Customer Personality Analysis

Group 2 - Hou Bo, Tian Shulin, Wang Yaoxuan

Content

Problem Formulation

About the Dataset

29 Columns & 2240 Entries

Numeric & Categorical Data

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

Project objective:

Summarize the customer segments &
Give advice on how to market different
types of products

Statistical Description

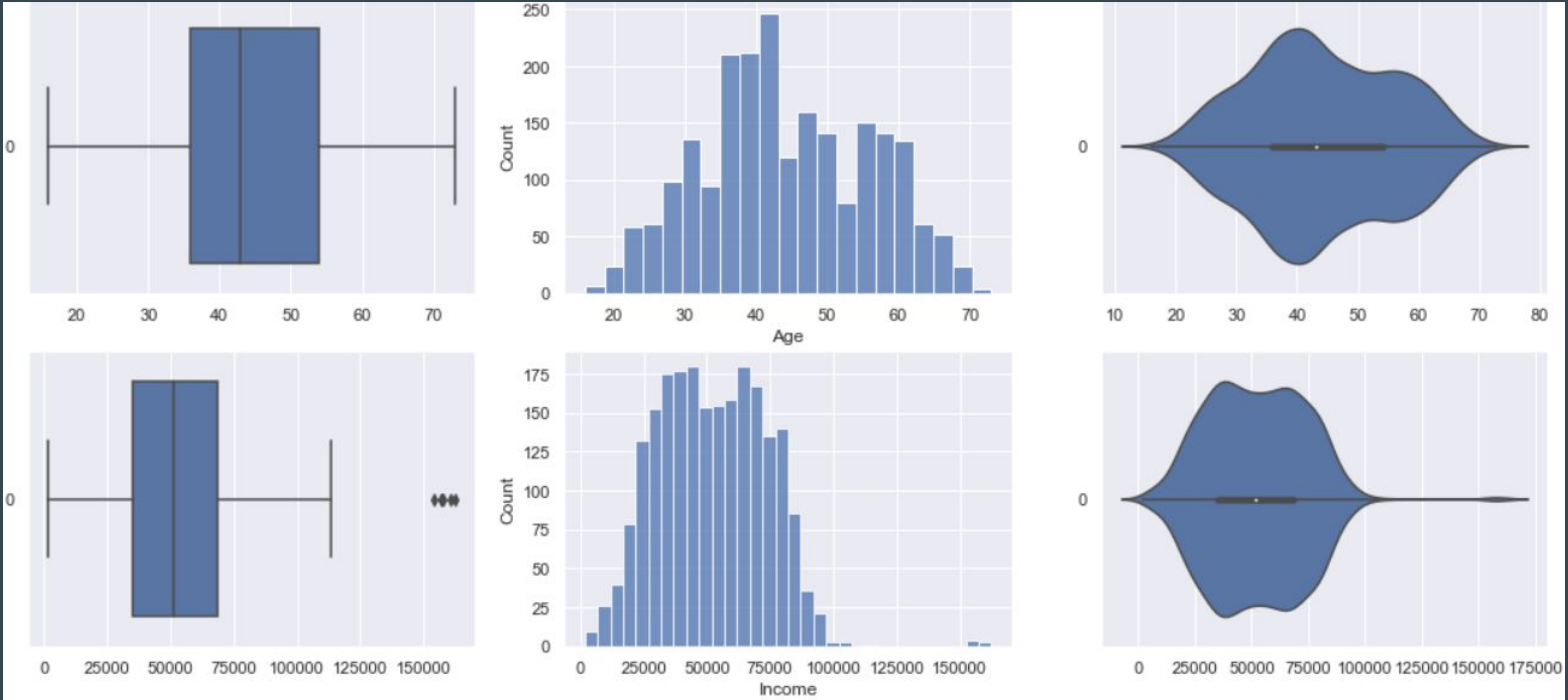
Exploratory Data Analysis

1. Numeric Data: Overall Descriptions

	Age	Income	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	Spent	NumDealsPurchases	NumWebPurchases
count	2212.00	2212.00	2212.00	2212.00	2212.00	2212.00	2212.00	2212.00	2212.00	2212.00	2212.00
mean	44.11	51958.81	305.29	26.33	167.03	37.65	27.05	43.93	607.27	2.32	4.09
std	11.74	21527.28	337.32	39.74	224.25	54.77	41.09	51.71	602.51	1.92	2.74
min	16.00	1730.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00
25%	36.00	35233.50	24.00	2.00	16.00	3.00	1.00	9.00	69.00	1.00	2.00
50%	43.00	51371.00	175.50	8.00	68.00	12.00	8.00	24.50	397.00	2.00	4.00
75%	54.00	68487.00	505.00	33.00	232.25	50.00	33.00	56.00	1048.00	3.00	6.00
max	73.00	162397.00	1493.00	199.00	1725.00	259.00	262.00	321.00	2525.00	15.00	27.00

