CUSTOMER

SEGMENTATION

by Guardian Tri Angoro

CUSTOMER SEGMENTATION

Customer segmentation is the practice of dividing a company's customers into groups that reflect similarity among customers in each group. The goal of segmenting customers is to decide how to relate to customers in each segment in order to maximize the value of each customer to the business

WHY CUSTOMER SEGMENTATION IMPORTANT?

Customer segmentation has the potential to allow marketers to address each customer in the most effective way. Customer segmentation analysis allows marketers to identify discrete groups of customers with a high degree of accuracy based on demographic, behavioral and other indicators.

OBJECTIVE

K-Means Clustering

Profiling Cluster

ABOUT DATASET

```
<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
```

2240

Entries

29

Columns

24

Missing Value In Income Column

0

Duplicate Data

DATA CLEANING

CONVERT DT_CUSTOMER DATA TYPE TO DATE

```
[ ]  data['Dt_Customer']
```

```
0       04-09-2012
1       08-03-2014
2       21-08-2013
3       10-02-2014
4       19-01-2014
           ...
2235    13-06-2013
2236    10-06-2014
2237    25-01-2014
2238    24-01-2014
2239    15-10-2012
Name: Dt_Customer, Length: 2240, dtype: object
```

```
[ ]  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 customer
     print('The newest customer date:',max(dates))
     print('The noldest customer date:',min(dates))
```

```
The newest customer date: 2014-12-06
The noldest customer date: 2012-01-08
```

DATA CLEANING

CREATE AGE COLUMN

```
[ ]  data['Age'] = 2022 - data['Year_Birth']
```

```
[ ]  data['Age']
```

```
0       65
1       68
2       57
3       38
4       41
       ..
2235    55
2236    76
2237    41
2238    66
2239    68
Name: Age, Length: 2240, dtype: int64
```

CREATE TOTAL SPENDING COLUMN

```
[ ]  data["Spent"] = data["MntWines"]+ data["MntFruits"]+ data["MntMeatProducts"]+data["MntFishProducts"]+ data["MntSweetProducts"]+ data["MntGoldProds"]
```

```
[ ]  data["Spent"]
```

```
0       1617
1         27
2        776
3         53
4        422
        ...
2235    1341
2236     444
2237    1241
2238     843
2239     172
Name: Spent, Length: 2240, dtype: int64
```

SIMPLIFYING MARTIAL STATUS

```
[ ]  data['Marital_Status'].value_counts()
```

```
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64
```

```
[ ]  data["MaritalStatus"]=data["Marital_Status"].replace({"Married":"Partner", "Together":"Partner",
                                                           "Absurd":"Alone", "Widow":"Alone", "YOLO":"Alone",
                                                           "Divorced":"Alone", "Single":"Alone",})
```

SIMPLIFYING EDUCATION

```
[ ]  data['Education'].value_counts()
```

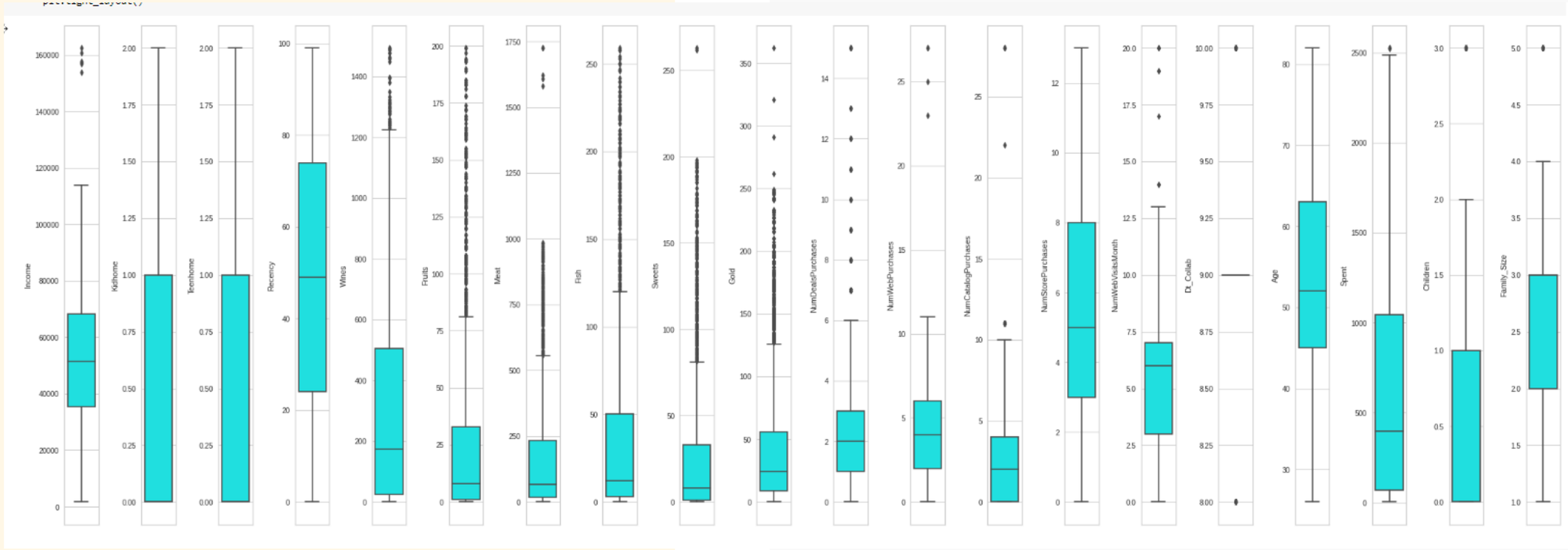
```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: Education, dtype: int64
```

```
[ ]  data['Education'] = data['Education'].replace({'Basic':'Undergraduate', '2n Cycle':'Undergraduate',
                                                    'Graduation':'Graduate', 'Master':'Graduate',
                                                    'PhD':'Graduate'})
```

```
[ ]  data['Education'].value_counts()
```

```
Graduate         1983
Undergraduate     257
Name: Education, dtype: int64
```

