Clustering Market Dynamics

ABSTRACT

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.

CONTENTS

Clustering Market Dynamics Using Unsupervised Clustering

Customer segmentation is a critical process for businesses seeking to understand and target specific groups of customers effectively. By utilizing unsupervised clustering techniques, businesses can derive valuable insights from large datasets to identify distinct customer segments based on their behaviors, preferences, and demographics. This allows for the development of tailored marketing strategies, personalized product offerings, and enhanced customer experiences.

Importing Libraries, Loading Data

Importing Libraries

At the beginning of the customer segmentation process, it's essential to import relevant data processing and machine learning libraries, such as Pandas, NumPy, and scikit-learn. These libraries provide powerful tools for data manipulation, analysis, and clustering algorithms.

Loading Data

Once the necessary libraries are imported, the next step involves loading the customer data from various sources, such as databases, spreadsheets, or APIs. Data integrity and quality checks are performed at this stage to ensure the accuracy and consistency of the information.

Data Cleaning

1

Identifying Data Anomalies

During the data cleaning phase, anomalies, outliers, and inconsistencies are identified and addressed to enhance the overall quality and reliability of the dataset.

2

Handling Missing Values

Strategies for handling missing data points, such as imputation or removal, are employed to ensure that the dataset used for clustering is complete and representative.

Data Preprocessing

Feature Engineering

Data preprocessing involves feature engineering to create new informative features and transform existing ones, resulting in a more effective representation of the underlying data.

Data Normalization

Normalization techniques, such as Min-Max scaling or Z-score standardization, are applied to ensure that all features contribute equally to the clustering process.

Dimensionality Reduction

1

Feature Selection

Identifying the most relevant features that significantly contribute to the variation within the data.

2

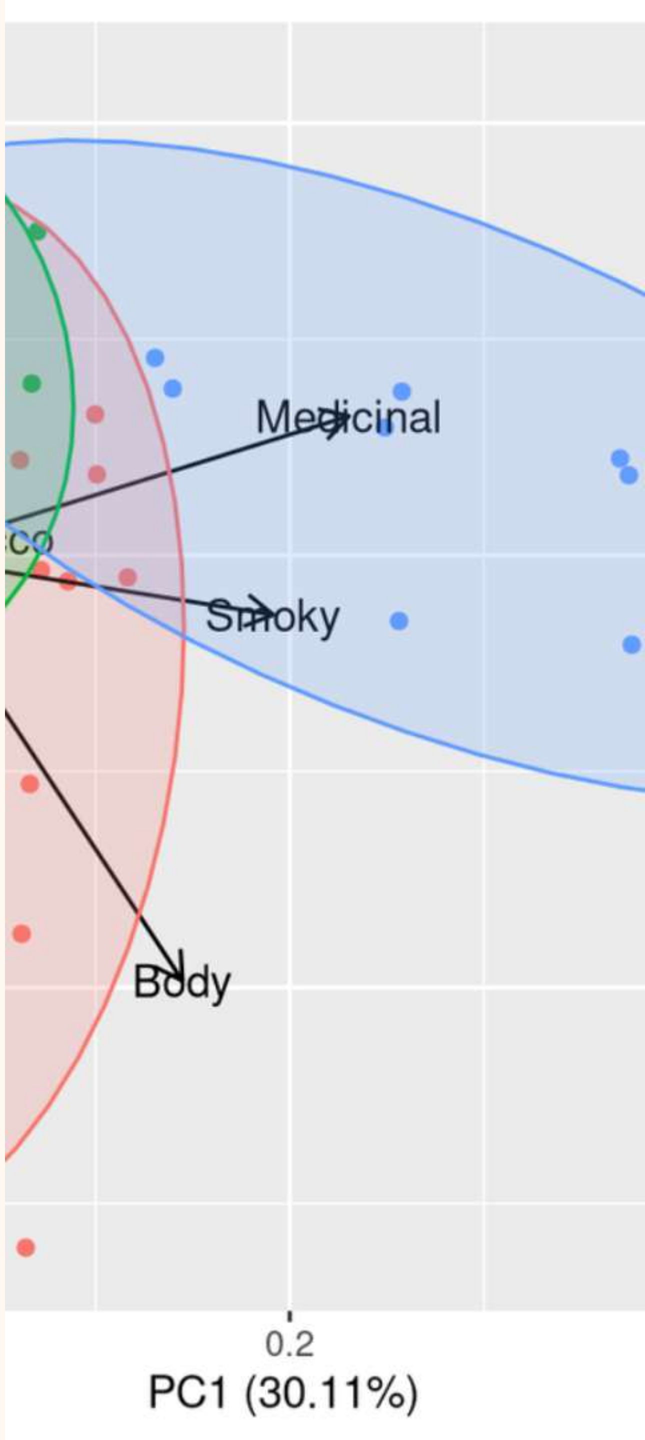
Principal Component Analysis (PCA)

PCA is used to reduce the dimensionality of the feature space while preserving the most vital information.

3

t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is employed to visualize high-dimensional data by embedding it in a lower-dimensional space.



Feature Scaling

1

Standardization

Application of standardization ensures that the features have a mean of 0 and a standard deviation of 1, making them comparable and eliminating bias due to differing scales.

2

Normalization

Normalization techniques are employed to scale the features within a specific range, preventing certain features from dominating the clustering algorithm due to their larger scales.