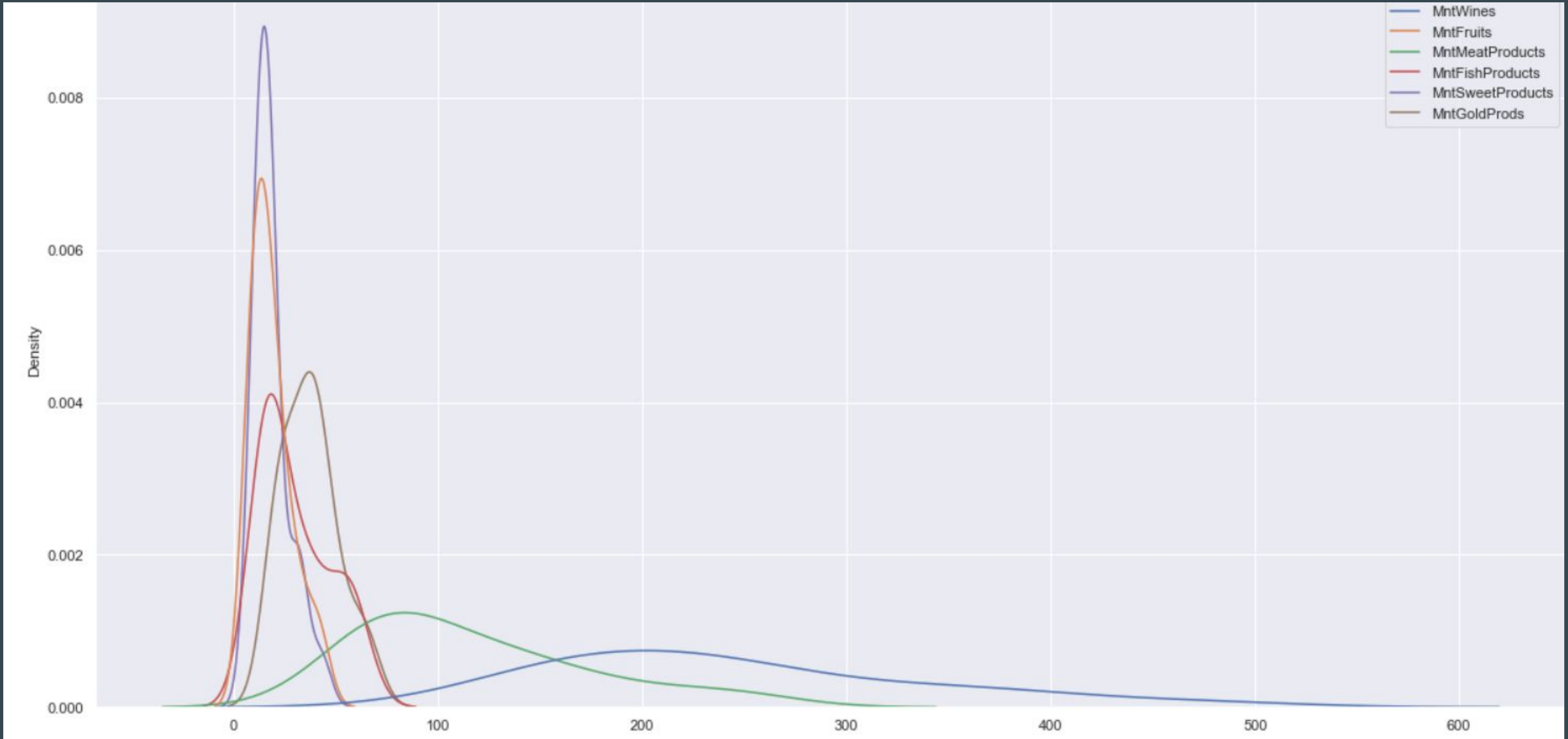
Exploratory Data Analysis

2. Numeric Data: Box-Plot + Histogram + Density



Exploratory Data Analysis

3. Numeric Data (Amount spent on different products): Distribution Plot

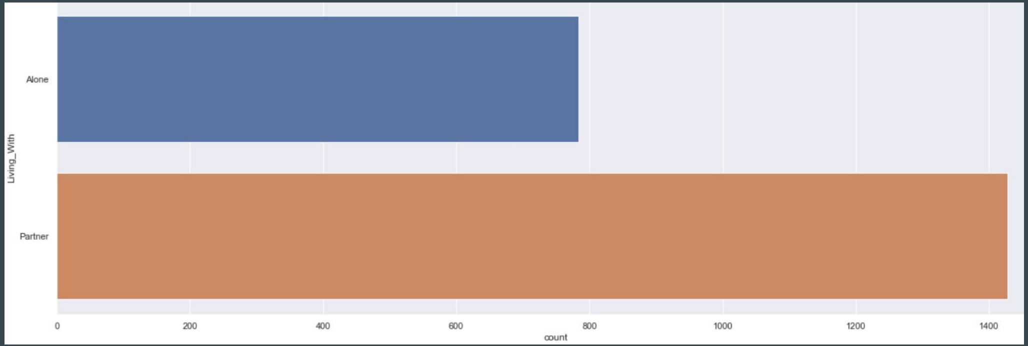


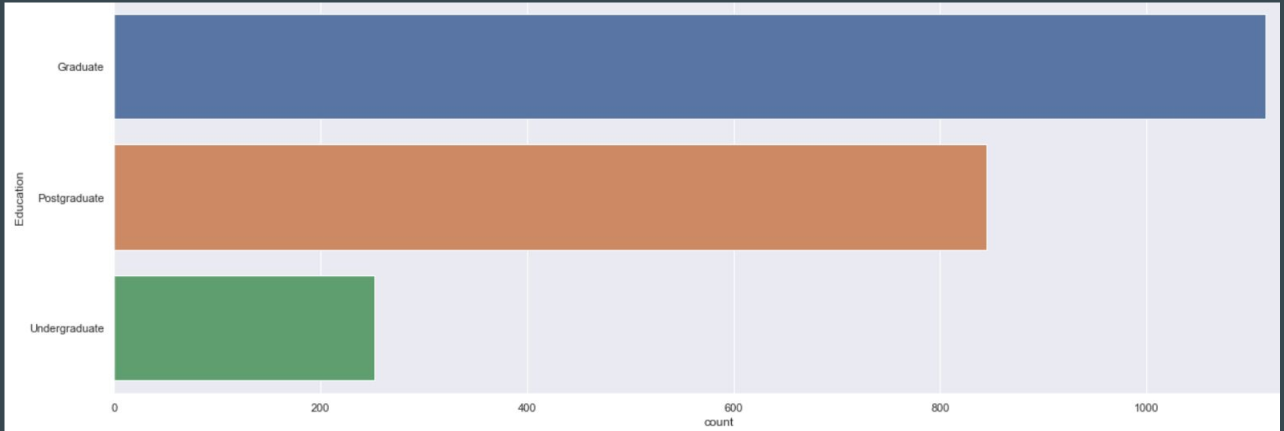
Exploratory Data Analysis

4. Numeric Data (Number of purchases in different places): Distribution Plot

Exploratory Data Analysis

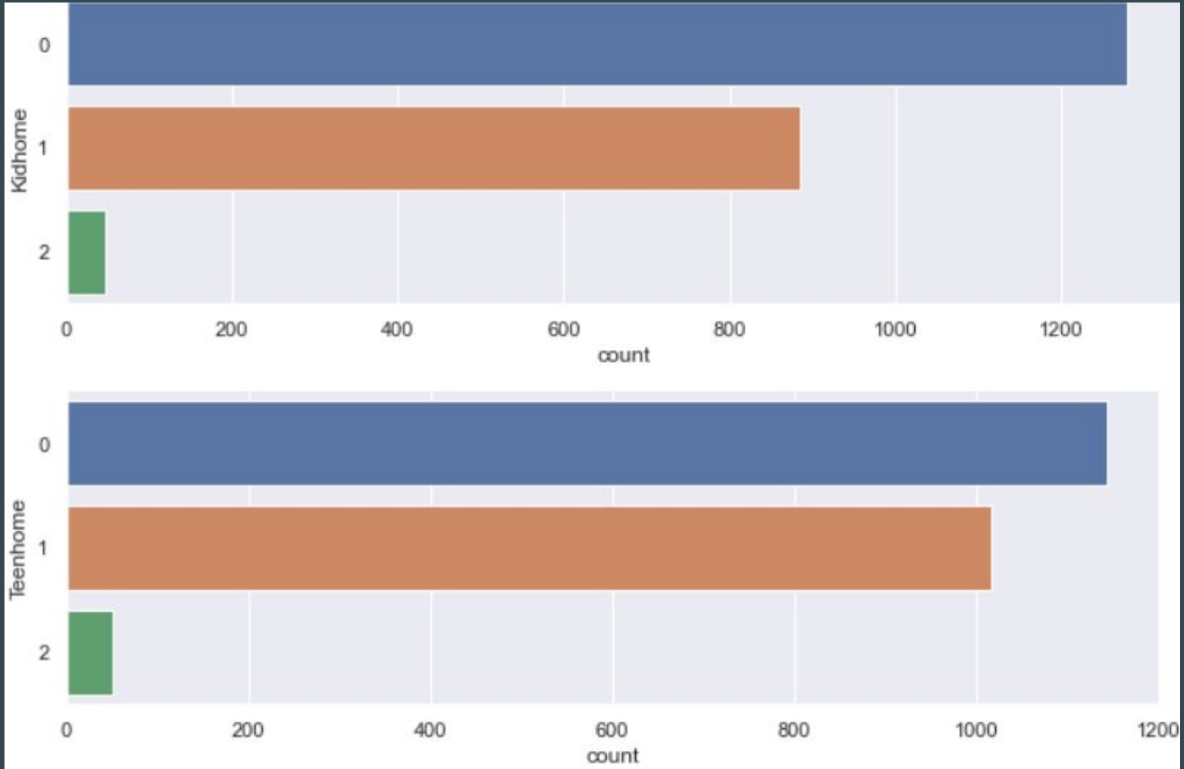
5. Categorical Data





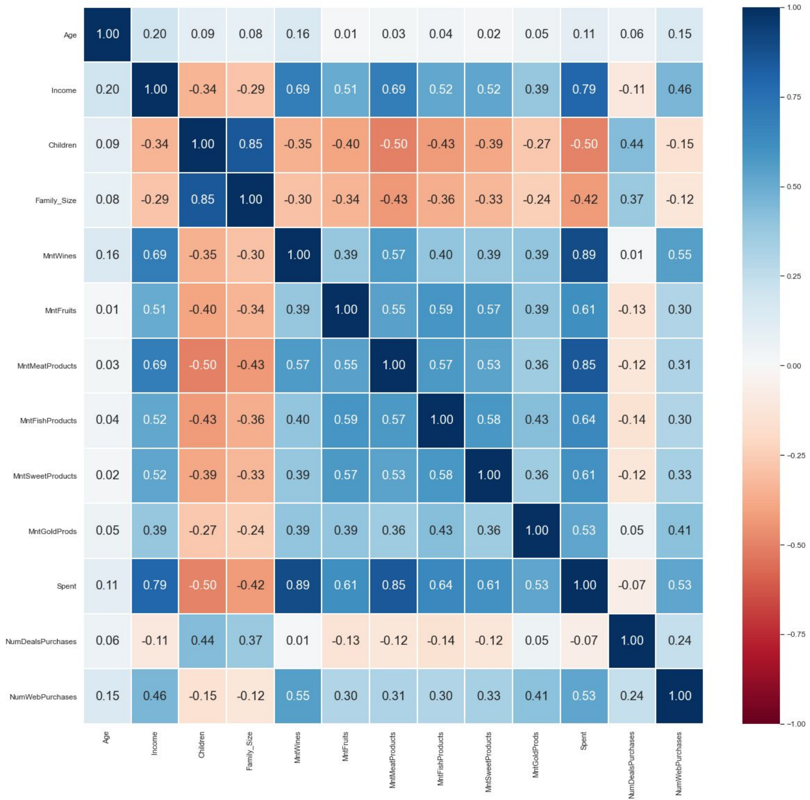
Exploratory Data Analysis

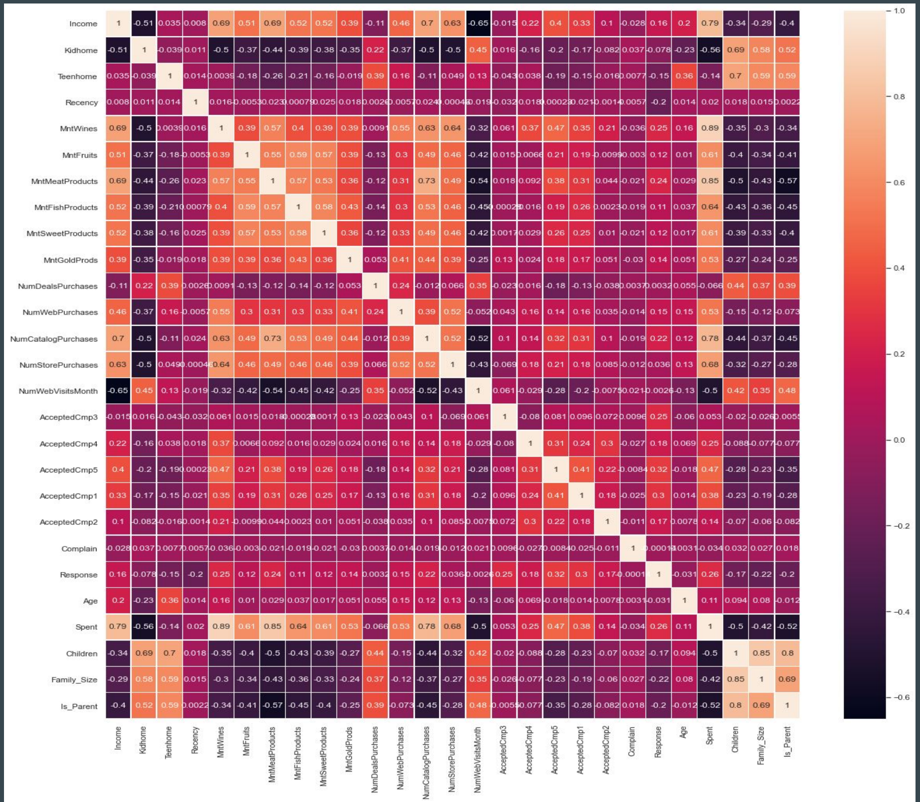
5. Categorical Data