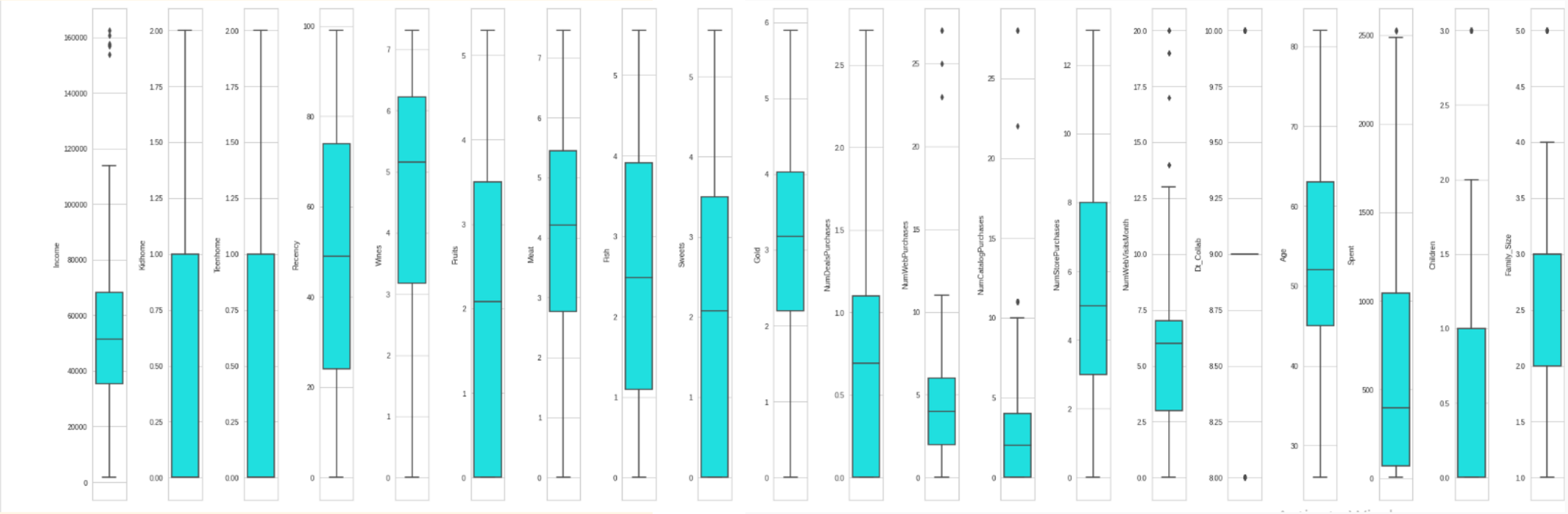
UNIVARIATE ANALYSIS
BOXPLOT



There are many outlier for some features

- Wines
- Fruits
- Meats
- Fish
- Sweets
- Gold
- NumDealsPurchases

UNIVARIATE ANALYSIS
BOXPLOT AFTER LOG TRANSFORMATION



There seems after treatment for outliers
with log transformation.
We have a better boxplot

VARIANCE INFLATION FACTOR (VIF)

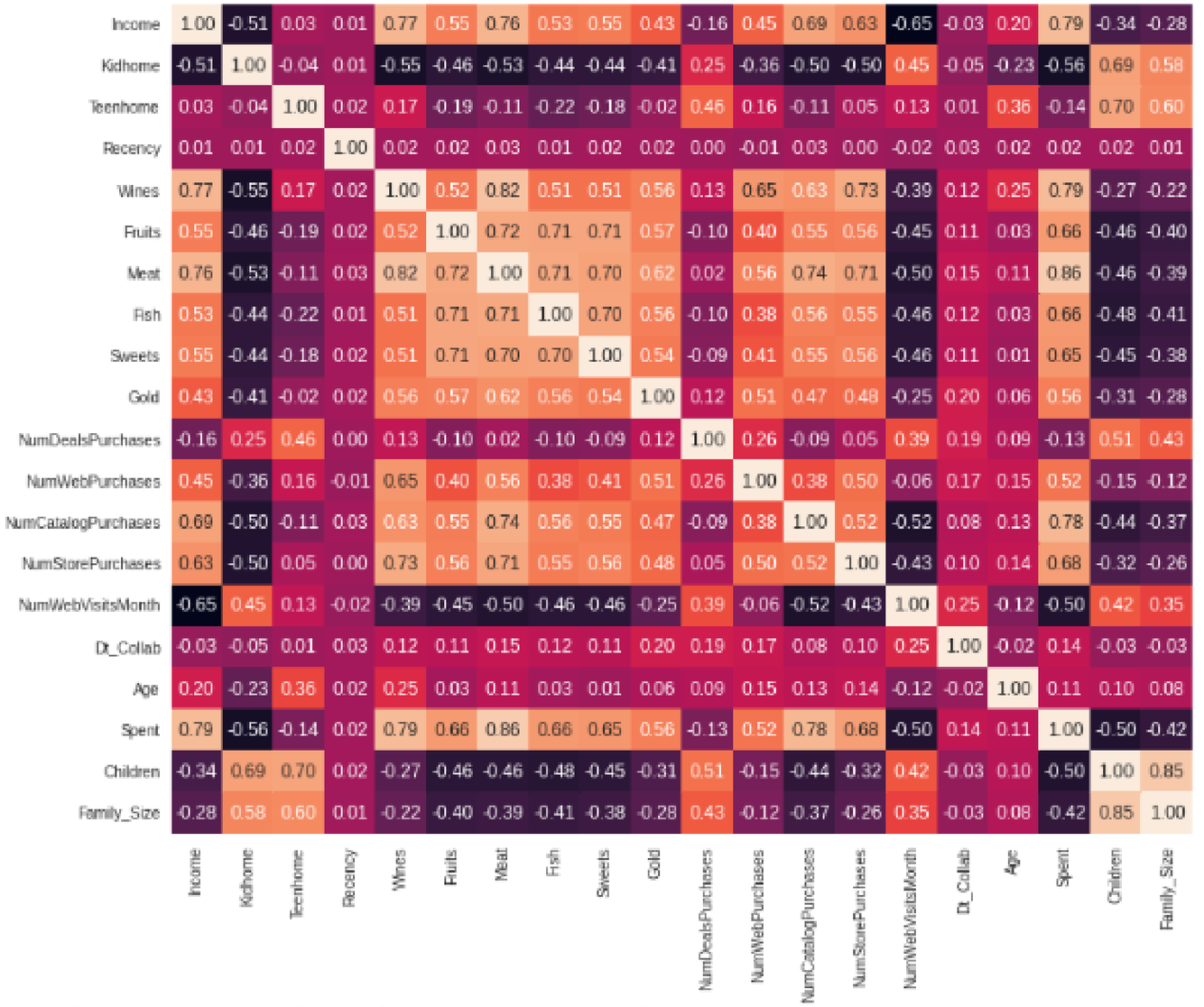
Interpretation :

