SC1015 MINI-PROJECT

Customer Personality Analysis

FCSC Group 7
Tam Yik Lock (U2222001C)
Xu Jialu (U2220758B)
Zhou Ziheng (U2222255F)

CONTENTS

PRACTICAL MOTIVATIONS

Customer personality analysis is the process of **understanding and categorising customers** based on their characteristics, preferences, behaviours, and other relevant factors.

It involves **gathering customer data** and then using techniques such as **machine learning clustering** to group customers into **distinct segments**.

CUSTOMER PERSONALITY ANALYSIS IS A POWERFUL TOOL

Customer Segmentation

Identify distinct customer groups based on shared characteristics or behaviours

Personalised Marketing

Tailor marketing messages and offers to specific customer segments for better engagement and conversion

Competitive Advantage

Stay ahead of competitors by understanding and meeting customer needs more effectively

Resource Optimisation

Allocate resources such as R&D and customer support more effectively by focusing on high-value segments

DATA SCIENCE PROBLEM

Given a dataset of customer details (age, income, family status etc.) and their corresponding activities history (such as spending)

Perform clustering to identify customer segments and generate insights for use in targeted campaigns

ABOUT THE kaggle DATASET

The dataset consists of **2240 datapoints** and **29 attributes** that can be categorised under **4 subsets**

Customer's Information

- ID
- Year_Birth
- Education
- Marital_Status
- Income
- Kidhome
- Teenhome
- Dt_Customer
- Recency
- Complain

Products

Amount spent on different products in the last 2 years:

- MntWines
- MntFruits
- MntMeatProducts
- MntFishProducts
- MntSweetProducts
- MntGoldProds

Promotion

- NumDealsPurchases
- AcceptedCmp1
- AcceptedCmp2
- AcceptedCmp3
- AcceptedCmp4
- AcceptedCmp5
- Response

Place

- NumWebPurchases
- NumCatalogPurchases
- NumStorePurchases
- NumWebVisitsMonth

DATA CLEANING

2240 entries of data
2216 non-null valid entries

Characteristics of Input data:

- Data incompleteness in customer income level
- Categorical variables such as Marital Status, Education need to be encoded numerically
- Features need simplifications and generalisation i.e. education status
- Date and Time are not parsed as DateTime objects

DATA CLEANING

Before we continue with analysis, we need to **numerically encode and simplify** some categorical variables, such as marital status and education

Exploring Categorical Values

Next, we take a closer look at the catergorical variables in the dataset.

In 24

```
print(
    "Total categories in the feature Marital_Status:\n",
    data["Marital_Status"].value_counts(),
    "\n",
)
print("Total categories in the feature Education:\n", data["Education"].value_counts())
```

Executed at 2024.04.21 23:57:15 in 49ms

```
Total categories in the feature Marital_Status:
 Marital_Status
Married     857
Together    573
Single      471
Divorced    232
Widow        76
Alone         3
Absurd        2
YOLO          2
Name: count, dtype: int64

Total categories in the feature Education:
 Education
Graduation    1116
PhD            481
Master         365
2n Cycle       200
Basic           54
Name: count, dtype: int64
```

DATA CLEANING

For **marital status** we simplify the categories into either living alone or with a partner

For **education** we also simplify and encode the categories into undergraduate, postgraduate and graduate

i.e. categorising all customers into EITHER “Alone” or “Partner”

```python
# Deriving living situation by marital status "Alone"
data["Living_With"] = data["Marital_Status"].replace(
    {
        "Married": "Partner",
        "Together": "Partner",
        "Absurd": "Alone",
        "Widow": "Alone",
        "YOLO": "Alone",
        "Divorced": "Alone",
        "Single": "Alone",
    }
)
print(data["Living_With"])
```

Executed at 2024.04.22 17:39:34 in 71ms

```
0         Alone
1         Alone
2       Partner
3       Partner
4       Partner
         ...
2235    Partner
2236    Partner
2237      Alone
2238    Partner
2239    Partner
Name: Living_With, Length: 2216, dtype: object
```

DATA CLEANING

We also modify the **Dt_Customer** variable, which describes the customer's date of enrolment to the outlet.

We engineer a feature called "**Customer_for**" instead, a numerical number of days the customer has been enrolled for, for future analysis.

```
data["Dt_Customer"] = pd.to_datetime(data["Dt_Customer"], format="%d-%m-%Y")
dates = []
for i in data["Dt_Customer"]:
    i = i.date()
    dates.append(i)
# Dates of the newest and oldest recorded customer
print("The newest customer's enrolment date in the records:", max(dates))
print("The oldest customer's enrolment date in the records:", min(dates))
```

Executed at 2024.04.22 17:39:34 in 88ms

```
The newest customer's enrolment date in the records: 2014-06-29
The oldest customer's enrolment date in the records: 2012-07-30
```

