A Micro project on

CLUSTERING MARKET DYNAMICS

Submitted in partial fulfillment of the

Computational Statistics Lab

GRIET Lab On Board (G-LOB)

By

A SHEERSHIKA (2XXX1A3203)

Under the esteemed guidance of

Aditya Sharma Panyam

Assistant Professor

Department of Computer Science and Business System

GOKARAJU RANGARAJU

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Autonomous under JNTUH)

Bachupally, Kukatpally, Hyderabad-500090.

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEM

CERTIFICATE

This is to certify that the micro project titled "**CLUSTERING MARKET DYNAMICS**" is a bonafide work done by **A SHEERSHIKA (2XXX1A3203)** under Computational Statistics Lab-GRIET Lab On Board (G-LOB) practice of our institute and that this work has not been submitted for the award of any other Degree/Diploma of any Institution/University.

Project Guide

Aditya Sharma Panyam

Assistant Professor

Table of Contents

1. Introduction

1.1. Need for the Project

As the business landscape undergoes dynamic shifts driven by technological advancements, the need for businesses to understand and adapt to consumer preferences becomes increasingly crucial. Traditional marketing strategies, which often generalize customer demographics, are proving ineffective in the face of diverse and discerning consumer bases. The need for a more nuanced understanding of customer behavior, encompassing individual preferences and purchasing patterns, is the driving force behind this project.

1.2. Project Description

Our project centers around harnessing the power of machine learning to unlock valuable insights from a comprehensive marketing campaign dataset. By employing advanced clustering algorithms, we aim to unveil hidden patterns and discernible segments within the customer data. This segmentation goes beyond traditional demographics, delving into nuanced characteristics and behaviors, enabling businesses to tailor their marketing initiatives with precision.

1.3 Components of the Project

The complexity of our project is addressed through a meticulous breakdown of its components:

Data Import and Libraries: The project initiation involves importing Python libraries essential for machine learning and loading the marketing campaign dataset. This phase sets the foundation for subsequent analyses.
Data Cleaning: To ensure the integrity of our results, a rigorous data cleaning process is undertaken. Missing values are handled, and feature engineering techniques are applied to enhance the dataset's quality.
Data Preprocessing: The raw data undergoes preprocessing steps such as label encoding, standard scaling, and other transformations to make it compatible with clustering algorithms.
Dimensionality Reduction: To streamline the computational process and enhance interpretability, dimensionality reduction techniques like Principal Component Analysis (PCA) are implemented.
Clustering: The core of our project involves the application of advanced clustering algorithms, including K-means and hierarchical clustering, to identify distinct customer segments based on a multitude of attributes.

Model Evaluation: The performance of our clustering models is rigorously evaluated using metrics such as silhouette score and adjusted Rand index. This ensures the reliability and accuracy of the identified clusters.
Visual Profiling: Going beyond numerical metrics, our project emphasizes the generation of visual profiles for each customer segment. These profiles serve as intuitive tools for businesses to understand the characteristics and preferences of each identified cluster.

2. Requirement Analysis

The requirement analysis phase is a pivotal stage where we delve deeper into the intricacies of our marketing dataset to ensure the robustness and relevance of our customer segmentation model. Beyond the identification of basic criteria, we meticulously examine the temporal and contextual dimensions of the data. This involves understanding the historical context of customer interactions and transactions, allowing us to identify patterns and trends that might influence current behaviors. By incorporating this historical perspective, our segmentation model gains a more nuanced understanding of customer preferences, enabling businesses to tailor their marketing strategies more effectively.

A critical aspect of our requirement analysis involves evaluating the dataset for any biases that might inadvertently affect the accuracy of our segmentation. Recognizing and mitigating these biases is crucial for ensuring fairness and preventing unintentional discrimination in marketing practices. Moreover, we assess the scalability requirements of our model, anticipating future increases in data volume and technological advancements. This forward-thinking approach not only enhances the precision of our segmentation but also future-proofs our model against potential challenges arising from evolving data landscapes.

3. Implementation

3.1 Libraries

1. NumPy (np):
Description: NumPy is a powerful numerical computing library in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.
Use in Code: NumPy is used for array operations and setting the random seed.
2. Pandas (pd):
Description: Pandas is a data manipulation and analysis library. It provides data structures like DataFrames for efficiently handling and analyzing structured data.
Use in Code: Likely used for loading and manipulating datasets.
3. Matplotlib:
Description: Matplotlib is a 2D plotting library for creating static, interactive, and animated visualizations in Python.
Use in Code: Used for creating various types of plots and charts for data visualization.
4. Seaborn:
Description: Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
Use in Code: Used for enhancing the visual aesthetics of Matplotlib plots.
5. Scikit-Learn:
Description: Scikit-Learn is a machine learning library that provides simple and efficient tools for data mining and data analysis. It includes various algorithms for classification, regression, clustering, and more.
Use in Code: Used for preprocessing data, dimensionality reduction, clustering, and model evaluation.
6. Yellowbrick:

Description: Yellowbrick is a visualization library that extends Scikit-Learn and Matplotlib to provide visual diagnostics and model selection tools for machine learning.
Use in Code: Used for visualizing the elbow method to find the optimal number of clusters.
7. mpl_toolkits.mplot3d (Axes3D):
Description: mpl_toolkits.mplot3d is a toolkit in Matplotlib that provides tools for 3D plotting. Axes3D is a class for creating 3D axes in Matplotlib.
Use in Code: Used for creating 3D plots, possibly to visualize clustered data in three dimensions.
8. AgglomerativeClustering:
Description: AgglomerativeClustering is a hierarchical clustering algorithm available in Scikit-Learn. It recursively merges the pair of clusters that minimally increases a given linkage distance.
Use in Code: Used for hierarchical clustering.
9. ListedColormap:
Description: ListedColormap is a class in Matplotlib that generates a colormap from a list of colors.
Use in Code: Potentially used for customizing the colors in visualizations.
10. Warnings and Sys:
Description: The warnings module provides a flexible way to handle warnings in Python. The sys module provides access to some variables used or maintained by the Python interpreter and to functions that interact strongly with the interpreter.
Use in Code: Used for handling warnings to maintain code cleanliness.

3.2 Feature Engineering
Age of Customer Today:
A new feature, "Age," is created by subtracting the customer's year of birth from the current year (assumed to be 2021).

Total Spending on Various Items:
Combines spending on different product categories ("Wines," "Fruits," "Meat," "Fish," "Sweets," "Gold") to create a new feature representing the total amount spent by each customer.

Deriving Living Situation by Marital Status ("Alone"):
Creates a new feature, "Living_With," indicating the living situation of the customer as either "Alone" or "With a Partner" based on their marital status.

Total Children Living in the Household:
Combines the count of children and teenagers in the household to create a new feature representing the total number of children.

Total Members in the Household:
Calculates the total number of members in the household, considering both the living situation and the number of children.

Parenthood Indicator:
Creates a binary indicator (1 or 0) to identify whether a customer is a parent based on the presence of children.

Segmenting Education Levels:
Segments education levels into three groups: "Undergraduate," "Graduate," and "Postgraduate" for simplicity and clarity.

Renaming Columns for Clarity:
Renames columns such as spending categories for better readability and clarity.

Dropping Redundant Features:
Removes redundant or unnecessary features from the dataset to reduce dimensionality and improve focus on relevant information.

3.3 Clustering and Profiling

1. Elbow Method for Determining Number of Clusters:
Objective:
Utilizing the Elbow Method to identify the optimal number of clusters for the subsequent clustering analysis.
Execution:
A KElbowVisualizer is employed with KMeans clustering to visualize the Elbow Method and help in determining the appropriate number of clusters.
The results are then displayed.
2. Agglomerative Clustering:
Objective:
Employing Agglomerative Clustering to cluster data points based on similarity.
Execution:
The Agglomerative Clustering model is initiated with the specified number of clusters (in this case, 4).
The model is fitted to the data, and predictions are made to assign cluster labels to each data point.
The resulting cluster labels are added to the original dataset under the column "Clusters."

3. 3D Scatter Plot of Clusters:
Objective:
Visualizing the clusters in a three-dimensional space.
Execution:
A 3D scatter plot is generated using Matplotlib, where the x, y, and z coordinates represent features from the dataset.
Points are colored according to their assigned clusters, providing a clear visualization of the clustering results.
The plot is displayed to illustrate the distribution of data points in different clusters.

Profiling
1. Visualizing Spending Patterns Based on Personal Features:
Objective:
Analyzing spending patterns by visualizing the relationships between personal features and total spending.
Execution:
For each personal feature (e.g., "Kidhome," "Teenhome," "Customer_For," "Age," "Children," "Family_Size," "Is_Parent," "Education," "Living_With"), a joint plot is created.
The joint plots use kernel density estimation (kde) to visualize the distribution of spending ("Spent") for different clusters, allowing insights into spending behavior based on personal characteristics.
Plots are displayed individually.

CODE

```
#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
```

[32]:

```
#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
```

[32]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	Acce
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635	...	7	0	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11	...	5	0	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426	...	4	0	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26	11	...	6	0	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	5	0	

