CUSTOMER SEGMENTATION

Alyssa Juarez

The Problem

- In this project I use K-Means, K-Modes and Agglomerative Clustering in order to divide a dataset of grocery store customers into different groups. In doing so It allows a company to understand who is buying what products and allows companies to more accurately cater to their customer base.

The Data

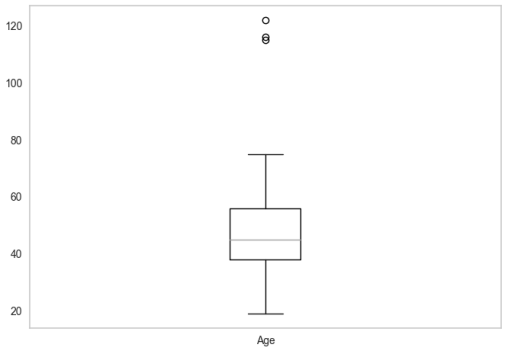
- Unsupervised
- Both Categorical and Numerical
- 2240 instances
- 29 features

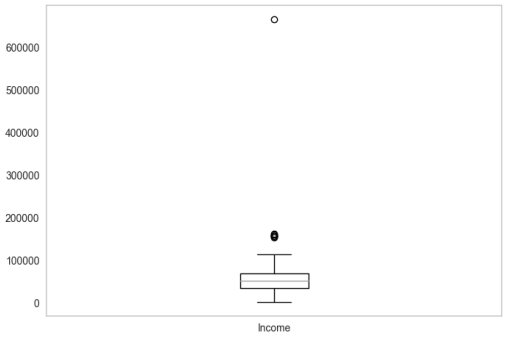
Feature	Description
ID	ID specific to each customer
Year_Birth	Customer year of birth
Education	Customer education
Marital_Status	Customers martial status
Income	Customers yearly income
Kidhome	# of small children
Teenhome	# of teenagers
Dt_Customer	Date they became a customer
Recency	Time since last visit
MntWines	Wines bought
MntFruits	Fruits bought
MntMeatProducts	Meat bought
MntFishProducts	Fish bought
MntSweetProducts	Sweets brought
MntGoldProds	Gold bought
NumDealsPurchases	# of promotion items purchased
NumWebPurchases	# of purchases through web
NumCatalogPurchases	# of purchases through catalog
NumWebVisitsMonth	# of visits this month
AcceptedCmp3	unknown
AcceptedCmp4	unknown
AcceptedCmp5	unknown
AcceptedCmp1	unknown
AcceptedCmp2	unknown
Complain	unknown
Z_CostContact	unknown
Z_Revenue	unknown
Response	unknown

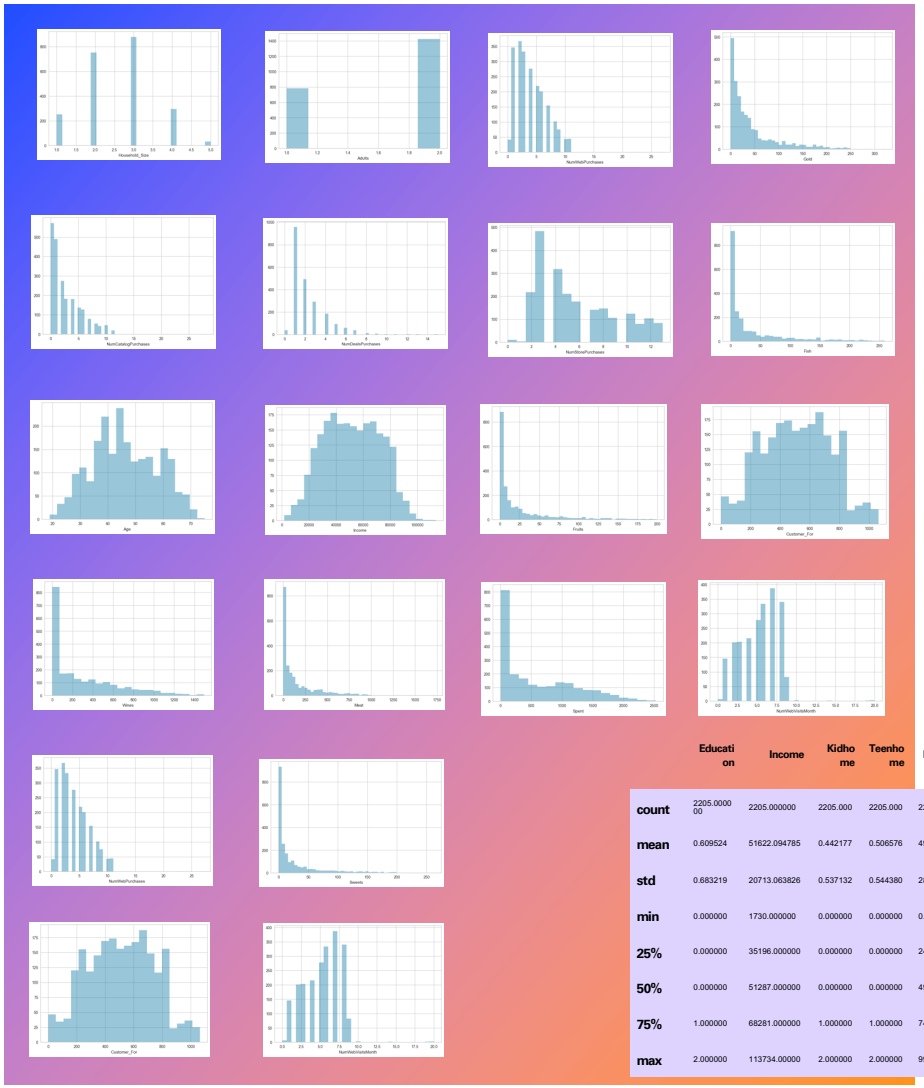
Data Cleaning

- Removing null values
- Removing Outliers
  - I only removed outliers such as income and Age
- Change categorical to numerical
- Feature engineering

Feature	Description
Customer_For	# of days they've been a customer
Age	Age of customer
Spent	Amount Spent
Adult	# of adults in household
Children	# of children in household
Household_Size	# of people in household







