Segmentation of Customer and Personality Analysis

Name - Samir kumar
Roll No- 2k22/IEM/09
Student of Mtech in Industrial Engineering and Management
Subject - Data Analytics

The problem

Company

The dataset is from a marketing campaign of a e-commerce company.
 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.

Whether their campaign is usefull or not? And which one?

From where there product is purchased?

Customer Personality

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.

DATA TYPES

People

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

Products

1. MntWines: Amount spent on wine in last 2 years
2. MntFruits: Amount spent on fruits in last 2 years
3. MntMeatProducts: Amount spent on meat in last 2 years
4. MntFishProducts: Amount spent on fish in last 2 years
5. MntSweetProducts: Amount spent on sweets in last 2 years
6. MntGoldProds: Amount spent on gold in last 2 years

DATA TYPES

Promotion

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

Place

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

Solution

More premium subscribers

- This is an unsupervised machine learning project where we have to identify customer segments using clustering technique by k-means method.
- We have to find out hidden insights of customer's personal traits based on various clusters.
- The dataset is from a marketing campaign of a company.
- Dataset source: https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis

Implementation
(Steps in the project and its details are given as)

Importing the required libraries

In this stage of project various libraries required for the project are imported, such as PANDAS, NUMPY, MATPLOTLIB, SEABORN, SCIKITLEARN.

Loading

This is a very simple stage of project, in which theS csv file is read.

Data cleaning and feature engineering

This was a very crucial stage of the project as it involved creation of new features from existing features and removing redundant features.

Data preprocessing

Label encoding, Removing redundant columns, Scaling the data before clustering (Standard scaler is used)

EDA on data(Exploratory Data Analysis)

In this part of the project two kinds of plots are used to identify relation of each feature with total_spending target

Dimensionality reduction

Even after removing redundant columns, still there are many columns.To cluster the data and visualization purpose PCA is used. (Priciple Component Analysis)

Clustering of data

Clustering methods are one of the most useful unsupervised ML methods. These methods are used to find similarity as well as the relationship patterns among data samples and then cluster those samples into groups having similarity based on features.

Customer personality analysis based on clusters found

Personality Analysis of different clusters.

Conclusion

Brief Summery of the data set for analysis in product management

Importing the required libraries

In this stage of project various libraries required for the project are imported, such as

- PANDAS, (data manipulation and analysis)
- NUMPY,(It provides a multidimensional array object)
- MATPLOTLIB, (Data visualization)
- SEABORN,(**Seaborn** helps you explore and understand your data)
- SCIKITLEARN.(ML Concept)

```
In [77]: import pandas as pd
         import numpy as np
         import seaborn as sns
         import matplotlib.pyplot as plt
         import warnings
         warnings.filterwarnings("ignore")
```

Loading the data

This is a very simple stage of project, in which the csv file is read.

1. Although it's simple file to read, after loading it in notebook it was found that the file is a tab separated one and not a comma separated.
2. Just by specifying sep = "\t" the file was correctly loaded into the Jupyter notebook.

```
In [78]: path = r'E:/kaggle/Customer_segmentation/marketing_campaign.csv'
```

```
In [79]: df = pd.read_csv(path,sep="\t")
```

Data cleaning

Handling missing values

```
In [84]: df.shape

Out[84]: (2240, 28)

In [85]: df=df.dropna()

In [86]: df.shape

Out[86]: (2216, 28)
```

Data Types

```
In [88]: df.dtypes

Out[88]: Year_Birth               int64
         Education               object
         Marital_Status          object
         Income                 float64
         Kidhome                  int64
         Teenhome                 int64
         Dt_Customer             object
         Recency                  int64
         MntWines                 int64
         MntFruits                int64
         MntMeatProducts          int64
         MntFishProducts          int64
         MntSweetProducts         int64
         MntGoldProds             int64
         NumDealsPurchases        int64
         NumWebPurchases          int64
         NumCatalogPurchases      int64
         NumStorePurchases        int64
         NumWebVisitsMonth        int64
         AcceptedCmp3             int64
         AcceptedCmp4             int64
         AcceptedCmp5             int64
         AcceptedCmp1             int64
         AcceptedCmp2             int64
         Complain                 int64
         Z_CostContact            int64
         Z_Revenue                int64
         Response                 int64
         dtype: object
```

From Uniqueness in coloumns. We Found that _CostContact and Z_Revenue is having only 1 unique values so we need to drop these columns

```
In [89]: df.nunique()

Out[89]: Year_Birth               59
         Education                 5
         Marital_Status            8
         Income                 1974
         Kidhome                   3
         Teenhome                  3
         Dt_Customer             662
         Recency                 100
         MntWines                776
         MntFruits               158
         MntMeatProducts         554
         MntFishProducts         182
         MntSweetProducts        176
         MntGoldProds            212
         NumDealsPurchases        15
         NumWebPurchases          15
         NumCatalogPurchases      14
         NumStorePurchases        14
         NumWebVisitsMonth        16
         AcceptedCmp3              2
         AcceptedCmp4              2
         AcceptedCmp5              2
         AcceptedCmp1              2
         AcceptedCmp2              2
         Complain                  2
         Z_CostContact             1
         Z_Revenue                 1
         Response                  2
         dtype: int64
```

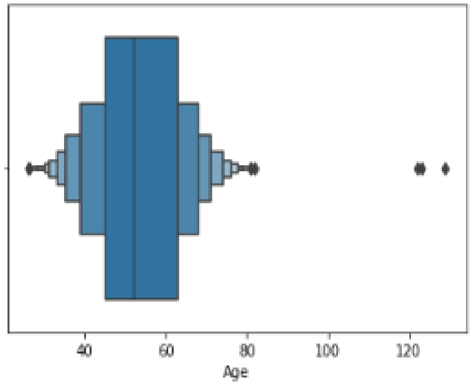
Data cleaning and feature engineering

This was a very crucial stage of the project as it involved creation of new features from existing features and removing redundant features.

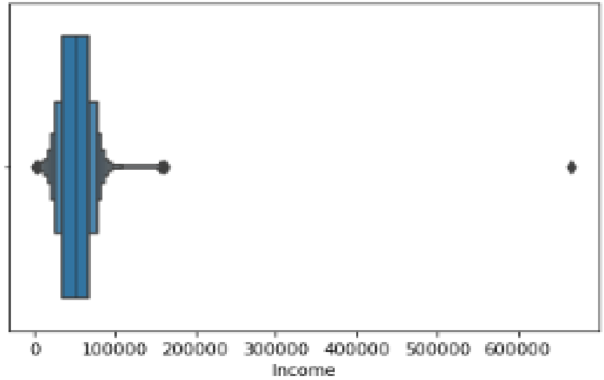
1. Converted birth year to age column
2. From Uniqueness in coloumns. We Found that _CostContact and Z_Revenue is having only 1 unique values so we need to drop these columns
3. Drop the missing rows in the column(From - (2240, 28) , to - (2216, 28))
4. Instead of multiple degree(Education) converted them all into either undergraduate, graduate or postgraduate(i.e Graduation 1116, PhD 481 ,Master 365 ,2n Cycle 200 ,Basic 54)
5. Converted Married, Together, Divorced, Widow, Single into either Partner or Alone.
6. Using teenhome and kidhome created a new column as children count.
7. Created a column family size using (living_with) column and children count
8. Date time conversion of dt_customer column
9. All the spending on wine, fruit, gold etc. are summed up and put into new column total_spending
10. A new column is created for customer duration by subtracting each date from newest date.
11. Removed outliers from Age (few people were having age > 120 years) and income (few people were having income > 600000)

There are some outliers looking in Income as well as in age because max age is 129

In [107]:
```
sns.boxenplot(df["Age"])
```
Out[107]: <AxesSubplot:xlabel='Age'>



In [108]:
```
sns.boxenplot(df["Income"])
```
Out[108]: <AxesSubplot:xlabel='Income'>



In [109]:
```
filt = ( df["Age"] <100 ) & (df["Income"] <600000  )
df=df.loc[filt]
```

In [110]:
```
df=df.reset_index().drop("index",1)
```

12. Now dropping unnecessary columns like Year_Birth","Marital_Status","duration","Date","Dt_Customer“.

Data preprocessing

1. Label encoding for Education column and living_with column
2. Removing redundant columns
3. Scaling the data before clustering (Standard scaler is used)

Label encoding

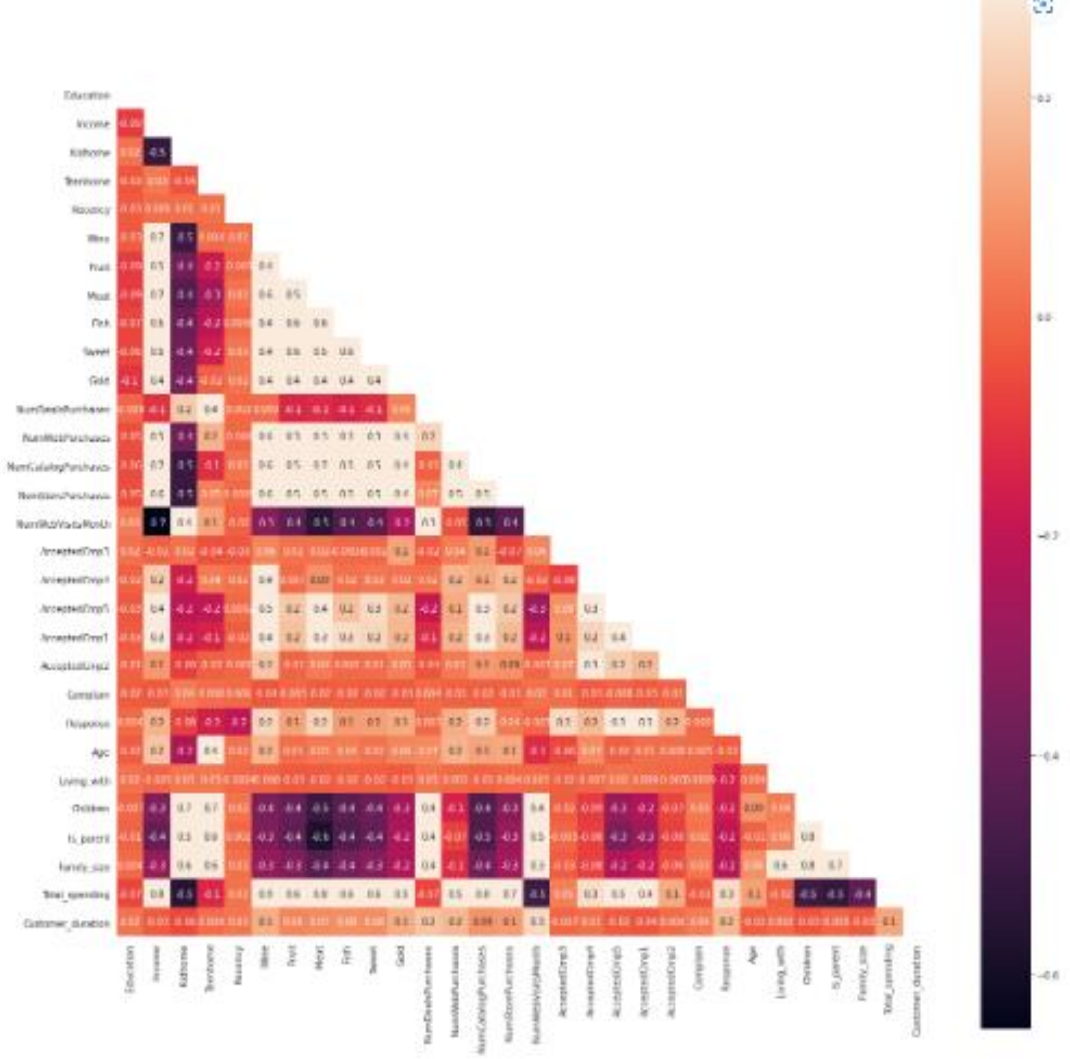
```
In [113]: from sklearn.preprocessing import LabelEncoder

In [114]: enc1 = LabelEncoder()
          enc2 = LabelEncoder()

In [115]: df["Education"] = enc1.fit_transform(df["Education"])
          df["Living_with"] = enc2.fit_transform(df["Living_with"])

In [116]: df.info()
```

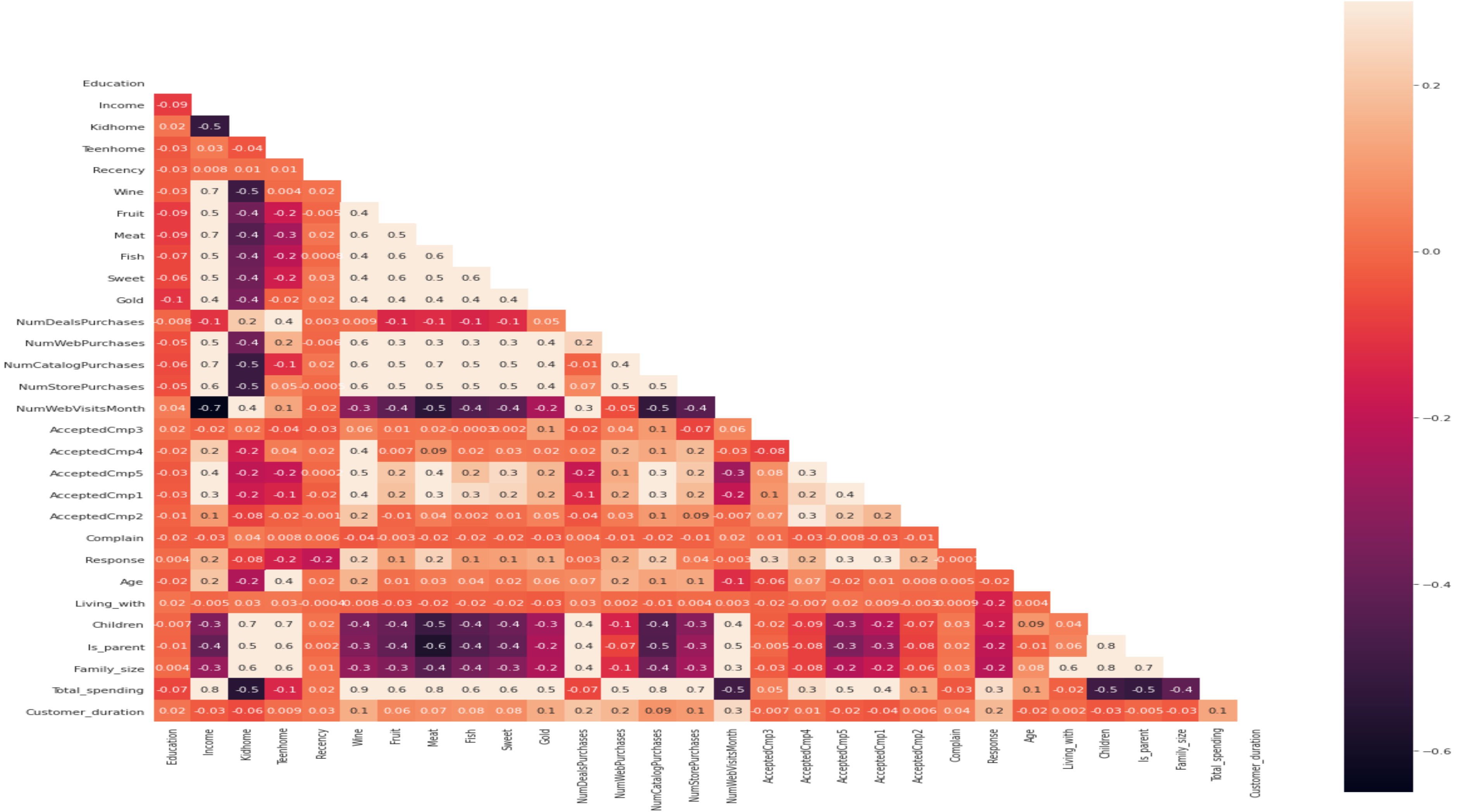
```
In [136]: from sklearn.preprocessing import StandardScaler

In [137]: scaler = StandardScaler()
```