There are some feature that have VIF score > 4

- Income
- Kidhome
- Teenhome
- Wines
- Meat
- Spent
- Children

	feature	vif_score
1	Income	4.492426
2	Kidhome	inf
3	Teenhome	inf
4	Recency	1.012607
5	Wines	6.424635
6	Fruits	2.694157
7	Meat	7.586649
8	Fish	2.815325
9	Sweets	2.626397
10	Gold	1.986033
11	NumDealsPurchases	2.147541
12	NumWebPurchases	2.039636
13	NumCatalogPurchases	2.957890
14	NumStorePurchases	2.563940
15	NumWebVisitsMonth	2.625376
16	Dt_Collab	1.261341
17	Age	1.251525
18	Spent	6.504530
19	Children	inf
20	Family_Size	3.650165

CORRELATION HEATMAP



Interpretation :

Variables inside cream rectangle are highly correlated each other

- Income have a high correlation with Wines, Meat and Spent
- Teenhome have a high correlation with Children
- Wines have a high correlation with Meat, NumStorePurchases, and Spent
- Fruits have a high correlation with Meat, Fish, and Sweets
- Meat have a high correlation with Fish, Sweets, NumCatalogPurchases, and NumStorePurchases.

This means they contain redundant information. We can choose only 1 of them to modelling process.

According VIF Score and Correlation Heatmap. I will drop some feature below here

- Kidhome
- TeenHome
- Wines
- Meat
- Children

DATA PREPROCESSING

- LABEL ENCODING FOR MARITAL STATUS AND EDUCATION

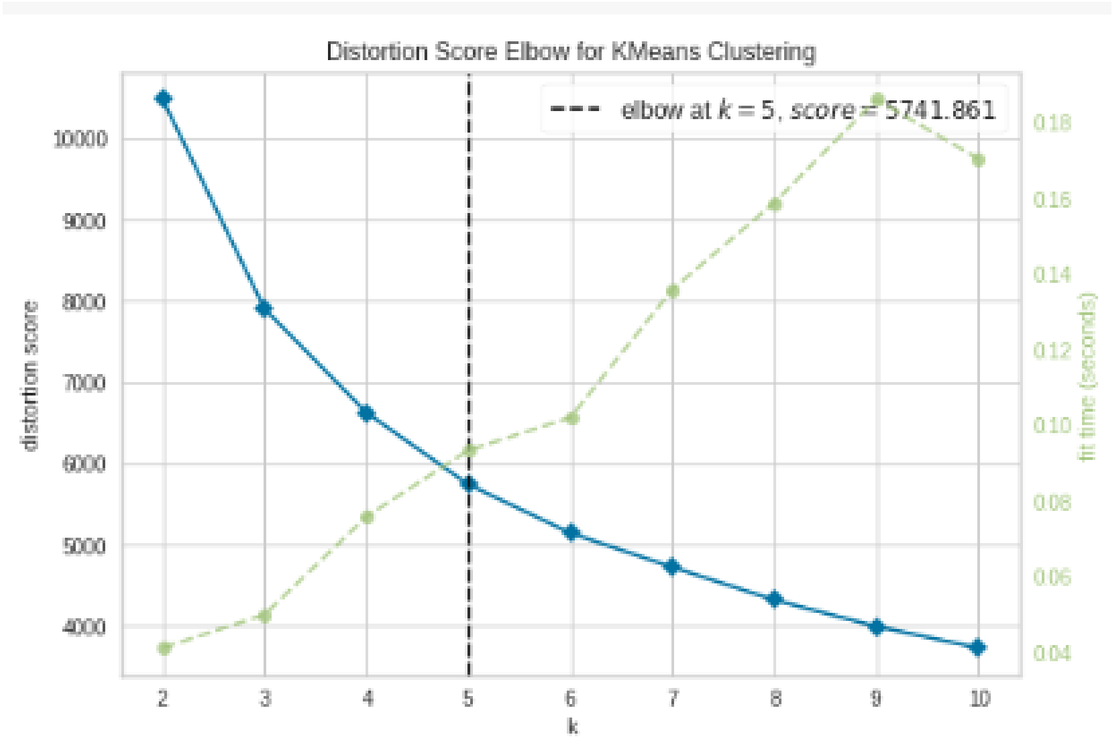
```
[ ] data['MaritalStatus'] = data['MaritalStatus'].replace("Partner", 0).replace("Alone", 1)
    data.head()
```

	Education	Income	Recency	Fruits	Fish	Sweets	Gold	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	Dt_Collab	Age	Spent	MaritalStatus	Family_Size
0	Graduate	58138.0	58	4.477337	5.147494	4.477337	4.477337	1.098612	8	10	4	7	10	65	1617	1	1
1	Graduate	46344.0	38	0.000000	0.693147	0.000000	1.791759	0.693147	1	1	2	5	8	68	27	1	3
2	Graduate	71613.0	26	3.891820	4.709530	3.044522	3.737670	0.000000	8	2	10	4	9	57	776	0	2
3	Graduate	26646.0	26	1.386294	2.302585	1.098612	1.609438	0.693147	2	0	4	6	8	38	53	0	3
4	Graduate	58293.0	94	3.761200	3.828641	3.295837	2.708050	1.609438	5	3	6	5	8	41	422	0	3

```
data['Education'] = data['Education'].replace("Undergraduate", 0).replace("Graduate", 1)
data.head()
```