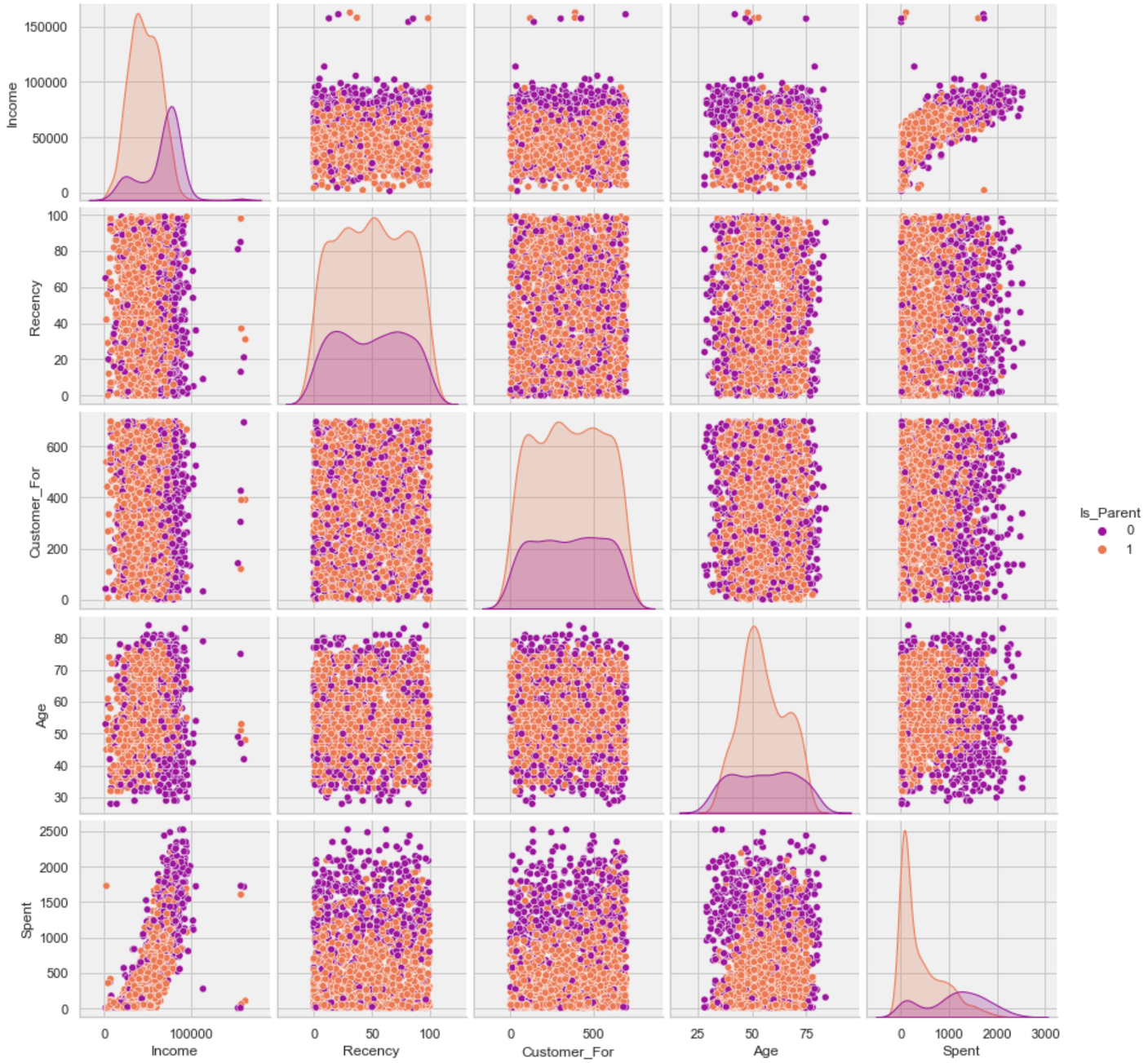
Creating a new feature Customer_For to record the number of numerical days the customer has been enrolled.

```
# Created a feature "Customer_For"
days = []
d1 = max(dates)  # taking it to be the newest customer
for i in dates:
    delta = d1 - i
    days.append(delta.days)
data["Customer_For"] = days
data["Customer_For"] = pd.to_numeric(data["Customer_For"], errors="coerce")
print(data["Customer_For"])
```