Exploratory Data Analysis

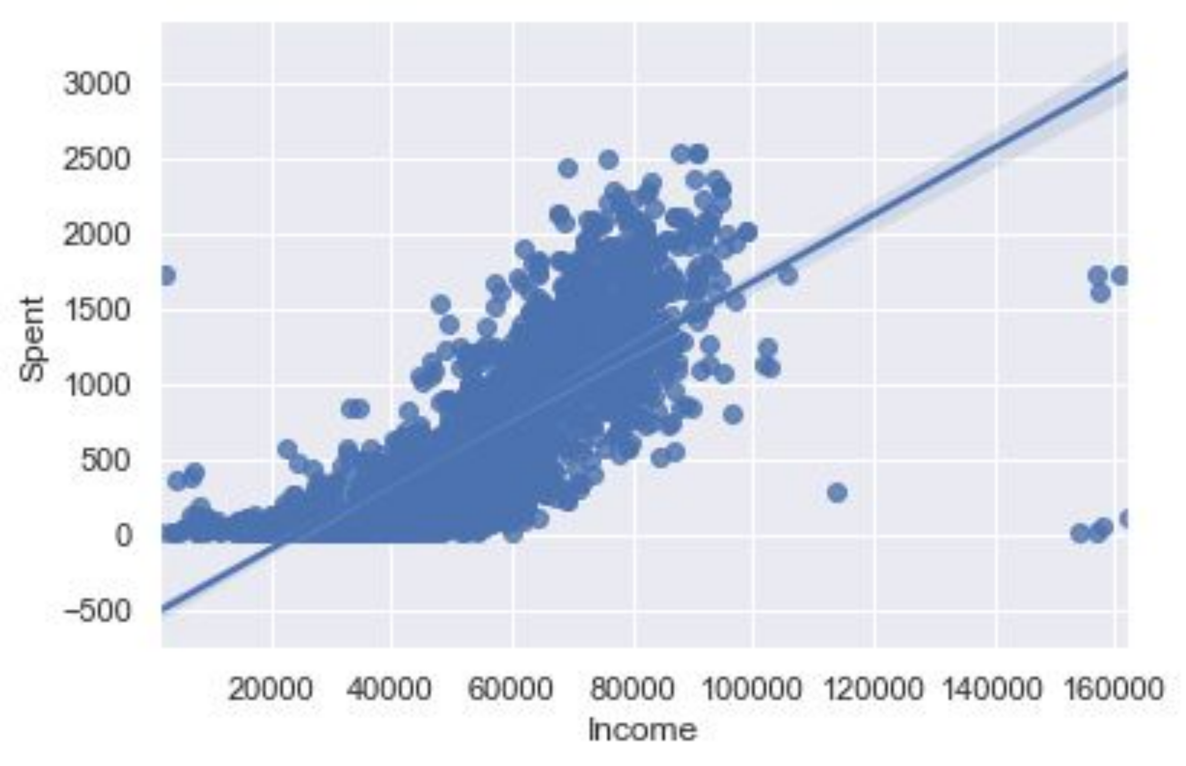
6. Correlation: Heat Map





Exploratory Data Analysis

7. Correlation: Regplot



Structure Detection -> Clustering

Data Preparation

1. NA Values

- 2240 Data points 🆚 24 NA values in “Income”
- => Drop

```
In [556]: custdata.isna().any()
```

```
Out[556]: ID                  False
          Year_Birth          False
          Education           False
          Marital_Status      False
          Income               True
          Kidhome             False
          Teenhome            False
          Dt_Customer         False
          Recency             False
```

