preprocess the data:

- Label encoding the categorical features.
- Scaling the features using the standard scaler.
- Creating a subset dataframe for dimensionality reduction.

```
# Get list of categorical variables
s = (data.dtypes == 'object')
object_cols = list(s[s].index)

print("Categorical variables in the dataset:", object_cols)
```

```
Categorical variables in the dataset: ['Education', 'Living_With']
```

```
# Label Encoding the object dtypes.
LE = LabelEncoder()
for i in object_cols:
    data[i] = data[[i]].apply(LE.fit_transform)

print("All features are now numerical")
```

```
All features are now numerical
```

```
# Creating a copy of data
ds = data.copy()
# Creating a subset of dataframe by dropping the features on deals
accepted and promotions
cols_del = ['AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5',
'AcceptedCmp1','AcceptedCmp2', 'Complain', 'Response']
ds = ds.drop(cols_del, axis=1)

# Scaling
scaler = StandardScaler()
scaler.fit(ds)
scaled_ds = pd.DataFrame(scaler.transform(ds), columns=ds.columns)
print("All features are now scaled")
```

```
All features are now scaled
```

```
# Scaled data to be used for reducing the dimensionality
print("Dataframe to be used for further modelling:")
scaled_ds.head()
```

```
Dataframe to be used for further modelling:
```

```
   Education    Income   Kidhome  Teenhome   Recency     Wines
Fruits  \
```

```
0  -0.893586  0.287105 -0.822754 -0.929699  0.310353  0.977660
1.552041
1  -0.893586 -0.260882  1.040021  0.908097 -0.380813 -0.872618 -
0.637461
2  -0.893586  0.913196 -0.822754 -0.929699 -0.795514  0.357935
0.570540
3  -0.893586 -1.176114  1.040021 -0.929699 -0.795514 -0.872618 -
0.561961
4   0.571657  0.294307  1.040021 -0.929699  1.554453 -0.392257
0.419540

       Meat      Fish    Sweets  ...  NumCatalogPurchases
NumStorePurchases  \
0  1.690293  2.453472  1.483713  ...             2.503607          -
0.555814
1 -0.718230 -0.651004 -0.634019  ...            -0.571340          -
1.171160
2 -0.178542  1.339513 -0.147184  ...            -0.229679
1.290224
3 -0.655787 -0.504911 -0.585335  ...            -0.913000          -
0.555814
4 -0.218684  0.152508 -0.001133  ...             0.111982
0.059532

   NumWebVisitsMonth  Customer_For       Age     Spent  Living_With
Children  \
0           0.692181      1.973583  1.018352  1.676245    -1.349603 -
1.264598
1          -0.132545     -1.665144  1.274785 -0.963297    -1.349603
1.404572
2          -0.544908     -0.172664  0.334530  0.280110     0.740959 -
1.264598
3           0.279818     -1.923210 -1.289547 -0.920135     0.740959
0.069987
4          -0.132545     -0.822130 -1.033114 -0.307562     0.740959
0.069987

   Family_Size  Is_Parent
0    -1.758359  -1.581139
1     0.449070   0.632456
2    -0.654644  -1.581139
3     0.449070   0.632456
4     0.449070   0.632456

[5 rows x 23 columns]
```

6. Reduce dimentions

In this problem, there are many factors on the basis of which the final classification will be done. These factors are basically attributes or features. The higher the number of features, the harder it is to work with it. Many of these features are correlated, and hence redundant. This is why I will be performing dimensionality reduction on the selected features before putting them through a classifier.

Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables.

Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss.

Steps in this section:

- Dimensionality reduction with PCA.
- Plotting the reduced dataframe.

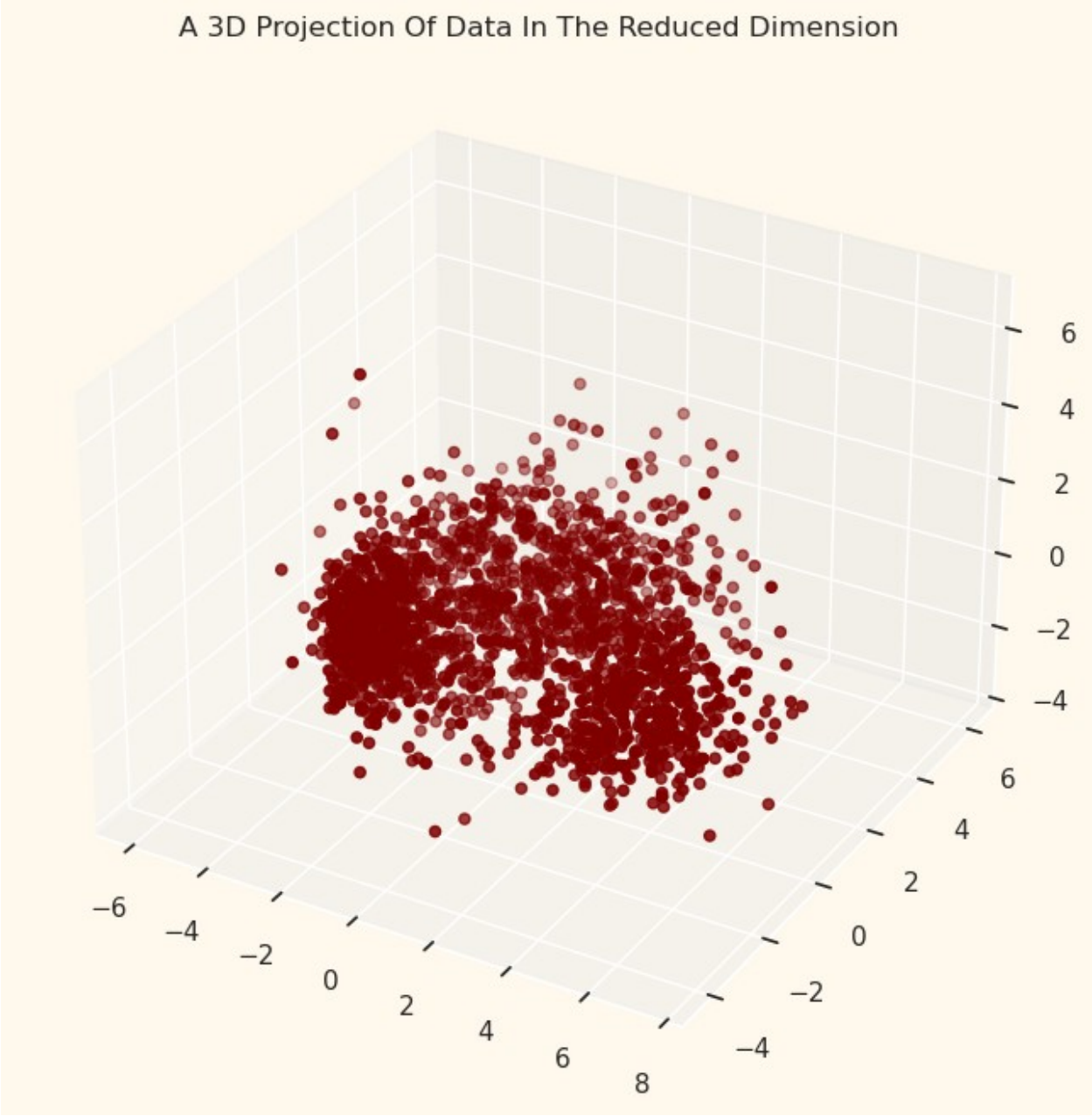
Dimensionality reduction with PCA

For this project, I will be reducing the dimensions to 3.

```
# Initiating PCA to reduce dimentions aka features to 3
pca = PCA(n_components=3)
pca.fit(scaled_ds)
PCA_ds = pd.DataFrame(pca.transform(scaled_ds),
columns=(["col1","col2", "col3"]))
PCA_ds.describe()
```