Unsupervised Clustering

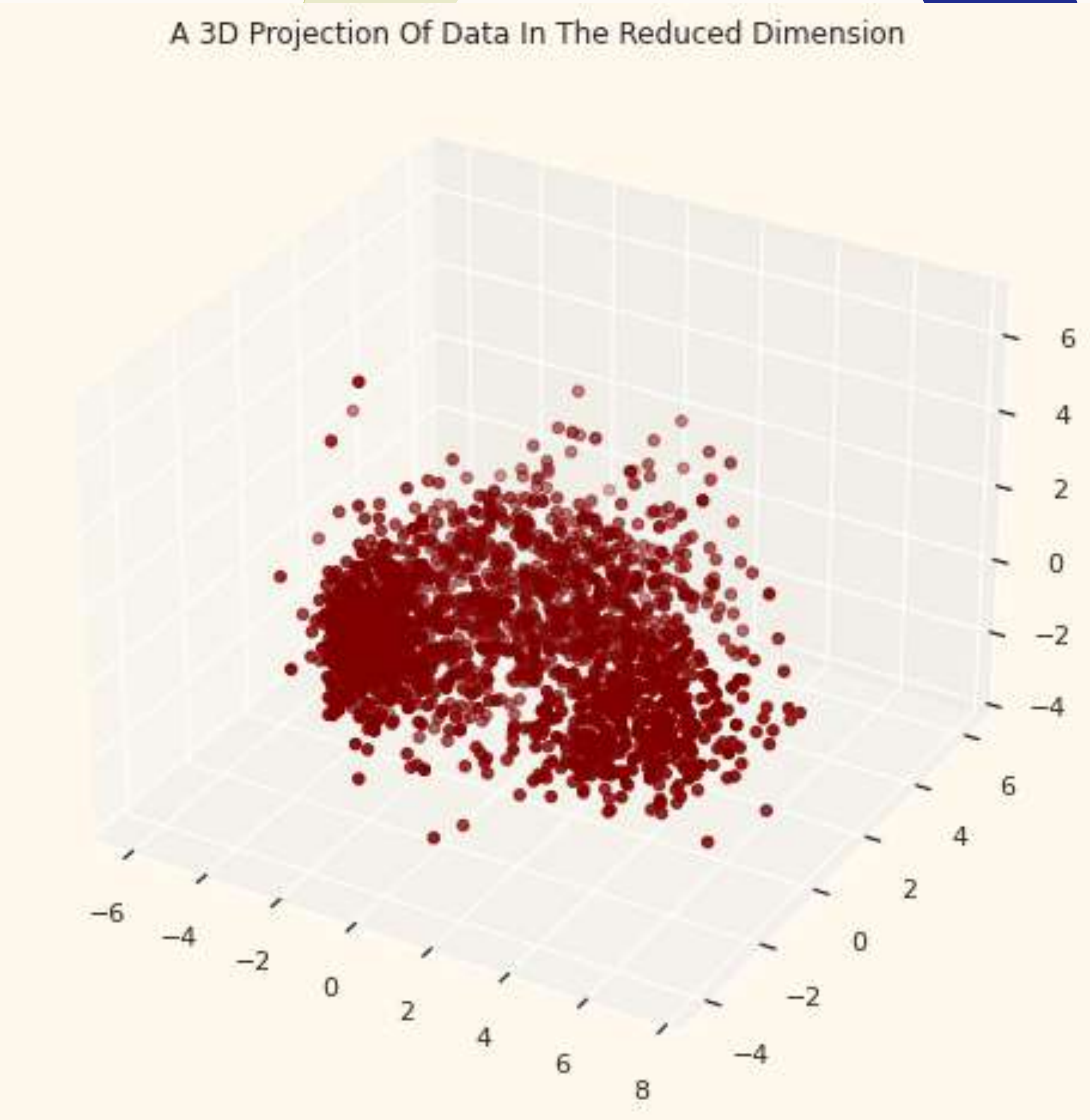
Algorithm Selection

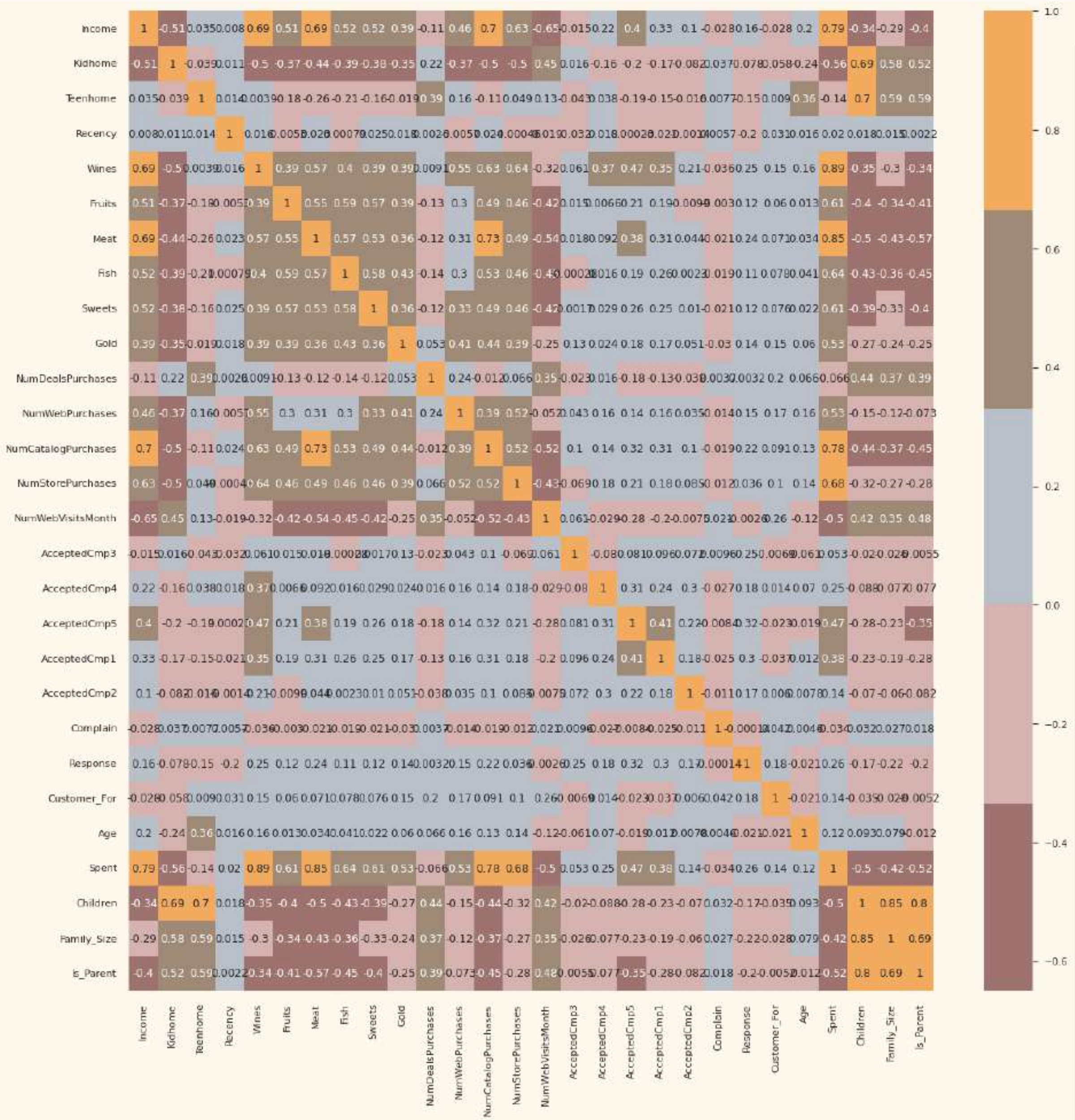
The selection of appropriate clustering algorithms, such as K-means, DBSCAN, or hierarchical clustering, plays a pivotal role in identifying meaningful clusters within the dataset.