Data Preparation

2. Convention Representation

- “Year_Birth” => “Age”

```python
# Age of customers
import datetime
for i in range(2216):
    #transform Dt_customer to standard timestamp
    custdata['Dt_Customer'][i] = datetime.datetime.strptime(str(custdata['Dt_Customer'][i]), "%d-%m-%Y").strftime(me("%Y-%m-%d")
    # access the YYYY of timestamp
    custdata['Dt_Customer'][i] = int(str(custdata['Dt_Customer'][i]).split('-')[0])

custdata["Age"] = custdata["Dt_Customer"] - custdata["Year_Birth"]
custdata["Age"] = custdata["Age"].astype(int)
custdata.info()
```

Data Preparation

3. Outliers

- Drop outliers by calculating “Z-Score”

1. Age

```
# Drop outliers of age by calculating Z-Score

from scipy import stats

custdata["z_value_age"] = np.abs(stats.zscore(custdata["Age"]))
custdata["z_value_age"]
```

```
In [9]: threshold = 3
        z1  = np.abs(stats.zscore(custdata["Age"]))
        np.where(z1>3)
```

```
Out[9]: (array([181, 228, 326]),)
```

```
In [10]: custdata.iloc[np.where(z1>3)]
```

Out[10]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	AcceptedCmp4	AcceptedCmp5	Accept
181	7829	1900	2n Cycle	Divorced	36640.0	1	0	2013	99	15	...	0	0	
228	11004	1893	2n Cycle	Single	60182.0	0	1	2014	23	8	...	0	0	
326	1150	1899	PhD	Together	83532.0	0	0	2013	36	755	...	0	1	