	col1	col2	col3
count	2.212000e+03	2.212000e+03	2.212000e+03
mean	-5.139550e-17	4.497106e-17	4.978939e-17
std	2.878377e+00	1.706839e+00	1.221956e+00
min	-5.969394e+00	-4.312196e+00	-3.530416e+00
25%	-2.538494e+00	-1.328316e+00	-8.290674e-01
50%	-7.804209e-01	-1.581233e-01	-2.269238e-02
75%	2.383290e+00	1.242289e+00	7.998952e-01
max	7.444305e+00	6.142721e+00	6.611222e+00

```
# A 3D Projection Of Data In The Reduced Dimension
x = PCA_ds["col1"]
y = PCA_ds["col2"]
z = PCA_ds["col3"]

# To plot
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection="3d")
ax.scatter(x,y,z, c="maroon", marker="o" )
ax.set_title("A 3D Projection Of Data In The Reduced Dimension")
plt.show()
```



7. Clustering

Now that I have reduced the attributes to three dimensions, I will be performing clustering via Agglomerative clustering. Agglomerative clustering is a hierarchical clustering method. It involves merging examples until the desired number of clusters is achieved.

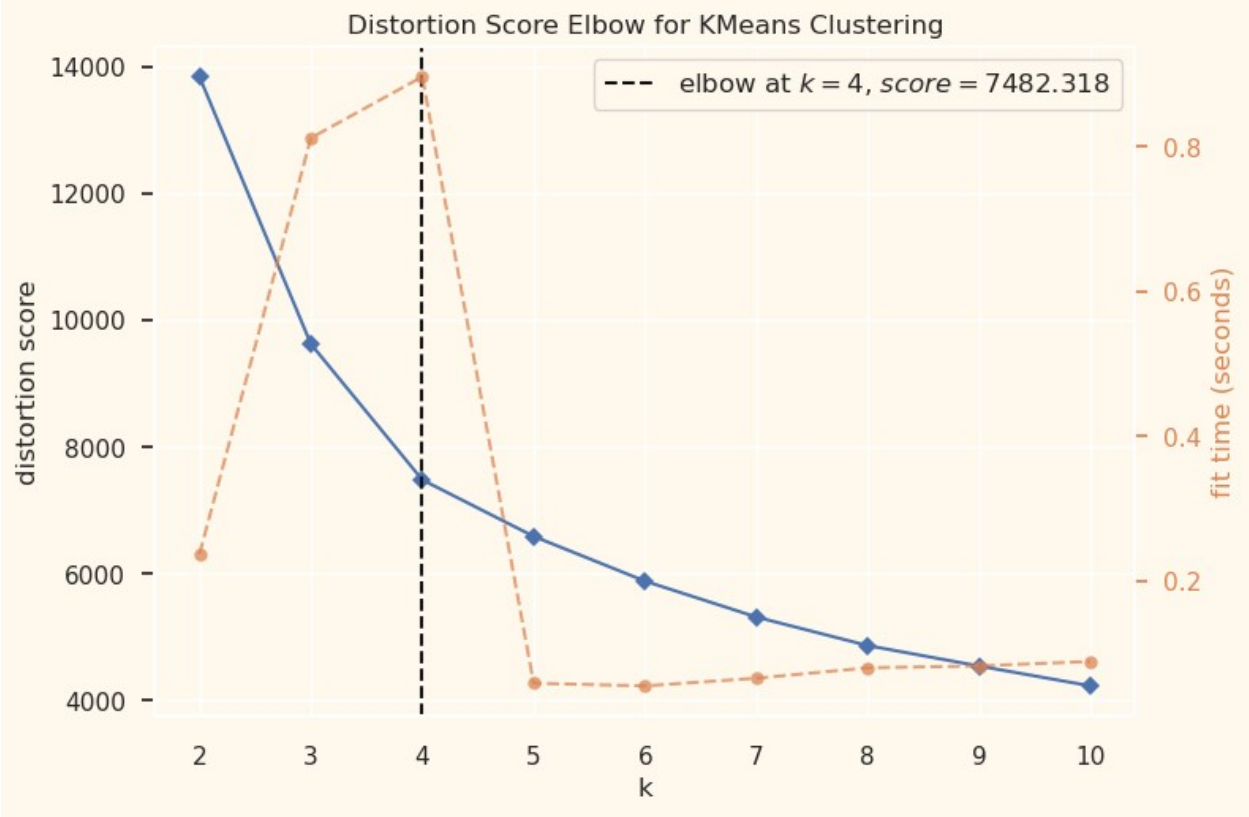
Steps involved in the Clustering

- Elbow Method to determine the number of clusters to be formed.
- Clustering via Agglomerative Clustering.

- Examining the clusters formed via scatter plot.

```
# Quick examination of elbow method to find numbers of clusters to
make.
print('Elbow Method to determine the number of clusters to be
formed:')
Elbow_M = KElbowVisualizer(KMeans(), k=10)
Elbow_M.fit(PCA_ds)
Elbow_M.show()
```

```
Elbow Method to determine the number of clusters to be formed:
```