Executed at 2024.04.22 17:39:34 in 77ms

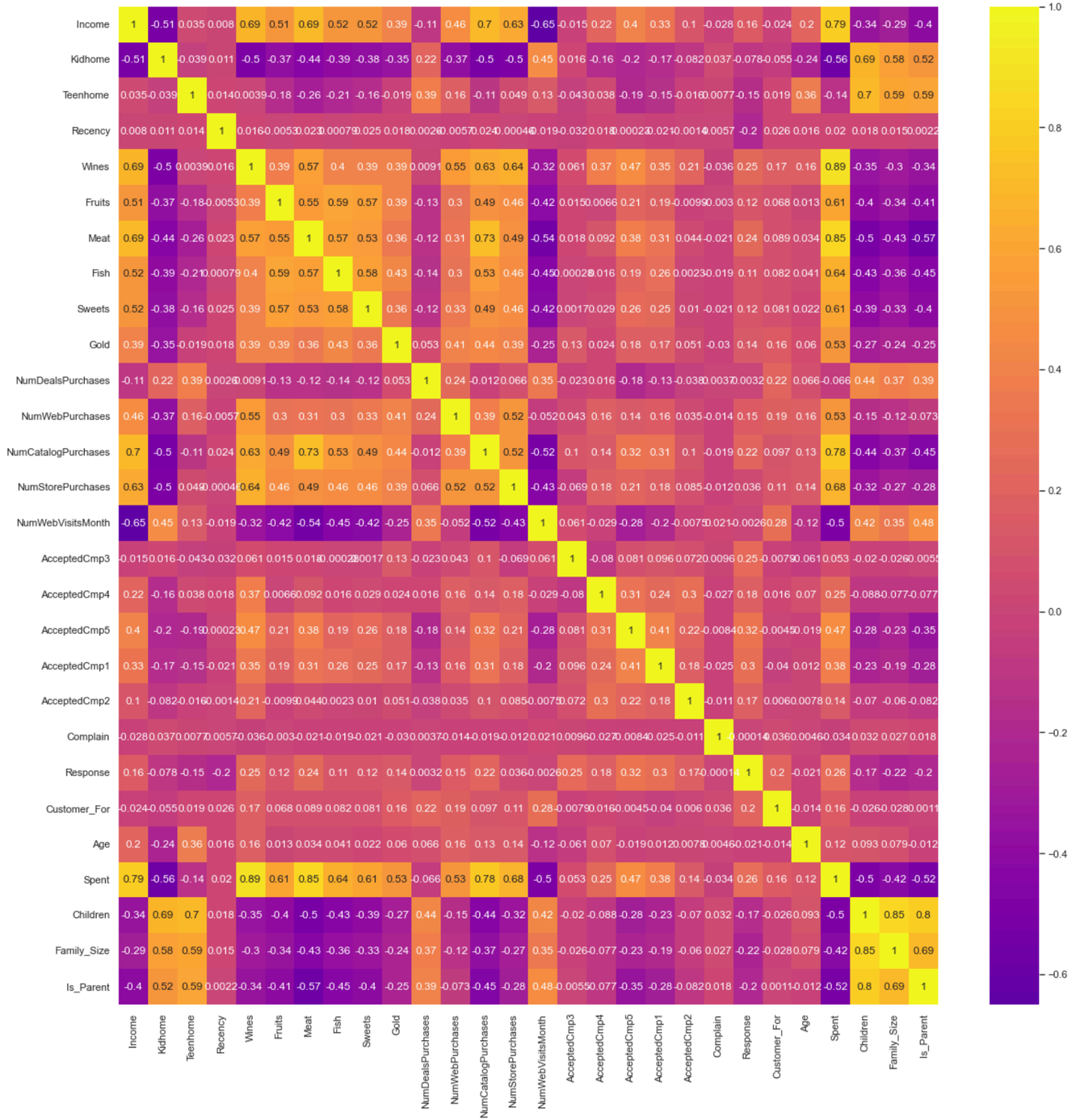
EXPLORATORY ANALYSIS

We use a pairplot to obtain a broad view of the data and it’s characteristics, and chose a few variables as our samples



CORRELATION MATRIX

To observe bivariate correlation patterns across the board, we use a correlation matrix to find strong correlations in both the positive and negative directions.



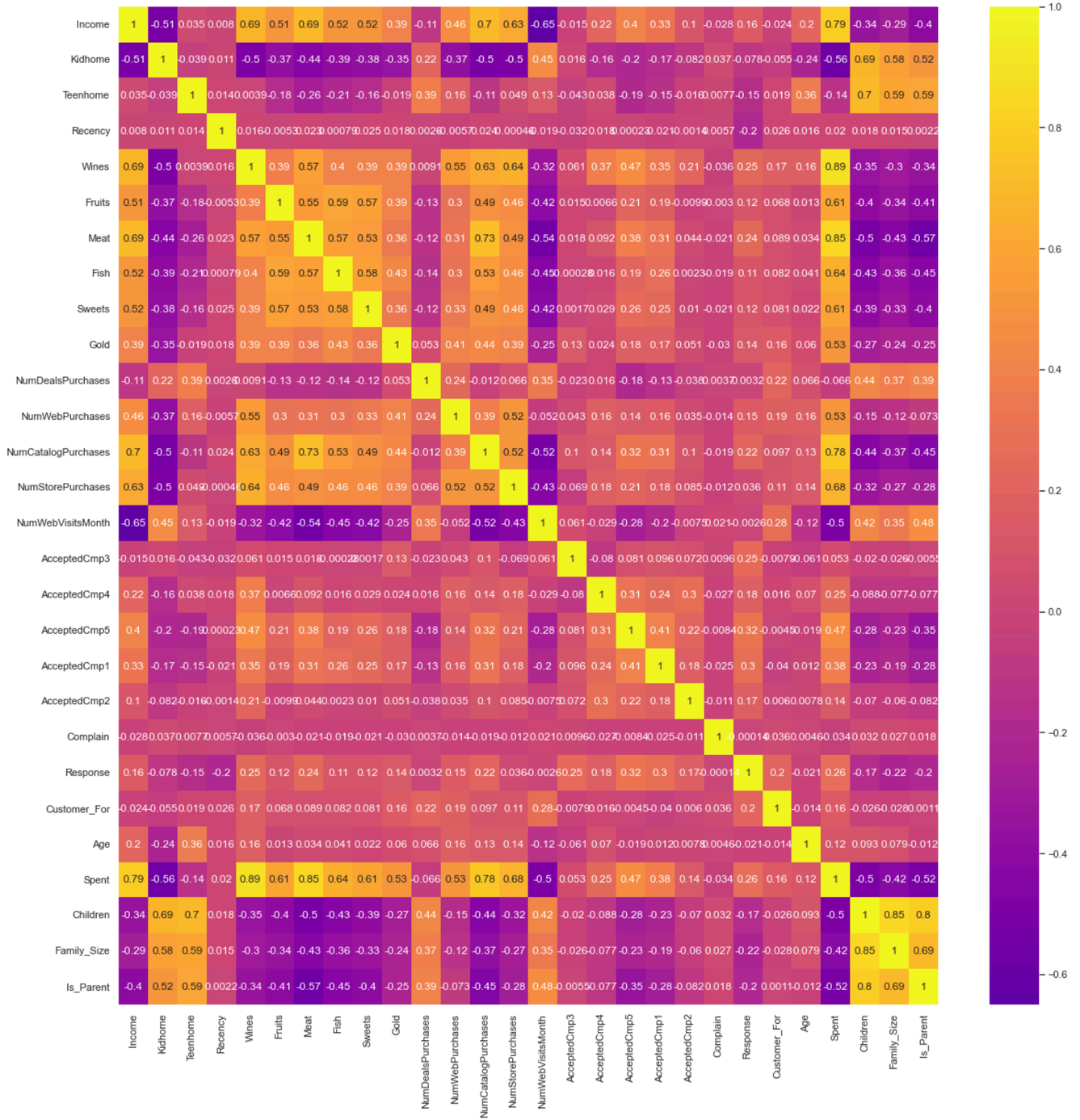
CORRELATION MATRIX

Insights

For preliminary analysis, we place more focus on spending amounts and patterns for different family compositions.

- **Income** shows a strong correlation of **Wines** and **Meat** purchases. (0.69)
- **Income** also shows strong correlation with **NumCatalogPurchases** (0.7)
- **NumWebVisitsMonth** shows negative correlation with all products listed in the dataset

...among other correlation patterns.