	Education	Income	Recency	Fruits	Fish	Sweets	Gold	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	Dt_Collab	Age	Spent	MaritalStatus	Family_Size
0	1	58138.0	58	4.477337	5.147494	4.477337	4.477337	1.098612	8	10	4	7	10	65	1617	1	1
1	1	46344.0	38	0.000000	0.693147	0.000000	1.791759	0.693147	1	1	2	5	8	68	27	1	3
2	1	71613.0	26	3.891820	4.709530	3.044522	3.737670	0.000000	8	2	10	4	9	57	776	0	2
3	1	26646.0	26	1.386294	2.302585	1.098612	1.609438	0.693147	2	0	4	6	8	38	53	0	3
4	1	58293.0	94	3.761200	3.828641	3.295837	2.708050	1.609438	5	3	6	5	8	41	422	0	3

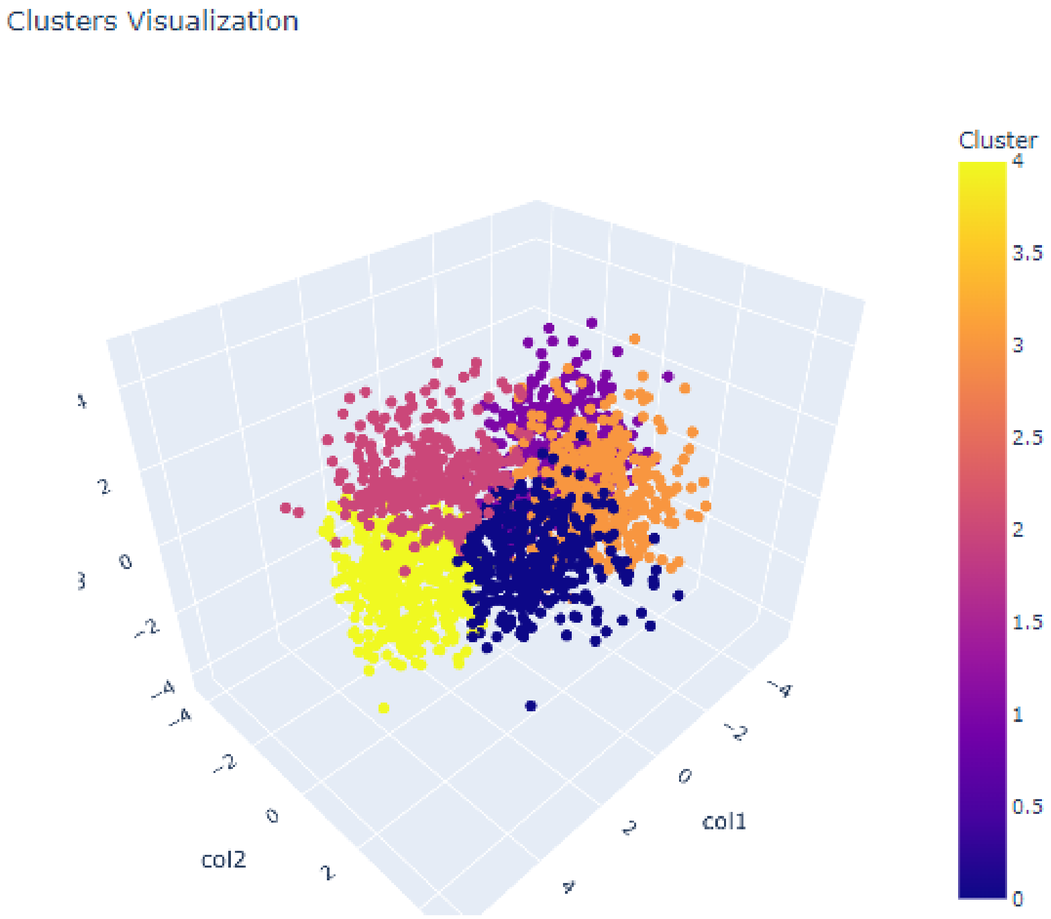
DISTORTION SCORE ELBOW FOR KMEANS CLUSTERING




```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc2b07a9610>
```

To determine the optimal number of clusters, i have to select the value of k at the “elbow” ie the point after which the distortion start decreasing in a linear. Thus for the given data, i have conclude that the optimal number of clusters for the data is 5.