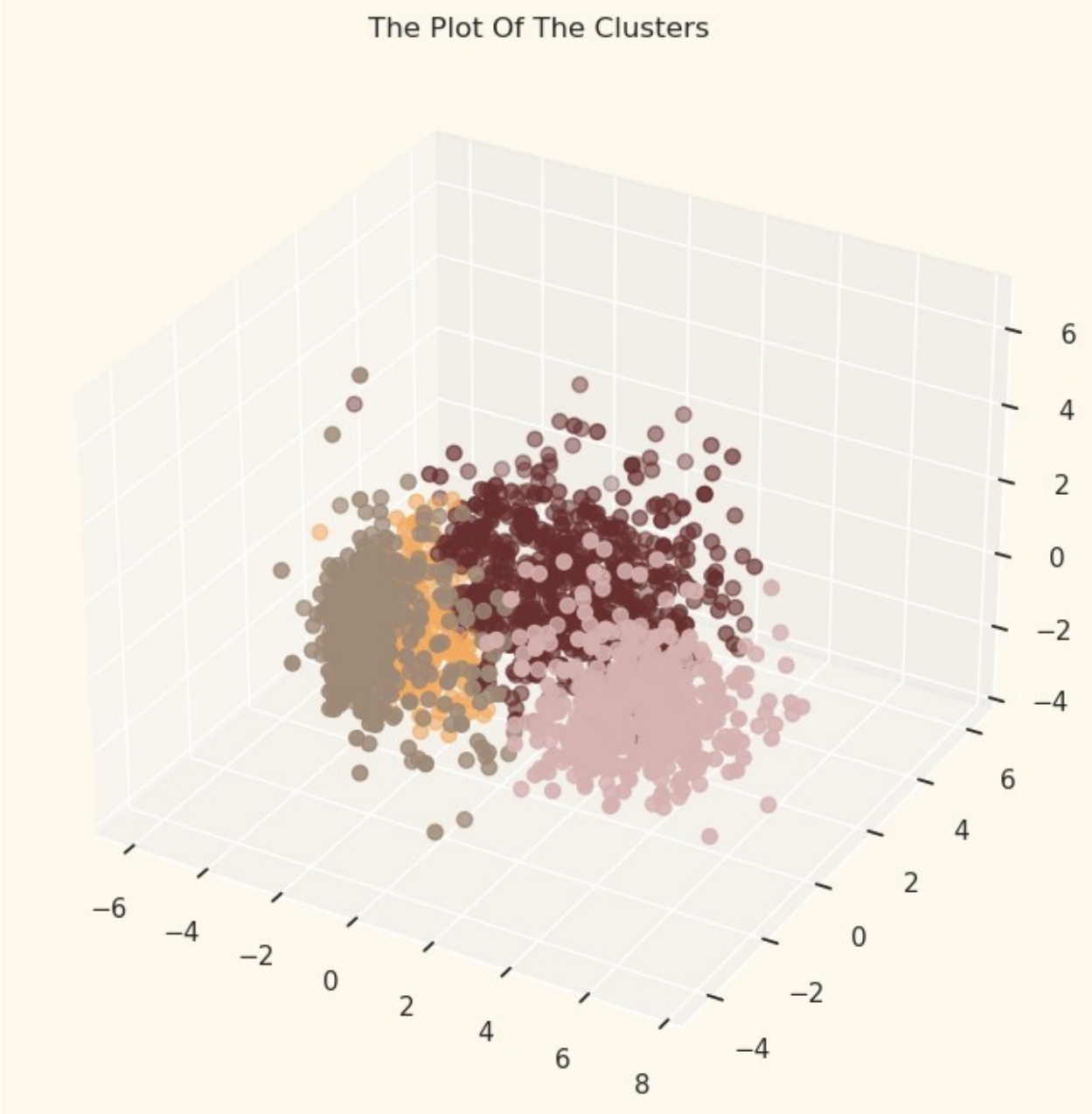

```
<Axes: title={'center': 'Distortion Score Elbow for KMeans
Clustering'}, xlabel='k', ylabel='distortion score'>
```

The above cell indicates that **four** will be an optimal number of clusters for this data. Next, we will be fitting the Agglomerative Clustering Model to get the final clusters.

```
# Initiating the Agglomerative Clustering model
AC = AgglomerativeClustering(n_clusters=4)
# Fit model and predict clusters
yhat_AC = AC.fit_predict(PCA_ds)
PCA_ds["Clusters"] = yhat_AC
# Adding the Clusters feature to the orignal dataframe.
data["Clusters"] = yhat_AC
```

To examine the clusters formed let's have a look at the 3-D distribution of the clusters.

```
# Plotting the clusters
fig = plt.figure(figsize=(10,8))
ax = plt.subplot(111, projection='3d', label="bla")
ax.scatter(x, y, z, s=40, c=PCA_ds["Clusters"], marker='o', cmap =
cmap )
ax.set_title("The Plot Of The Clusters")
plt.show()
```



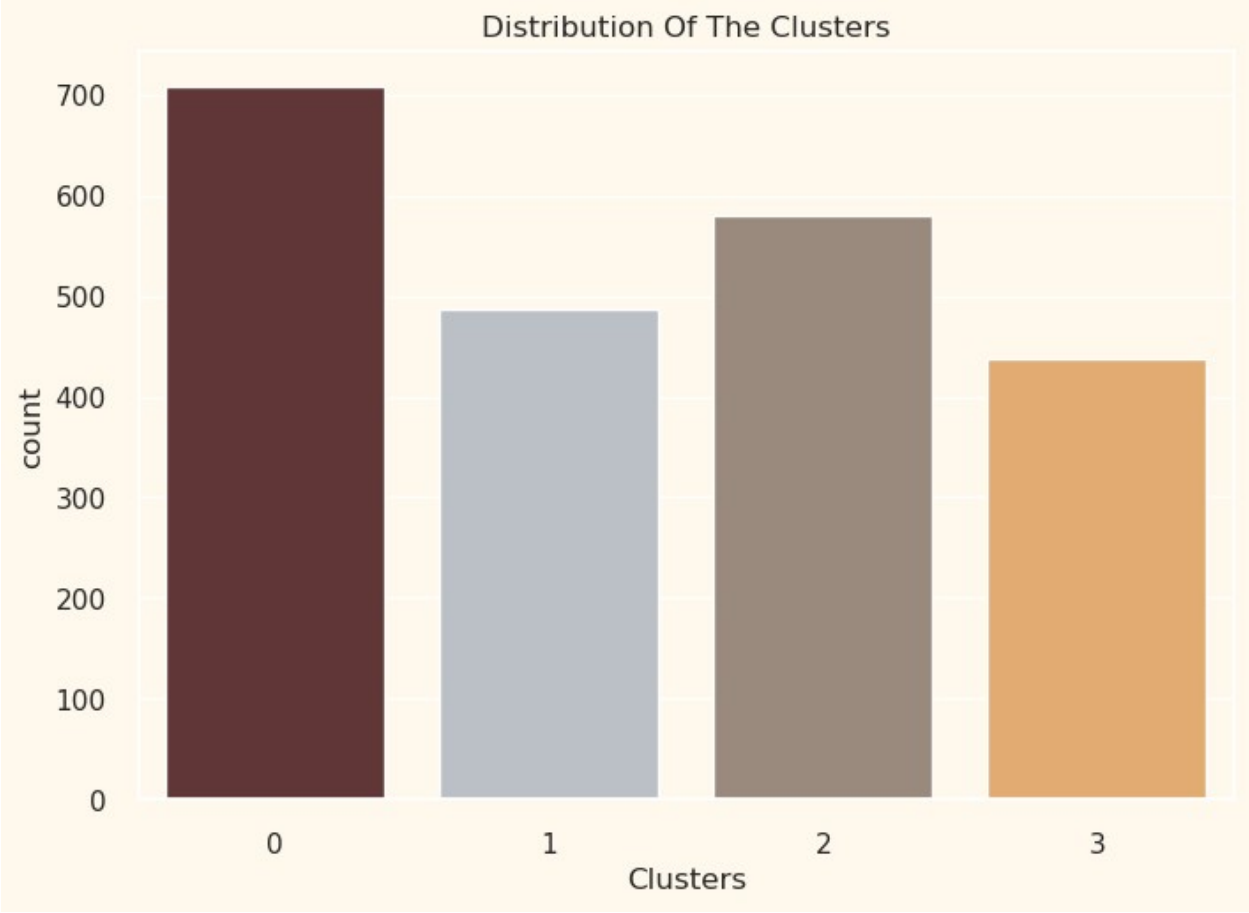
8. Evaluate models

Since this is an unsupervised clustering. We do not have a tagged feature to evaluate or score our model. The purpose of this section is to study the patterns in the clusters formed and determine the nature of the clusters' patterns.