5 rows × 29 columns

[33]:
```python
#Information on features
data.info()
```

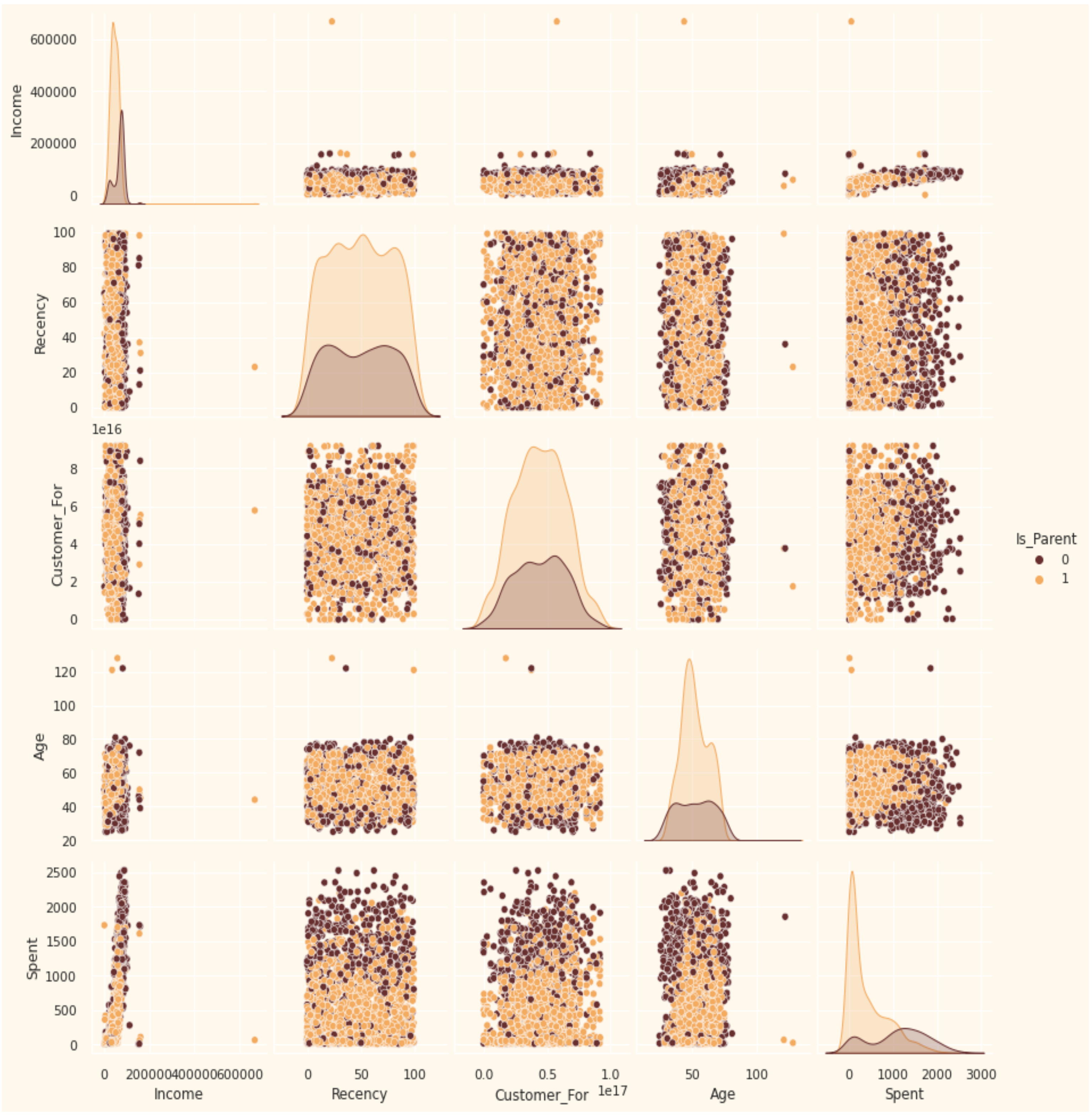
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ID                   2240 non-null   int64  
 1   Year_Birth           2240 non-null   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  