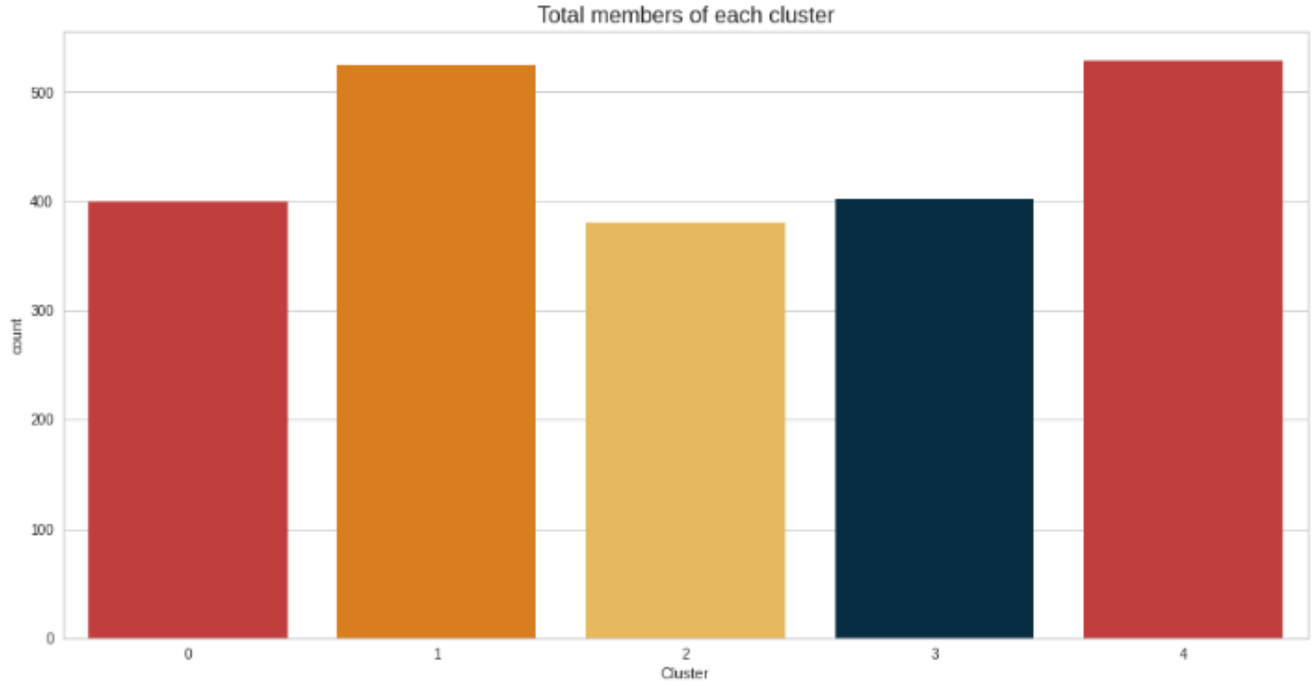
RESULT KMEANS CLUSTERING



Visualization with PCA along 3 main Col. the clusters are very mixed

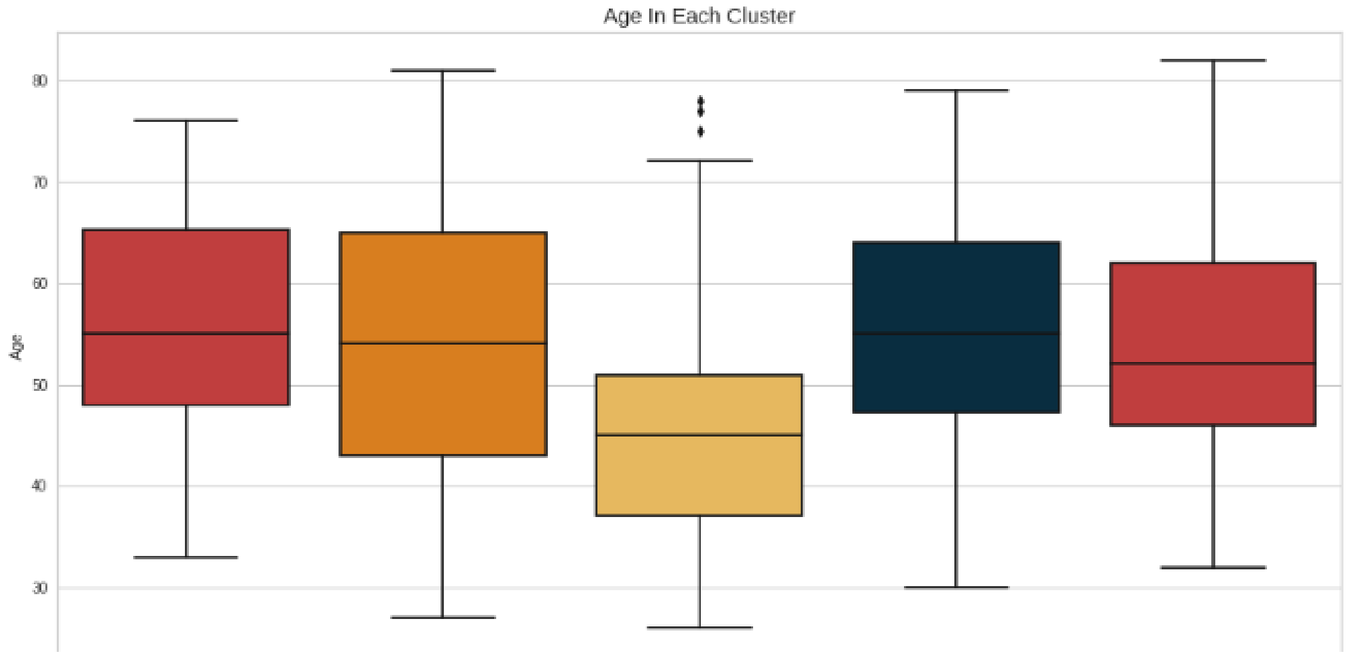
PROFILING
TOTAL MEMBER OF CUSTOMER

```
Total members of each cluster :
4    529
1    525
3    402
0    400
2    380
Name: Cluster, dtype: int64
**************************************************
Text(0.5, 1.0, 'Total members of each cluster')
```



In profiling total member of customer, Cluster 4 has the higgest total member with 529 member. Followed by cluster 1 with 525 member, cluster 3 with 402 member, and cluster 0 with 400 member.