For that, we will be having a look at the data in light of clusters via exploratory data analysis and drawing conclusions.

Firstly, let us have a look at the group distribution of clustring.

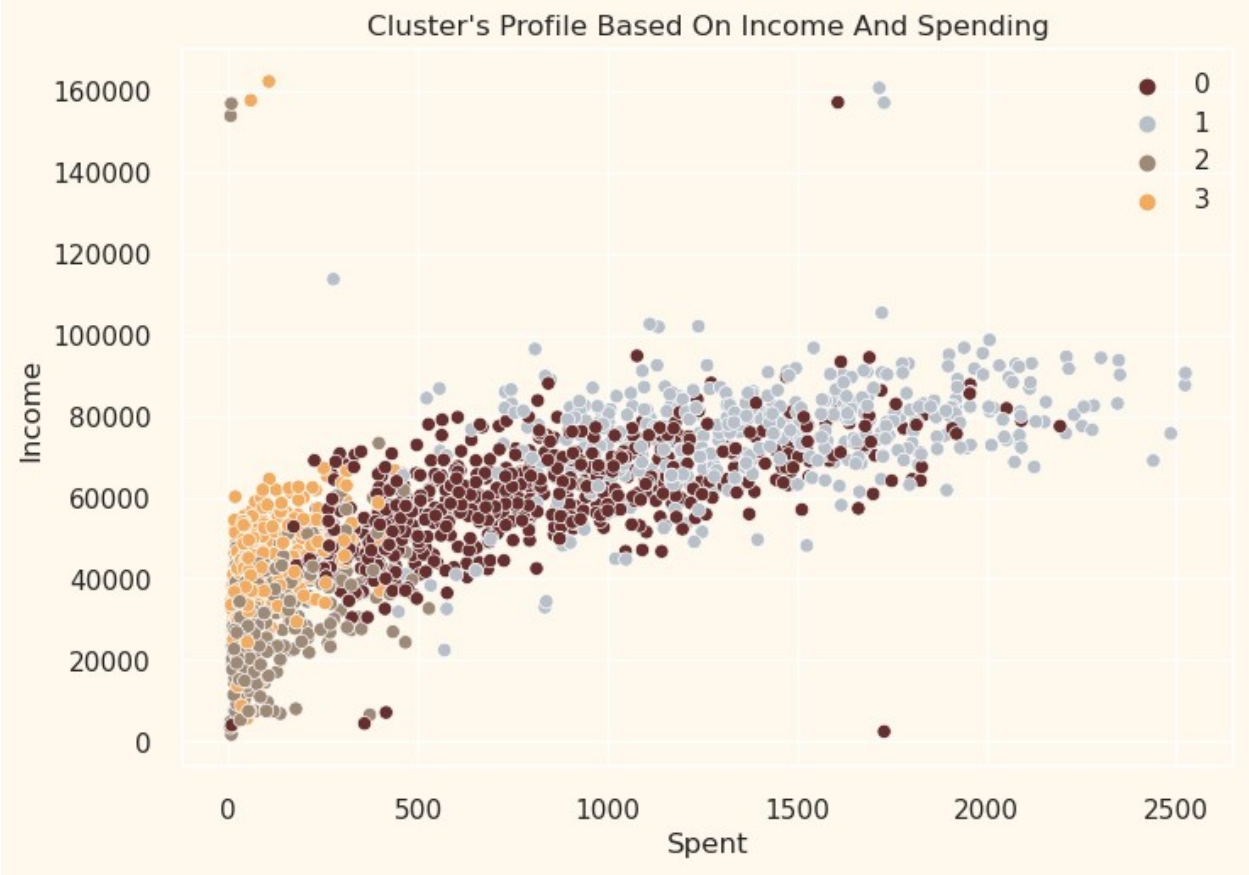
```
# Plotting countplot of clusters
pal = ["#682F2F","#B9C0C9", "#9F8A78","#F3AB60"]
pl = sns.countplot(x=data["Clusters"], palette= pal)
pl.set_title("Distribution Of The Clusters")
plt.show()
```



The clusters seem to be fairly distributed.

```
pl = sns.scatterplot(data=data, x=data["Spent"], y=data["Income"],
hue=data["Clusters"], palette=pal)
```

```
pl.set_title("Cluster's Profile Based On Income And Spending")
plt.legend()
plt.show()
```