```

[35]:
```python
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))
```

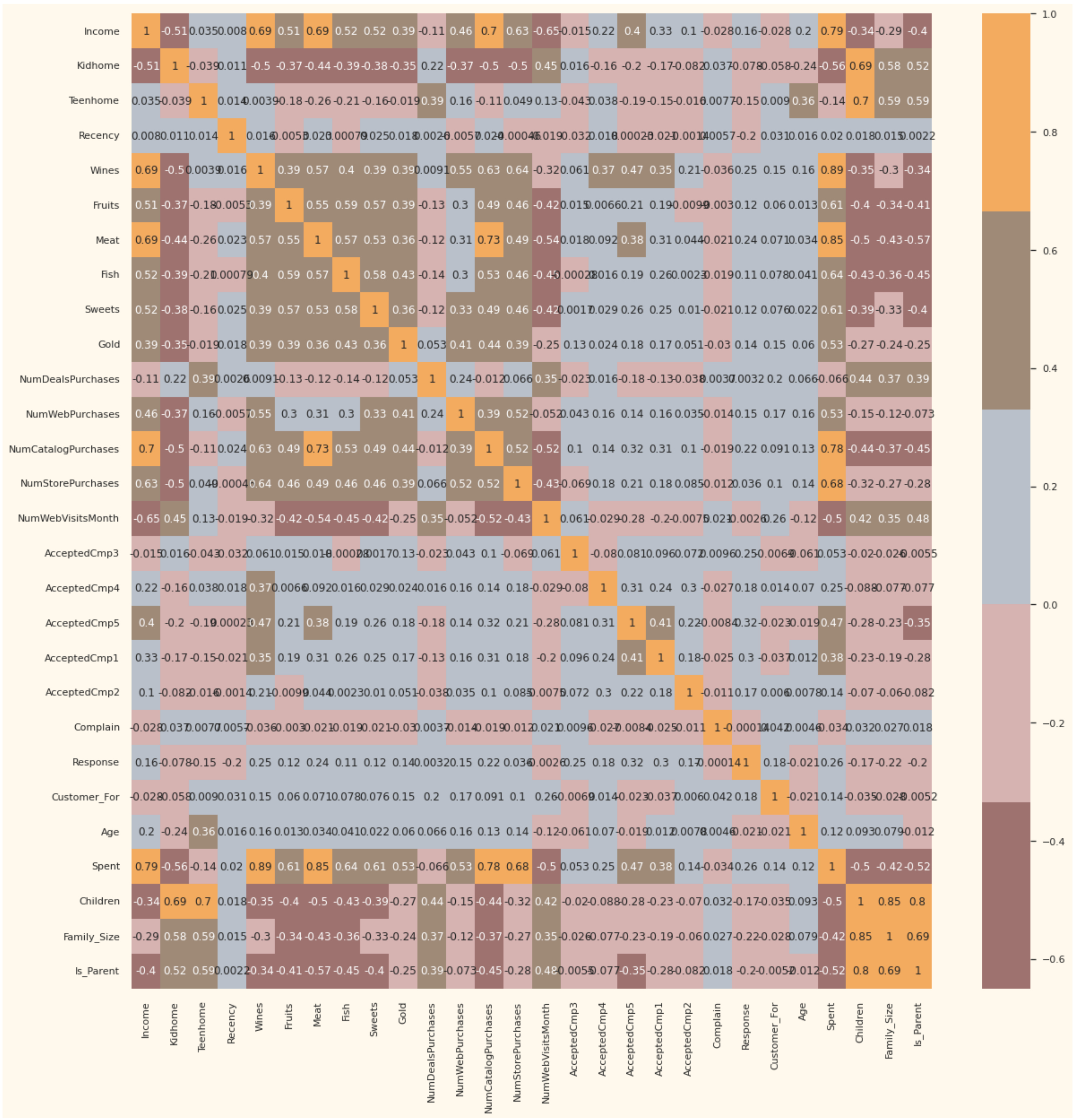
```
The newest customer's enrolment date in therecords: 2014-12-06
The oldest customer's enrolment date in the records: 2012-01-08
```