Let's check for correlation

After this , Dropping Deals and response columns

remove_cols = ['AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5',
'AcceptedCmp1', 'AcceptedCmp2', 'Complain', 'Response']





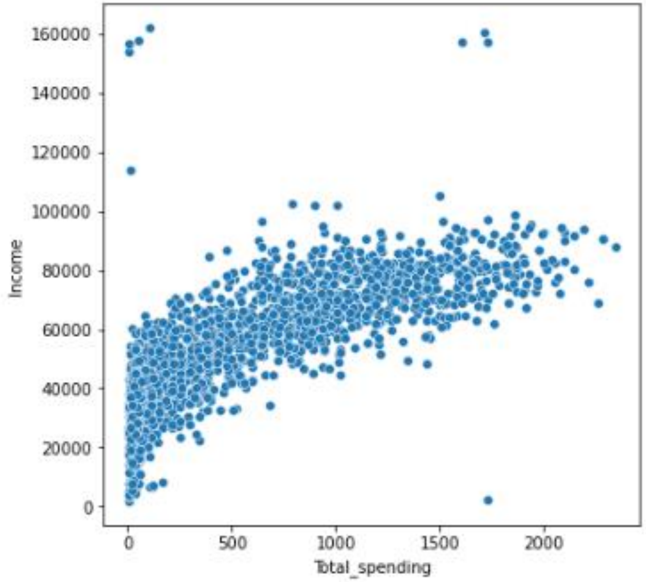
Education
Income
Kidhome
Teenhome
Recency
Wine
Fruit
Meat
Fish
Sweet
Gold
NumDealsPurchases
NumWebPurchases
NumCatalogPurchases
NumStorePurchases
NumWebVisitsMonth
AcceptedCmp3
AcceptedCmp4
AcceptedCmp5
AcceptedCmp1
AcceptedCmp2
Complain
Response
Age
Living_with
Children
Is_parent
Family_size
Total_spending
Customer_duration
0.2
0.0
−0.2
−0.4
−0.6

Let's do some **EDA(Exploratory Data Analysis)**

We will be plotting data with respect to total spending and only relevant feature plotting will be done.
For continuous data we will be using scatter plot and for categorical data bar plot

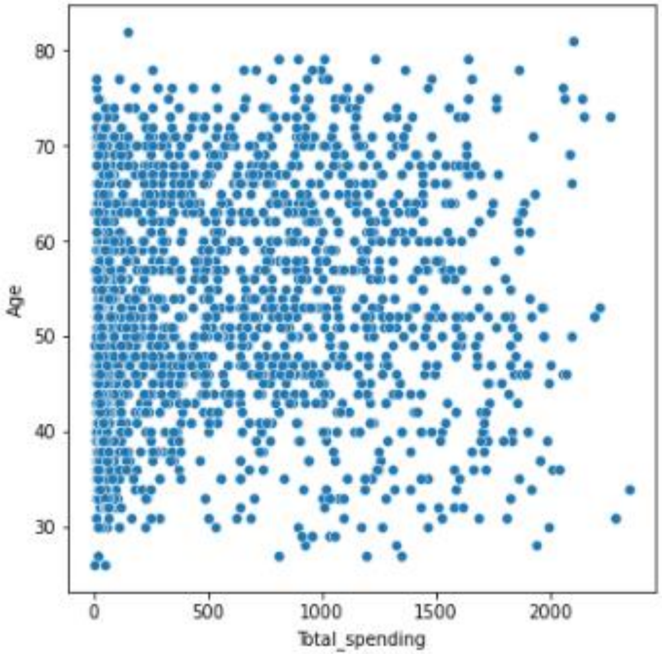
Scatter plot

```
In [125]: plt.figure(figsize=(6,6))
          sns.scatterplot(data=cust,x="Total_spending",y="Income")
          plt.show()
```



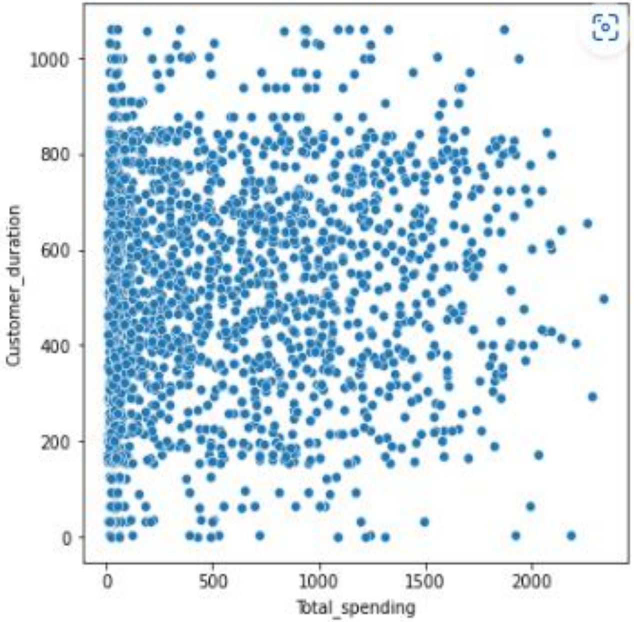
There is linear relationship present here

In [126]:
```
plt.figure(figsize=(6,6))
sns.scatterplot(data=cust,x="Total_spending",y="Age")
plt.show()
```



No linear linearship

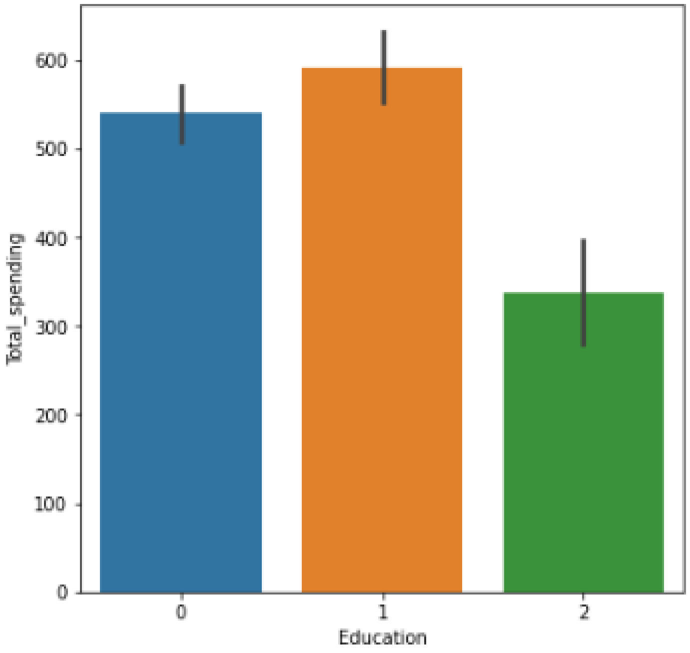
In [127]:
```
plt.figure(figsize=(6,6))
sns.scatterplot(data=cust,x="Total_spending",y="Customer_duration")
plt.show()
```



No linear relationship

Barplot

```
In [129]: plt.figure(figsize=(6,6))
          sns.barplot(data=cust,x="Education",y="Total_spending")
          plt.show()
```



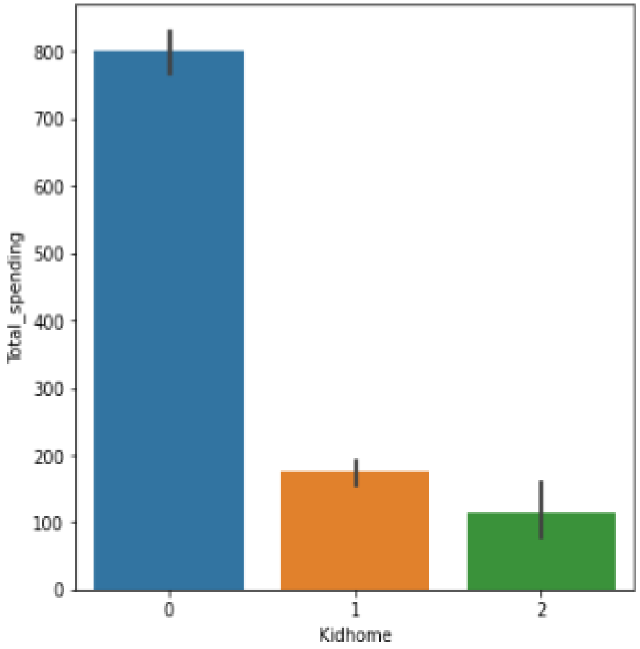
0: Graduate, 1: Postgraduate, 2:Undergraduate

Note:-
----> Undergraduate spends less compared to others

Barplot

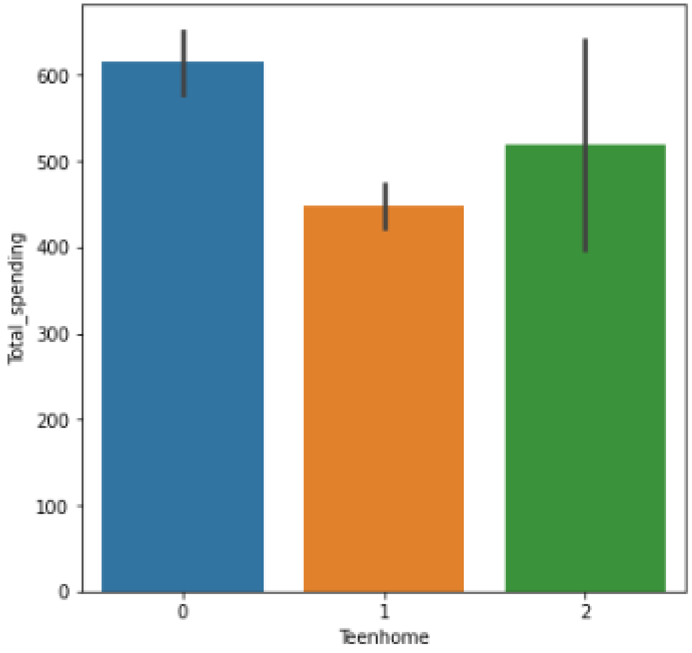
Note:-
Customer with no “kid_home” tends to spend more than others

In [130]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=cust,x="Kidhome",y="Total_spending")
plt.show()
```

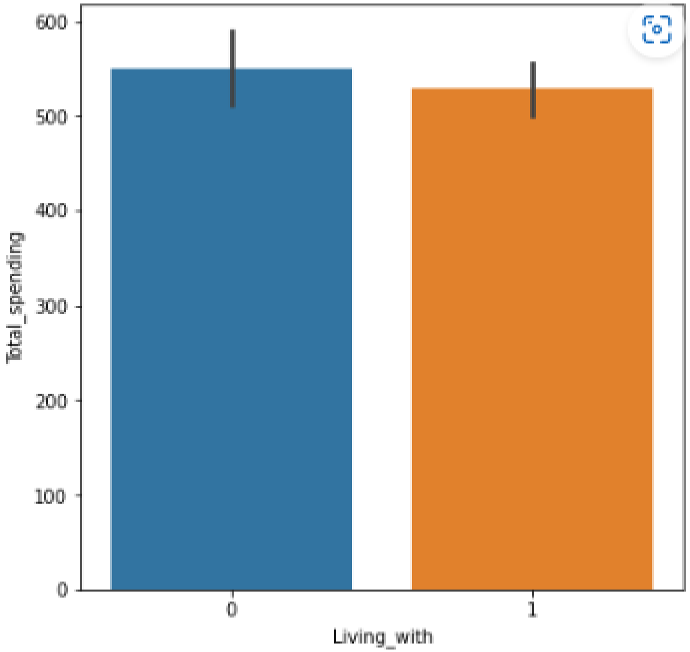


Customer with no kid_home tends to spend more than others

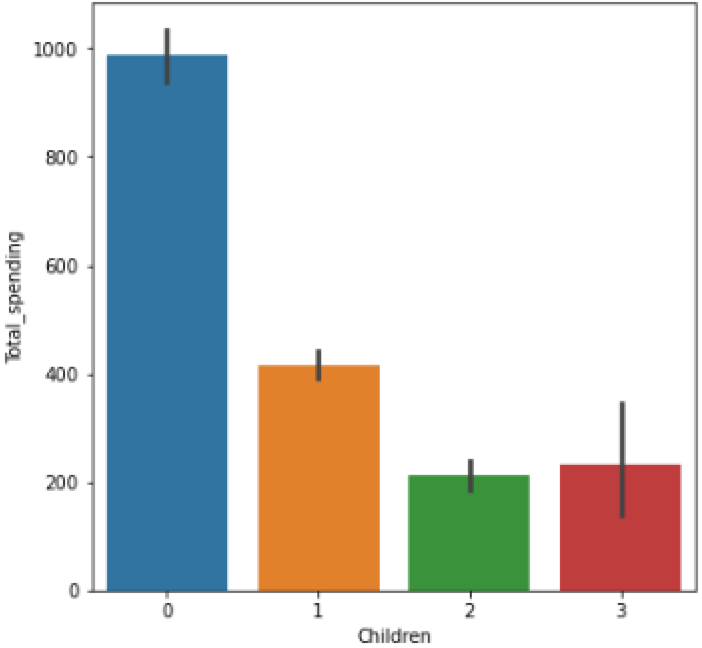
In [131]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=cust,x="Teenhome",y="Total_spending")
plt.show()
```