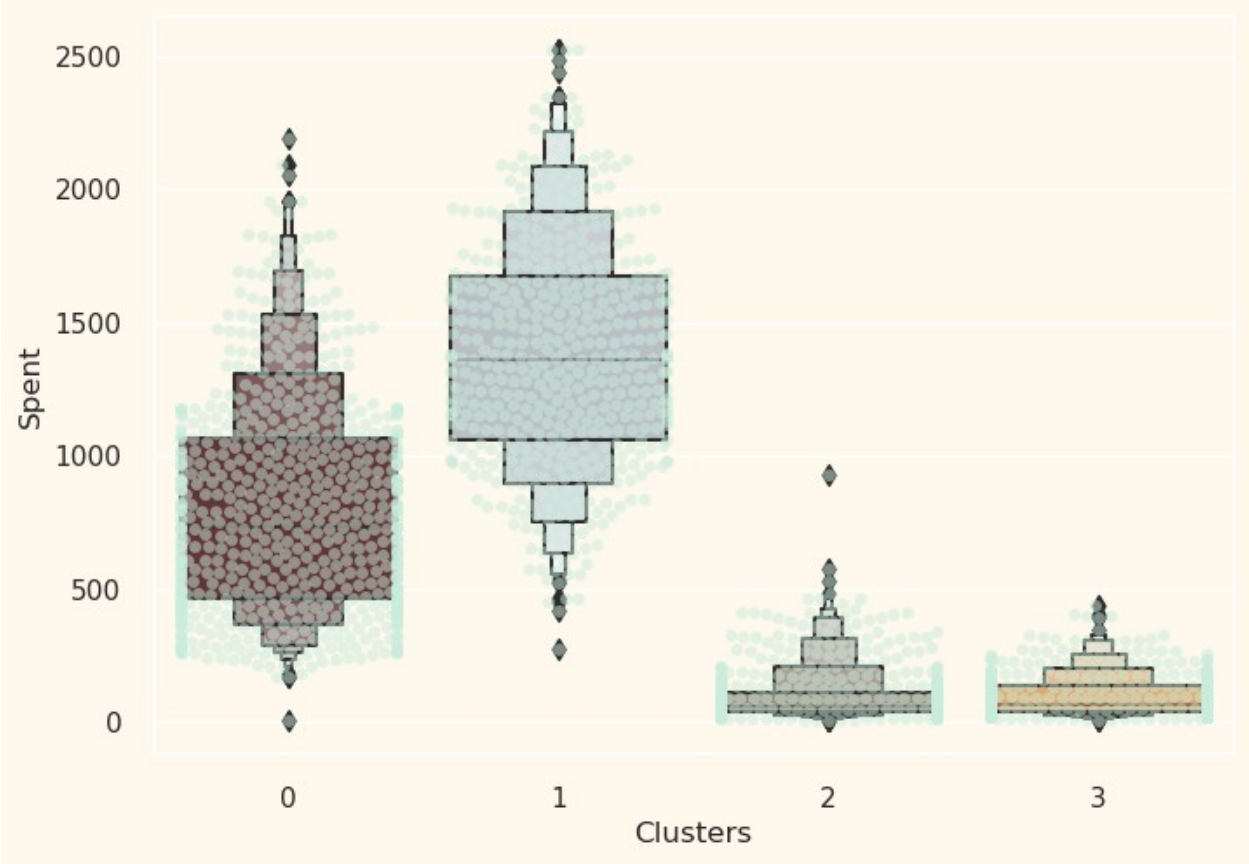


Income vs spending plot shows the clusters pattern

- Group 0: high spending & average income
- Group 1: high spending & high income
- Group 2: low spending & low income
- Group 3: high spending & low income

Next, I will be looking at the detailed distribution of clusters as per the various products in the data. Namely: Wines, Fruits, Meat, Fish, Sweets and Gold.

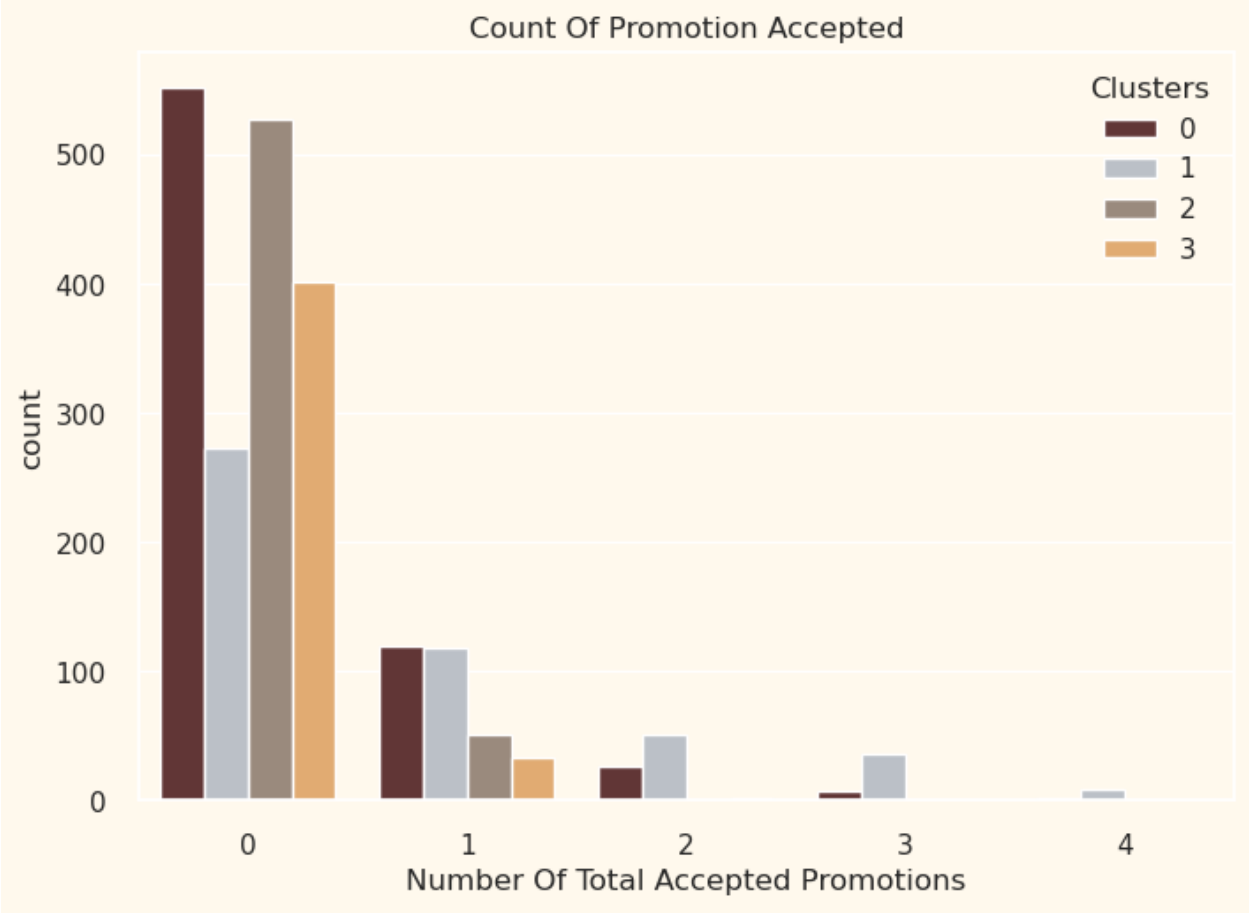
```
plt.figure()
pl=sns.swarmplot(x=data["Clusters"], y=data["Spent"], color="#CBEDDD",
alpha=0.5 )
pl=sns.boxenplot(x=data["Clusters"], y=data["Spent"], palette=pal)
plt.show()
```



From the above plot, it can be clearly seen that cluster 1 is our biggest set of customers closely followed by cluster 0. We can explore what each cluster is spending on for the targeted marketing strategies.