```python
#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"]))
#Taking hue
plt.show()
```

```
Reletive Plot Of Some Selected Features: A Data Subset
<Figure size 576x396 with 0 Axes>
```



Income
Recency
Customer_For
Age
Spent
1e16
1e17
Is_Parent
0
1


Income
Kidhome
Teenhome
Recency
Wines
Fruits
Meat
Fish
Sweets
Gold
NumDealsPurchases
NumWebPurchases
NumCatalogPurchases
NumStorePurchases
NumWebVisitsMonth
AcceptedCmp3
AcceptedCmp4
AcceptedCmp5
AcceptedCmp1
AcceptedCmp2
Complain
Response
Customer_For
Age
Spent
Children
Family_Size
Is_Parent
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

[47]:

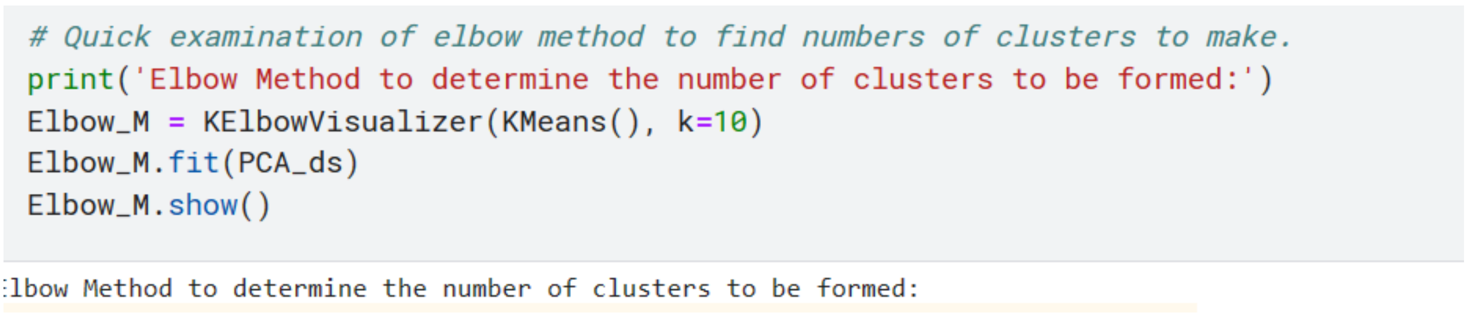
```python
#Initiating PCA to reduce dimentions aka features to 3
pca = PCA(n_components=3)
pca.fit(scaled_ds)
PCA_ds = pd.DataFrame(pca.transform(scaled_ds), columns=(["col1","col2", "col3"]))
PCA_ds.describe().T
```

[47]:

	count	mean	std	min	25%	50%	75%	max
col1	2212.0	-1.116246e-16	2.878377	-5.969394	-2.538494	-0.780421	2.383290	7.444305
col2	2212.0	1.105204e-16	1.706839	-4.312196	-1.328316	-0.158123	1.242289	6.142721
col3	2212.0	3.049098e-17	1.221956	-3.530416	-0.829067	-0.022692	0.799895	6.611222

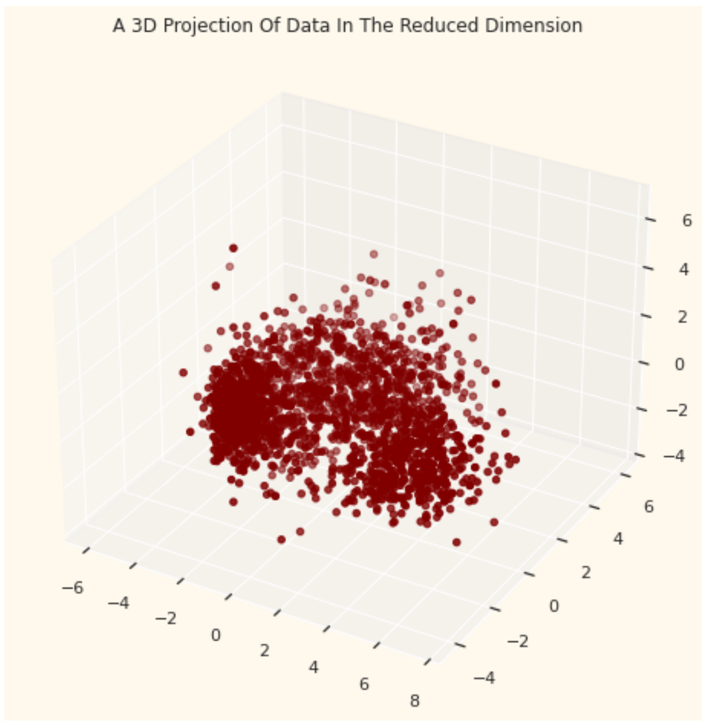
[48]:

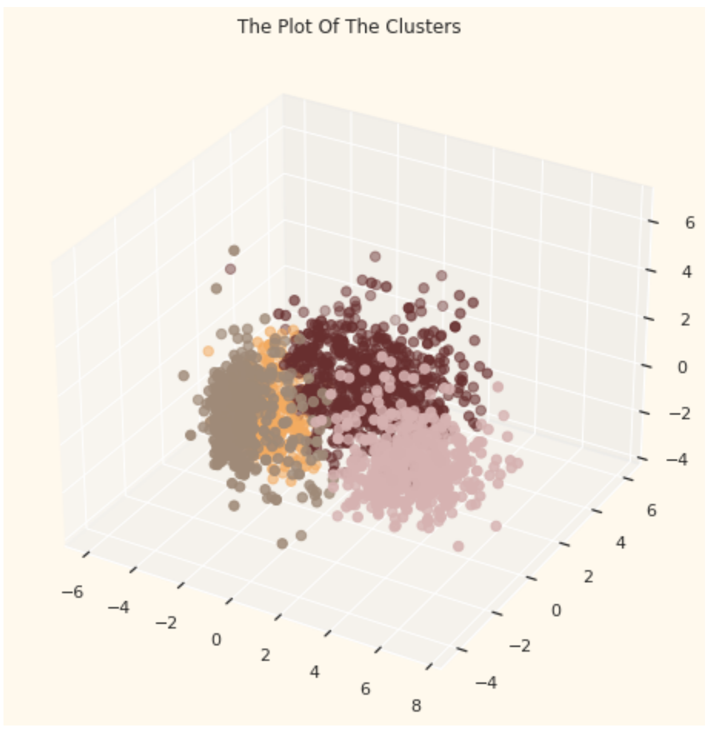
```python
#A 3D Projection Of Data In The Reduced Dimension
x =PCA_ds["col1"]
y =PCA_ds["col2"]
z =PCA_ds["col3"]
#To plot
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection="3d")
ax.scatter(x,y,z, c="maroon", marker="o" )
ax.set_title("A 3D Projection Of Data In The Reduced Dimension")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