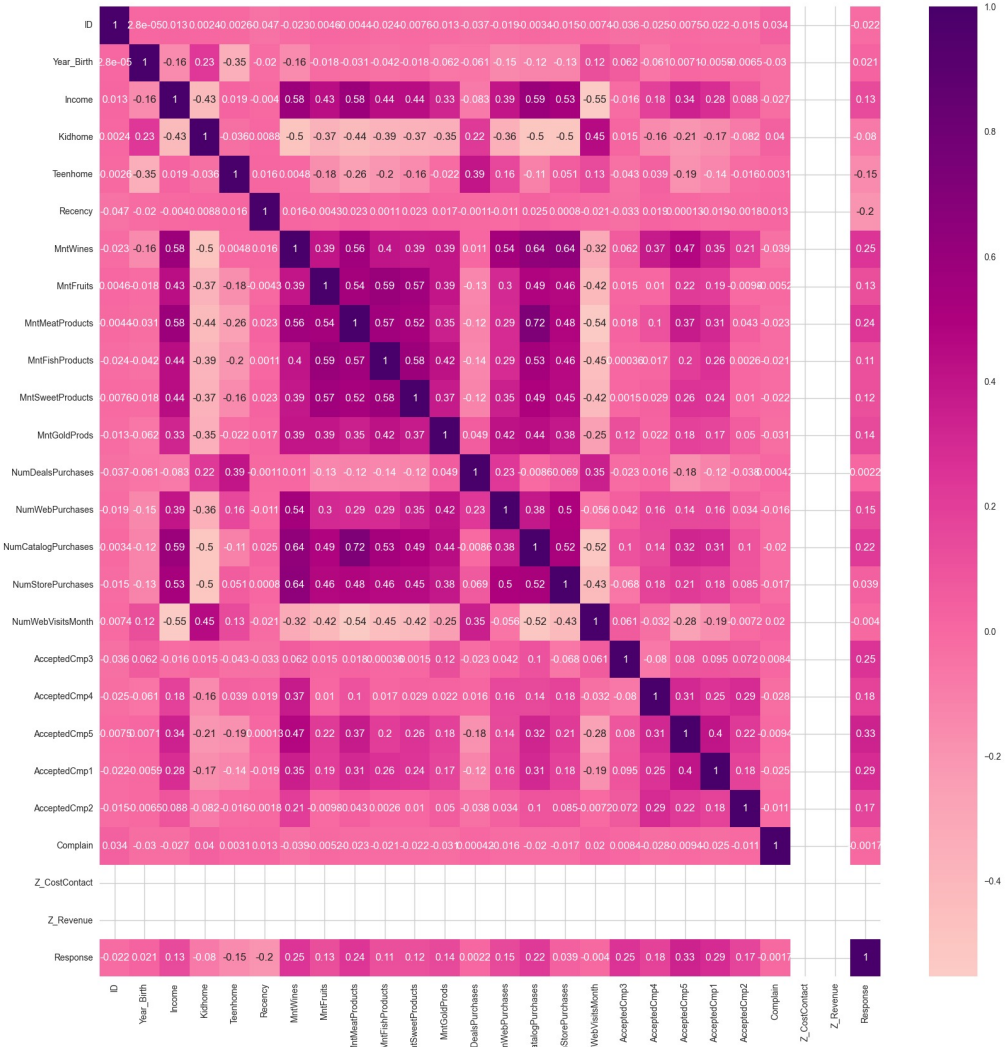
Data Analysis

- Density plots
- Acquiring descriptive statistics

	Education	Income	Kidhome	Teenhome	Recency	Wines	Fruits	Meat	Fish	Sweets	...	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	Customer_For	Age	Spent	Adults	Children	Household_Size
count	2205.000000	2205.000000	2205.000	2205.000	2205.00000	2205.00000	2205.0000	2205.00000	2205.000000	2205.000000	...	2205.000000	2205.000000	2205.000000	2205.0	2205.000000	2205.0000	2205.000000	2205.0000	2205.000	2205.000
mean	0.609524	51622.094785	0.442177	0.506576	49.009070	306.164626	26.403175	165.312018	37.756463	27.128345	...	4.100680	2.645351	5.823583	5.33696	512.062585	46.095692	606.821769	1.644898	0.948753	2.593651
std	0.683219	20713.063826	0.537132	0.544380	28.932111	337.493839	39.784484	217.784507	54.824635	41.130468	...	2.737424	2.798647	3.241796	2.41353	232.528808	11.705801	601.675284	0.478653	0.749231	0.906197
min	0.000000	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.0000	0.000000	19.000000	5.000000	1.000000	0.000000	1.000000
25%	0.000000	35196.000000	0.000000	0.000000	24.000000	24.000000	2.000000	16.000000	3.000000	1.000000	...	2.000000	0.000000	3.000000	3.0000	340.000000	38.000000	69.000000	1.000000	0.000000	2.000000
50%	0.000000	51287.000000	0.000000	0.000000	49.000000	178.000000	8.000000	68.000000	12.000000	8.000000	...	4.000000	2.000000	5.000000	6.0000	513.000000	45.000000	397.000000	2.000000	1.000000	3.000000
75%	1.000000	68281.000000	1.000000	1.000000	74.000000	507.000000	33.000000	232.000000	50.000000	34.000000	...	6.000000	4.000000	8.000000	7.0000	686.000000	56.000000	1047.000000	2.000000	1.000000	3.000000
max	2.000000	113734.00000	2.000000	2.000000	99.000000	1493.00000	199.00000	1725.00000	259.00000	262.00000	...	27.00000	28.00000	13.00000	20.000	1063.000	75.000000	2525.000000	2.000000	3.000000	5.000000

Dimension Reduction

	PCA1	PCA2	PCA3
0	4.792799	-0.201735	2.665035
1	-2.805690	0.089870	-1.882891
2	2.296644	-0.504974	-0.288823
3	-2.681957	-1.547154	-0.950586
4	-0.499809	0.093429	0.001852
...	...	...	...
2200	2.639373	2.040107	0.476451
2201	-2.864289	4.407249	-1.486880
2202	2.315013	-1.769889	0.578566
2203	1.788713	1.412011	-1.655871
2204	-2.604216	1.875229	-0.279369