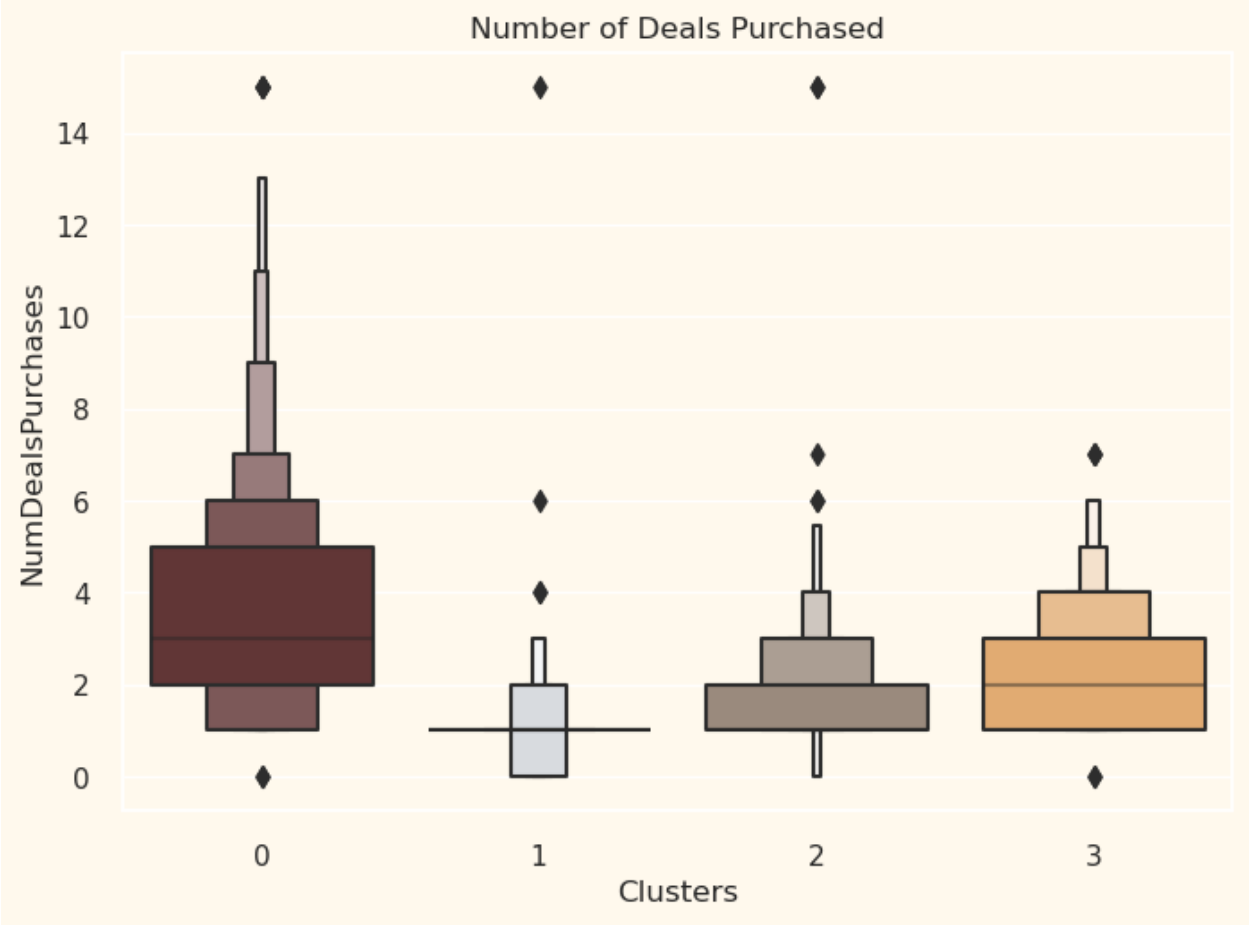
Let us next explore how did our campaigns do in the past.

```
# Creating a feature to get a sum of accepted promotions
data["Total_Promos"] = data["AcceptedCmp1"] + data["AcceptedCmp2"] +
data["AcceptedCmp3"] + data["AcceptedCmp4"] + data["AcceptedCmp5"]
# Plotting count of total campaign accepted.
plt.figure()
pl = sns.countplot(x=data["Total_Promos"], hue=data["Clusters"],
palette=pal)
pl.set_title("Count Of Promotion Accepted")
pl.set_xlabel("Number Of Total Accepted Promotions")
plt.show()
```



There has not been an overwhelming response to the campaigns so far. Very few participants overall. Moreover, no one part take in all 5 of them. Perhaps better-targeted and well-planned campaigns are required to boost sales.

```
# Plotting the number of deals purchased
plt.figure()
pl=sns.boxenplot(y=data["NumDealsPurchases"],x=data["Clusters"],
palette= pal)
pl.set_title("Number of Deals Purchased")
plt.show()
```



Unlike campaigns, the deals offered did well. It has best outcome with cluster 0 and cluster 3. However, our star customers cluster 1 are not much into the deals. Nothing seems to attract cluster 2 overwhelmingly

9. Profiling

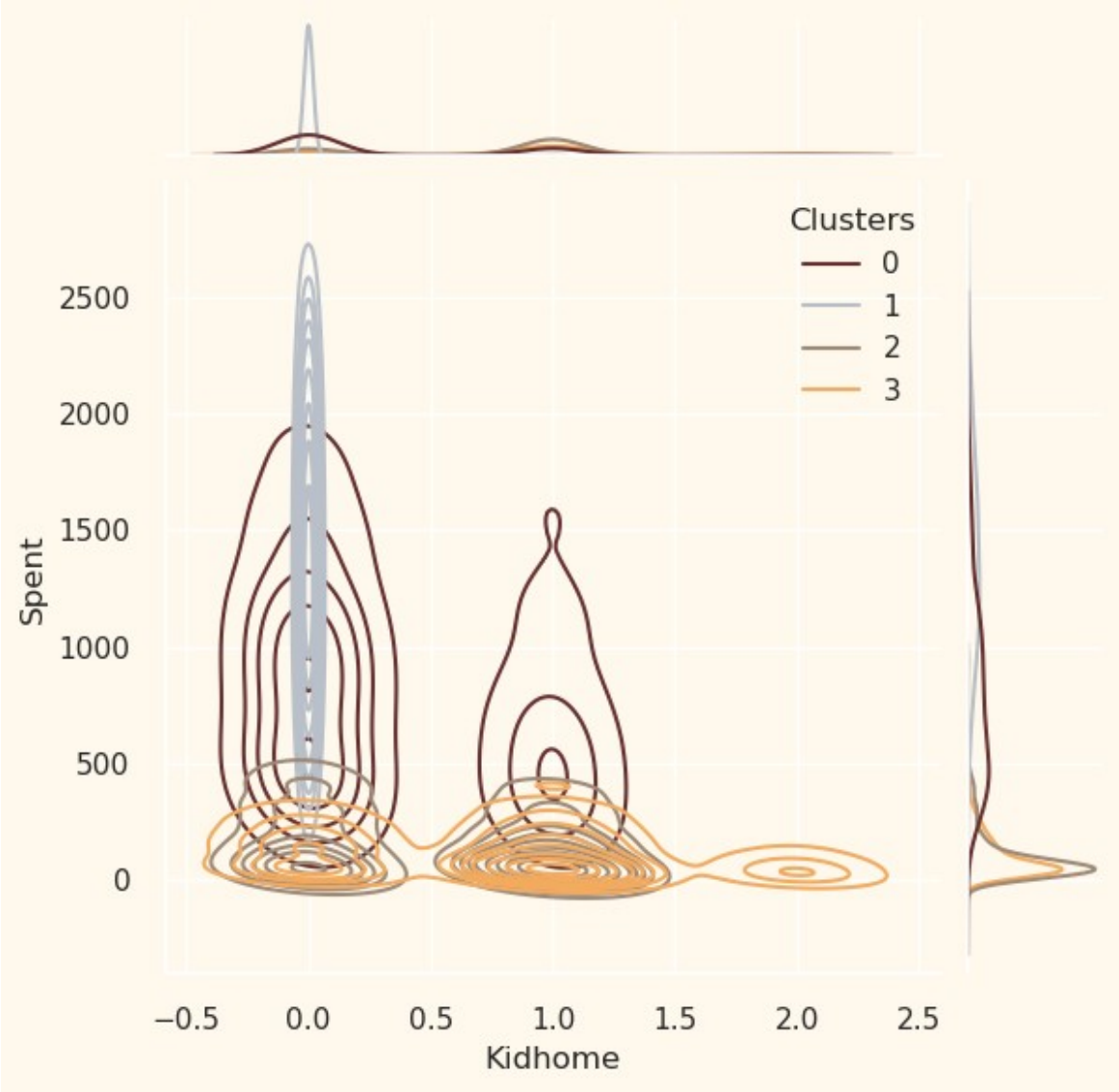
Now that we have formed the clusters and looked at their purchasing habits. Let us see who all are there in these clusters. For that, we will be profiling the clusters formed and come to a conclusion about who is our star customer and who needs more attention from the retail store's marketing team.