Data Preparation

4. Feature Engineering

- Add attributes and use conventional representation

2. Feature Engineering

```
#Total spendings on various items
custdata["Spent"] = custdata["MntWines"]+ custdata["MntFruits"]+ custdata["MntMeatProducts"]+ custdata["MntFishProducts

#Deriving living situation by marital status"Alone"
custdata["Living_With"]=custdata["Marital_Status"].replace({"Married":"Partner", "Together":"Partner", "Absurd":"Alone"

#Feature indicating total children living in the household
custdata["Children"]=custdata["Kidhome"]+custdata["Teenhome"]

#Feature for total members in the householde
custdata["Family_Size"] = custdata["Living_With"].replace({"Alone": 1, "Partner":2})+ custdata["Children"]

#Feature pertaining parenthood
custdata["Is_Parent"] = np.where(custdata.Children> 0, 1, 0)

#Segmenting education levels in three groups
custdata["Education"]=custdata["Education"].replace({"Basic":"Undergraduate","2n Cycle":"Undergraduate", "Graduation":"

#Dropping some of the redundant features
to_drop = ["ID", "Year_Birth", "Marital_Status", "Dt_Customer", "Z_CostContact", "Z_Revenue", "z_value_age", "z_value_i
custdata = custdata.drop(to_drop, axis=1)
```

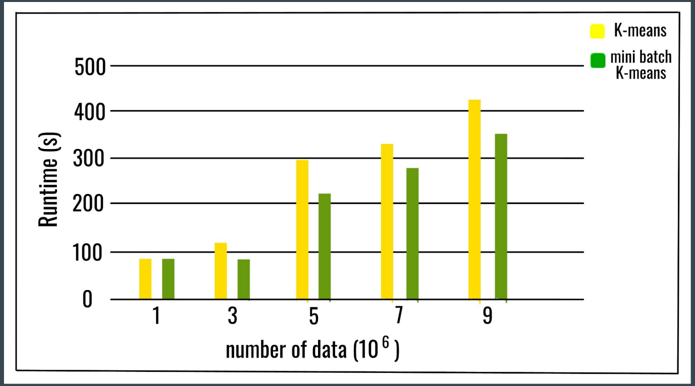
Machine Learning

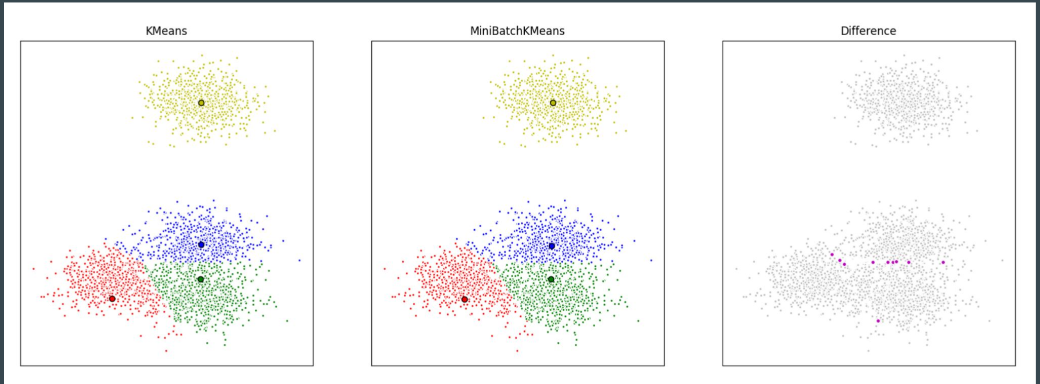
K-Means++ 4 steps to initialize centroids

Phase 1: Select	Select a random first centroid point from the given dataset.
Phase 2: Calculate	Calculate the distance from every instance to the closest, previously chosen centroid.
Phase 3: Select	Select the following centroid（the likelihood of picking a point as centroid is corresponding to the distance from phase 2.)
Phase 4: Repeat	Last 2 steps are repeated until you get k mean points.

MiniBatch K-Means

Phase 1: Draw random sample	Small random batches of data of fixed size
Phase 2: Iteration	Update clusters
Phase 3: Convergence	No changes in the clusters

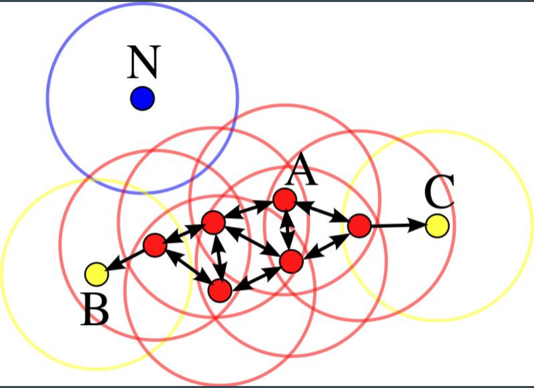




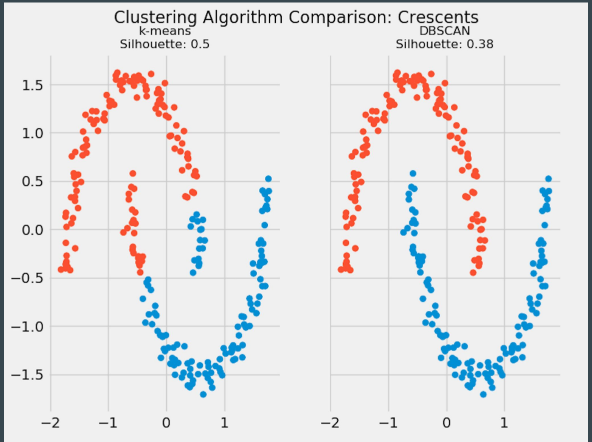
Sources: https://www.geeksforgeeks.org/ml-mini-batch-k-means-clustering-algorithm/