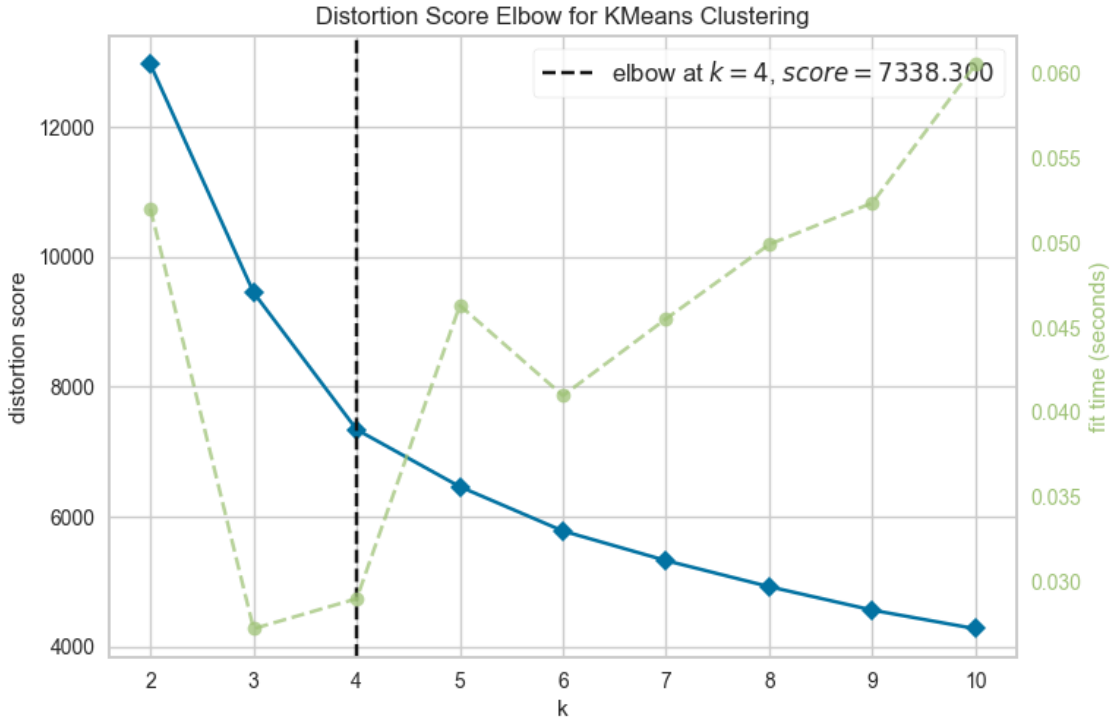


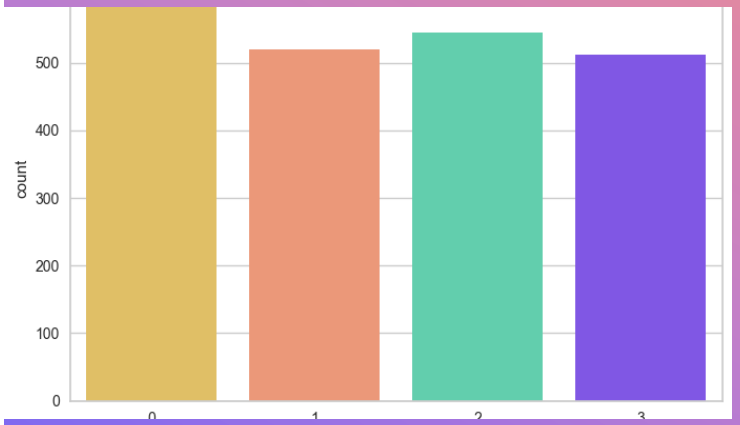
Model Selection

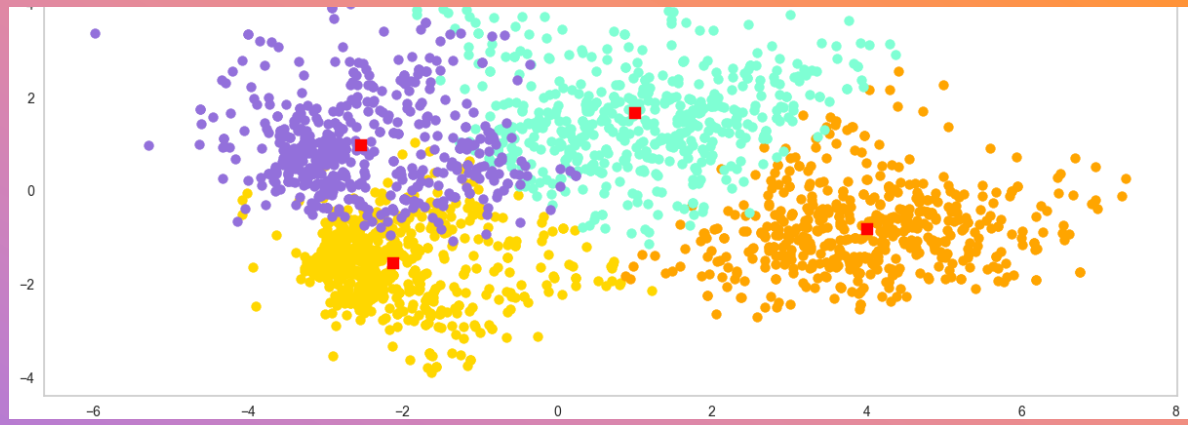
- For this project, I need an unsupervised clustering model which leaves me with the following:
  - K-Means
  - Agglomerative Clustering (Hierarchal)
  - K-Modes

K-Means

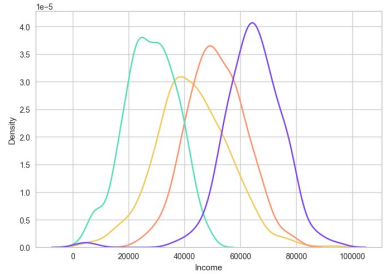
Finding The Number of Clusters

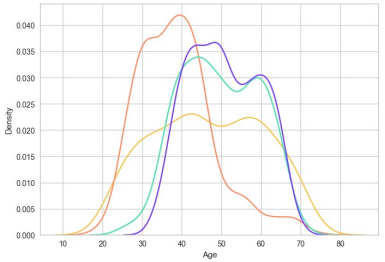


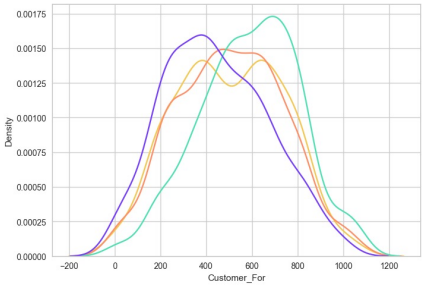


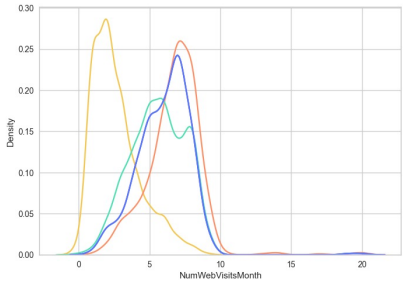


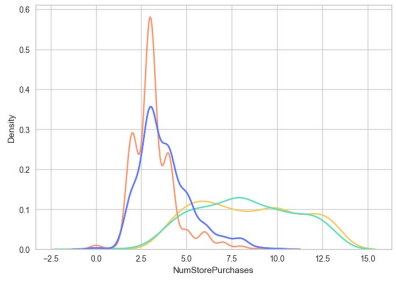
K-MEANS DISTRIBUTION

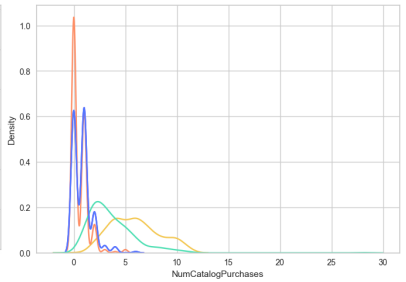

1e−5
Density
Income

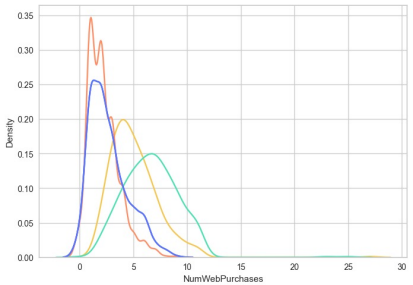

Density
Age

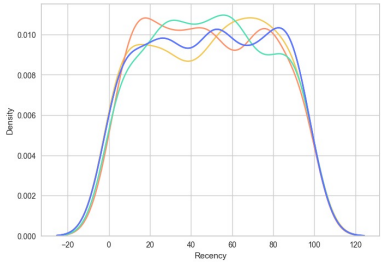

Density
Customer_For

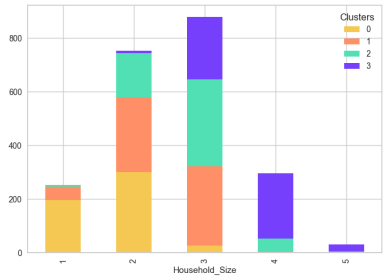

Density
NumWebVisitsMonth

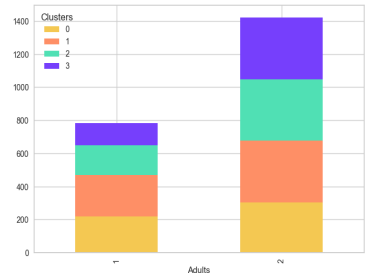

Density
NumStorePurchases

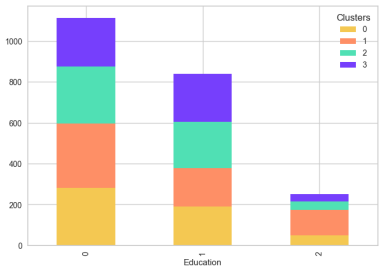

Density
NumCatalogPurchases


Density
NumWebPurchases


Density
Recency


Clusters
0
1
2
3
Household_Size


Clusters
0
1
2
3
Adults


Clusters
0
1
2
3
Education


Cluster's Profile Based On Income And Spending
0
1
2
3
Spent
Income


Cluster's Profile Based On Income And Spending
Spent
Sweets
0
1
2
3


Cluster's Profile Based On Income And Spending
Spent
Fish
0
1
2
3


Cluster's Profile Based On Income And Spending
Spent
Fruits
0
1
2
3


Cluster's Profile Based On Income And Spending
Spent
Meat
0
1
2
3


Cluster's Profile Based On Income And Spending
Spent
Wines
0
1
2
3


Cluster's Profile Based On Income And Spending
Spent
Gold
0
1
2
3

Group 1

- Spend the most
- Income 20-65k
- Shops in store the most
- Highest spenders
- Smallest household (1-2)
- Largest group

Product	Mean
Wine	612.12
Fruit	66.93
Meat	460.27
Fish	98.87
Sweets	70.08
Gold	76.15

Group 2

- 20-80k income
- Youngest (20-50)
- Most likely to shop online
- 2-3 person household
- Spend the least
- Has child at home