PROFILING AGE



Avg Age of each cluster :

Cluster	
0	56.265000
1	54.011429
2	45.342105
3	55.549751
4	53.521739

Maximum Age of each cluster :

Cluster	
0	76
1	81
2	78
3	79
4	82

Minimum Age of each cluster :

Cluster	
0	33
1	27
2	26
3	30
4	32

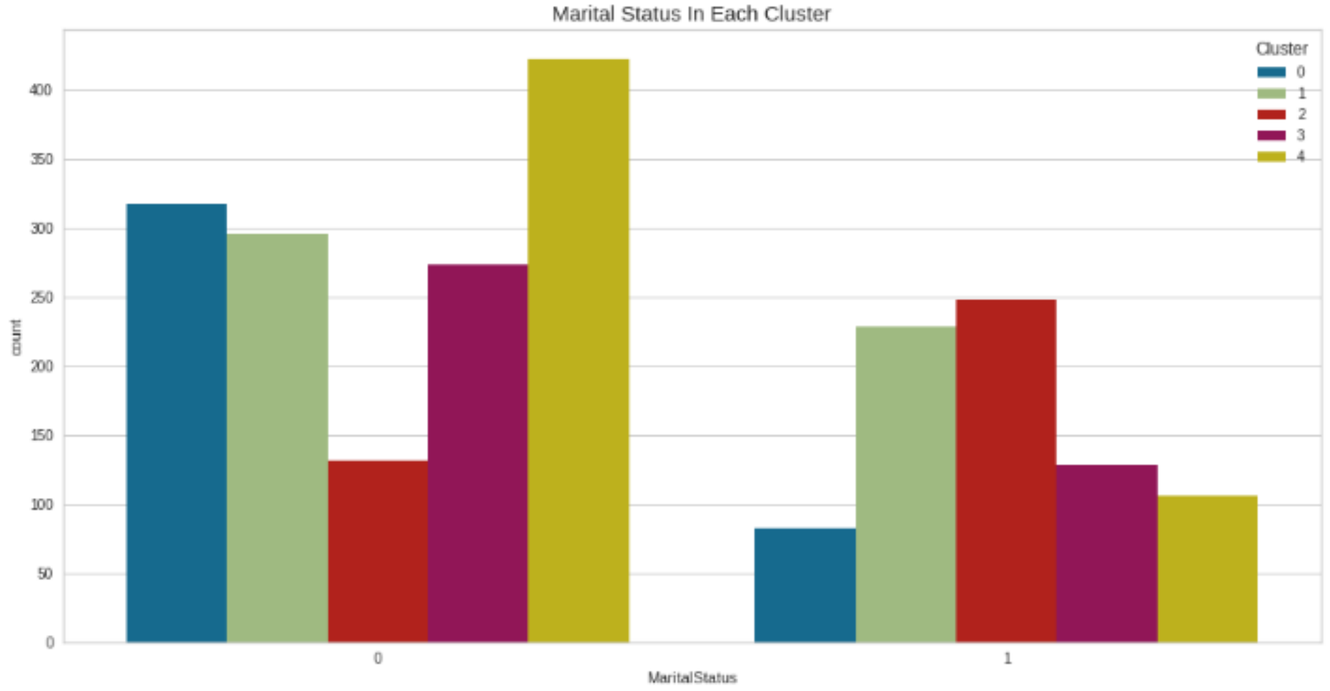
In profiling Age

By averange age, Cluster 0 has the higgest averange age with 56 years old. followed by maxium age and minimum age with 76 years old and 33 years old.

But the youngest member is in cluster 1 with 27 years old. And the oldest member is in cluster 4 with 82 years old

PROFILING MARITAL STATUS

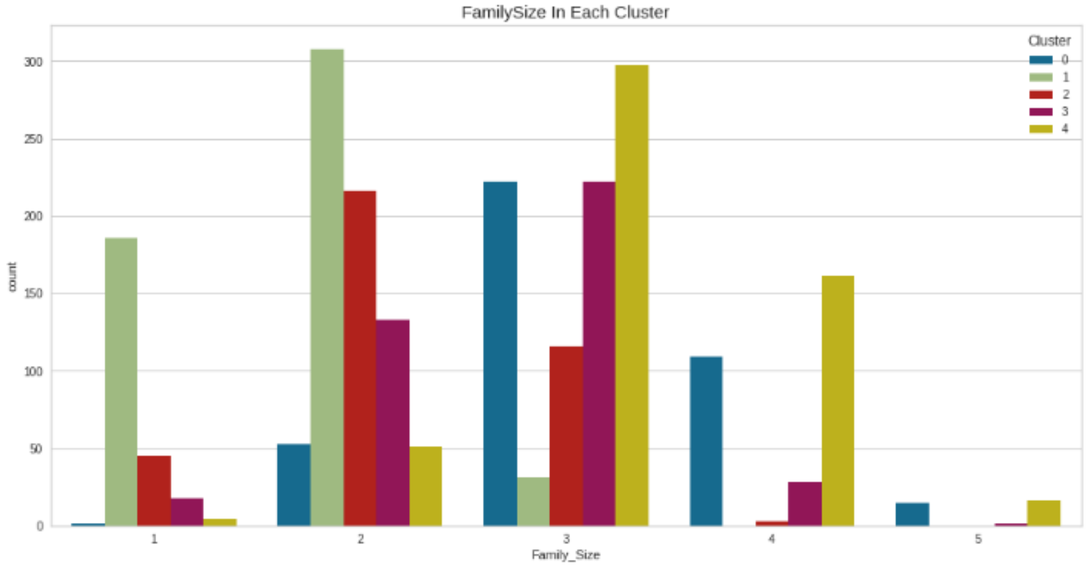
```
Cluster  MaritalStatus
0        0                317
         1                 83
1        0                296
         1                229
2        0                132
         1                248
3        0                274
         1                128
4        0                423
         1                106
dtype: int64
**************************************************
Text(0.5, 1.0, 'Marital Status In Each Cluster')
```



In profiling marital status, Cluster 4 has the higgest total member with 529 member. Followed by cluster 1 with 525 member, cluster 3 with 402 member, and cluster 0 with 400 member.

PROFILING FAMILY SIZE

```
Cluster  Family_Size
0        1              1
         2             53
         3            222
         4            109
         5             15
1        1            186
         2            308
         3             31
2        1             45
         2            216
         3            116
         4              3
3        1             18
         2            133
         3            222
         4             28
         5              1
4        1              4
         2             51
         3            297
         4            161
         5             16
dtype: int64
Text(0.5, 1.0, 'FamilySize In Each Cluster')
```



In profiling family size,

Cluster 0 has the most family size with 3 members in one family with a value of 205.