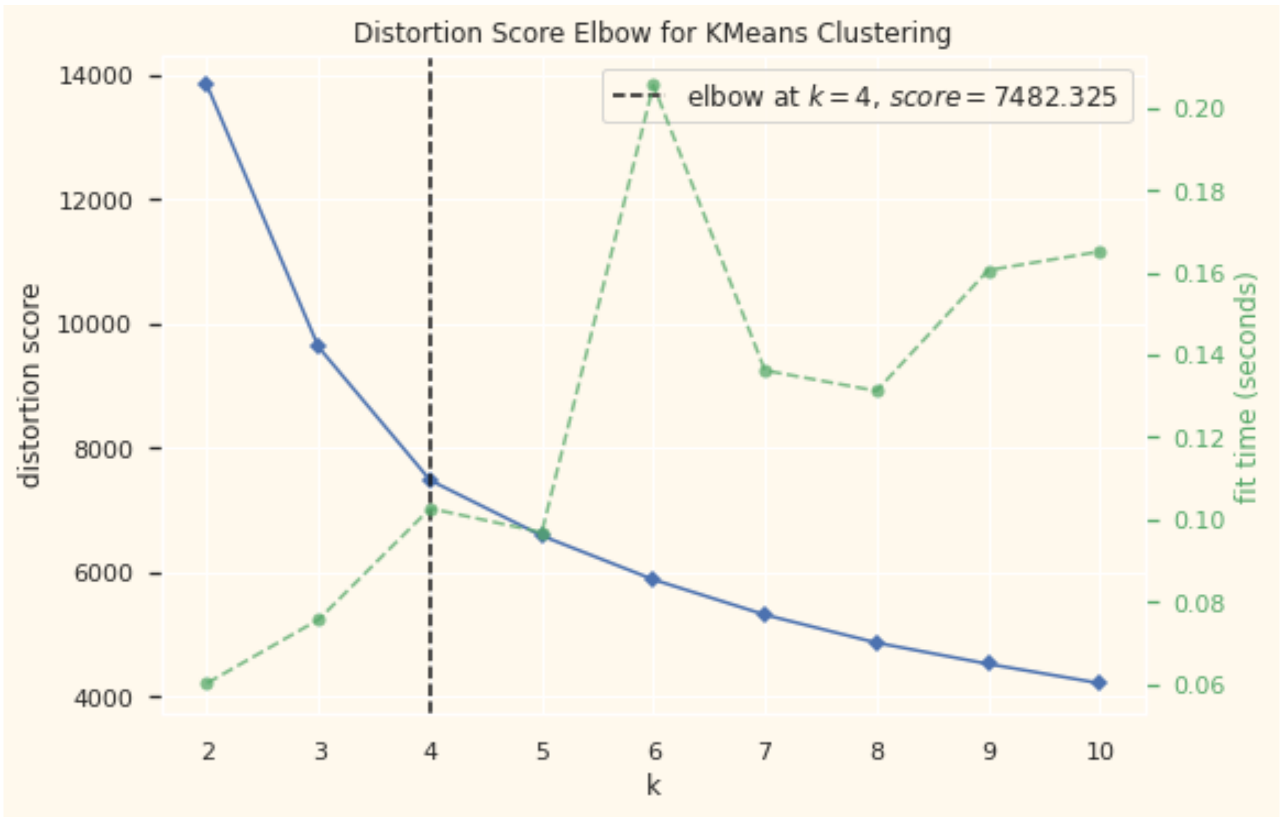


```python
# Quick examination of elbow method to find numbers of clusters to make.
print('Elbow Method to determine the number of clusters to be formed:')
Elbow_M = KElbowVisualizer(KMeans(), k=10)
Elbow_M.fit(PCA_ds)
Elbow_M.show()
```

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





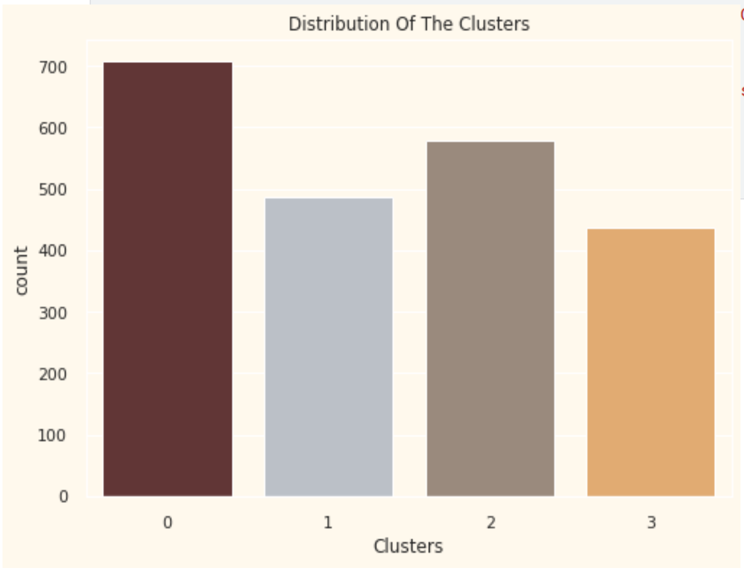


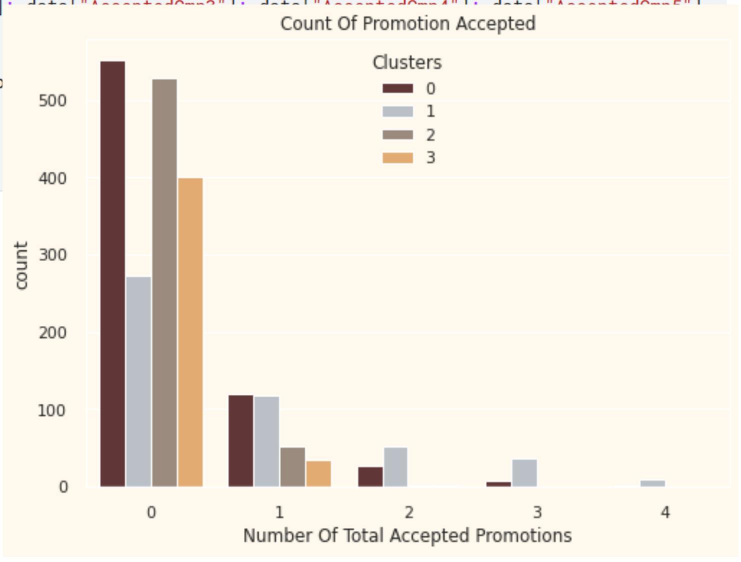
[52]:
```
#Plotting countplot of clusters
pal = ["#682F2F","#B9C0C9", "#9F8A78","#F3AB60"]
pl = sns.countplot(x=data["Clusters"], palette= pal)
pl.set_title("Distribution Of The Clusters")
plt.show()
```