Product	Mean
Wine	36.98
Fruit	6.64
Meat	26.00
Fish	10.25
Sweets	6.77
Gold	18.02

Group 3

- Lowest Income (10-50k)
- 40-65 years old
- Longest customers
- 2nd highest spenders
- Online shoppers

Product	Mean
Wine	524.90
Fruit	31.31
Meat	170.78
Fish	41.01
Sweets	31.27
Gold	70.03

Group 4

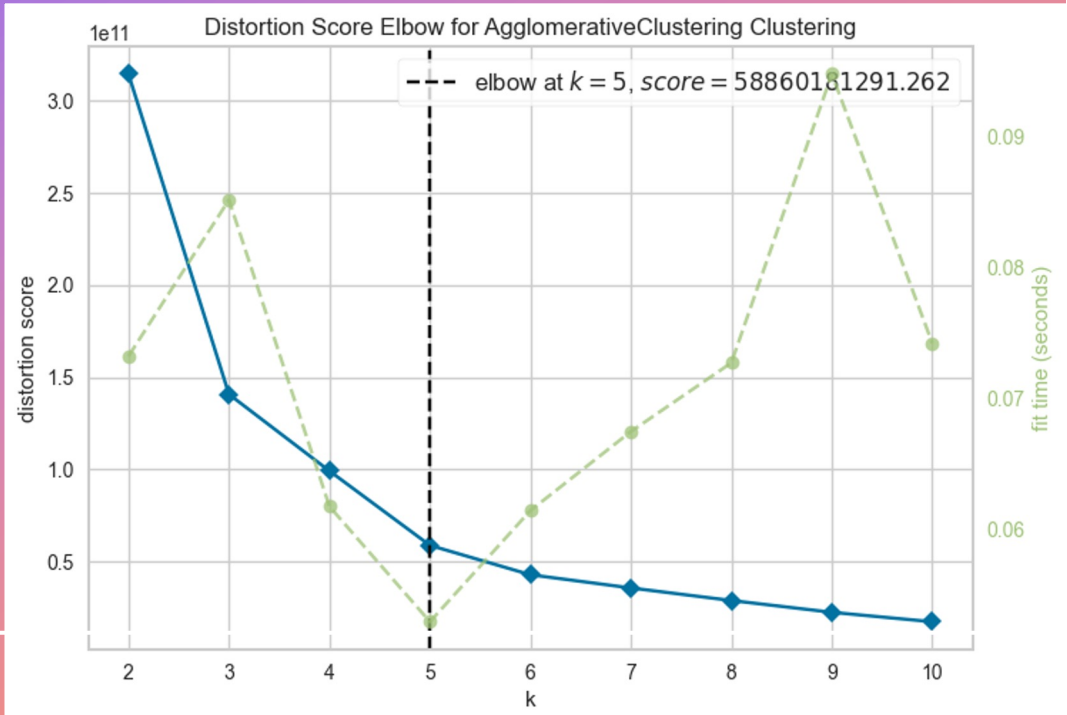
- Largest income (0-100k)
- Mostly 2 adults
- 3-4 person household
- Oldest (40-70)
- Uses most coupons
- Spend the 2nd least
- Has a teenager

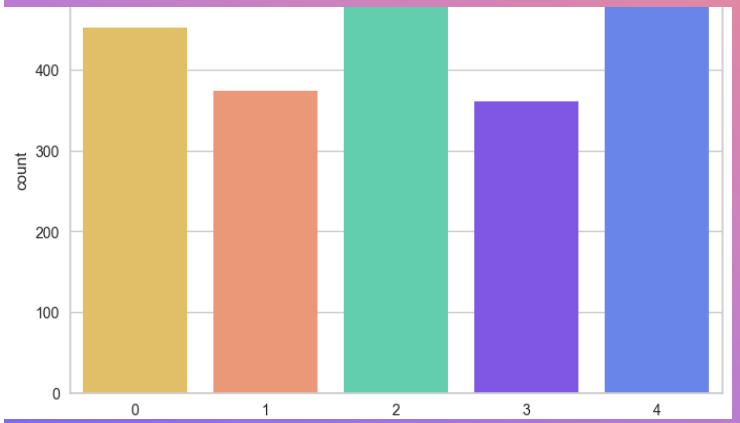
Product	Mean
Wine	90.96
Fruit	4.14
Meat	30.22
Fish	5.82
Sweets	3.95
Gold	15.52

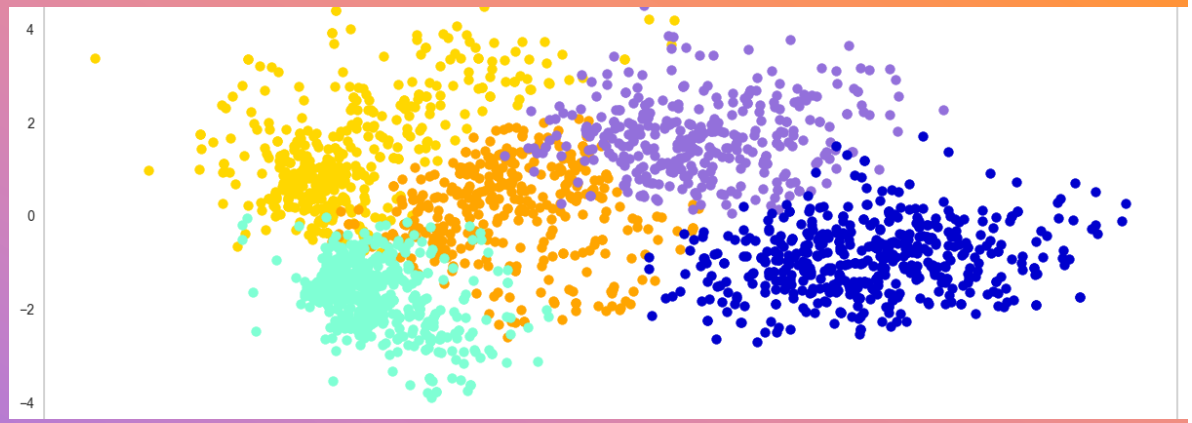
The Groups

- Group 1
  - mostly doesn't have kids
  - Buys the most luxury items
  - Has excess income
  - Is the majority of shoppers
  - Coupon motivated
- Group 2
  - is a small, new family
  - Buys the least luxury items
- Group 3
  - Is likely a single child matured family
  - Buys luxury
- Group 4
  - is a large family
  - Doesn't buy much luxury
  - Spends less on grocery

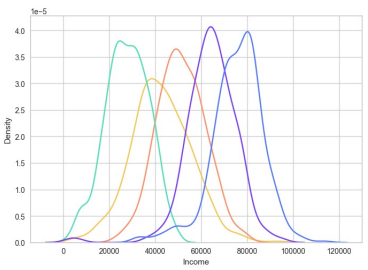
AGGLOMERATIVE (HIEARCHAL)