CORRELATION MATRIX

We observe that there is strong correlation between multiple variables, some of them we know have multicollinearity relationships

For example, **Family_Size, Children and Is_Parent** are 3 predictors that show strong correlation with each other.

- Strong correlations across many variables indicate that there is redundancy in information. **Principal Component Analysis** can be used to identify which directions carry more information than the others.

- Multicollinearity in predictors, such as **Family_Size** and **Children**, cause data trends to be difficult to be traced to a single predictor for analysis. Multicollinearity between predictors can also lead to inflated standard errors.

DATA PREPROCESSING

1. Identify **Categorical Variables** and **convert** them **to numerical values** using Label Encoding.
2. Dropped data irrelevant to clustering operation, such as campaigns and complaints (as we are trying to identify customer profiles), and **Scale data** with sklearn StandardScaler, such that **mean is normalised to 0 and standard deviation to 1.** This makes it easier to perform clustering operations.

```
# Extract the list of categorical variables
obj_type = data.dtypes == "object"
cat_vars = list(obj_type[obj_type].index)

print("Categorical variables in dataset:", cat_vars)

# Label encode the object dtypes to transform them to numerical dtype
Label_Enc = LabelEncoder()
for i in cat_vars:
    data[i] = data[[i]].apply(Label_Enc.fit_transform)
```

✓ 0.0s

```
Categorical variables in dataset: ['Education', 'Living_With']
```

- Categorical vars ‘Education’ and ‘Living_With’ are label encoded

```
ds = data.copy()

# Creating a subset cols_del of dataframe to be dropped
cols_del = [
    "AcceptedCmp1", "AcceptedCmp2",
    "AcceptedCmp3", "AcceptedCmp4",
    "AcceptedCmp5", "Complain",
    "Response",
]
ds = ds.drop(cols_del, axis=1)

# Perform scaling on all features. Normalise mean to 0 and standard deviation to 1
scaler = StandardScaler()
scaler.fit(ds)
scaled_data = pd.DataFrame(scaler.transform(ds), columns=ds.columns)
```

✓ 0.0s

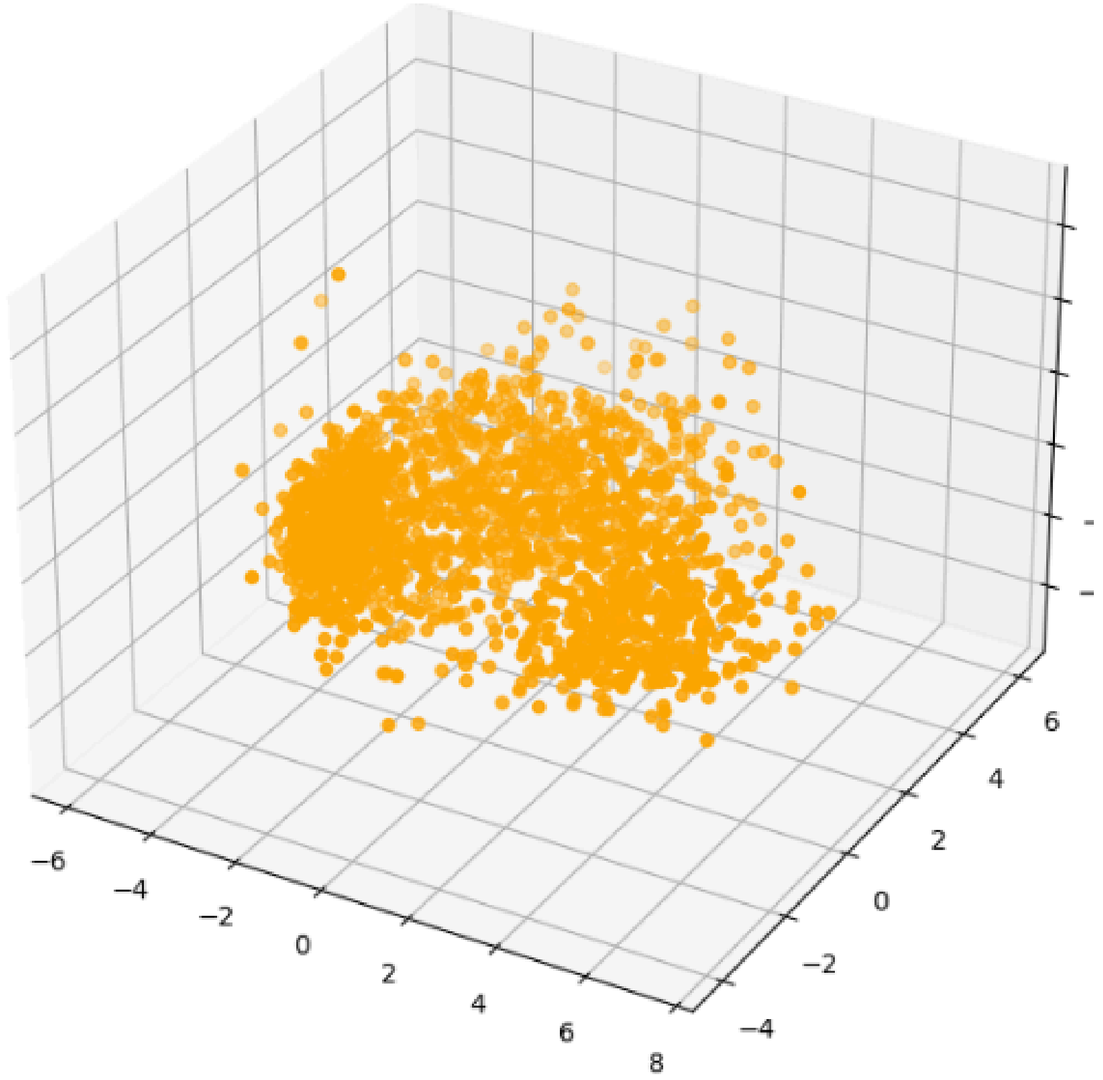
- All variables are scaled using scaler object (sklearn StandardScaler)

DATA PREPROCESSING

3. **Dimensionality reduction** using **Principal Component Analysis**

- **Large number of features**, many which are **correlated** (based on EDA)
- Used PCA to **reduce the number of features/dimension to 3 dimensions** while preserving as much information, to allow for clustering and visualization.
- PCA **transforms data** into Principal components, each a **linear combination of original** features.