DBSCAN

Phase 1: Parameters	ε & minimum number of points to form a cluster
Phase 2: Formation	Samples/instances are located within the ε-neighborhood
Phase 3: Noise Detection	Data points that are not within the ε-neighborhood



Sources: https://en.wikipedia.org/wiki/DBSCAN



Sources:https://realpython.com/k-means-clustering-python/

BIRCH (balanced iterative reducing and clustering using hierarchies)

Phase 1	• Build a clustering feature (CF) tree
Phase 2 (Optional)	• Scan all leaf entries to rebuild a smaller CF tree
Phase 3	• Obtain clusters
Phase 4 (Optional)	• Redistribute data points

Which one to choose

- Calculating silhouette score (S_score)

1. K-Means++

```
In [51]: S_score(X_labeled_KM, labels, metric='euclidean', sample_size=None, random_state=None)
Out[51]: 0.25036933749708906
```

2. Mini Batch K-Means

```
In [63]: S_score(X_labeled_MBK, labels, metric='euclidean', sample_size=None, random_state=None)
Out[63]: 0.18440301494692635
```

3. DBSCAN

```
In [74]: S_score(X_labeled_DP, labels, metric='euclidean', sample_size=None, random_state=None)
Out[74]: 0.15981687335223357
```

4. BIRCH

```
In [82]: S_score(scaled_features_df, result, metric='euclidean', sample_size=None, random_state=None)
Out[82]: 0.11816302751380603
```

What we have learned so far

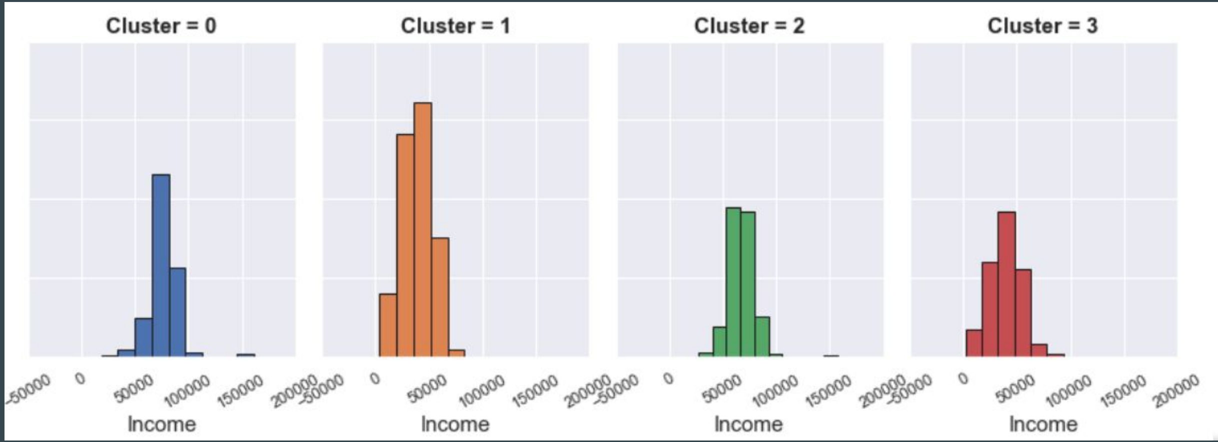
1. Machine learning methods
2. EDA visualization methods
3. Evaluate the effectiveness by calculating numeric parameters.

Cluster Analysis & Conclusion

Cluster 0: has the highest income & highest spending.