In [132]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=cust,x="Living_with",y="Total_spending")
plt.show()
```

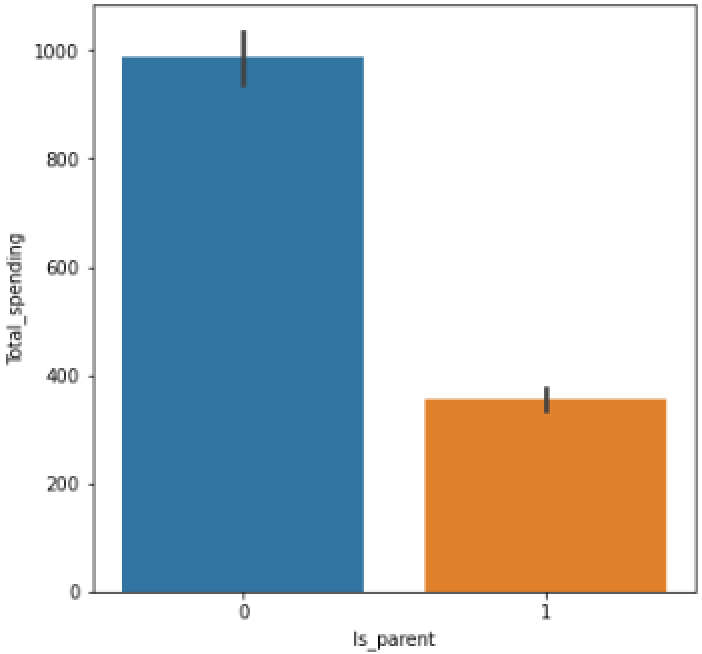


```python
In [133]: plt.figure(figsize=(6,6))
sns.barplot(data=cust,x="Children",y="Total_spending")
plt.show()
```

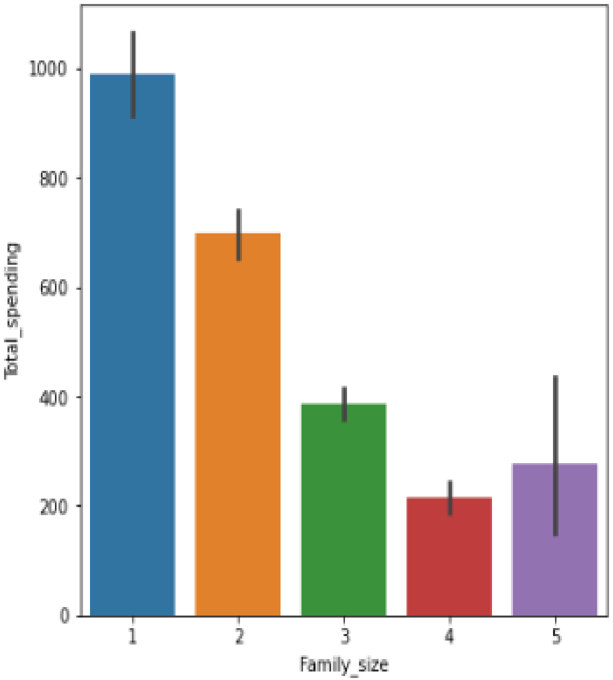


Note: -
Customer with no children spends more than others

In [134]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=cust,x="Is_parent",y="Total_spending")
plt.show()
```



In [135]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=cust,x="Family_size",y="Total_spending")
plt.show()
```



Same conclusion as for children

PCA(Principle Component Analysis)

There are many columns and many of them are correlated with each other so lets do dimentionality reduction before finding clusters

1. Doing PCA before clustering analysis is also useful for dimensionality reduction as a feature extractor and visualize / reveal clusters.
2. Doing PCA after clustering can validate the clustering algorithm (reference: Kernel principal component analysis).
3. PCA is sometimes applied to reduce the dimensionality of the dataset prior to clustering.

In [173]:
```
from sklearn.decomposition import PCA

#keeping 80% of explained variance
pca = PCA(n_components=0.8)
```

In [141]:
```
pca.fit_transform(scaled_cust).shape
```

Out[141]:
```
(2212, 9)
```

In [142]:
```
scaled_pca = pd.DataFrame(pca.fit_transform(scaled_cust),columns=["PCA1","PCA2","PCA3","PCA4","PCA5","PCA6","PCA7","PCA8","PCA9"]
```

Clustering

- It is basically a type of unsupervised learning method.
- An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

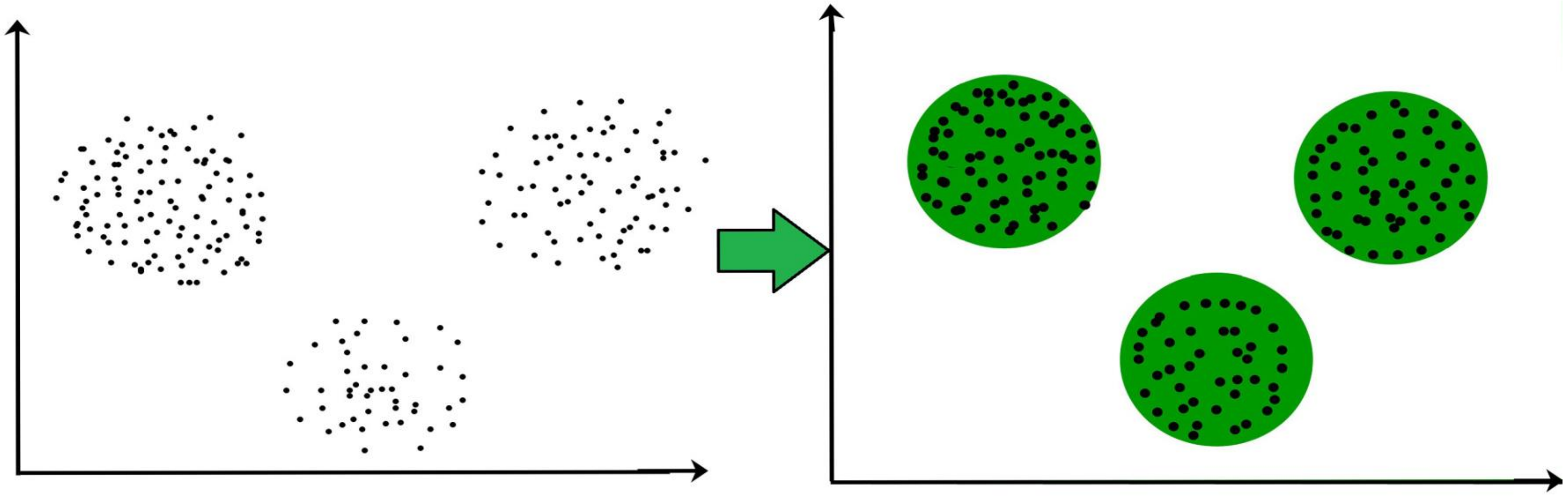
Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

What is Clustering?

"**Clustering** is the process of dividing the datasets into groups, consisting of similar data-points"

- Points in the same group are as similar as possible
- Points in different group are as dissimilar as possible

For ex– The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.



K-means

It is the simplest unsupervised learning algorithm that solves clustering problem.K-means algorithm partitions n observations into k clusters where each observation belongs to the cluster with the nearest mean serving as a prototype of the cluster.

Advantages of k-means

- Relatively simple to implement.
- Scales to large data sets.
- Can warm-start the positions of centroids.
- Easily adapts to new examples.
- Generalizes to clusters of different shapes and sizes, such as elliptical clusters. as elliptical clusters.

Disadvantages of K-means

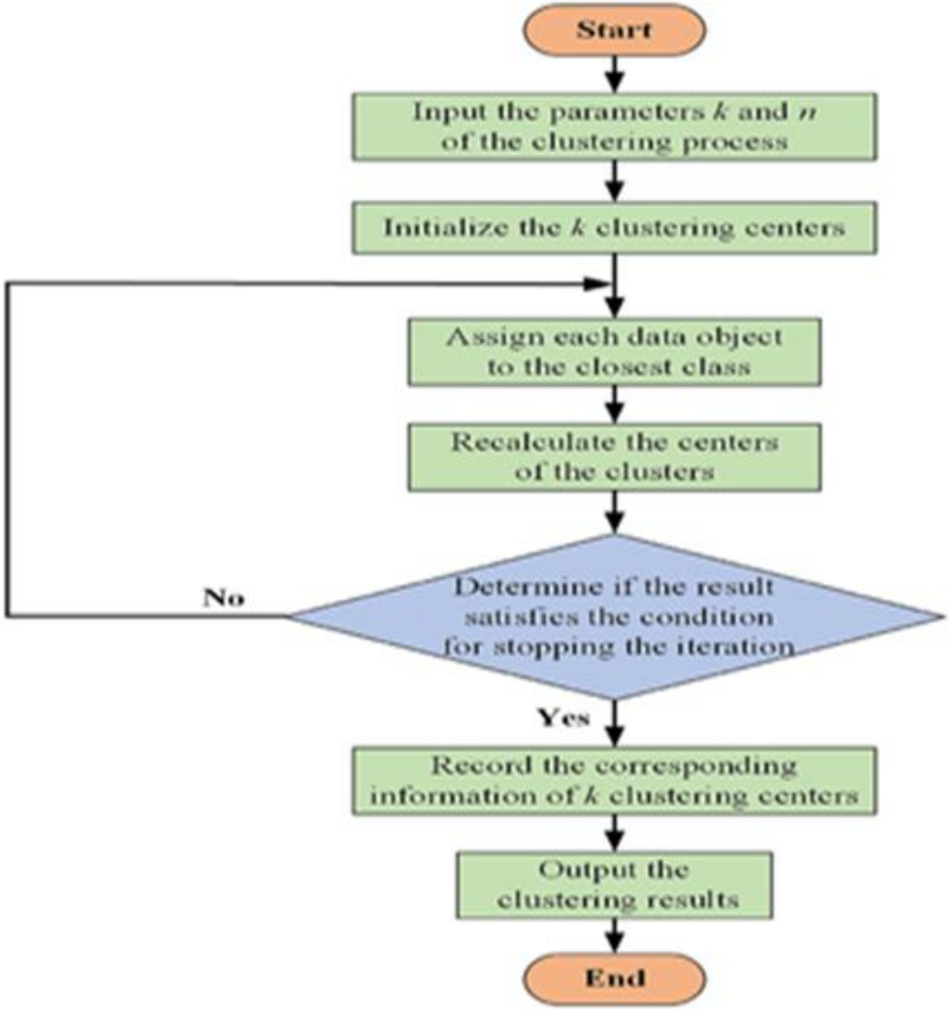
- It is sensitive to the outliers.
- Choosing the k values manually is a tough job.
- As the number of dimensions increases its scalability decreases.

edureka!

What is K-Means Clustering?

Pile of dirty clothes

Algorithm



Start
Input the parameters k and n of the clustering process
Initialize the k clustering centers
Assign each data object to the closest class
Recalculate the centers of the clusters
Determine if the result satisfies the condition for stopping the iteration
No
Yes
Record the corresponding information of k clustering centers
Output the clustering results
End

Elbow Method

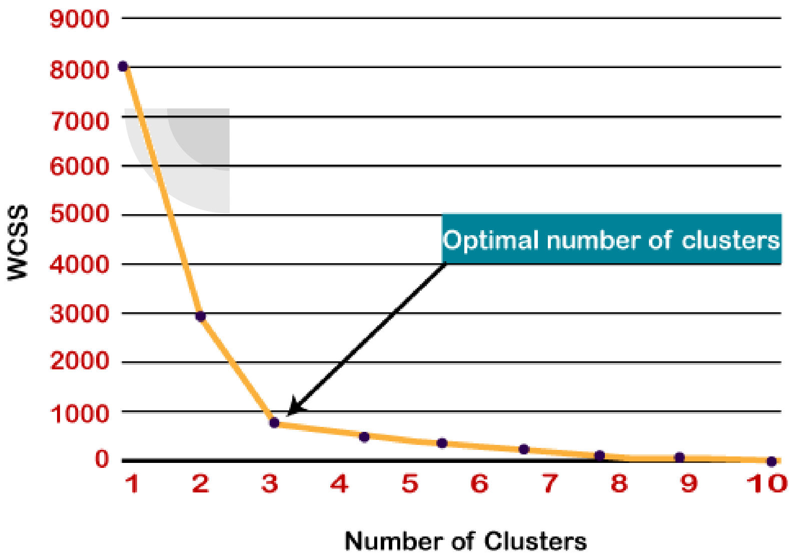
The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

WCSS= ∑Pi in Cluster1 distance(Pi C1)2 +∑Pi in Cluster2distance(Pi C2)2+∑Pi in CLuster3 distance(Pi C3)2

In the above formula of WCSS,

∑Pi in Cluster1 distance(Pi C1)2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.



To find the optimal value of clusters, the elbow method follows the below steps:

1. It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
2. For each value of K, calculates the WCSS value.
3. Plots a curve between calculated WCSS values and the number of clusters K.

The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

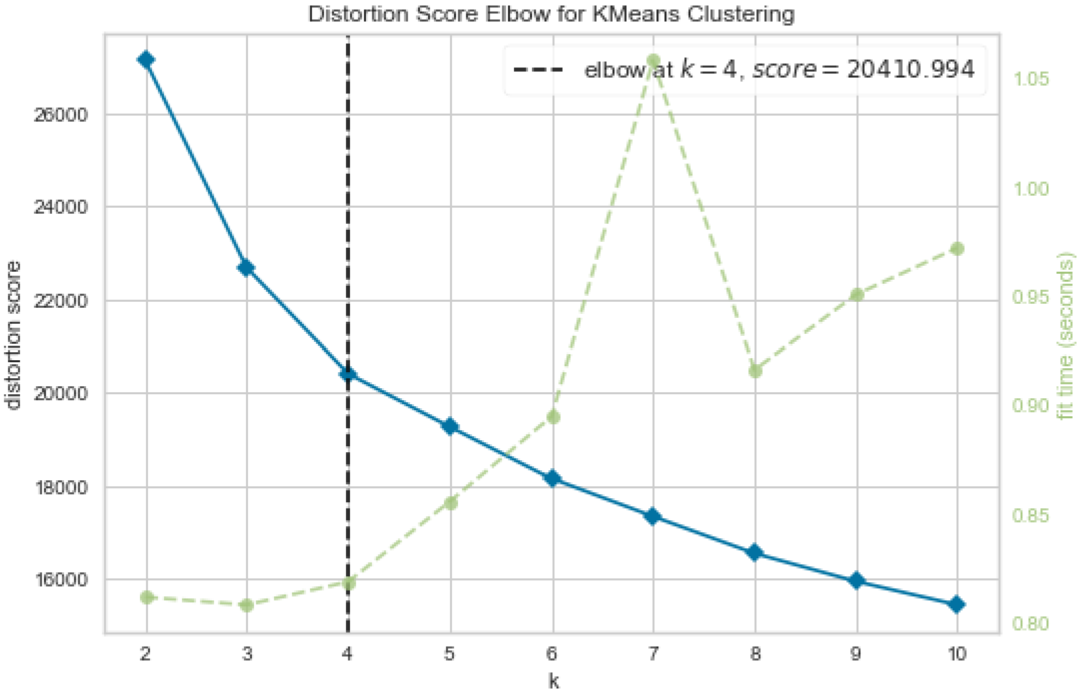
Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:

```
In [149]: cls_data_1 =scaled_pca[["PCA1","PCA2","PCA3","PCA4","PCA5","PCA6","PCA7","PCA8","PCA9"]]
```

```
In [150]: from yellowbrick.cluster import KElbowVisualizer
          from sklearn.cluster import KMeans
```

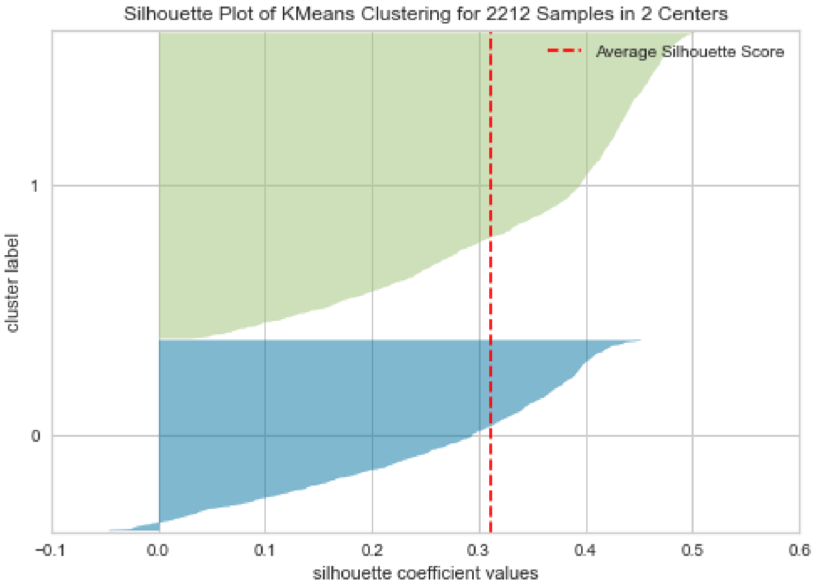
```
In [151]: elbow = KElbowVisualizer(KMeans(), k=10)
          elbow.fit(cls_data_1)
          elbow.show()
          plt.show()
```

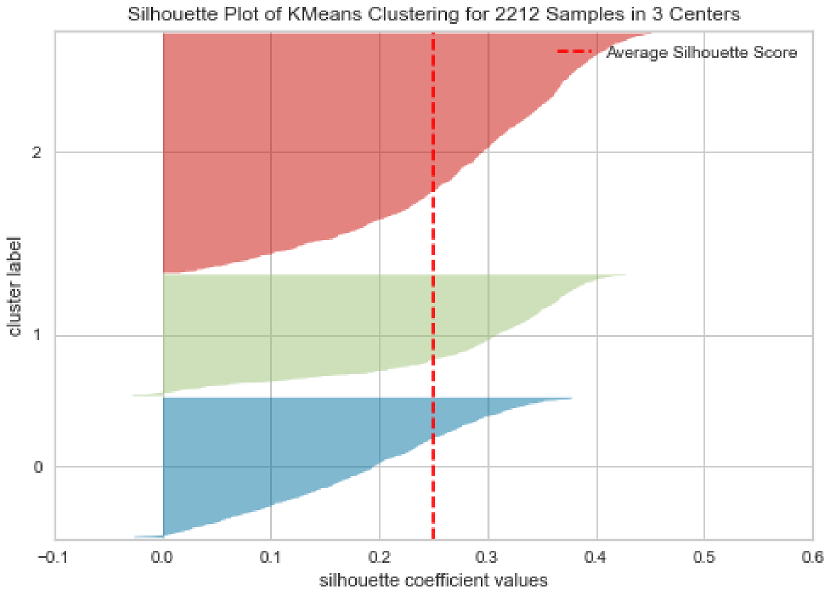
Let's do clustering

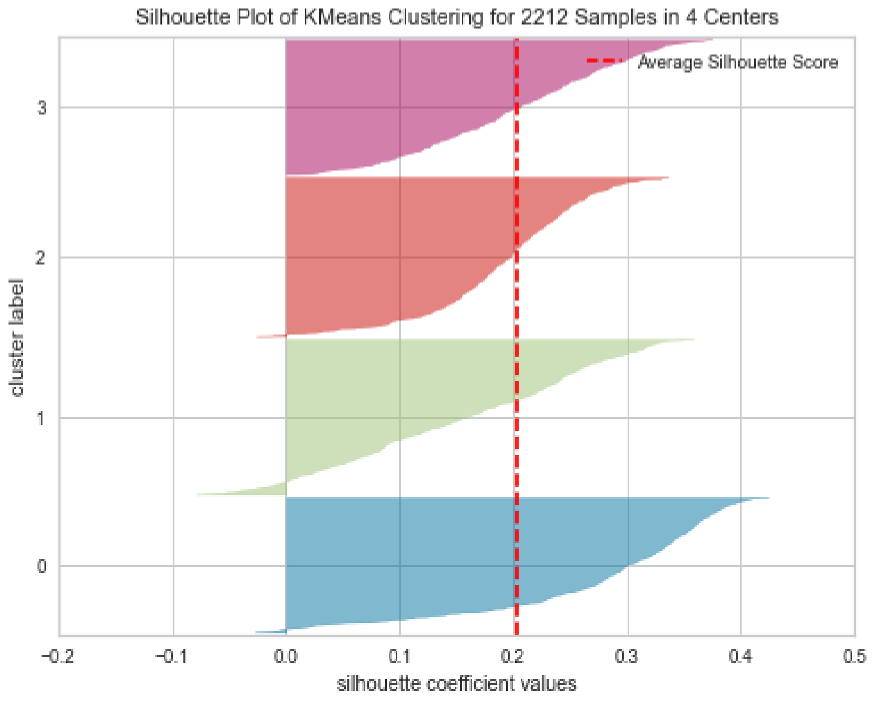


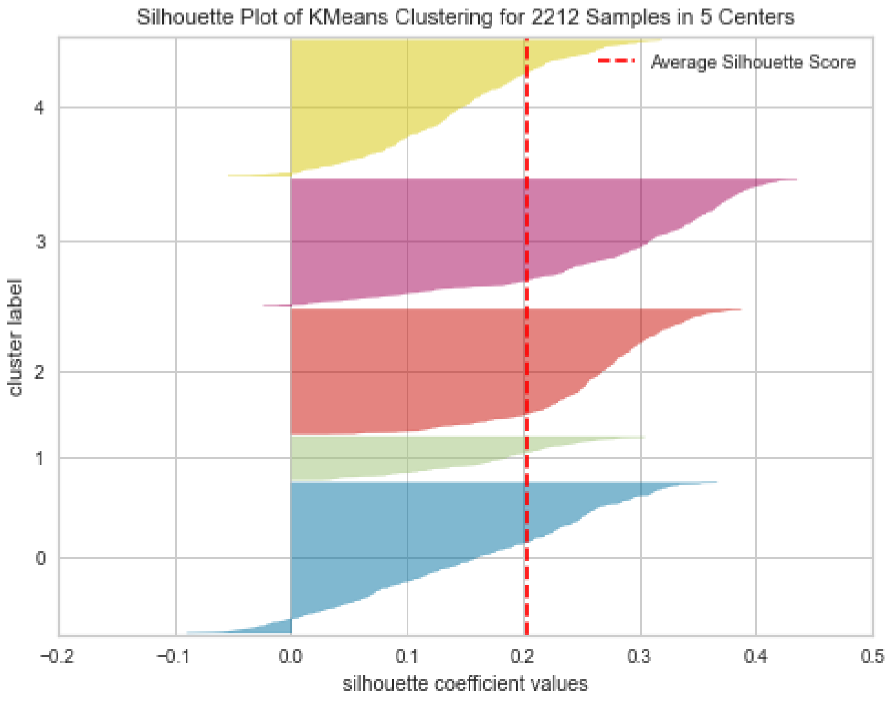
Automated library for elbow plot says that 4 clusters are best and visually also k = 4 looks good

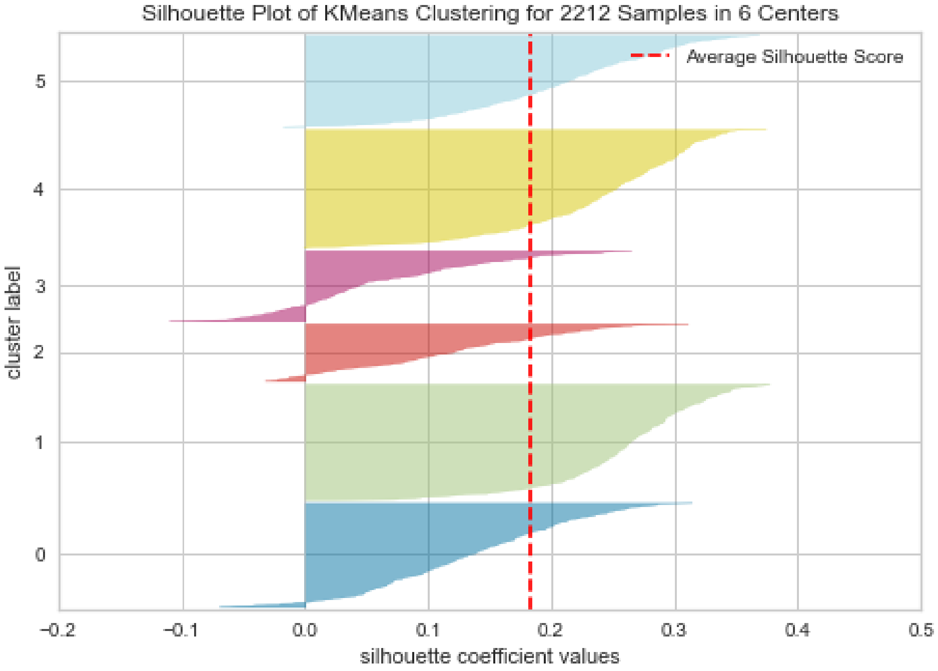
Let's check for silhoutte score plot



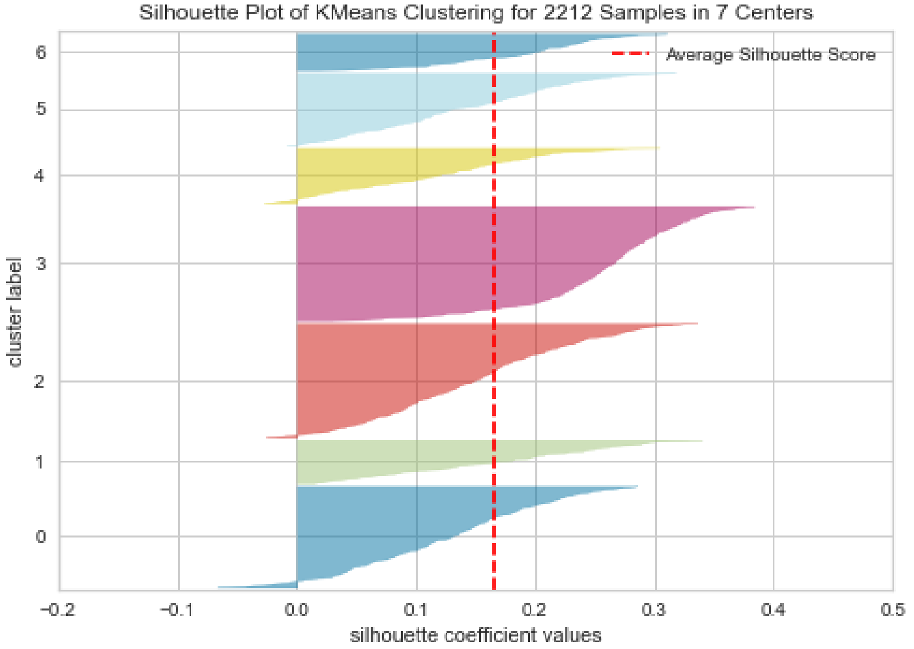



Silhouette Plot of KMeans Clustering for 2212 Samples in 4 Centers
Average Silhouette Score
cluster label
3
2
1
0
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
silhouette coefficient values

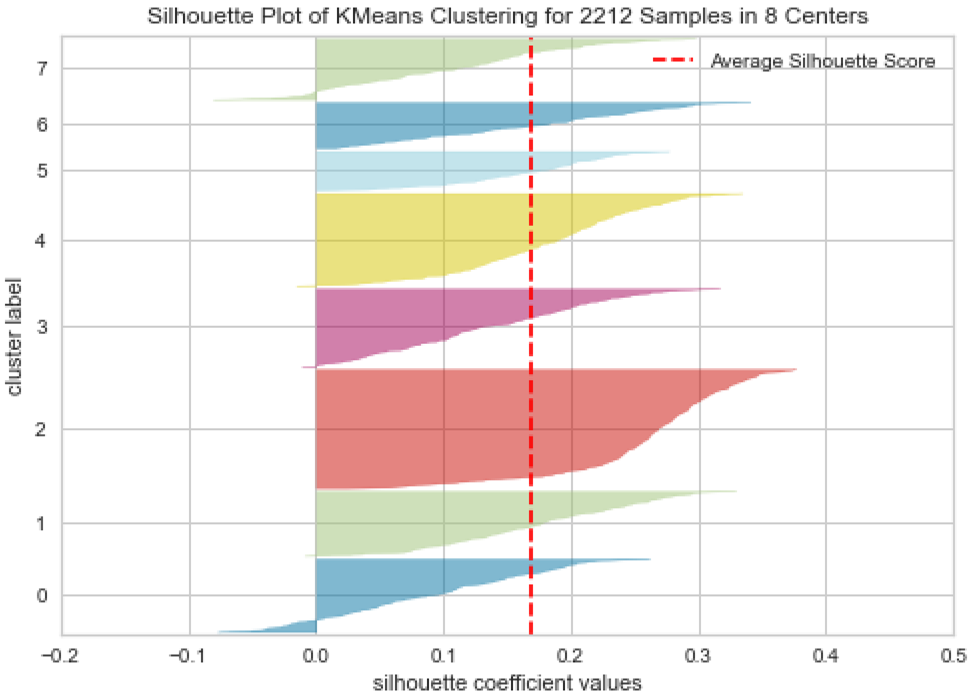

Silhouette Plot of KMeans Clustering for 2212 Samples in 5 Centers
Average Silhouette Score
cluster label
4
3
2
1
0
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
silhouette coefficient values