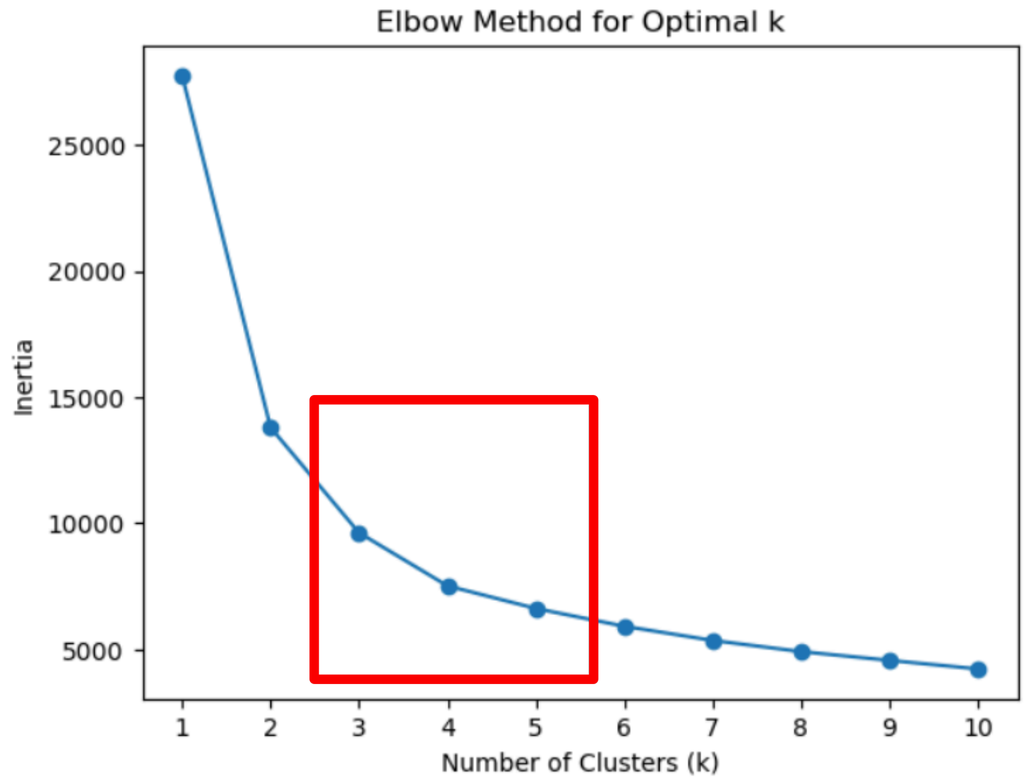
Visualisation of dataset after Dimensionality Reduction:



CLUSTERING MODEL

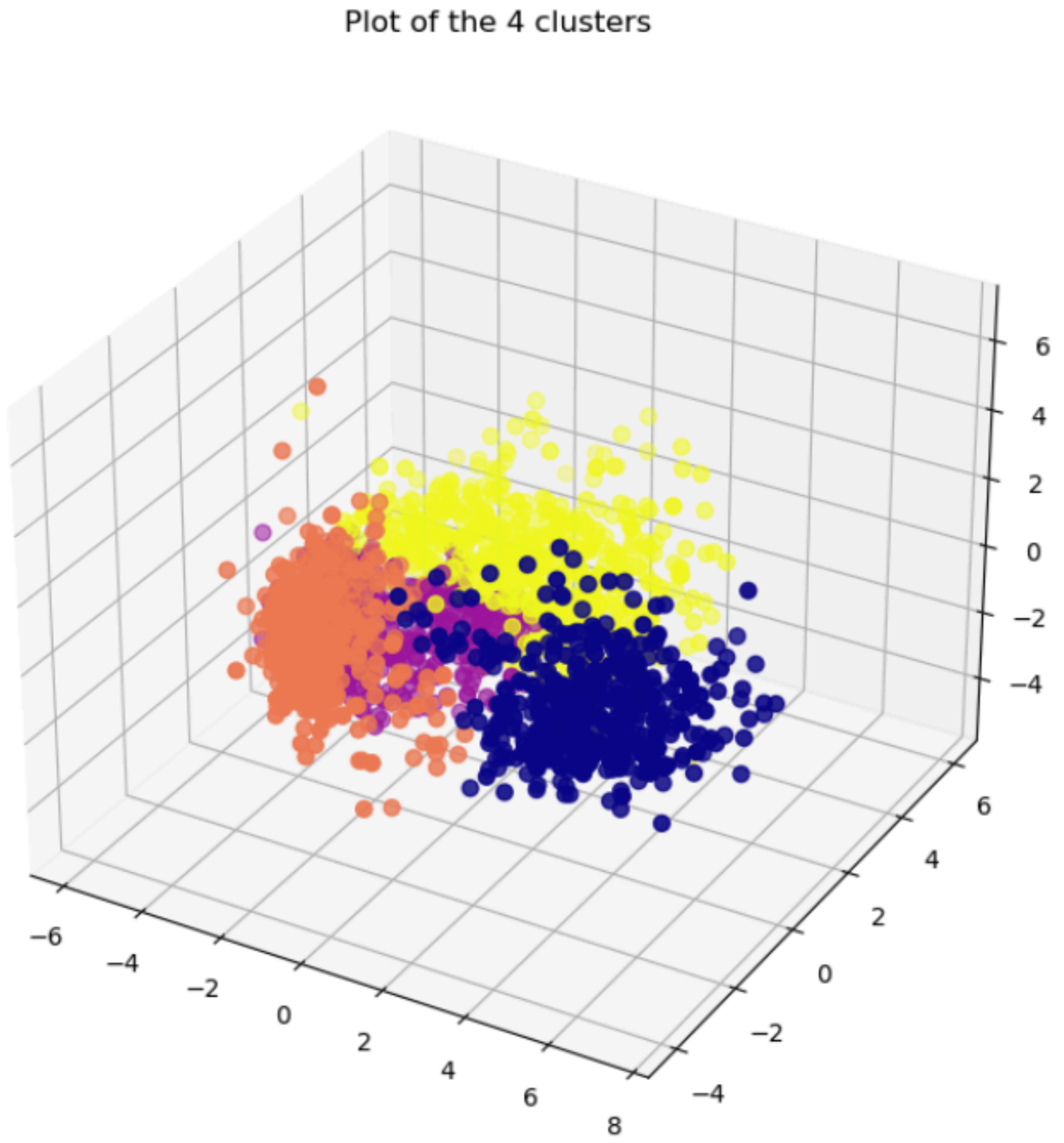
- Determine **number of clusters** to form
  - Using “Elbow method” to find when inertia (**sum of squares within cluster**) **decreases at slower rate**
- As seen from the graph, the **rate of inertia decrease slows** down above 4 clusters. Hence we select the number of clusters to be 4.

```
inertia_values = []
k_values = range(1, 11)  # Specify the range of k values to try

for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=52)
    kmeans.fit(PCA_ds)  # Fit KMeans to the data
    inertia_values.append(kmeans.inertia_)  # Append the inertia value to the list

# Plotting the elbow curve
plt.plot(k_values, inertia_values, marker="o")
plt.xlabel("Number of Clusters (k)")
plt.ylabel("Inertia")
plt.title("Elbow Method for Optimal k")
plt.xticks(k_values)
plt.show()
```