Cluster Analysis

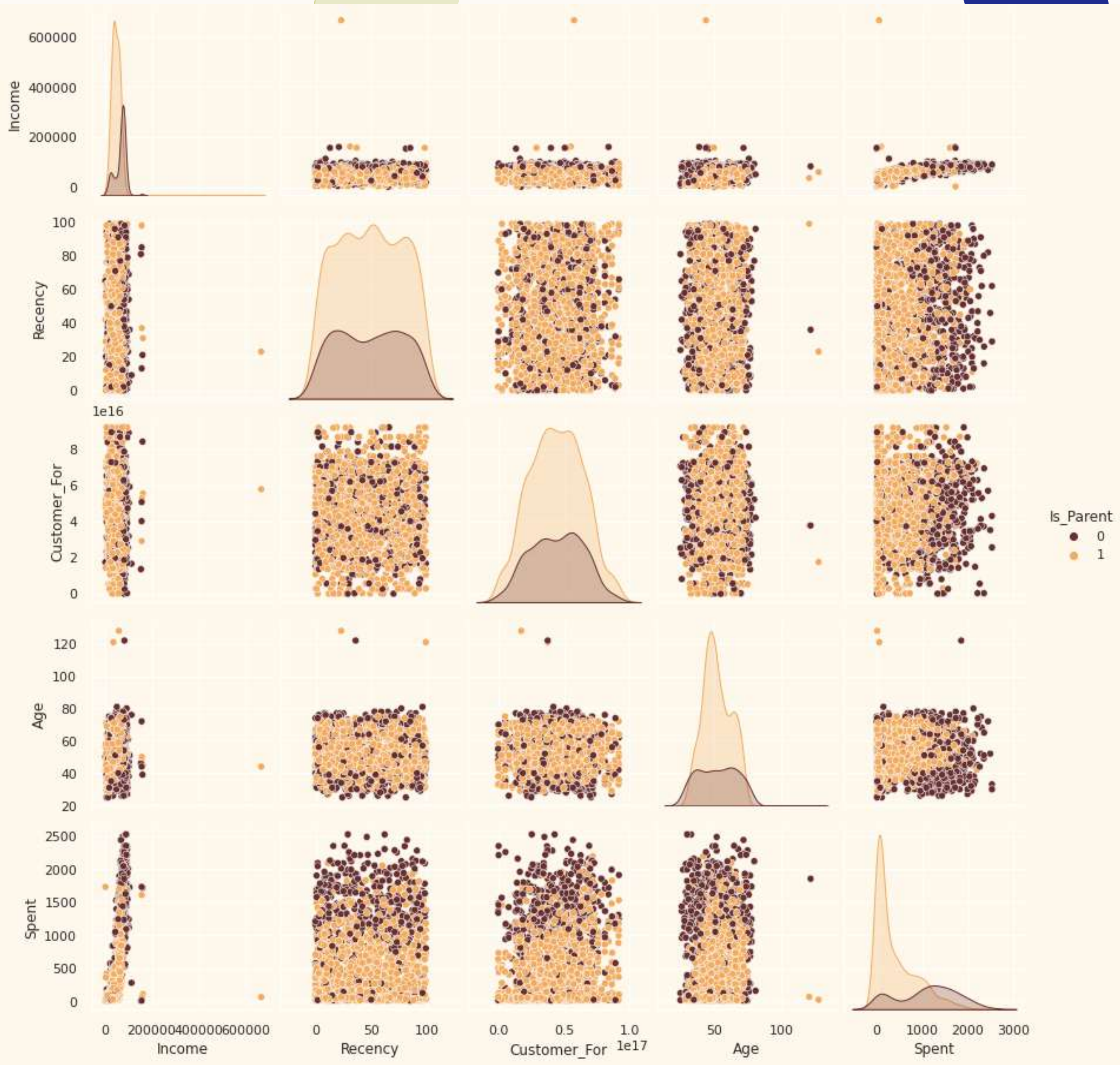
The application of clustering algorithms to the preprocessed and scaled data results in the identification of distinct customer segments based on similar characteristics and behaviors.

```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()
```