Cluster 2: the 2nd highest
Cluster 1: the 3rd highest

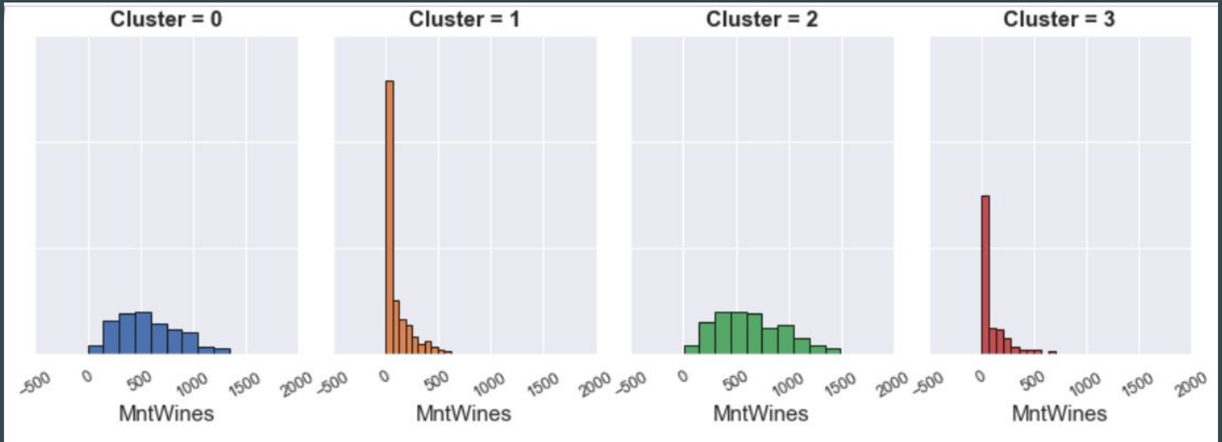
Cluster3: lowest income & spending

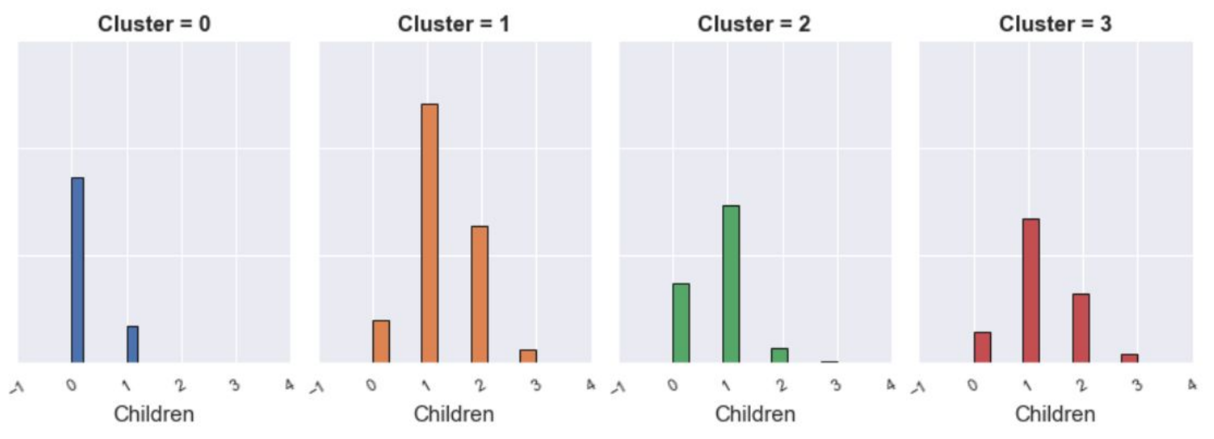


Conclusions & Fun Facts

Most popular product: Wines

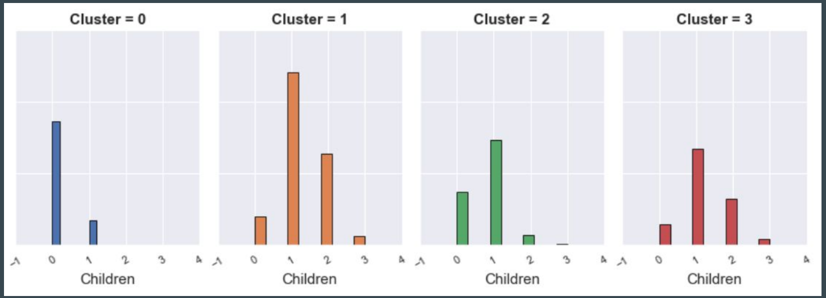
Wines and kids

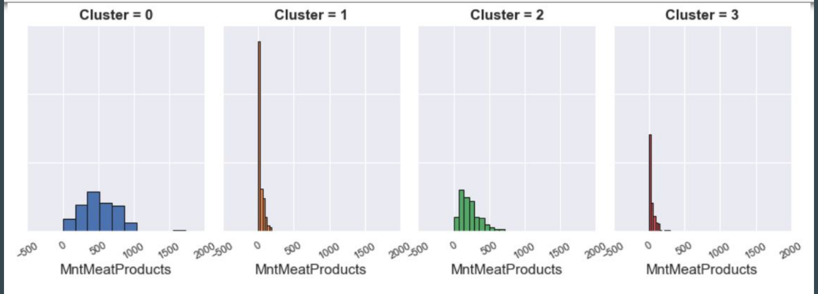




Conclusions & Fun Facts

Family size
and
purchasing amount





The Team & Task Distribution

Hou Bo

1. Exploratory Data Analysis: Categorical Data Visualization
2. Machine Learning: K-Means++
3. Cluster Analysis: Cluster Distribution

Tian Shulin

1. Data Preparation
2. Exploratory Data Analysis: Numeric Data Overall Visualization & Heat Map
3. Machine Learning: MiniBatch KMeans & DBSCAN
4. Cluster Analysis: Cluster Interpretation

Wang Yaoxuan

1. Exploratory Data Analysis: Distribution Plots & Regression Plots
2. Machine Learning: BIRCH
3. Cluster Analysis: Products Distribution by Clusters