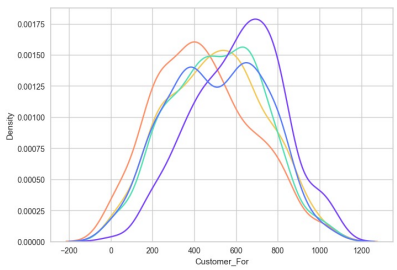


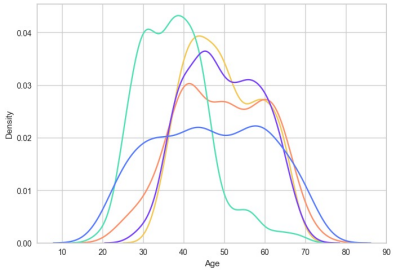


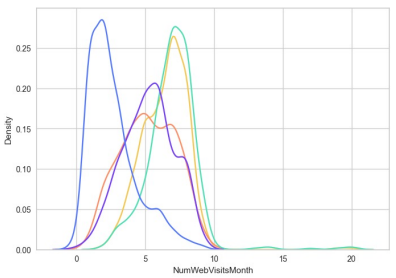


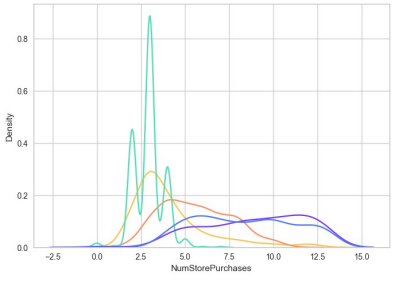
AGGLOMERATIVE DISTRIBUTION

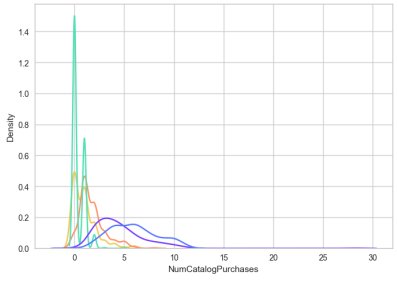

1e−5
Density
Income

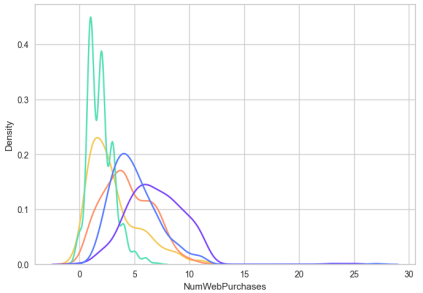

Density
Customer_For

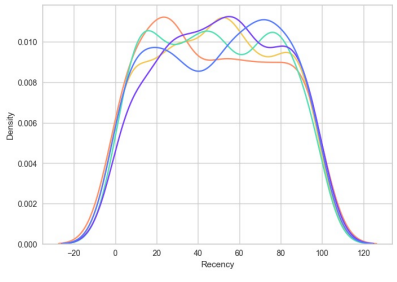

Density
Age

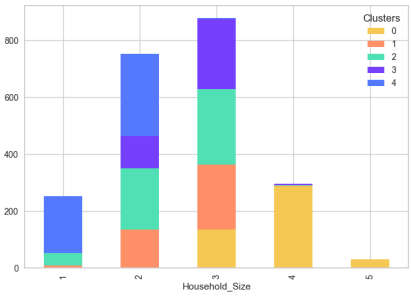

Density
NumWebVisitsMonth

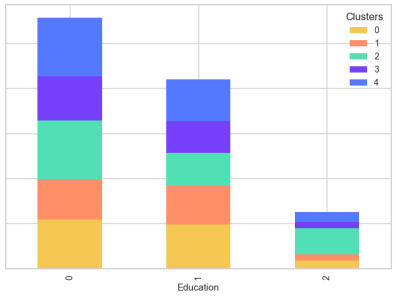

Density
NumStorePurchases

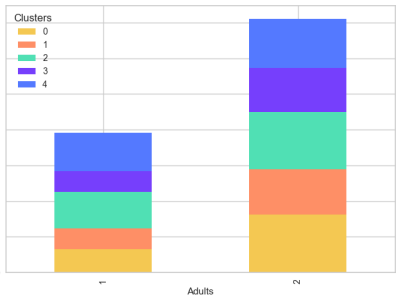

Density
NumCatalogPurchases


Density
NumWebPurchases

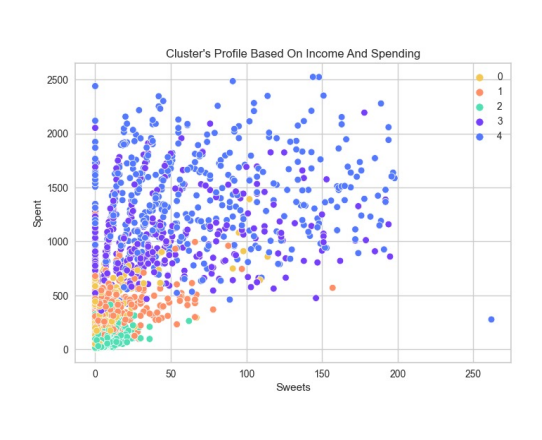

Density
Recency

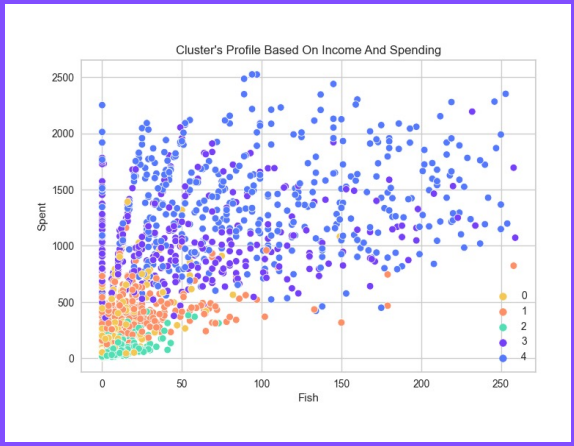

Clusters
0
1
2
3
4
Household_Size


Clusters
0
1
2
3
4
Education

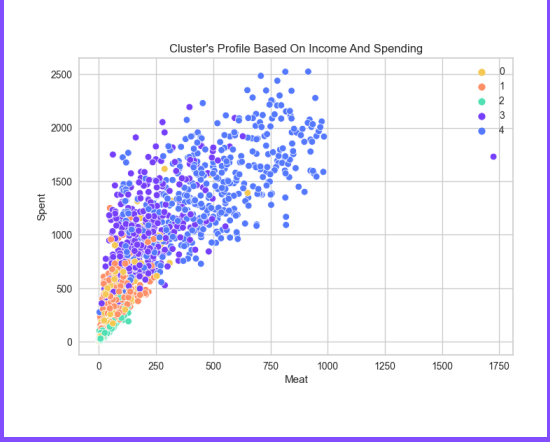

Clusters
0
1
2
3
4
Adults

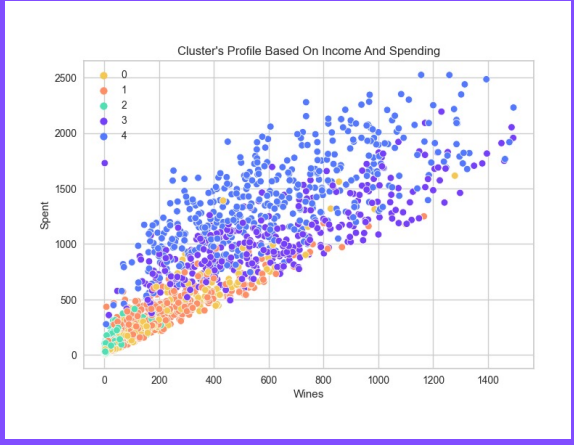

Cluster's Profile Based On Income And Spending
0
1
2
3
4
Spent
Income

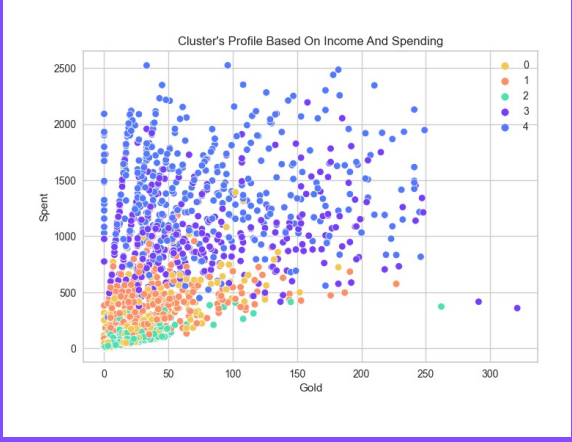

Cluster's Profile Based On Income And Spending
Spent
Sweets
0
1
2
3
4

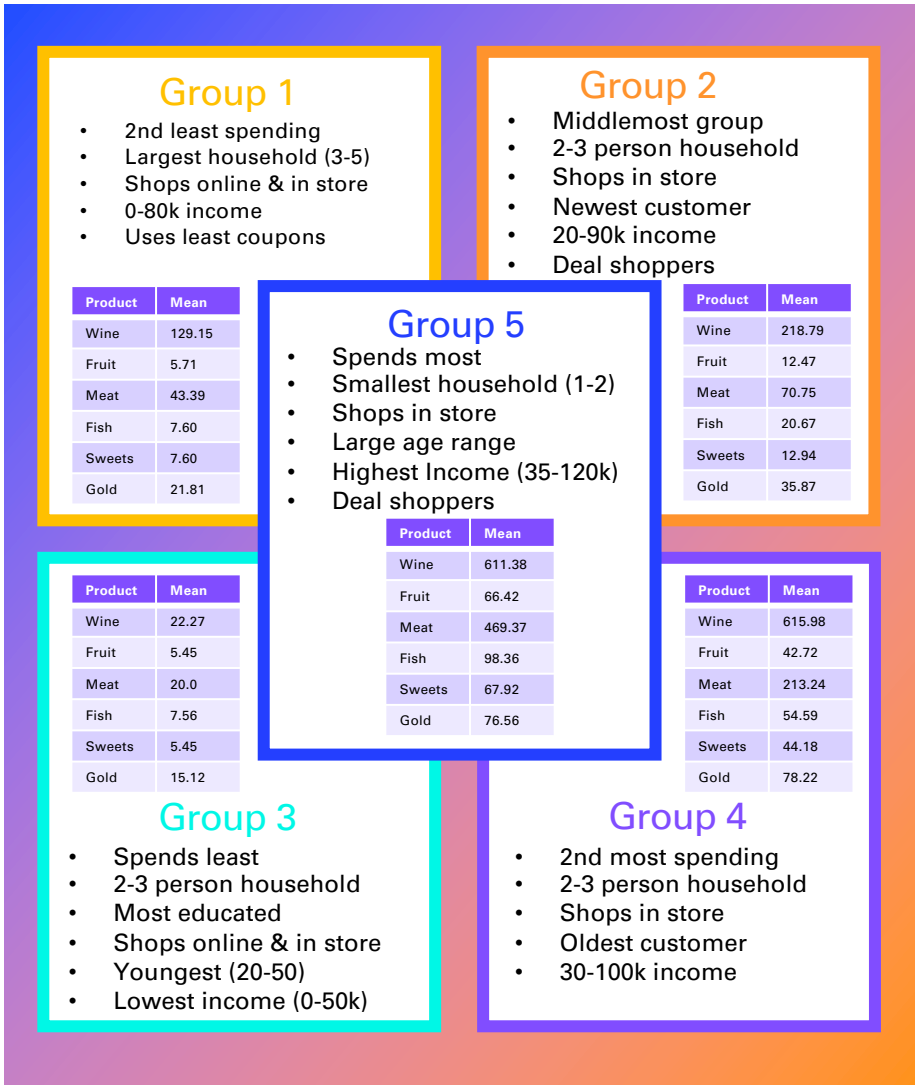

Cluster's Profile Based On Income And Spending
Spent
Fish
0
1
2
3
4