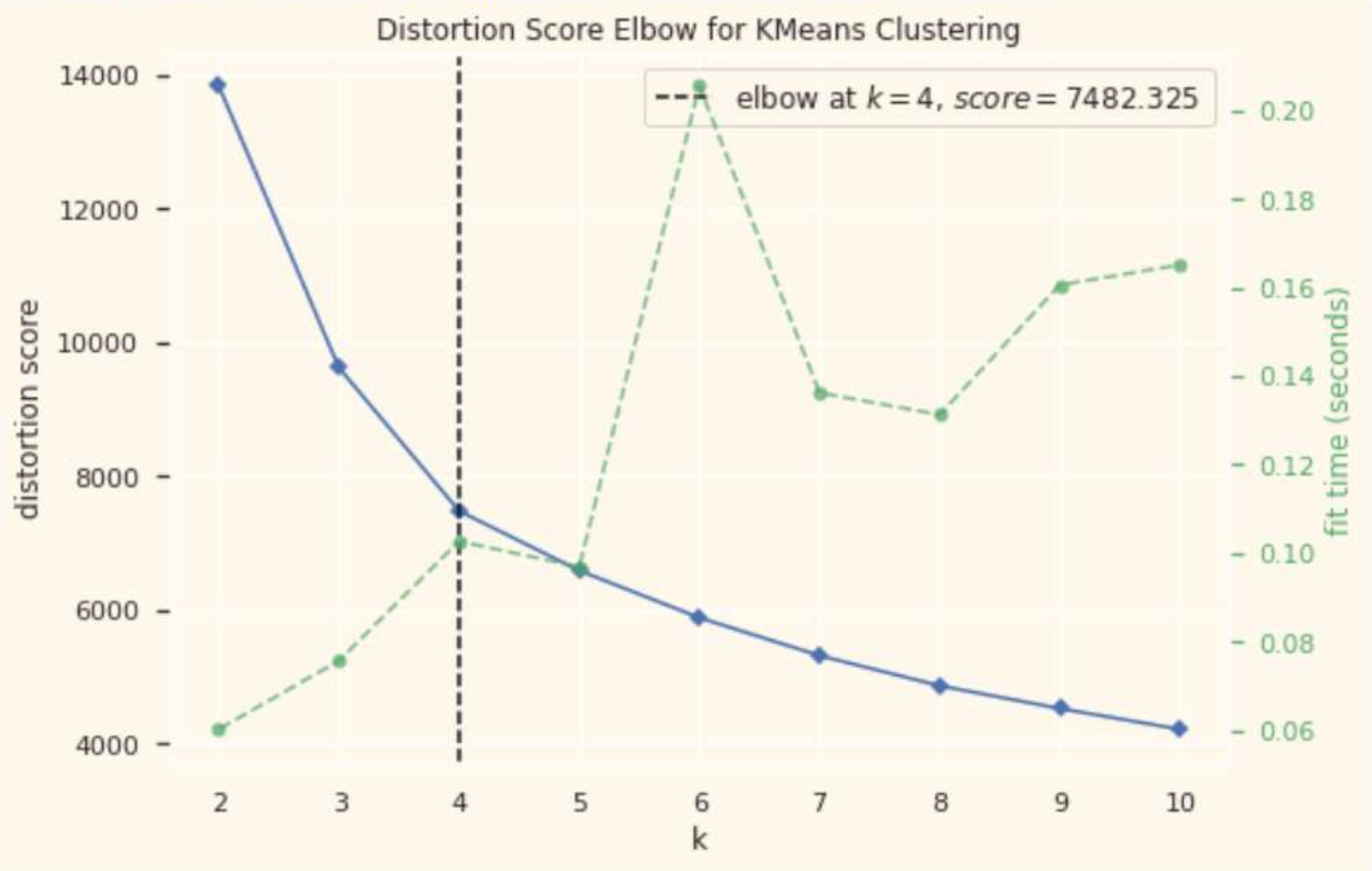
Income
Kidhome
Teenhome
Recency
Wines
Fruits
Meat
Fish
Sweets
Gold
NumDealsPurchases
NumWebPurchases
NumCatalogPurchases
NumStorePurchases
NumWebVisitsMonth
AcceptedCmp3
AcceptedCmp4
AcceptedCmp5
AcceptedCmp1
AcceptedCmp2
Complain
Response
Customer_For
Age
Spent
Children
Family_Size
Is_Parent



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

```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()
```