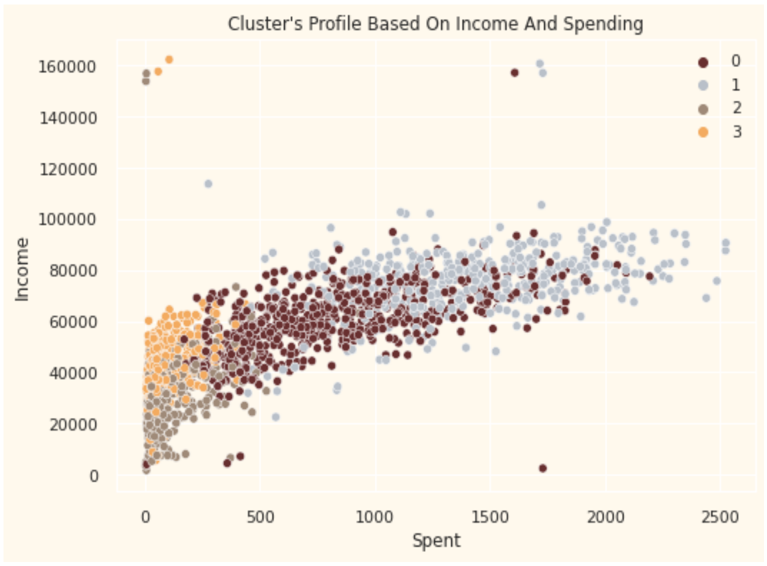
[53]:
```
pl = sns.scatterplot(data = data,x=data["Spent"], y=data["Income"],hue=data["Clusters"], palette= pal)
pl.set_title("Cluster's Profile Based On Income And Spending")
plt.legend()
plt.show()
```

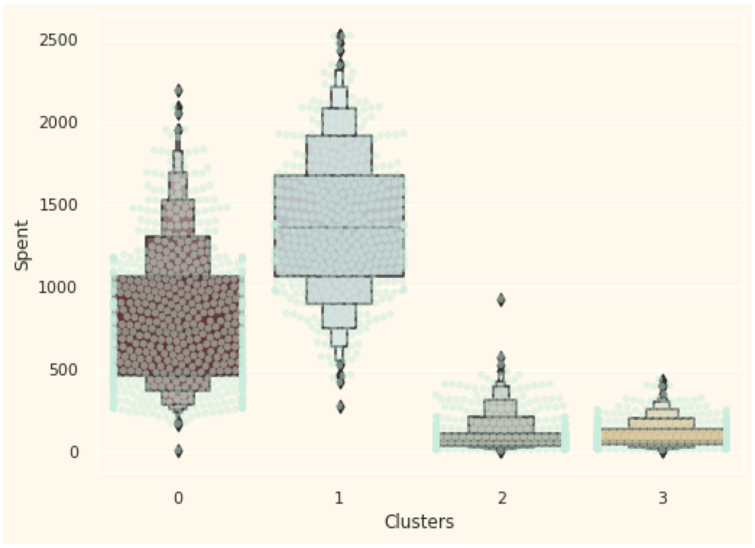
[54]:
```
plt.figure()
pl=sns.swarmplot(x=data["Clusters"], y=data["Spent"], color= "#CBEDDD", alpha=0.5 )
pl=sns.boxenplot(x=data["Clusters"], y=data["Spent"], palette=pal)
plt.show()
```

```
[55]: #Creating a feature to get a sum of accepted promotions
      Cmp2"]
      s"], p
```