Cluster 1 has the most family size with 2 members in one family with a value of 383.

Cluster 2 has the most family size with 2 members in one family with a value of 216.

Cluster 3 has the most family size with 3 members in one family with a value of 222.

Cluster 4 has the most family size with 3 members in one family with a value of 287.

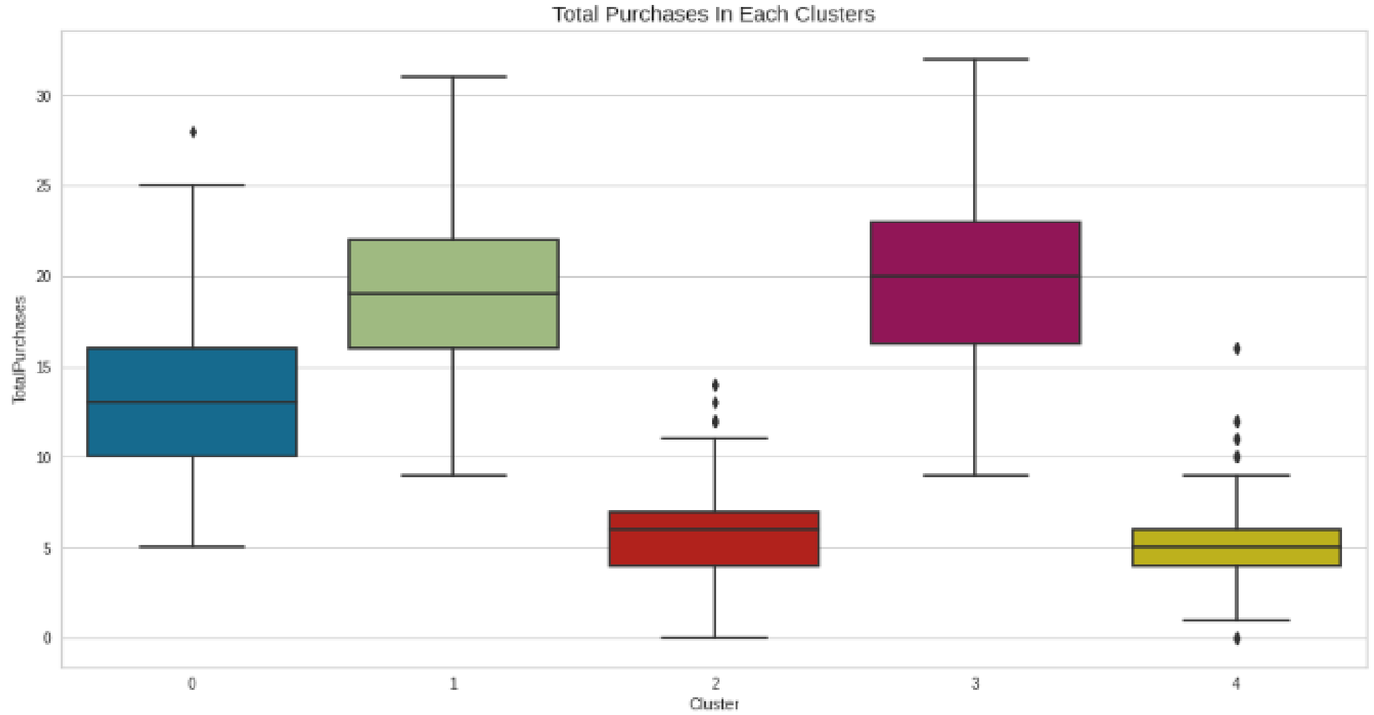
PROFILING SPENT OF THE MEMBER COMPARED BY INCOME

```
                Income   Spent
Cluster
0         19379256.0  179620
1         40126794.5  699706
2         11092153.5   41063
3         25326640.0  398009
4         20241201.0   36588
**************************************************
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

Text(0.5, 1.0, 'Income vs Spent of each cluster')
```

In Profiling Spent and Income By comparing Spent and Income of member, Cluster 1 has the highest spent and income compared to other clusters. But cluster 4 has the smallest spent and cluster 2 has the smallest income

PROFILING NUMBER OF PURCHASES IN EACH CLUSTER

```
Cluster
0    5207
1    9990
2    2226
3    7898
4    2733
Name: TotalPurchases, dtype: int64
Text(0.5, 1.0, 'Total Purchases In Each Clusters')
```