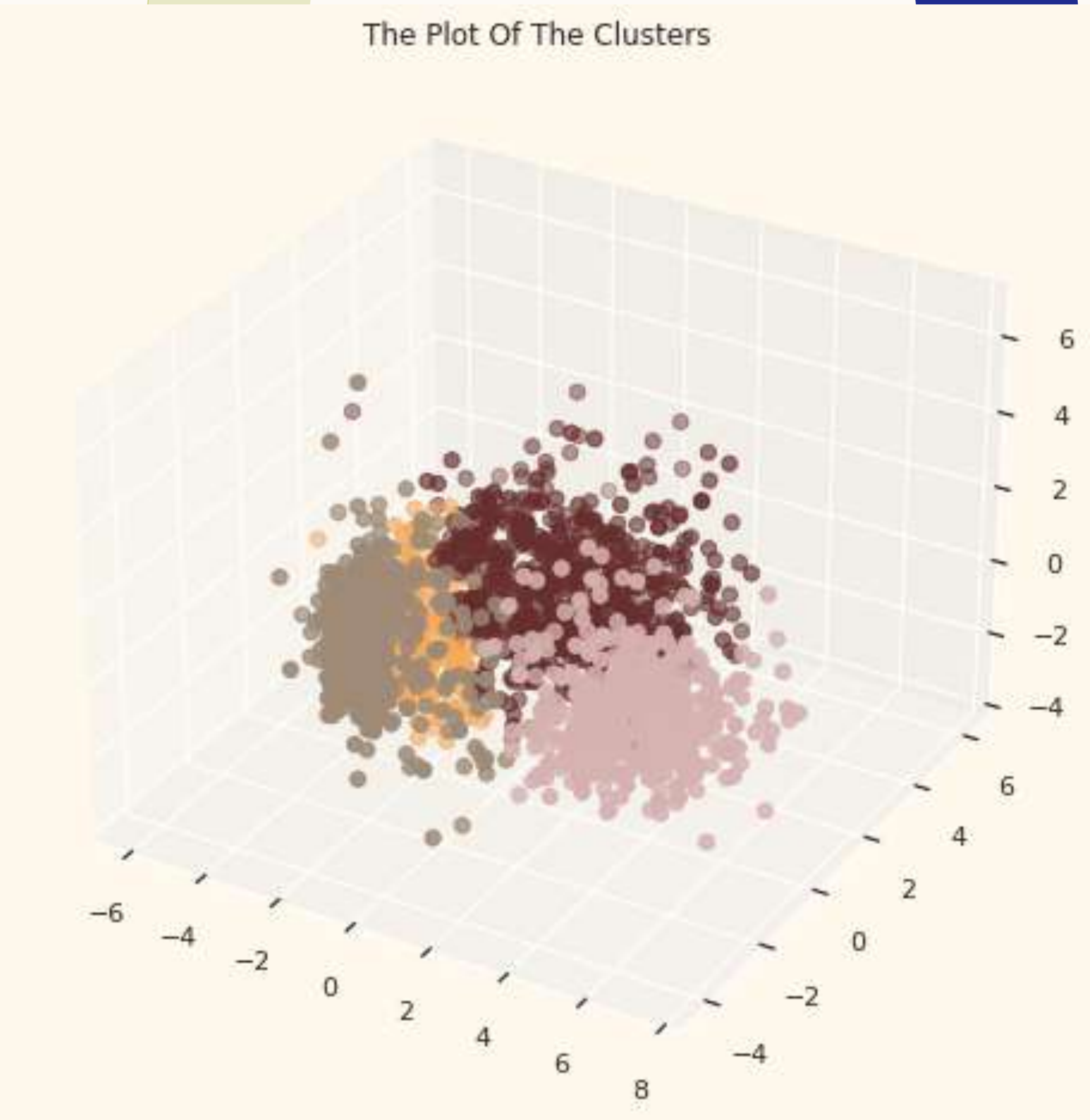
```
Elbow Method to determine the number of clusters to be formed:
```