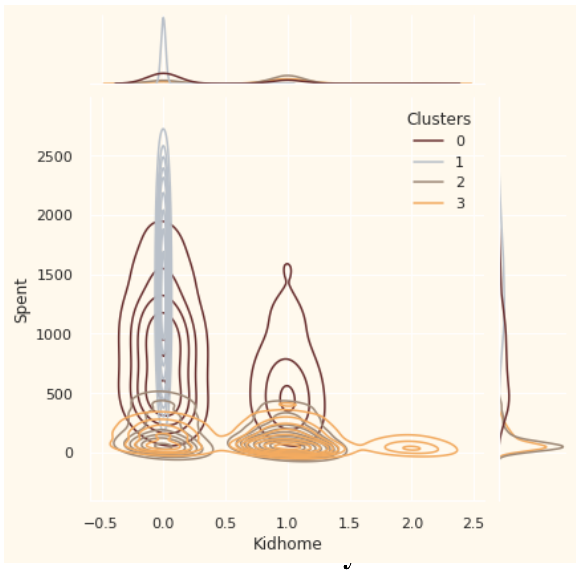


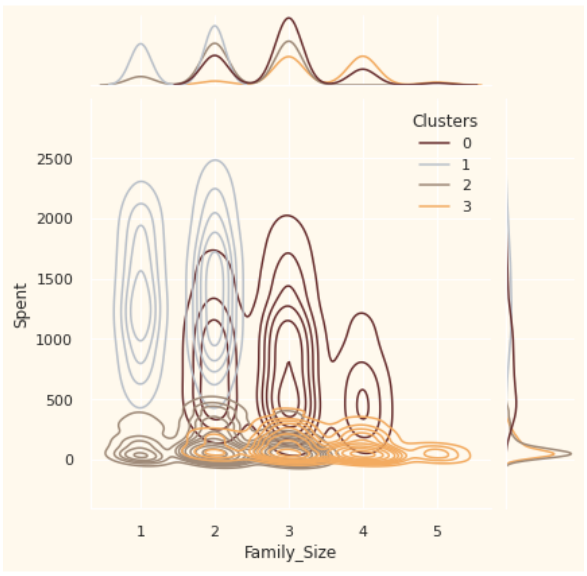


```
[58]: Personal = [ "Kidhome","Teenhome","Customer_For", "Age", "Children", "Family_Size", "Is_Parent", "Education","Living_With"]

      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 576x396 with 0 Axes>
```





The application of the Elbow Method aids in determining the optimal number of clusters for subsequent analysis. By visualizing the trade-off between the number of clusters and the within-cluster sum of squares, the "elbow" point signifies a suitable balance, influencing the selection of an appropriate cluster count.

4.2 Agglomerative Clustering Results:
Agglomerative Clustering, a hierarchical clustering technique, is employed to group data points into clusters. With a predetermined number of clusters set at four, this method allows for the exploration of inherent patterns and similarities within the dataset.

4.3 3D Scatter Plot:
The 3D scatter plot provides a visual representation of the clustered data. Each point is colored based on its assigned cluster, offering an intuitive understanding of how data points are spatially distributed within the identified clusters.

4.4 Personal Features and Spending Patterns:
Individual joint plots are generated for personal features such as "Kidhome," "Teenhome," "Customer_For," "Age," "Children," "Family_Size," "Is_Parent," "Education," and "Living_With." These plots utilize kernel density estimation to showcase the distribution of spending across different clusters, revealing nuanced spending patterns based on diverse customer characteristics.

5. Conclusion:

The comprehensive clustering analysis based on personal features has provided valuable insights into customer segmentation and spending patterns. The exploration of diverse aspects, from age and living arrangements to educational backgrounds and parenthood status, has uncovered distinct clusters within the customer base. These clusters offer a nuanced understanding of the diverse behaviors and preferences exhibited by different groups of customers.

The key findings from the clustering analysis can significantly impact strategic decision-making in marketing and customer engagement. By acknowledging the heterogeneity within the customer base, businesses can tailor their approaches to better resonate with specific segments. This personalized approach is critical for enhancing customer satisfaction, loyalty, and overall experience.

5.1 Implications for Marketing Strategies:

Targeted Campaigns: With identified clusters, targeted marketing campaigns can be designed to address the unique preferences of each segment.
Personalized Recommendations: Understanding spending patterns allows for the creation of personalized product recommendations, increasing the likelihood of customer engagement.

5.2 Operational Insights:

Resource Allocation: Insights from the clustering analysis can guide resource allocation, ensuring that marketing efforts are concentrated where they are most likely to yield positive results.
Product Development: Identification of high-spending segments can inform product development strategies, aligning offerings with the needs and preferences of specific customer groups.

5.3 Enhancing Customer Experience:

Tailored Communications: Clusters provide a basis for tailored communication strategies, ensuring that messages resonate with the unique characteristics of each customer segment.
Customer Retention: Understanding the factors influencing spending behavior facilitates the development of retention strategies, reducing churn and fostering long-term customer relationships.
In conclusion, the clustering analysis not only offers a snapshot of the current customer landscape but also provides a roadmap for strategic decision-making. The personalized insights gained from this analysis empower businesses to create more targeted, efficient, and customer-centric approaches, ultimately contributing to overall business success. Continued monitoring and adaptation based on customer behavior will be crucial for maintaining relevance in an ever-evolving market landscape.

6. References:

- A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects Absalom E. Ezugwu a, Abiodun M. Ikotun a, Olaide O. Oyelade a, Laith Abualigah b c, Jeffery O. Agushaka a, Christopher I. Eke d, Andronicus A. Akinyelu.
- A review of clustering techniques and developments Amit Saxena a, Mukesh Prasad b, Akshansh Gupta c, Neha Bharill d, Om Prakash Patel d, Aruna Tiwari d, Er Meng Joo e, Ding Weiping f, Lin Chin-Teng b.