Cluster's Profile Based On Income And Spending
Spent
Fruits
0
1
2
3
4


Cluster's Profile Based On Income And Spending
Spent
Meat
0
1
2
3
4


Cluster's Profile Based On Income And Spending
Spent
Wines
0
1
2
3
4

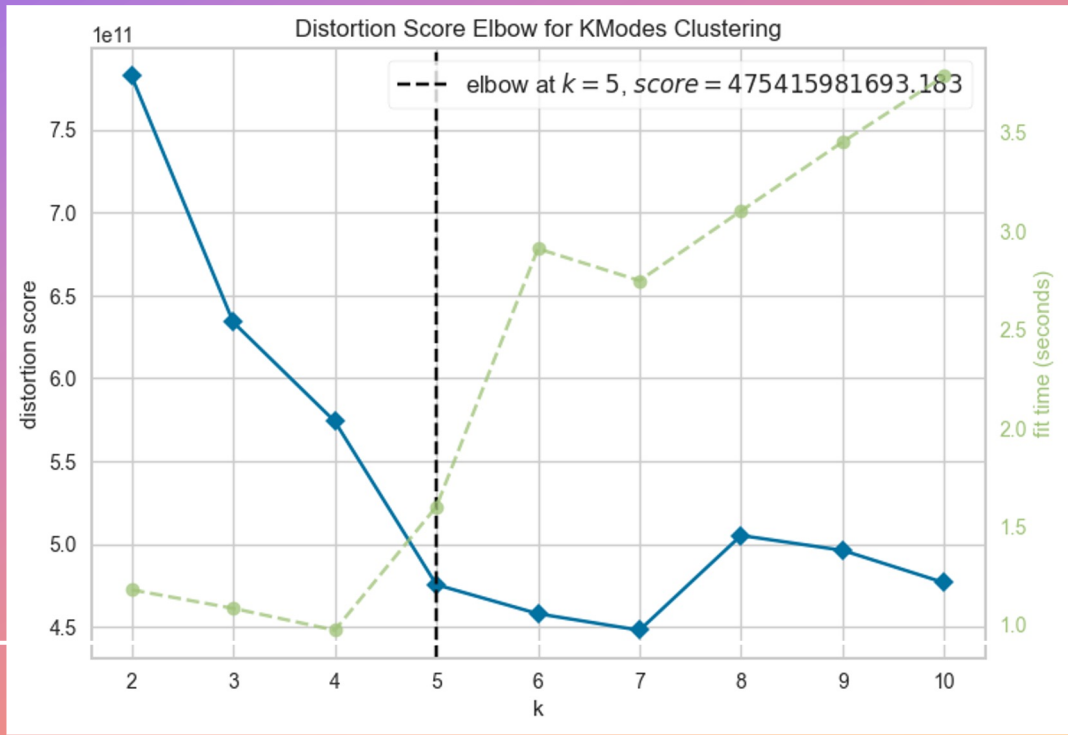

Cluster's Profile Based On Income And Spending
Spent
Gold
0
1
2
3
4

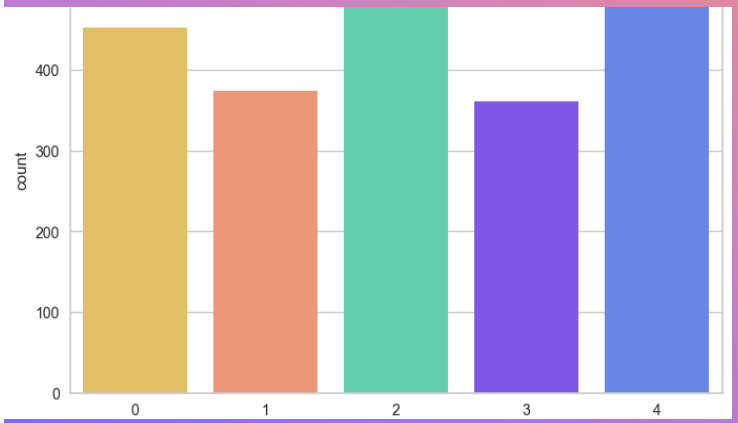


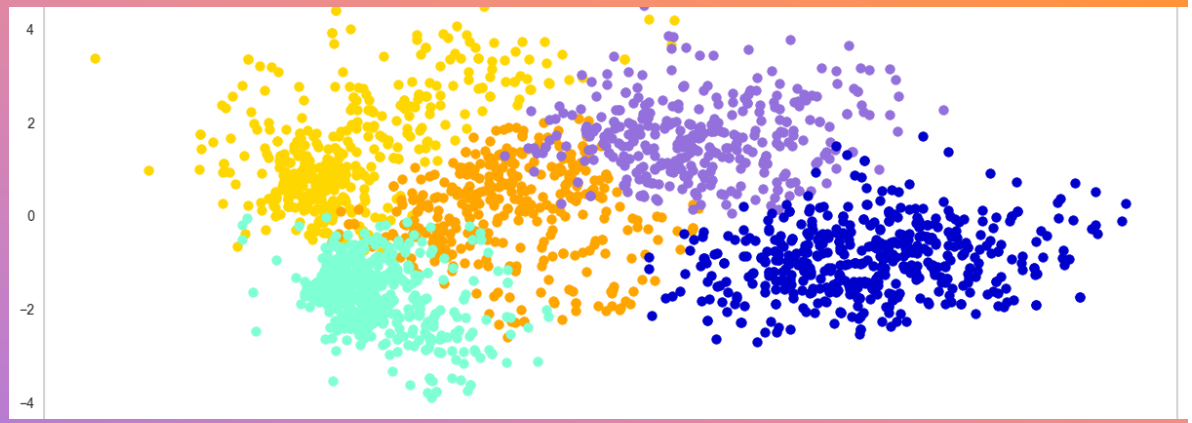
The Groups

- Group 1
  - Large Family
  - Purchase some luxury
- Group 2
  - Older family
  - Buys more luxury
  - Possibly buys better cuts of meat
- Group 3
  - mostly young families
  - Spends equally on grocery and luxury but doesn't purchase much
  - Possibly buys cheapest meats
- Group 4
  - Older family
  - seems to mainly purchase luxury items
- Group 5
  - Adults without kids
  - seems to do all their shopping at the grocery store
  - Most excess income
  - Prefers to spend on groceries than general luxury

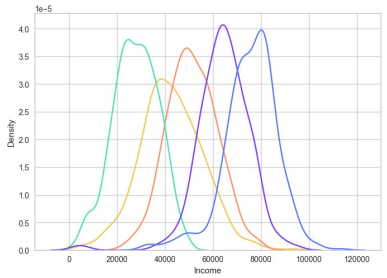
K-MODES

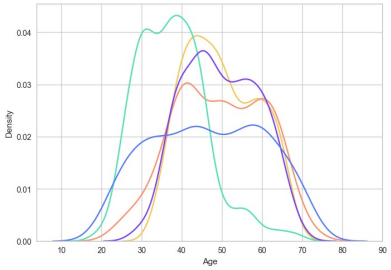

Distortion Score Elbow for KModes Clustering
1e11
elbow at $k = 5$, $score = 475415981693.183$
distortion score
fit time (seconds)
k
7.5
7.0
6.5
6.0
5.5
5.0
4.5
3.5
3.0
2.5
2.0
1.5
1.0
2
3
4
5
6
7
8
9
10