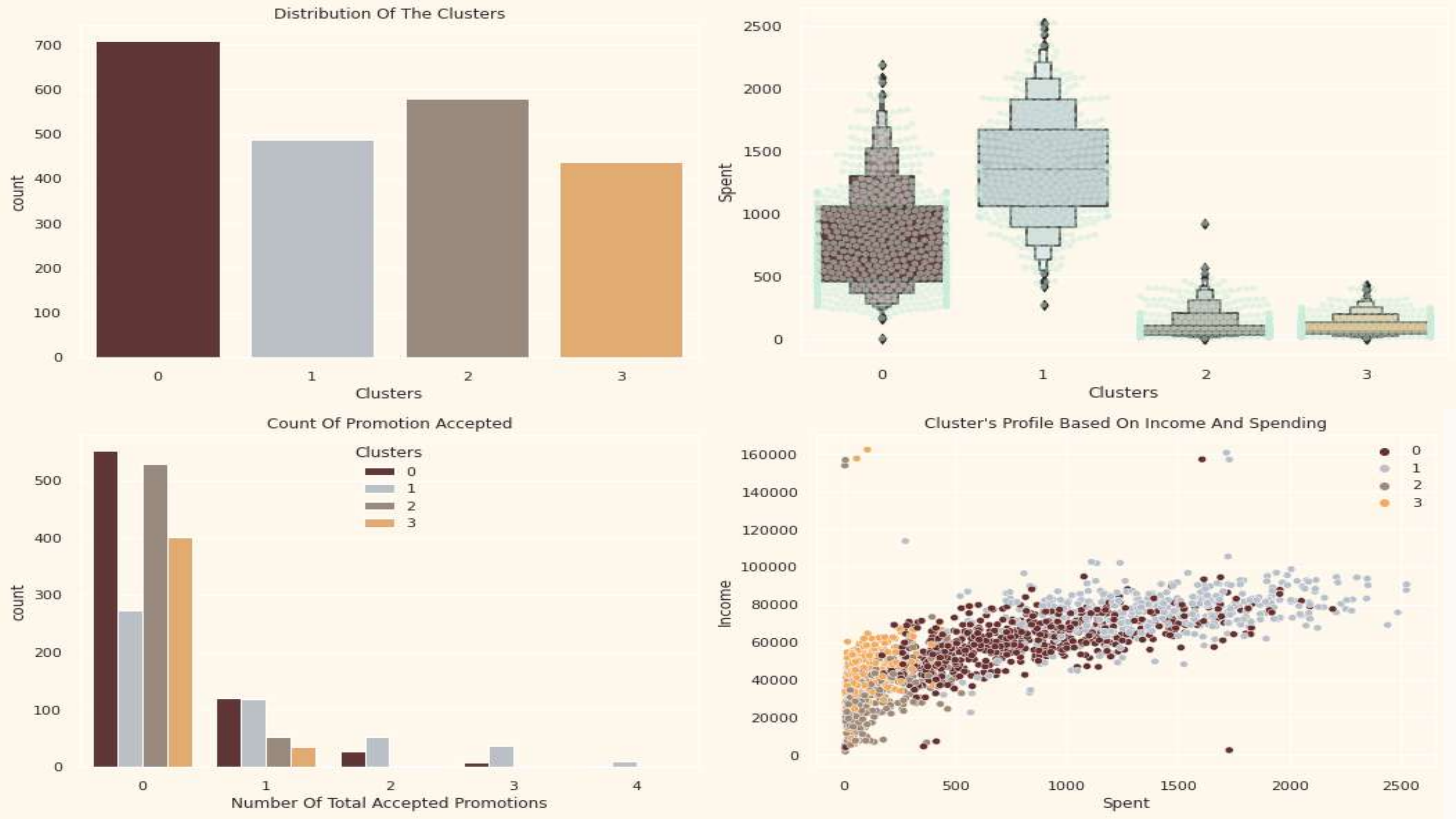


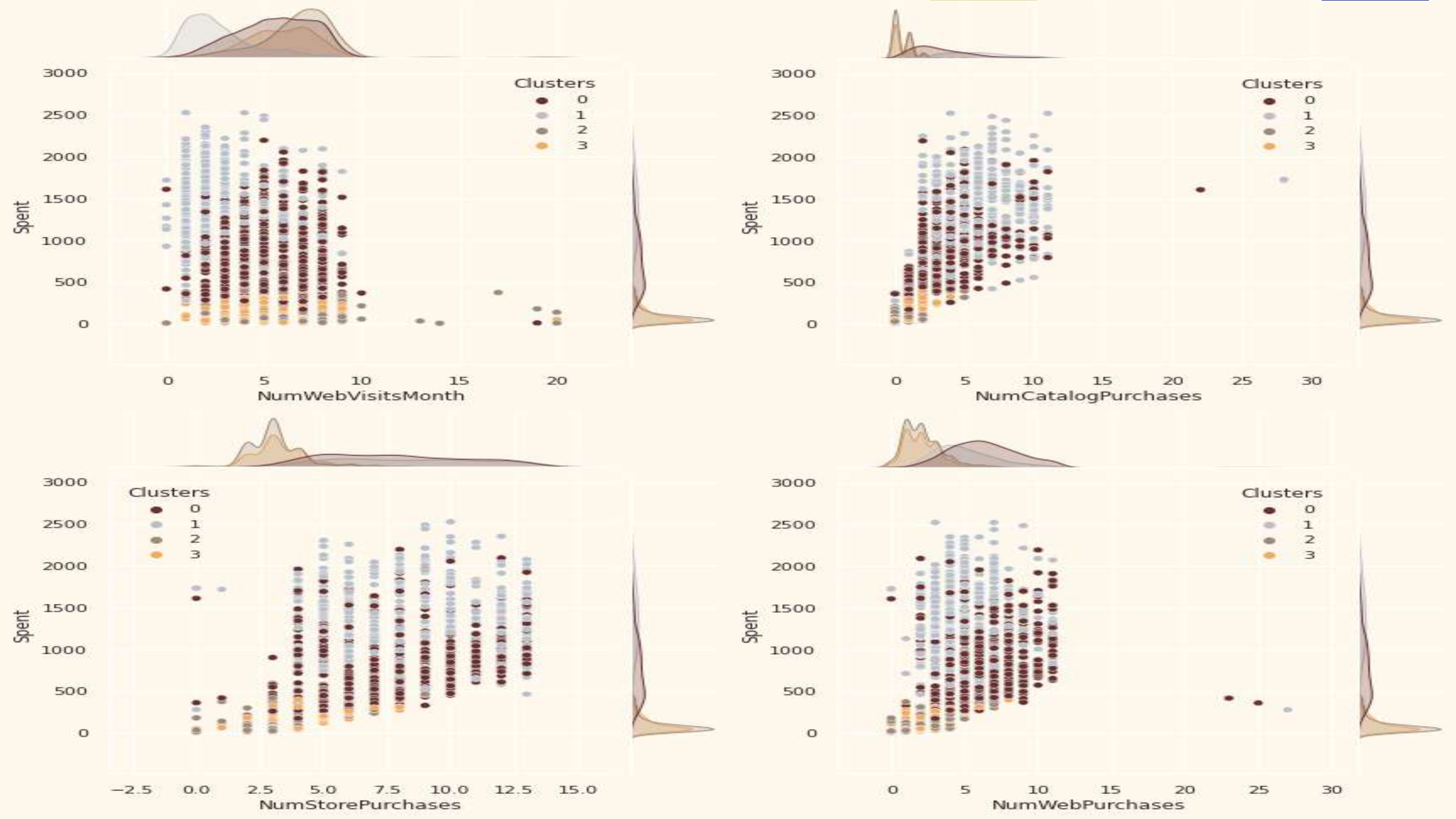

```
<AxesSubplot:title={'center':'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='distortion score'>
```