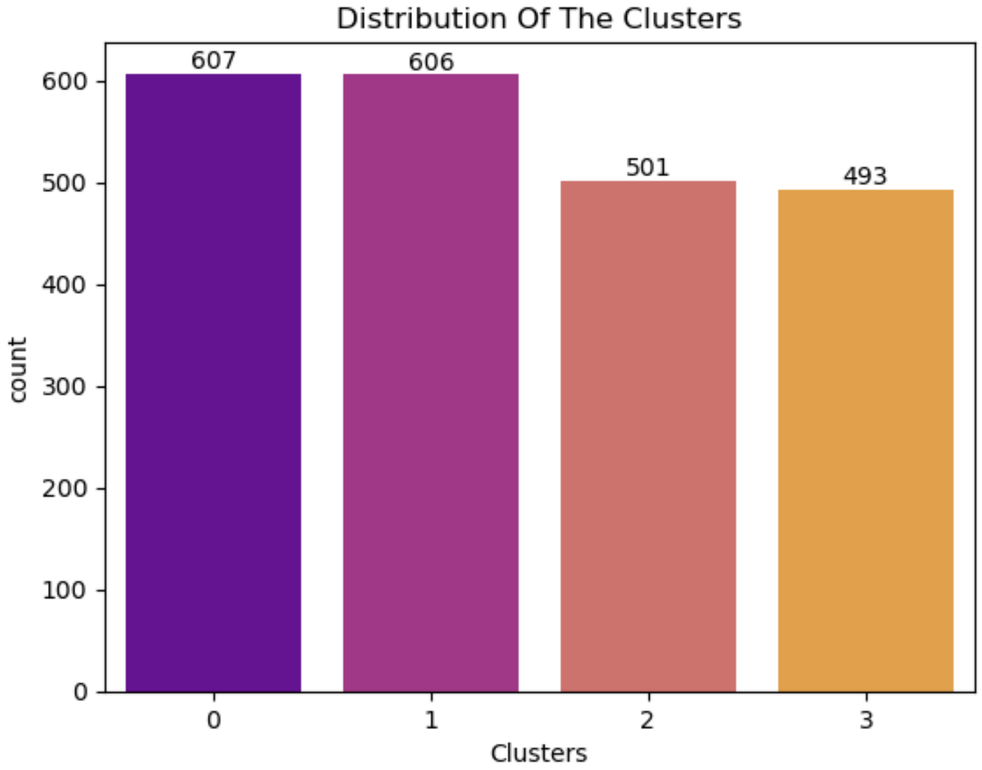
CLUSTERING MODEL

- Fit data to identified number of clusters with sklearn **Agglomerative Clustering**
- **Agglomerative Clustering** is an **unsupervised ML** technique that **merges similar “clusters”** of data, until the **target number of clusters** (4, in this case) are formed. Each data point starts of being its own “cluster”.



MODEL EVALUATION

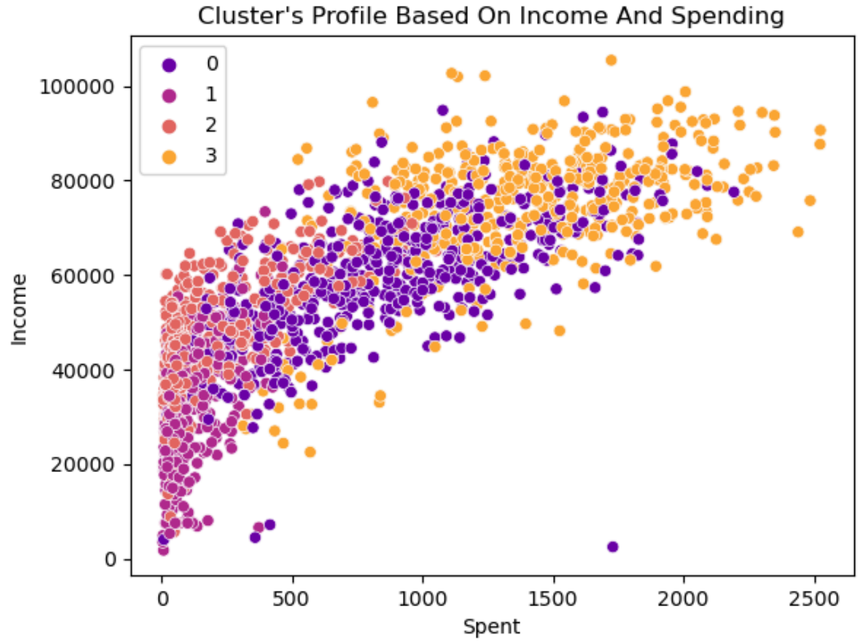
- Unsupervised clustering means **no target feature** for scoring
- Conduct **exploratory data analysis** to study patterns and draw insights
- Recommend **data-driven strategies** to help the business grow