To decide that I will be plotting some of the features that are indicative of the customer's personal traits in light of the cluster they are in. On the basis of the outcomes, I will be arriving at the conclusions.

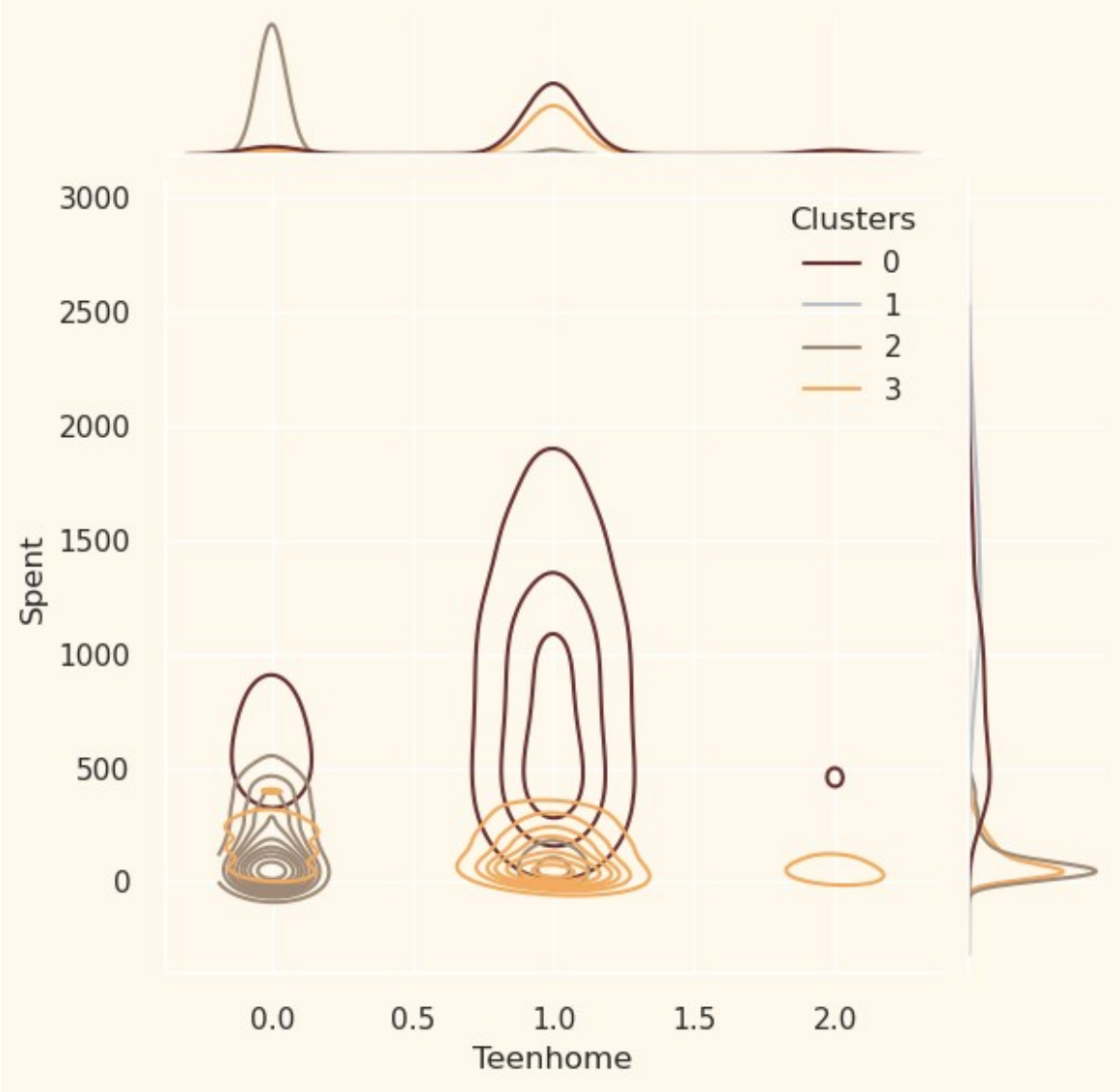
```
Personal = ["Kidhome","Teenhome","Customer_For", "Age", "Children",
"Family_Size", "Is_Parent", "Education","Living_With"]

for i in Personal:
    plt.figure()
```

```
    sns.jointplot(x=data[i], y=data["Spent"], hue=data["Clusters"],
kind="kde", palette=pal)
    plt.show()
```