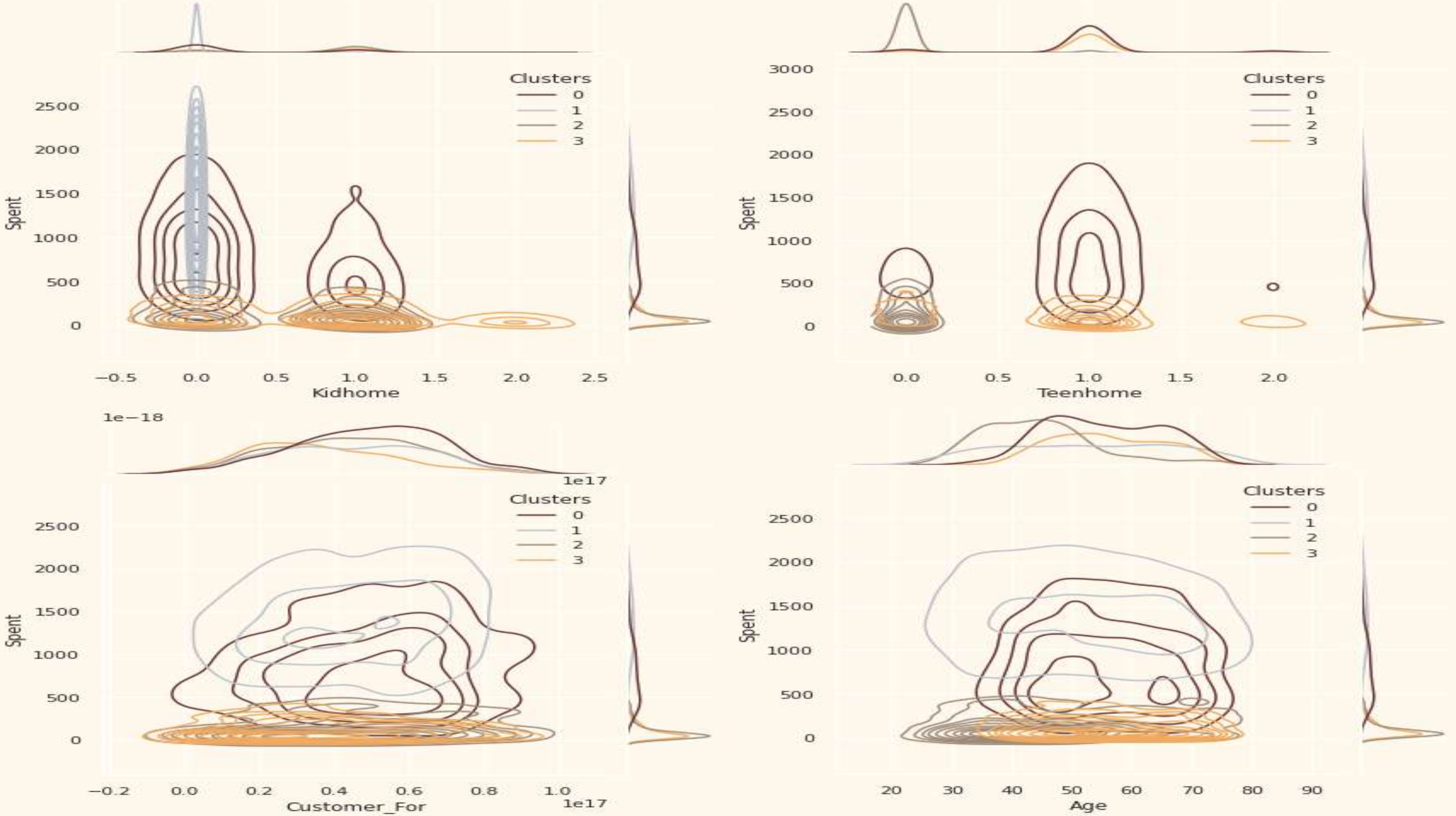
```python
#Initiating the Agglomerative Clustering model
AC = AgglomerativeClustering(n_clusters=4)
# fit model and predict clusters
yhat_AC = AC.fit_predict(PCA_ds)
PCA_ds["Clusters"] = yhat_AC
#Adding the Clusters feature to the orignal dataframe.
data["Clusters"]= yhat_AC
```