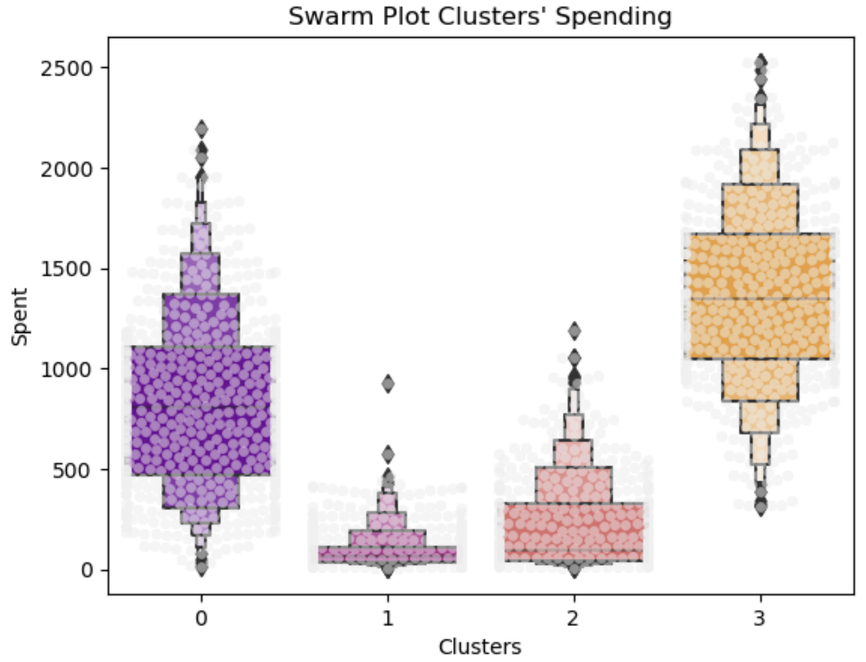


INCOME INSIGHTS

- Group 0 has **medium-high income** and **medium spending**
- Group 1 has **low income** and **low spending**
- Group 2 has **medium income** and **medium-low spending**
- Group 3 has **high income** and **high spending**

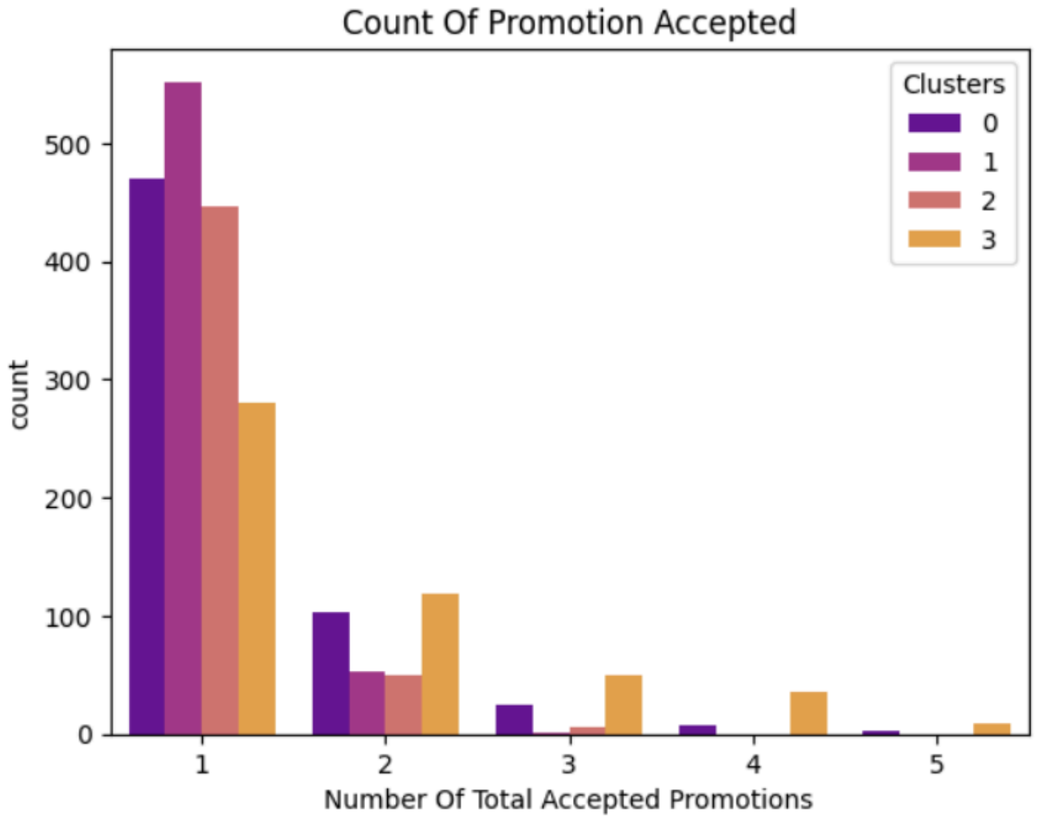
Group 3 is our **most valuable segment**, followed closely by Group 0