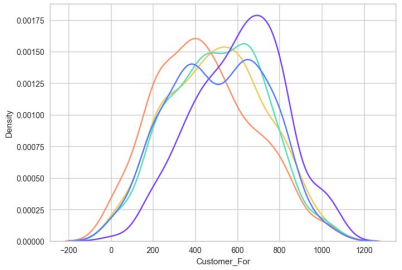


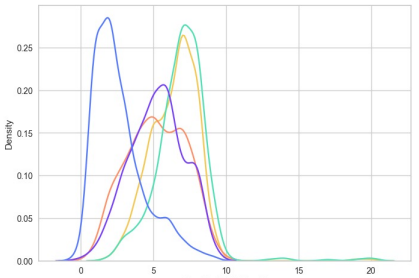


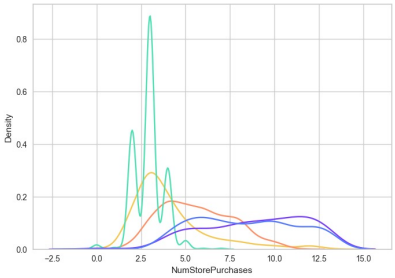
K-MODES

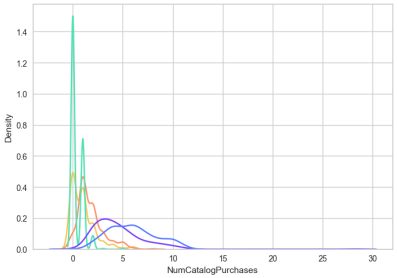

1e−5
Density
Income

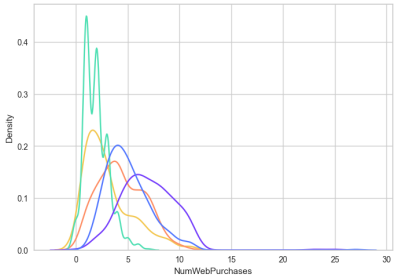

Density
Age

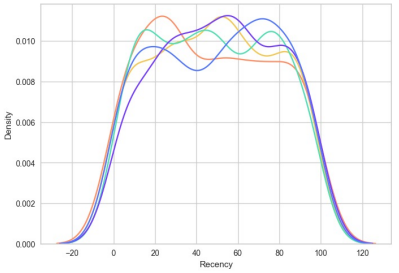

Density
Customer_For

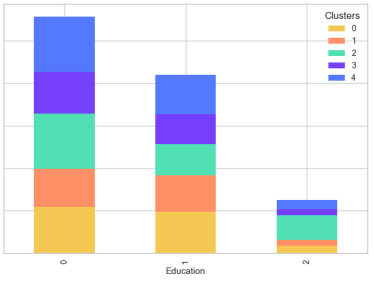

Density
NumWebVisitsMonth

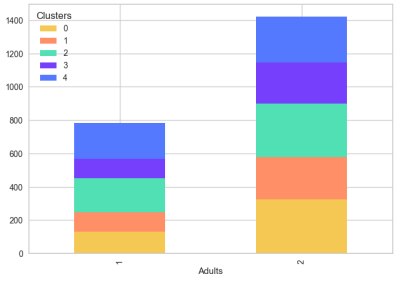

Density
NumStorePurchases

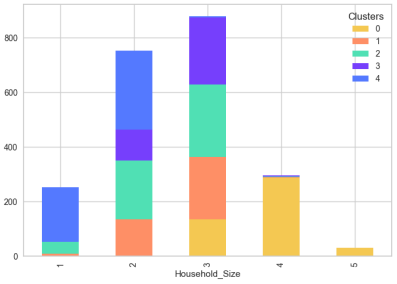

Density
NumCatalogPurchases

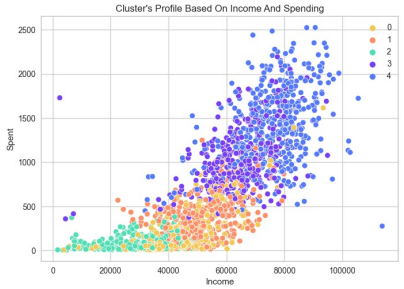

Density
NumWebPurchases

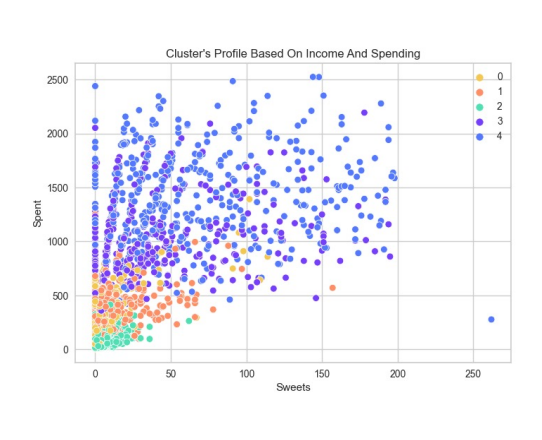

Density
Recency

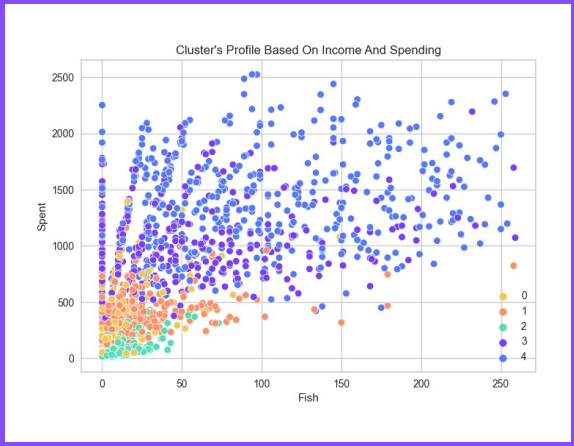

Clusters
0
1
2
3
4
Education


Clusters
0
1
2
3
4
Adults

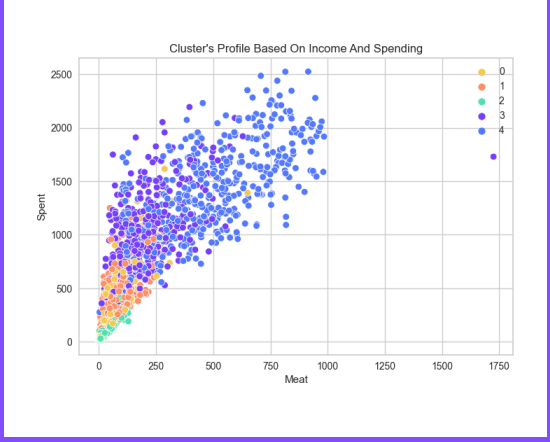

Clusters
0
1
2
3
4
Household_Size

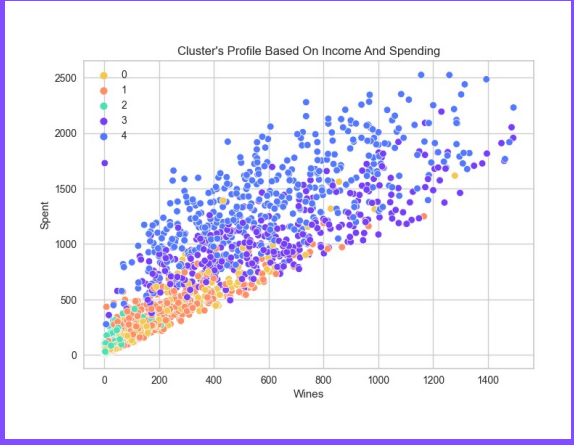

Cluster's Profile Based On Income And Spending
Spent
Income
0
1
2
3
4

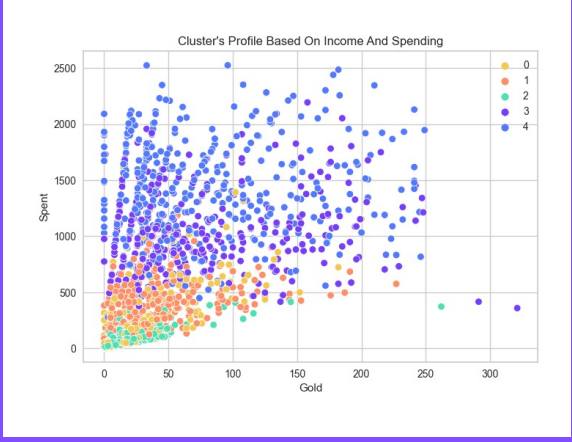

Cluster's Profile Based On Income And Spending
Spent
Sweets
0
1
2
3
4

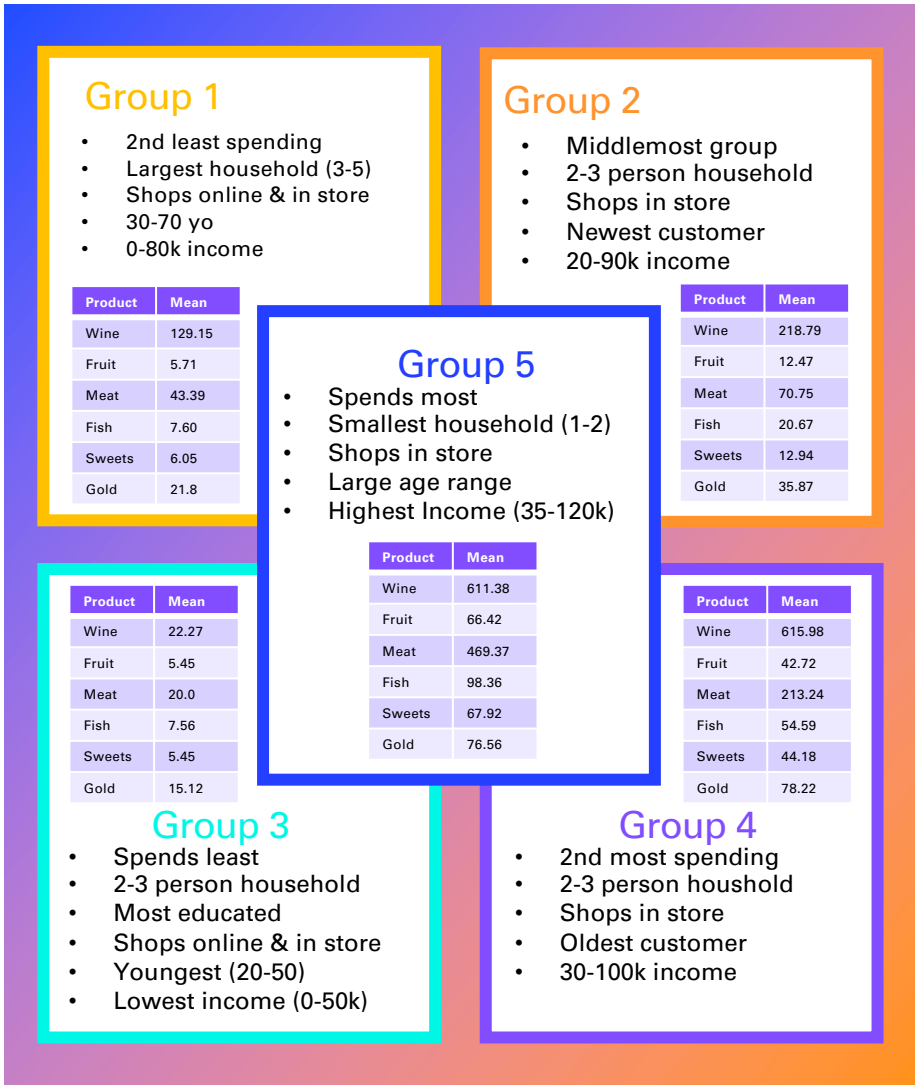

Cluster's Profile Based On Income And Spending
Spent
Fish
0
1
2
3
4


Cluster's Profile Based On Income And Spending
Spent
Fruits
0
1
2
3
4


Cluster's Profile Based On Income And Spending
Spent
Meat
0
1
2
3
4


Cluster's Profile Based On Income And Spending
Spent
Wines
0
1
2
3
4


Cluster's Profile Based On Income And Spending
Spent
Gold
0
1
2
3
4



The Groups

- These groups are Identical to our agglomerative model results.

Strengths/Limitations

- I feel that my dataset ws limited and with a larger dataset I would be able to derive more
- K-Modes was temperamental and was not instant, unlike the other sets.

Conclusion

- Overall I appreciated the K-Means model the most. It gave me the least trouble and gave me an overall better insight to customer interests.
- K-Modes was my least favorite, as it was highly temperamental

Accuracy, Precision, Recall, F1 and when you use them?

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

- Precision tests actual positives over all positives
- Recall tests actual if positives were marked correctly
- Accuracy is related to the number of true negatives
- F1 provides a balance and works best in order to reduce potential loss of resources