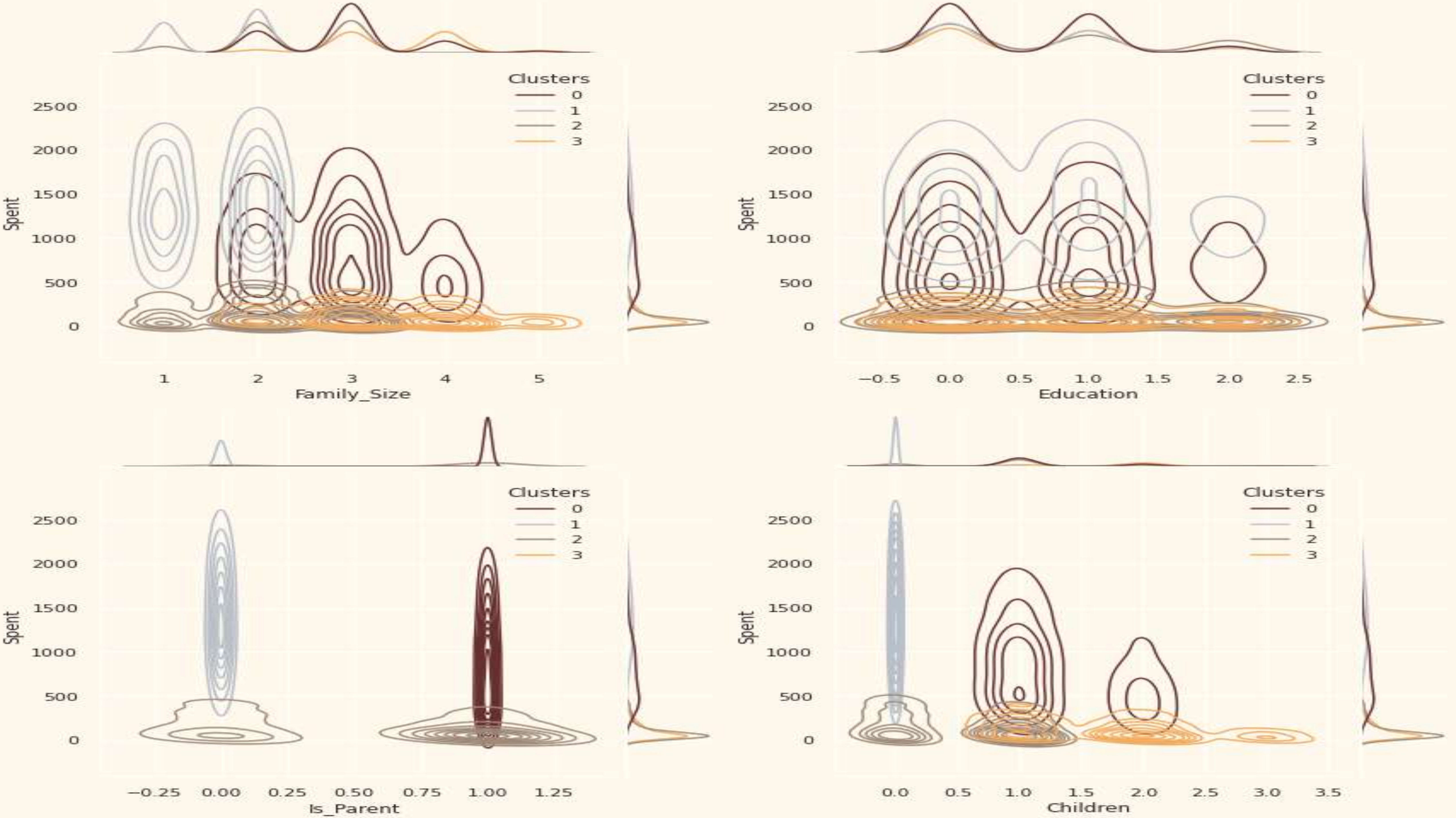
```python
#Plotting the clusters
fig = plt.figure(figsize=(10,8))
ax = plt.subplot(111, projection='3d', label="bla")
ax.scatter(x, y, z, s=40, c=PCA_ds["Clusters"], marker='o', cmap = cmap )
ax.set_title("The Plot Of The Clusters")
plt.show()
```




Distribution Of The Clusters
count
Clusters
0
1
2
3
Spent
Clusters
Count Of Promotion Accepted
Clusters
0
1
2
3
count
Number Of Total Accepted Promotions
Cluster's Profile Based On Income And Spending
Income
Spent
0
1
2
3


Clusters
0
1
2
3
Spent
NumWebVisitsMonth
NumCatalogPurchases
NumStorePurchases
NumWebPurchases


Clusters
0
1
2
3
Spent
Kidhome
Teenhome
1e−18
1e17
Customer_For
Age



Clusters
0
1
2
3
Spent
2500
2000
1500
1000
500
0
1
2
3
4
5
Family_Size
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Education
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Is_Parent
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Children

Evaluating Clustering Performance

87% Silhouette Score

Clustering Accuracy

The accuracy of the clustering process is evaluated to determine how well the identified clusters represent the actual patterns and structures within the data.

Internal Validation

Internal validation metrics, like the Silhouette Score, provide insights into the cohesion and separation of clusters, aiding in the assessment of clustering quality.

Interpreting Cluster Results

Cluster 1	Diverse age group, high purchase frequency
Cluster 2	Young adults, moderate purchase frequency
Cluster 3	Elderly population, low purchase frequency

Conclusion

1

Insights and Recommendations

Based on the interpreted cluster results, businesses can derive actionable insights and formulate targeted marketing strategies to cater to the specific needs and preferences of each customer segment.

2

Future Scope

The process of customer segmentation through unsupervised clustering opens avenues for further analysis, personalization, and enhanced customer-centric approaches in diverse business domains.

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