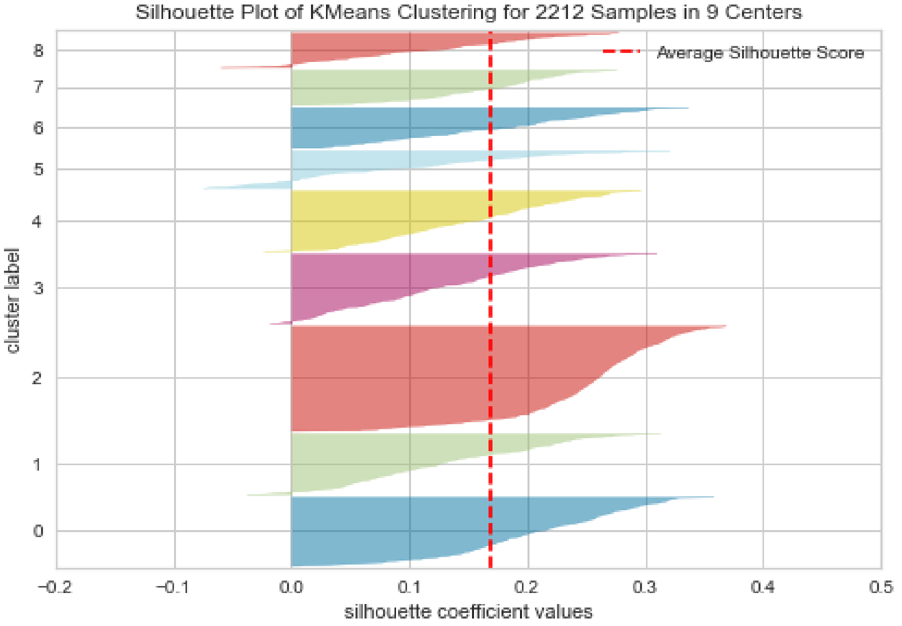
Silhouette Plot of KMeans Clustering for 2212 Samples in 6 Centers
Average Silhouette Score
cluster label
silhouette coefficient values
0
1
2
3
4
5
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5



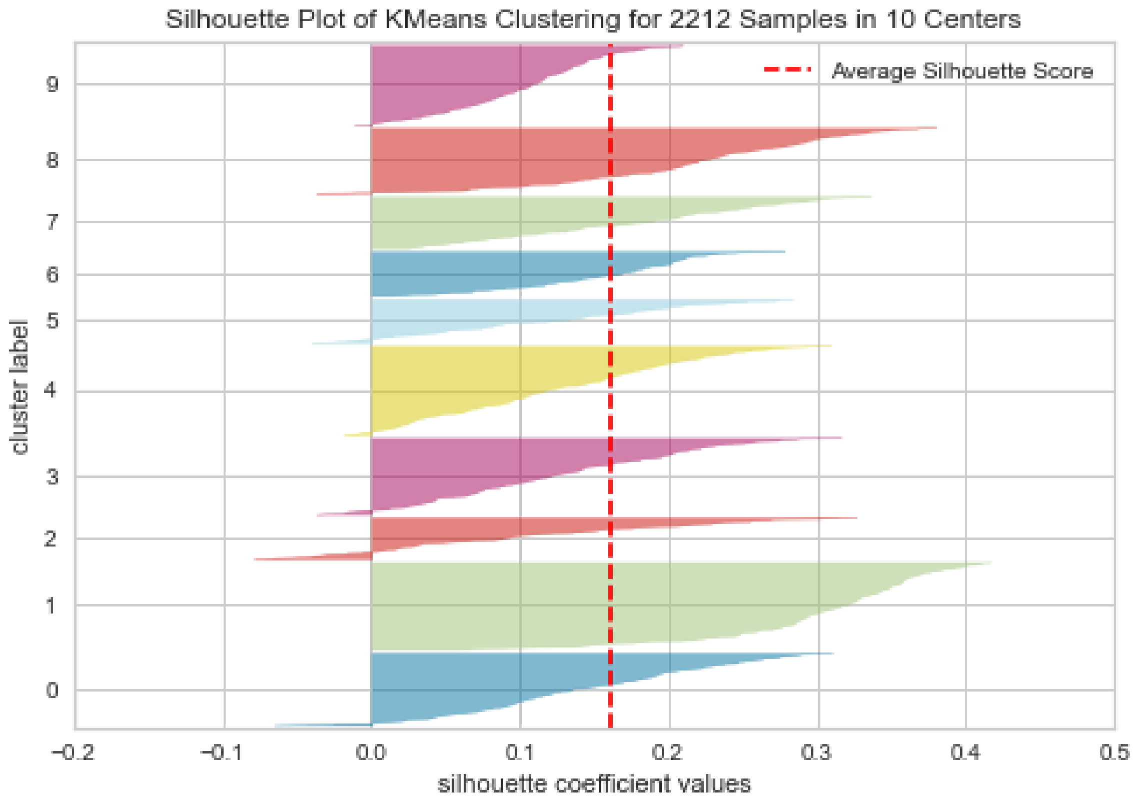
Silhouette Plot of KMeans Clustering for 2212 Samples in 7 Centers
Average Silhouette Score
cluster label
silhouette coefficient values
0
1
2
3
4
5
6
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5



Silhouette Plot of KMeans Clustering for 2212 Samples in 8 Centers
Average Silhouette Score
cluster label
silhouette coefficient values



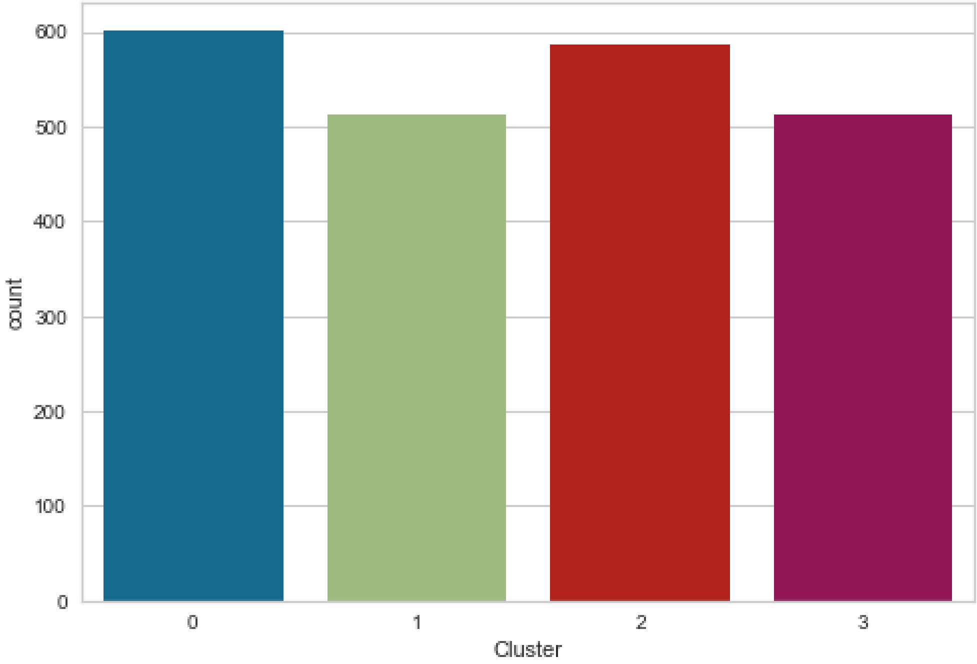
Silhouette Plot of KMeans Clustering for 2212 Samples in 9 Centers
Average Silhouette Score
cluster label
silhouette coefficient values



For k=3 the cluster 2 looks big, The cluster size looks similar for all clusters when k = 4.

Let's check cluster size

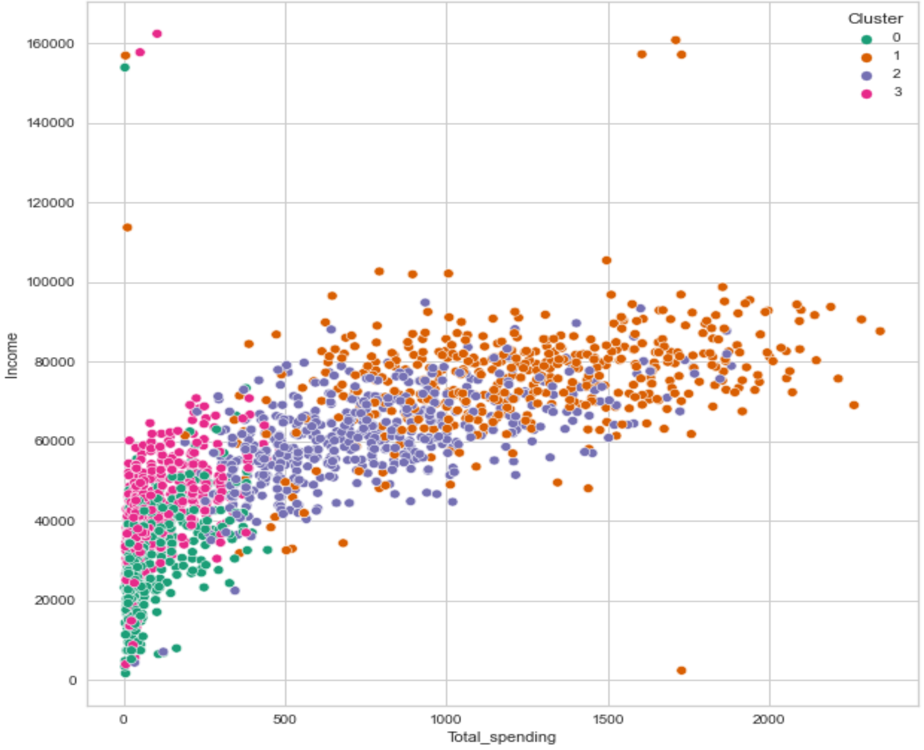
Cluster size for all clusters, more or less looks similar



Let's check clusters on income and spent scatter plot

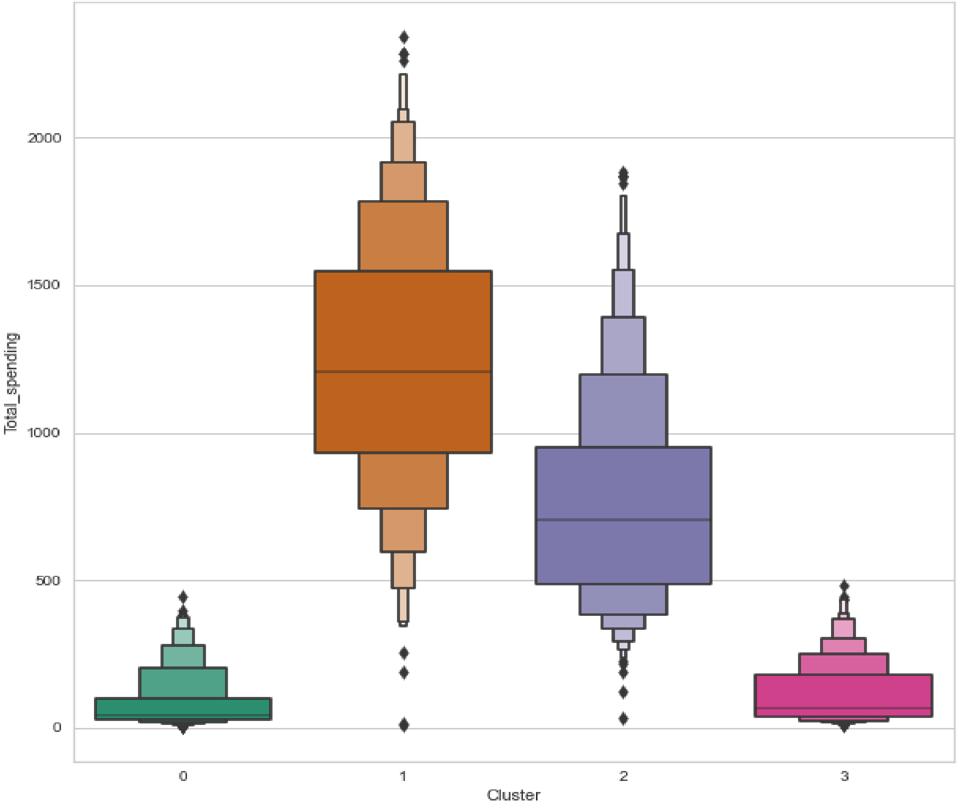
This plot reveals that there are Majorly four customer segments