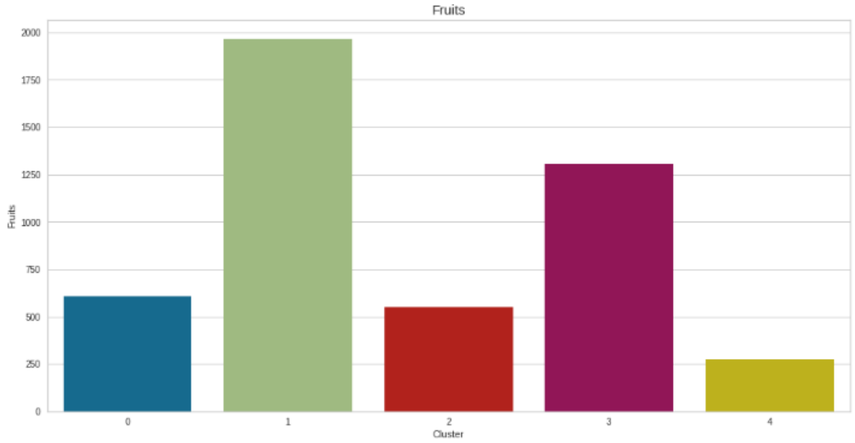


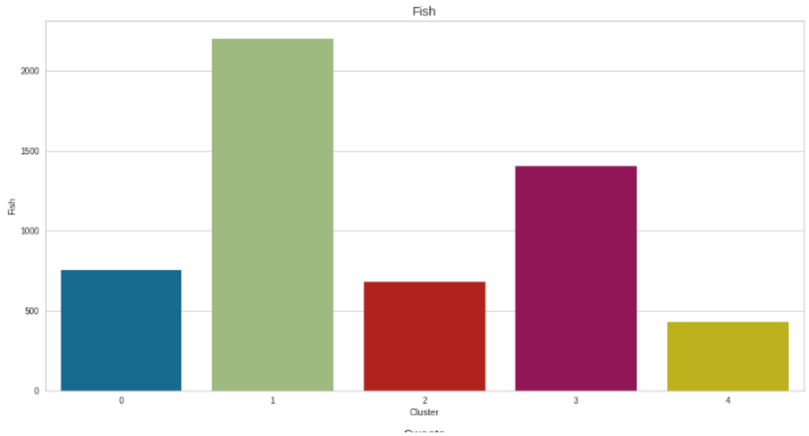
In Profiling number of purchases . Cluster 1 has the highest purchases value with 9990. Followed by cluster 3 with 7898, cluster 0 with 5207, cluster 4 with 2733 and the last is cluster 2 with 2226 number of purchases

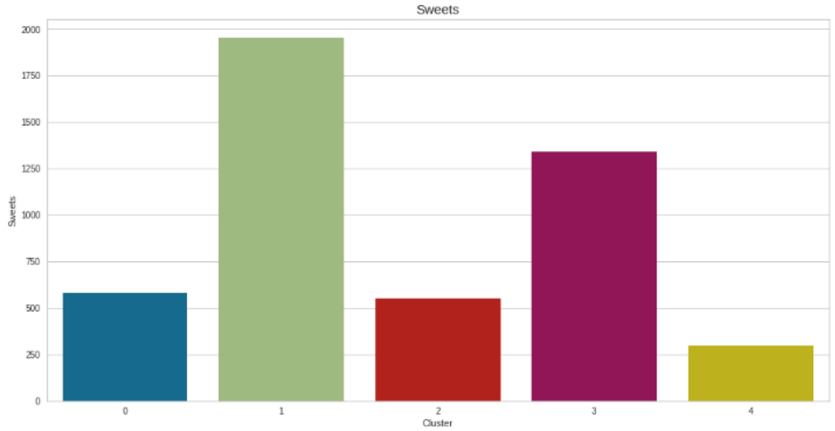
PROFILING
WHAT ARE EACH CLUSTER INTERESTED IN BUYING ?

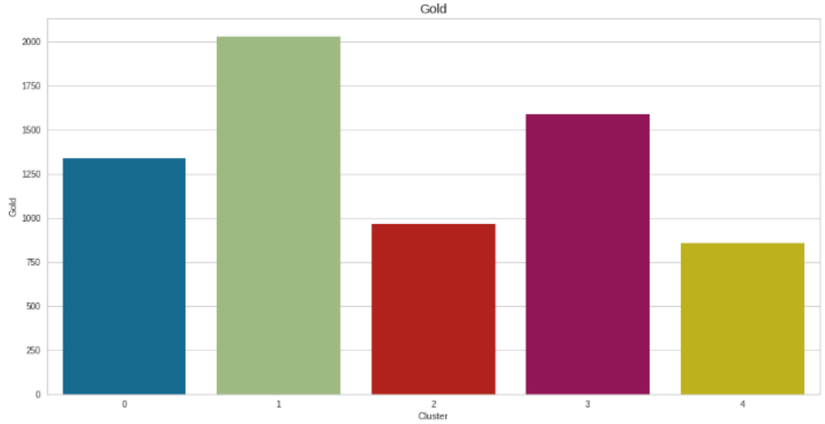
Cluster	Fruits	Fish	Sweets	Gold
0	607.289827	751.737284	581.827525	1339.309136
1	1967.940869	2201.459027	1955.680578	2029.913530
2	551.899997	681.927179	550.767919	965.452446
3	1307.191892	1402.062850	1338.356054	1586.869359
4	276.633122	427.943636	295.897087	854.563869

****









In profiling the insterest of buying for each cluster,

buying gold is the most favorite thing to buy for cluster 0, cluster 2, cluster 3, and cluster 4. Which is diffrent form cluster 1 who prefers to buy sweets

SUMMARY

Cluster 0

1. Total member of this cluster is 400
2. Compared to other clusters, they have avg Income and low spending
3. Compared to other clusters, they have been 5207 product
4. Their Family Size is between 1 and 5 people.
5. They are mostly married.
6. Their Age is between 33 and 76 years (most of them are around 56 years old).
7. They mostly has been graduated
8. They mostly interested buying gold

Cluster 1

1. Total member of this cluster is 525
2. Compared to other clusters, they have avg high Income and high spending
3. Compared to other clusters, they have been 9990 product
4. Their Family Size is between 1 and 3 people.
5. They are mostly married.
6. Their Age is between 27 and 81 years (most of them are around 54 years old).
7. They mostly has been graduated
8. They mostly interested buying fish

Cluster 2

1. Total member of this cluster is 380.
2. Compared to other clusters, they have avg Income and avg spending.
3. Compared to other clusters, they have been 2226 product.
4. Their Family Size is between 1 and 4 people.
5. They are mostly single parents.
6. Their Age is between 26 and 78 years (most of them are around 55 years old).
7. They mostly has been graduated.
8. They mostly interested buying gold.

Cluster 3

1. Total member of this cluster is 402
2. Compared to other clusters, they have high Income and high spending
3. Compared to other clusters, they have been 7898 product
4. Their Family Size is between 1 and 5 people.
5. They are mostly married.
6. Their Age is between 30 and 79 years (most of them are around 56 years old).
7. They mostly has been graduated
8. They mostly interested buying gold.

Cluster 4

1. Total member of this cluster is 529
2. Compared to other clusters, they have low Income and low spending
3. Compared to other clusters, they have been 2733 product
4. Their Family Size is between 1 and 5 people.
5. They are mostly married.
6. Their Age is between 32 and 82 years (most of them are around 53 years old).
7. They mostly has been graduated
8. They mostly interested buying gold.

THANK YOU