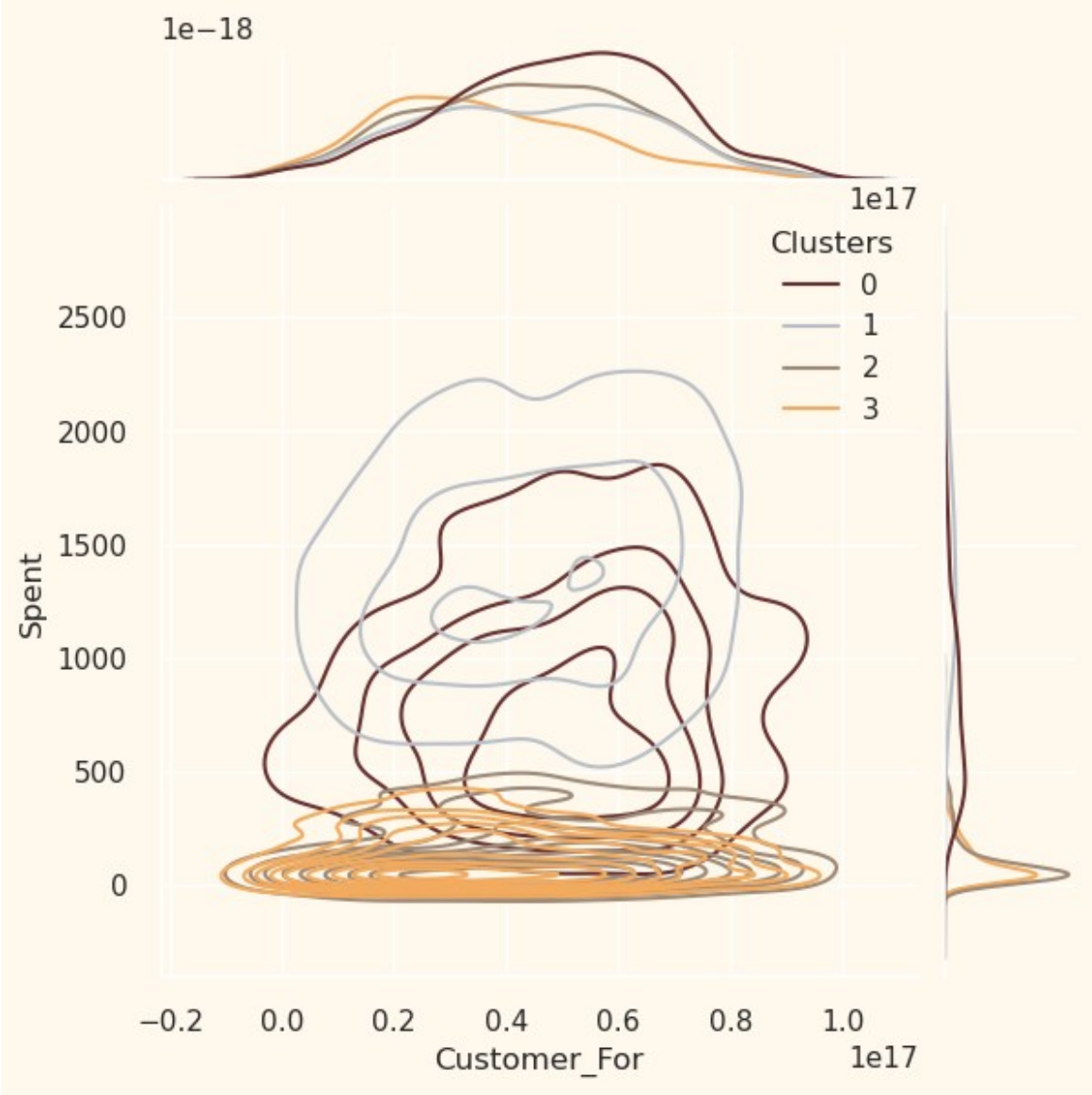
```
<Figure size 800x550 with 0 Axes>
```



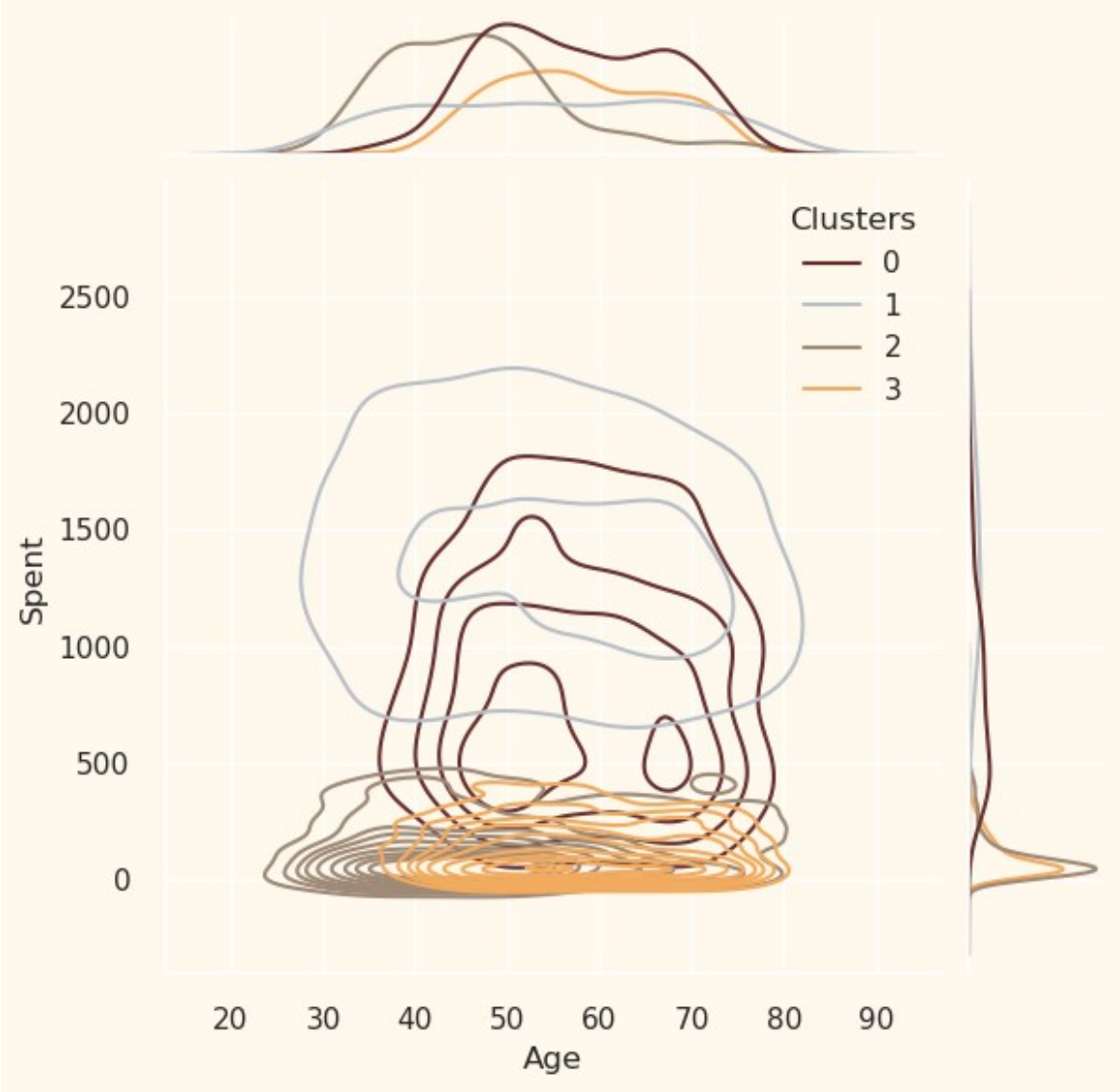

```
<Figure size 800x550 with 0 Axes>
```




```
<Figure size 800x550 with 0 Axes>
```




```
<Figure size 800x550 with 0 Axes>
```



Points to be noted

The following information can be deduced about the customers in different clusters.

Cluster 0:

- Are a definitely a parent.
- At the max have 4 members in the family and at least 2.
- Single parents are a subset of this group.
- Most have a teenager at home.
- Relatively older.

Cluster 1:

- Are definitely not a parent.
- At the max are only 2 members in the family.
- A single majority of couples over single people.

- Span all ages.
- A high income group.

Cluster 2:

- The majority of these people are parents.
- At the max are 3 members in the family.
- They majorly have one kid (and not teenagers, typically).
- Relatively younger.

Cluster 3:

- They are definitely a parent.
- At the max are 5 members in the family and at least 2.
- Majority of them have a teenager at home.
- Relatively older.
- A lower-income group.