0: Low income low spending
1: High income high spending
2: Medium income medium-high spending
3: Medium income low spending



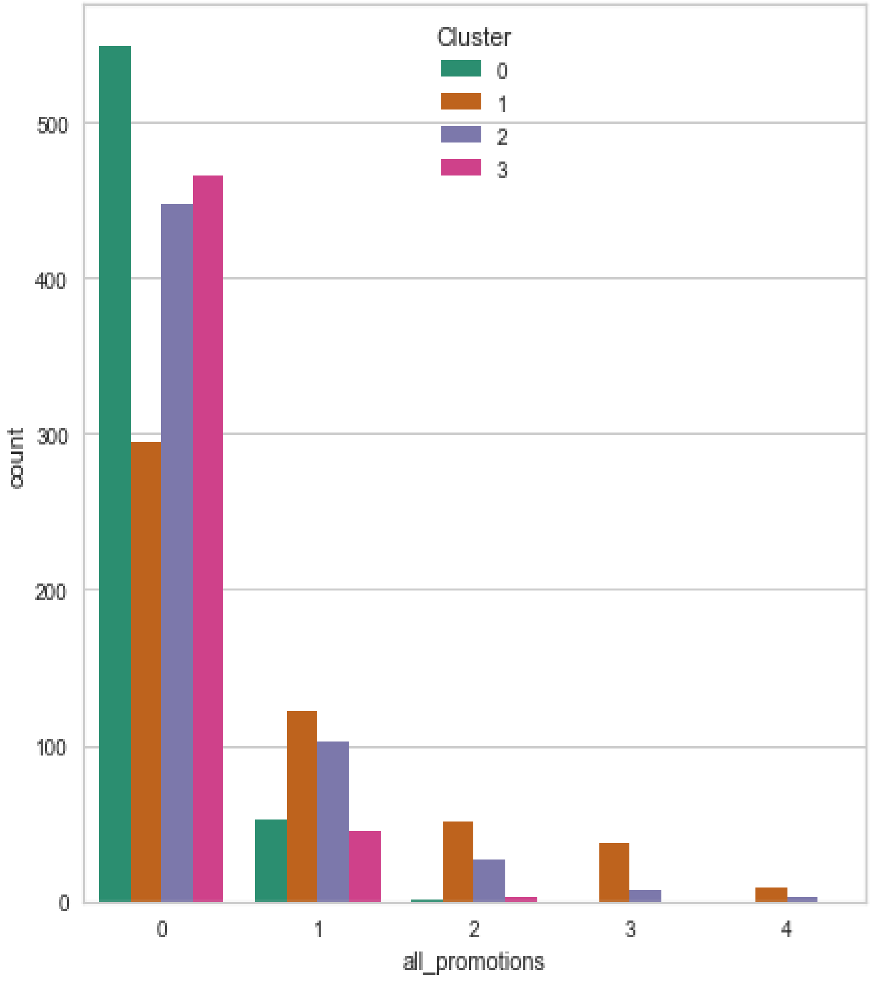
Now let's look at spending volume for each cluster

The above plot shows that cluster 1 is biggest customer segment for us and cluster 2 is second biggest customer segment



Promotions analysis

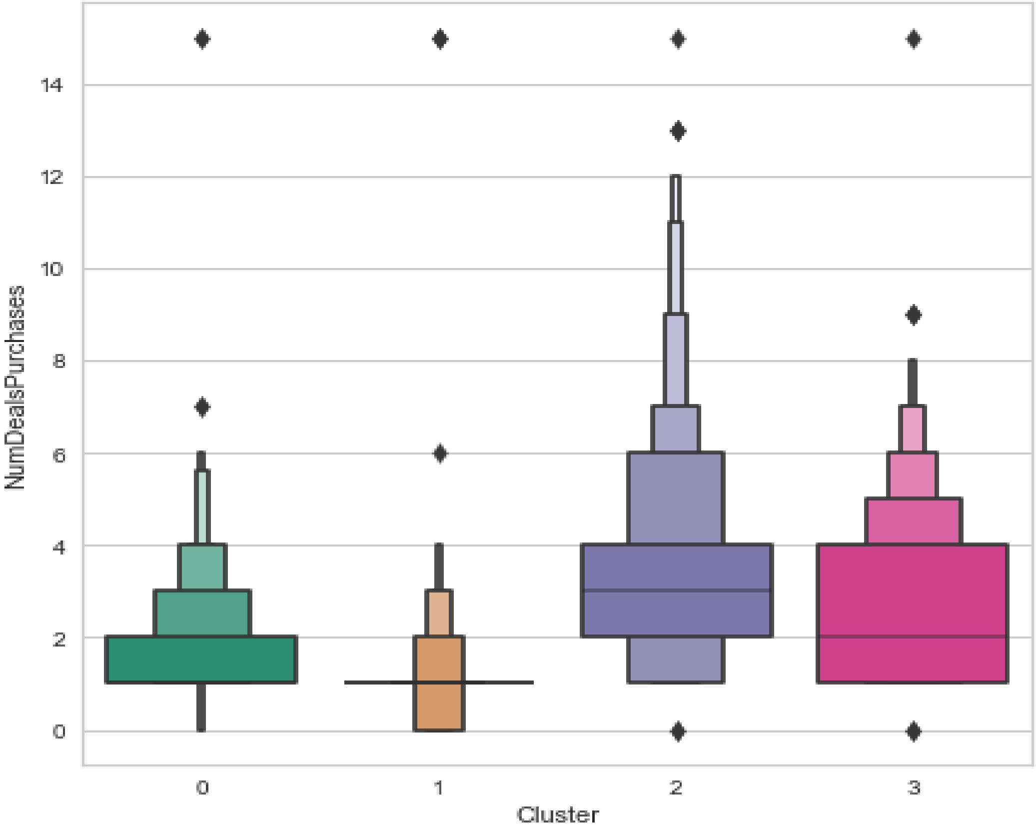
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise



Note:- The promotion acceptance is very low for all the clusters

Now checkling purchase analysis

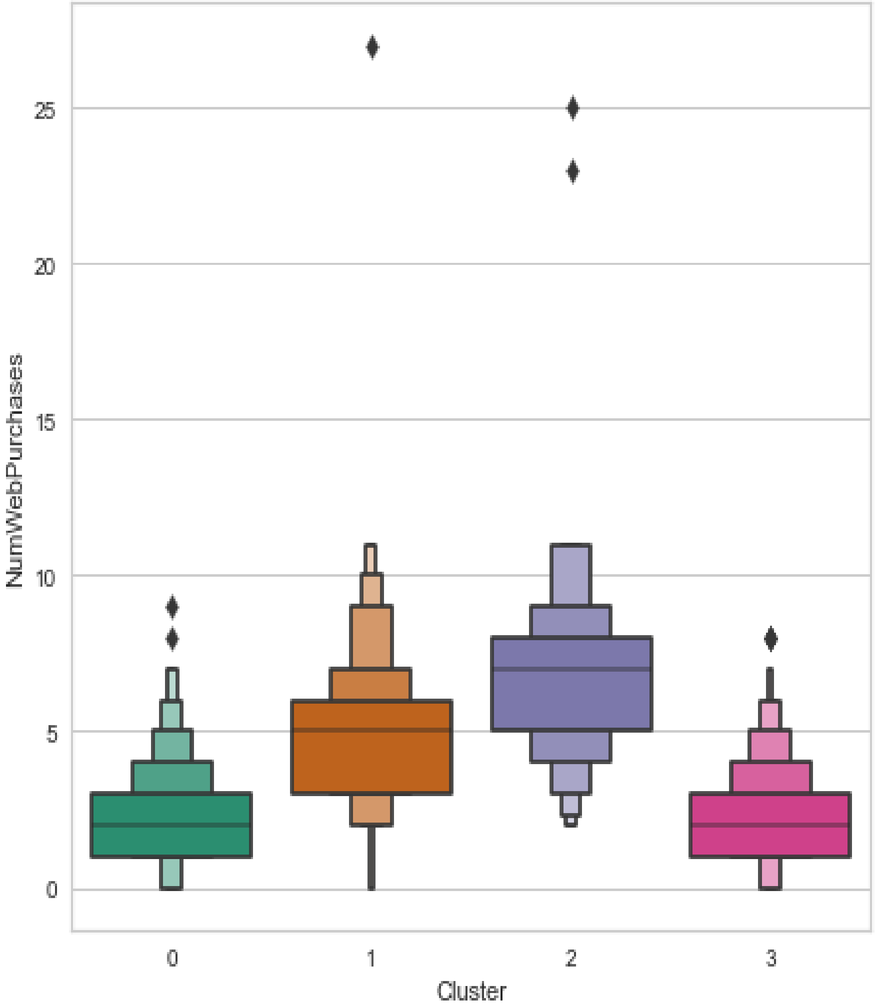
NumDealsPurchases: Number of purchases made with a discount



Note :- Most deals are purchased by cluster 2, followed by cluster 3

Web Purchases

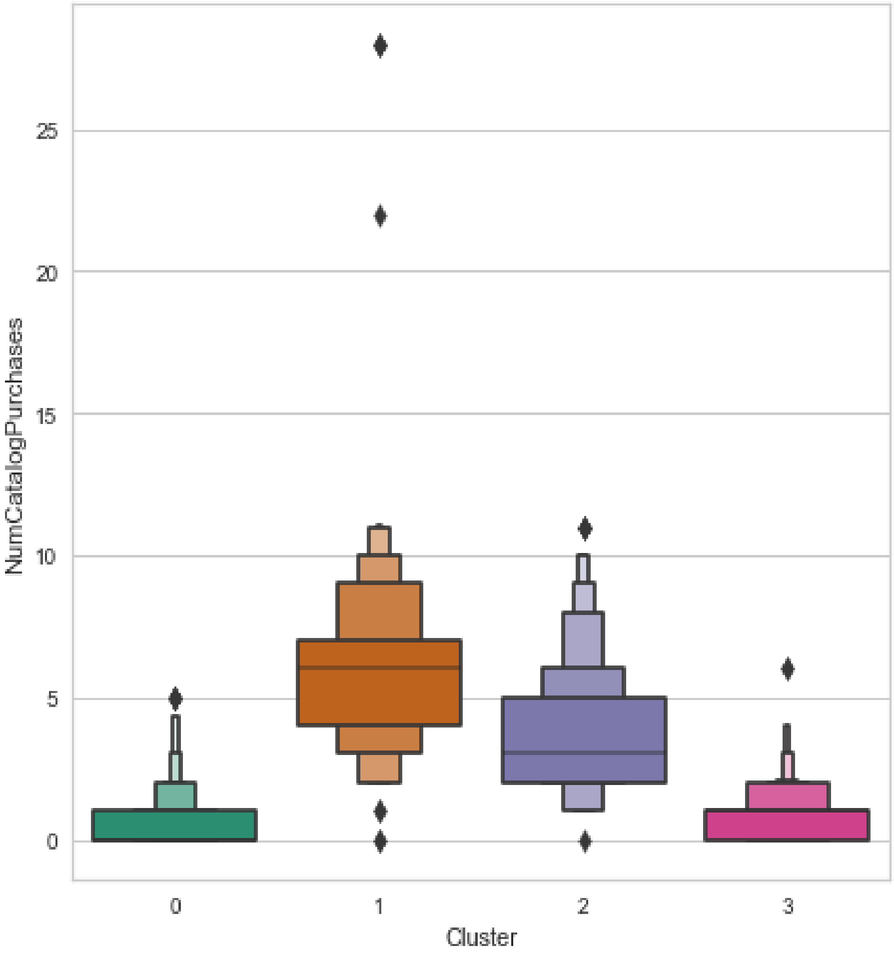
NumWebPurchases: Number of purchases made through the company's website



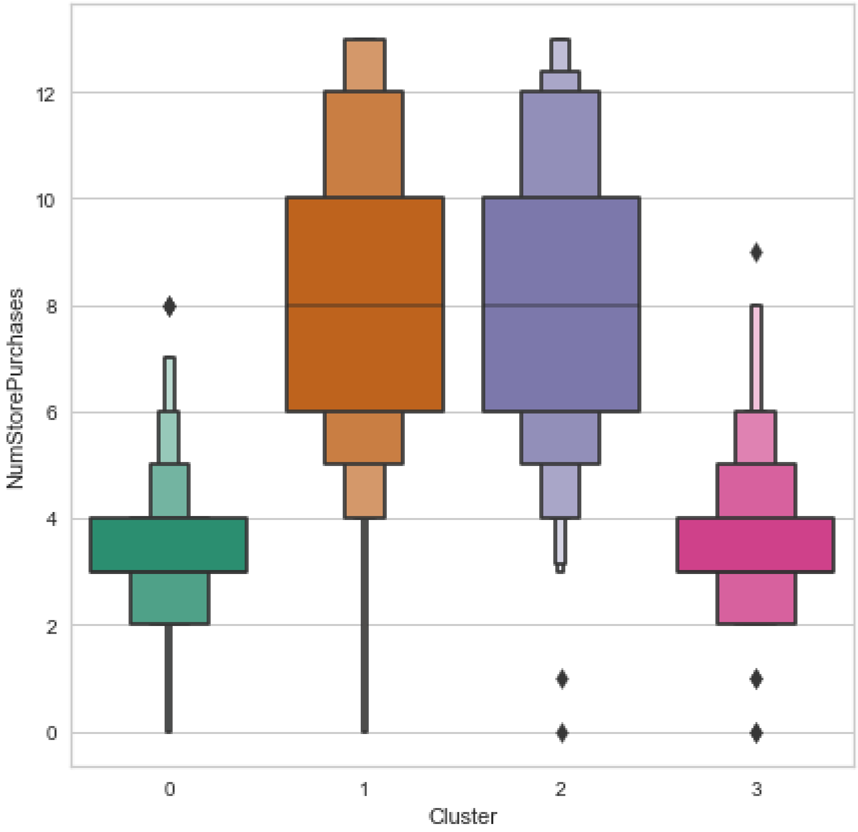
Note:- Most web based purchase are by cluster 2, followed by cluster 1

Catalog based purchase

Most catalog based purchase are by cluster 1, followed by cluster 2



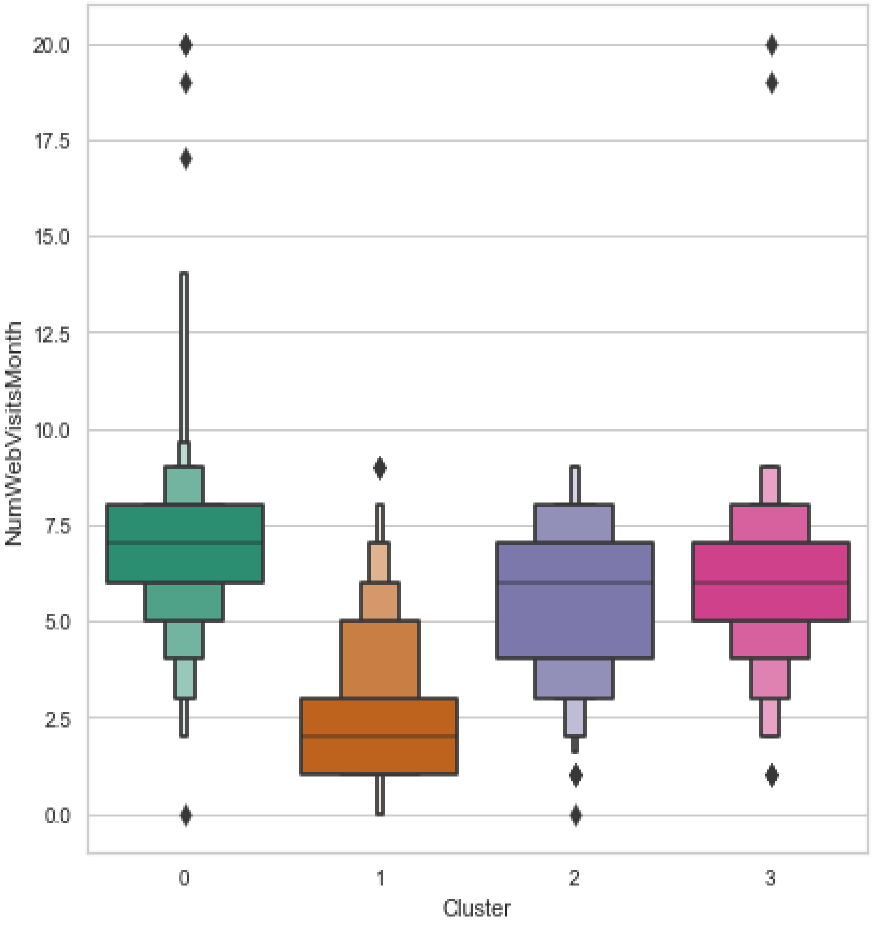
Store based purchases



Most store based purchases are by cluster 1 and 2

Number of visits to company's website in the last month

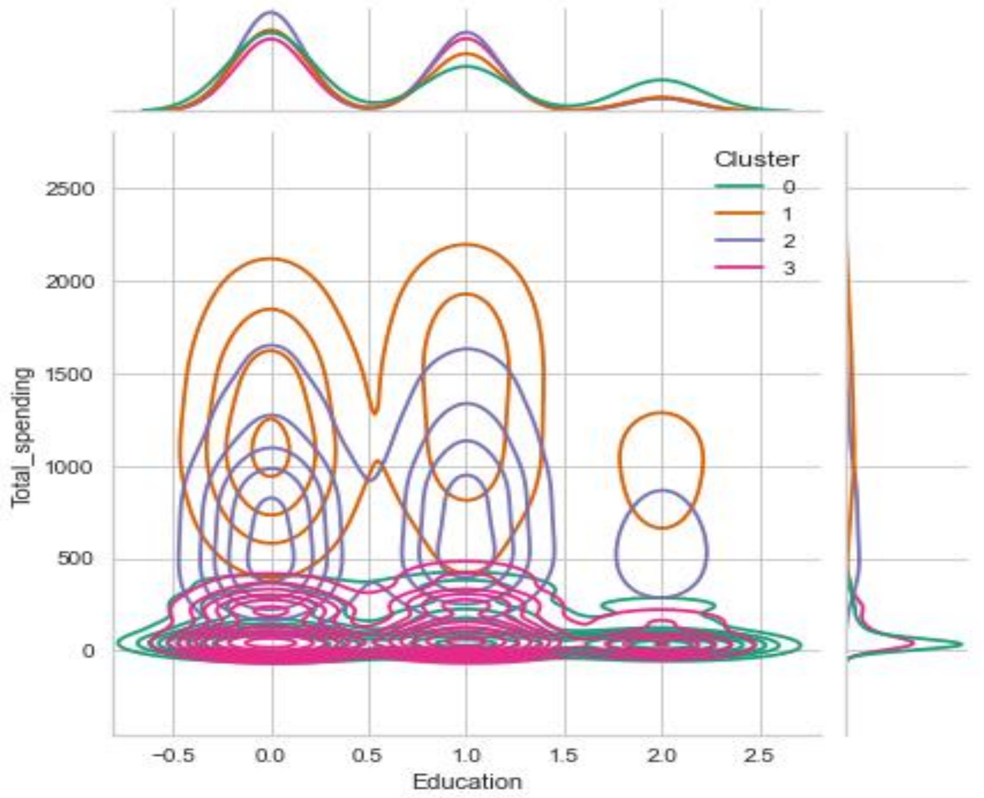
Most website visits per month are by cluster 0 and 3, although most website based purchases are by cluster 2 and 1

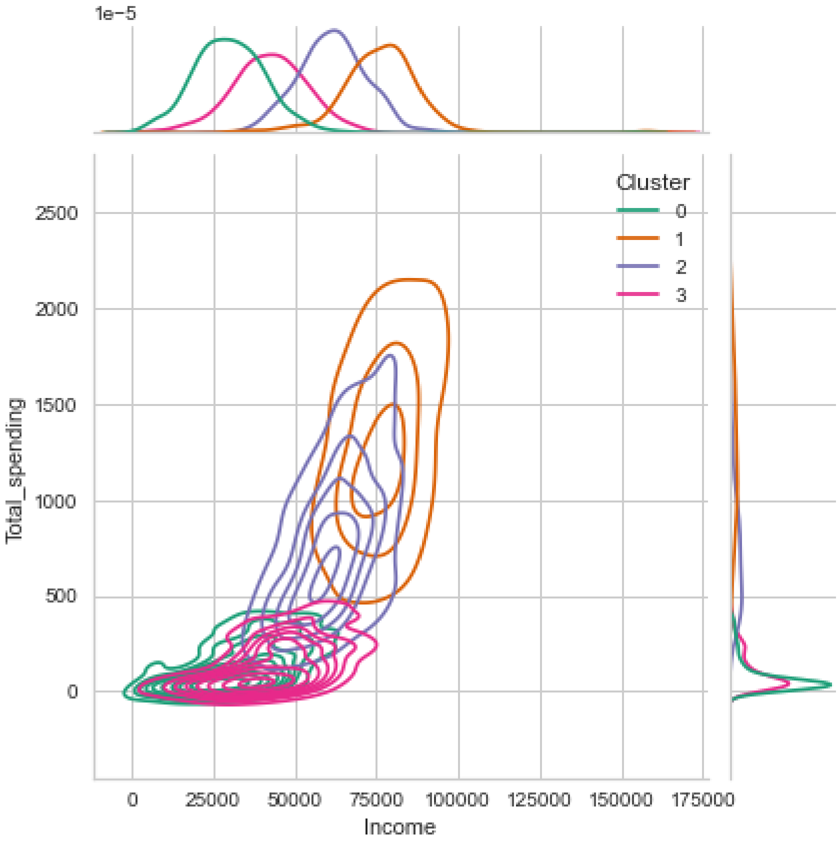


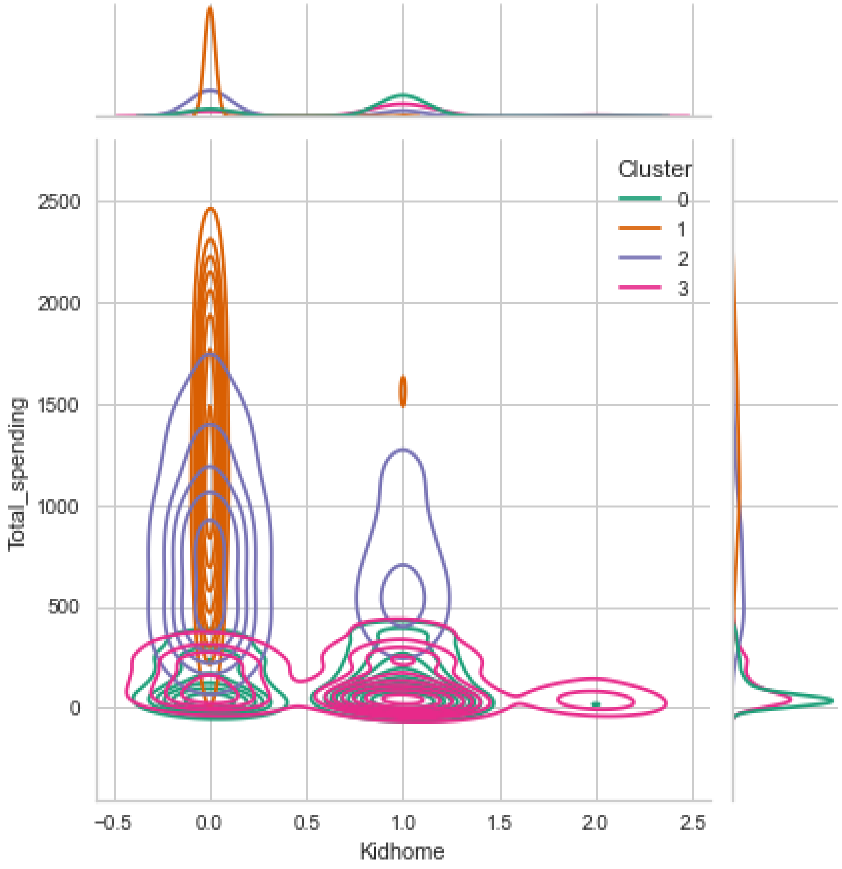
Now let's find out hidden information about each cluster based on their personal attributes

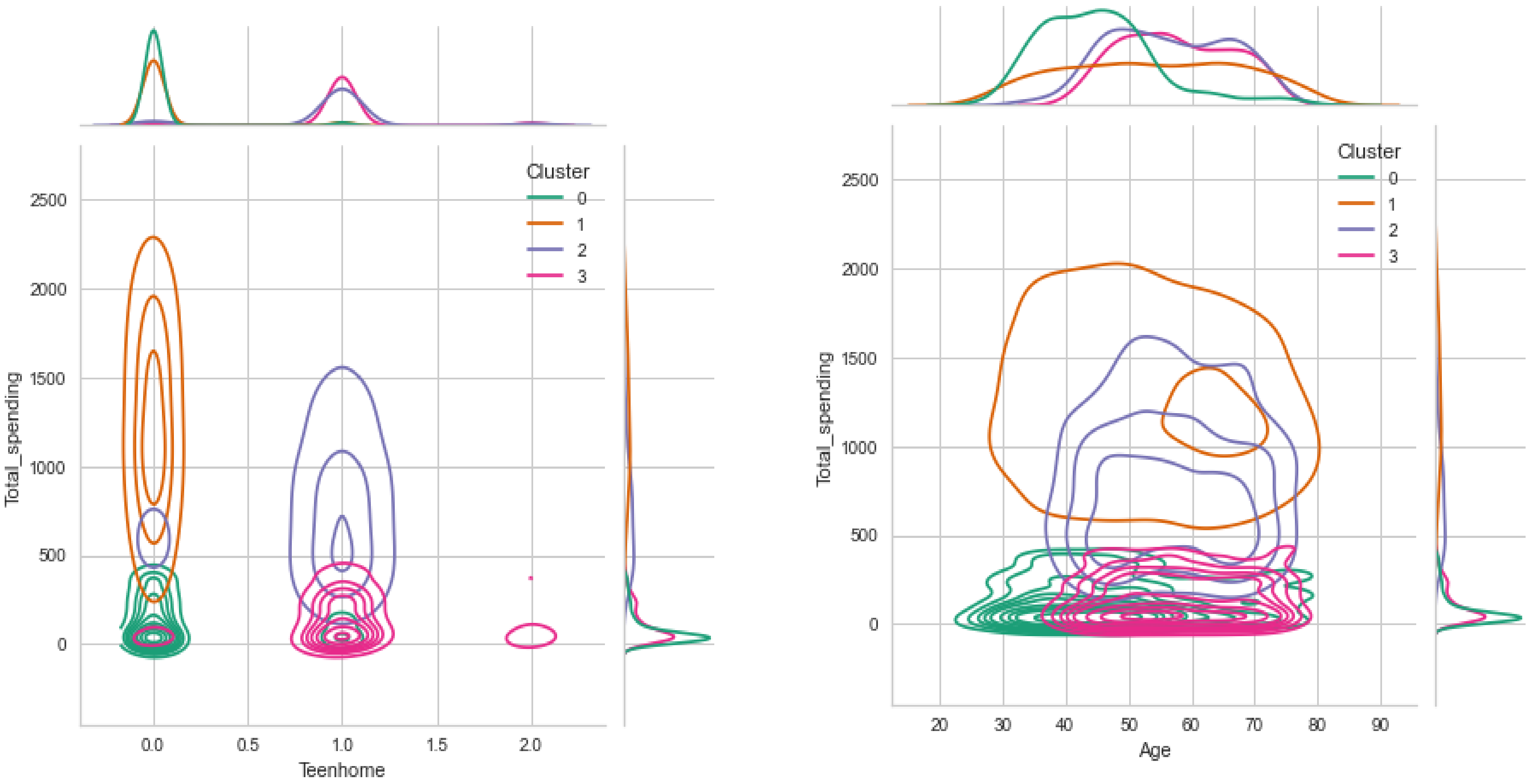
Some personal_attributes = ['Education', 'Income', 'Kidhome', 'Teenhome', 'Age', 'Living_with', 'Children', 'Is_parent', 'Family_size','Customer_duration']

Joint plots

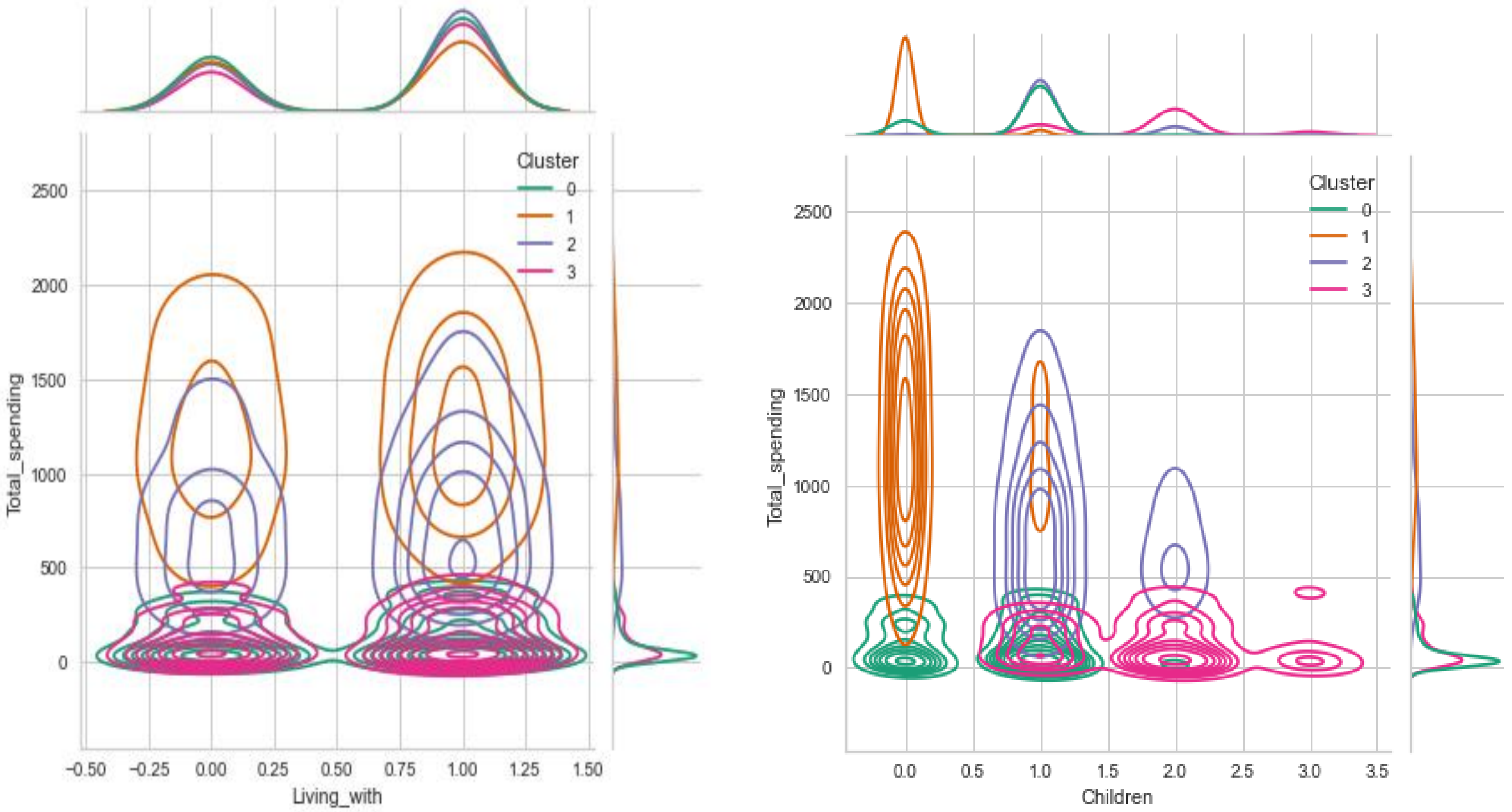


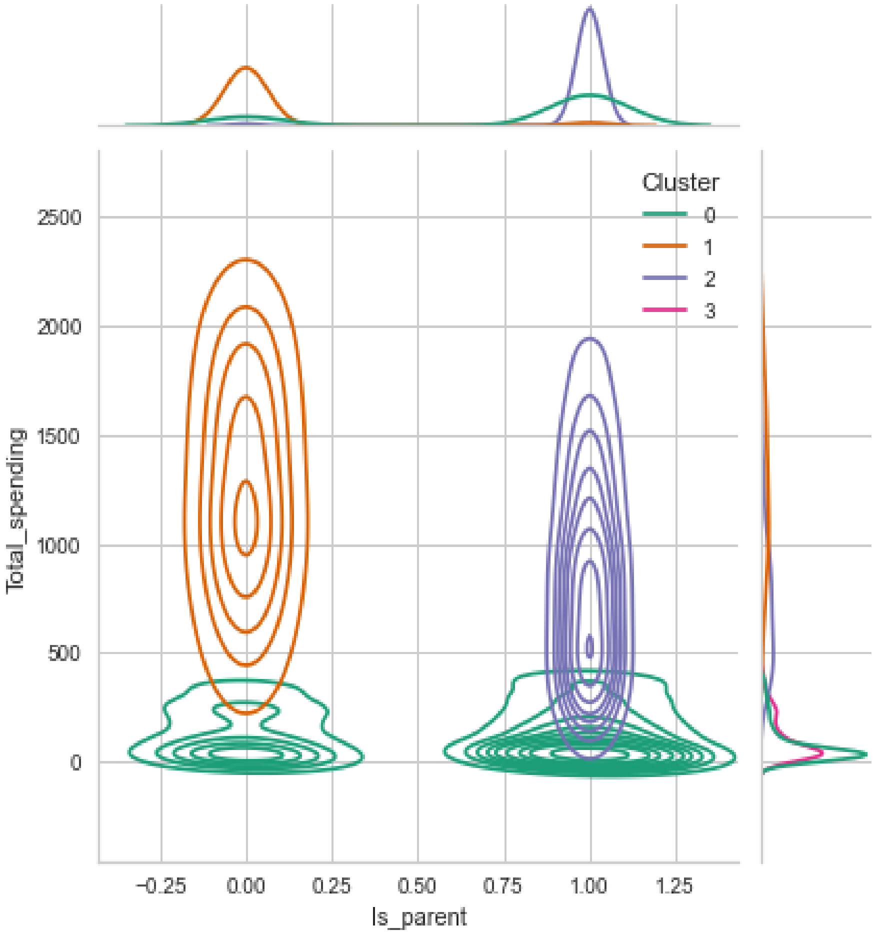

1e−5
Cluster
0
1
2
3
Total_spending
2500
2000
1500
1000
500
0
0
25000
50000
75000
100000
125000
150000
175000
Income


Cluster
0
1
2
3
Total_spending
2500
2000
1500
1000
500
0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Kidhome

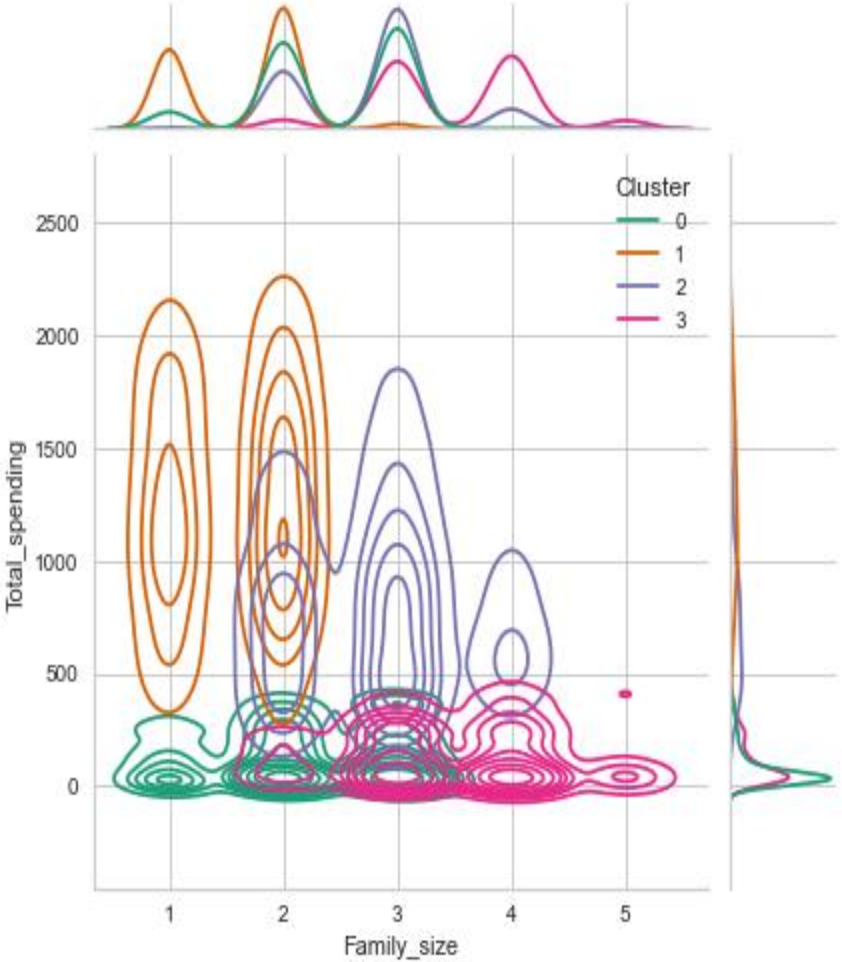


Cluster
0
1
2
3
2500
2000
1500
1000
500
0
Total_spending
0.0
0.5
1.0
1.5
2.0
Teenhome
Cluster
0
1
2
3
2500
2000
1500
1000
500
0
Total_spending
20
30
40
50
60
70
80
90
Age

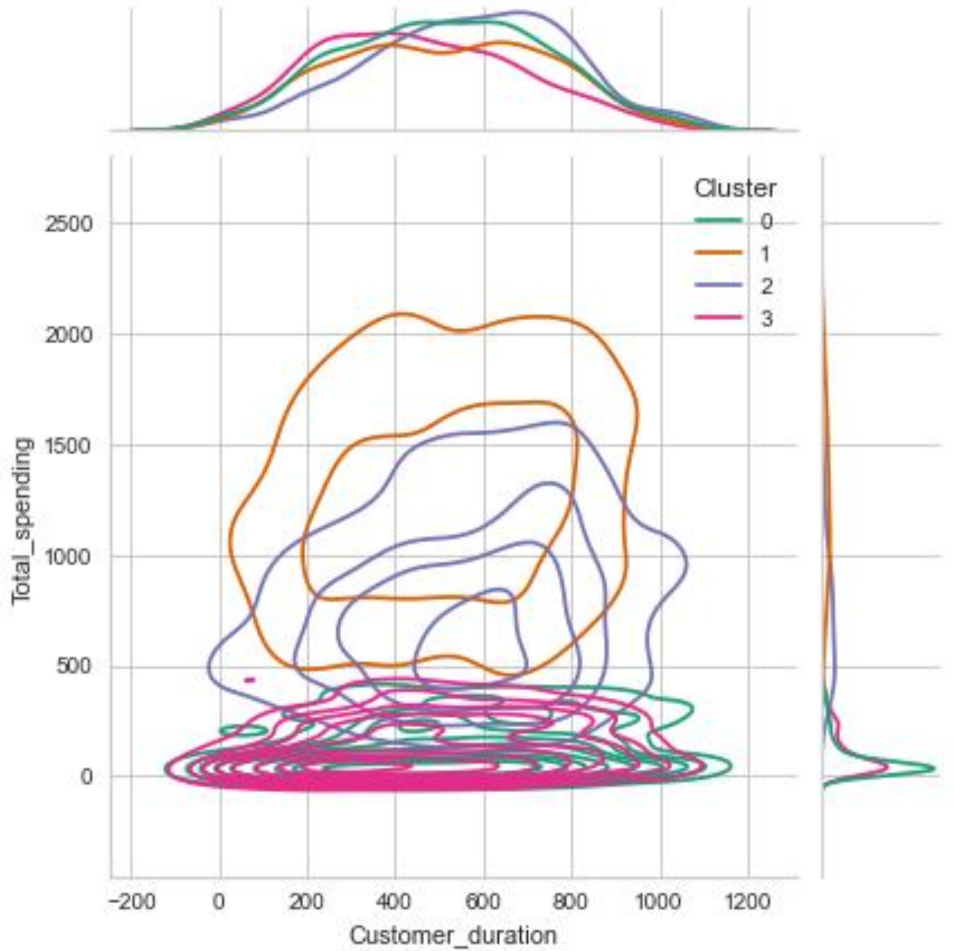

Cluster
0
1
2
3
Total_spending
2500
2000
1500
1000
500
0
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
Living_with
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Children



Cluster
0
1
2
3
Total_spending
2500
2000
1500
1000
500
0
-0.25
0.00
0.25
0.50
0.75
1.00
1.25
Is_parent



Cluster
0
1
2
3
Total_spending
2500
2000
1500
1000
500
0
1
2
3
4
5
Family_size



Cluster 1 and cluster 2 spends most of the time on searching the product of the company

Conclusions for each clusters

Cluster “0” :-

1. Low income low spending group
2. Comparatively younger
3. 0 or 1 child
4. Majority living with partner
5. Majority are parent
6. Family size 1 to 3

Cluster 1 :-

- High income high spending group
- No children
- All age range
- More people living with partner than alone
- Definitely not a parent
- Family size 1 to 2

Conclusions for each clusters

Cluster 2 -

1. Medium income medium-high spending group
2. Majority have 1 teen at home
3. Comparatively high aged
4. 1 - 2 children
5. Definitely a parent
6. Family size 2 to 4

Cluster 3 -

1. Medium income low spending group
2. Comparatively high aged
3. Majority living with partner
4. 1 to 3 children
5. Family size 3 to 5