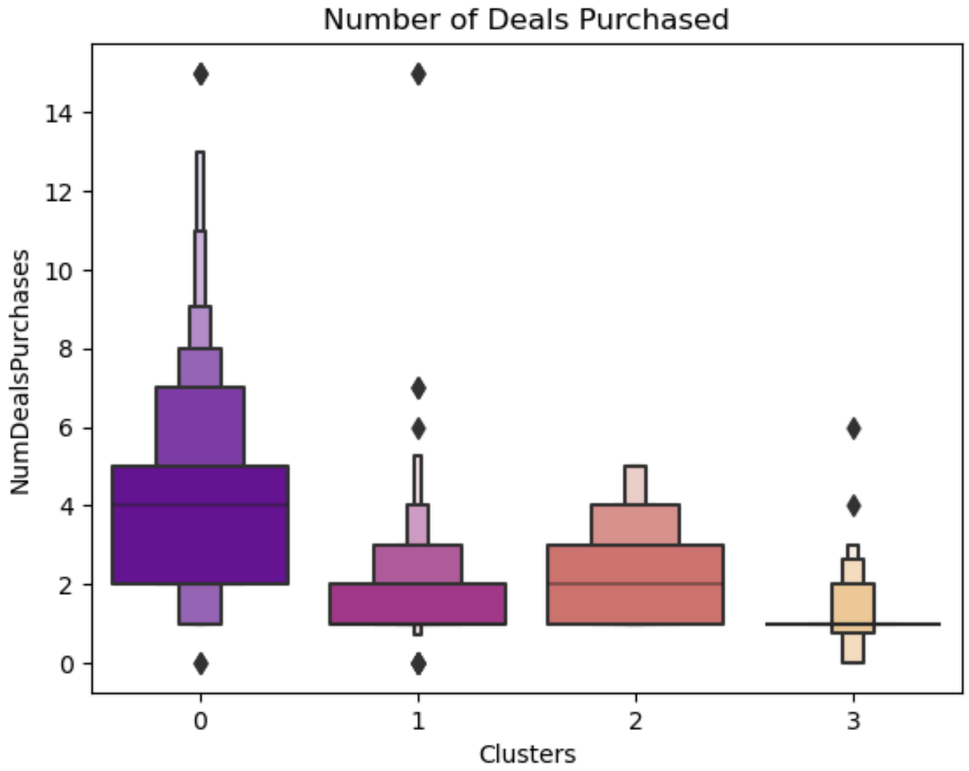
CAMPAIGNS INSIGHTS

- **Poor performance** for all campaigns across all clusters
- **Drastic drop** in the number of promotions accepted after the **first campaign**
- **Minimal participants** in all clusters from the **third campaign** onwards
- Current campaign strategies are **suboptimal** and should be **revamped**



DISCOUNT INSIGHTS

- **Poor performance** across all clusters
- Most valuable **Group 3** has the poorest response
- This could be attributed to their **high income background**
- Second most valuable **Group 0** has the best response



PROFILING

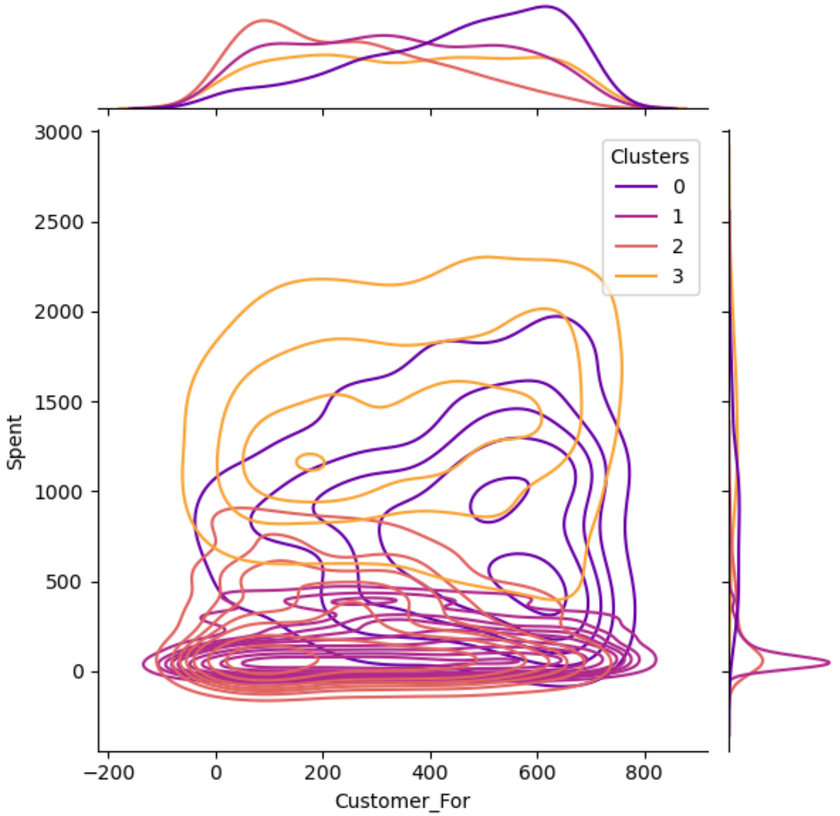
- **Profile and identify** each cluster and its type of customers
- Plot features indicative of a **customer's personal traits** in the cluster they are in

data-insights.ipynb

```
personal_traits = [
    "Kidhome",
    "Teenhome",
    "Customer_For",
    "Age",
    "Children",
    "Family_Size",
    "Is_Parent",
    "Education",
    "Living_With",
]

for i in personal_traits:
    plt.figure()
    sns.jointplot(
        x=data[i],
        y=data["Spent"],
        hue=data["Clusters"],
        kind="kde"
    )
    plt.show()
```

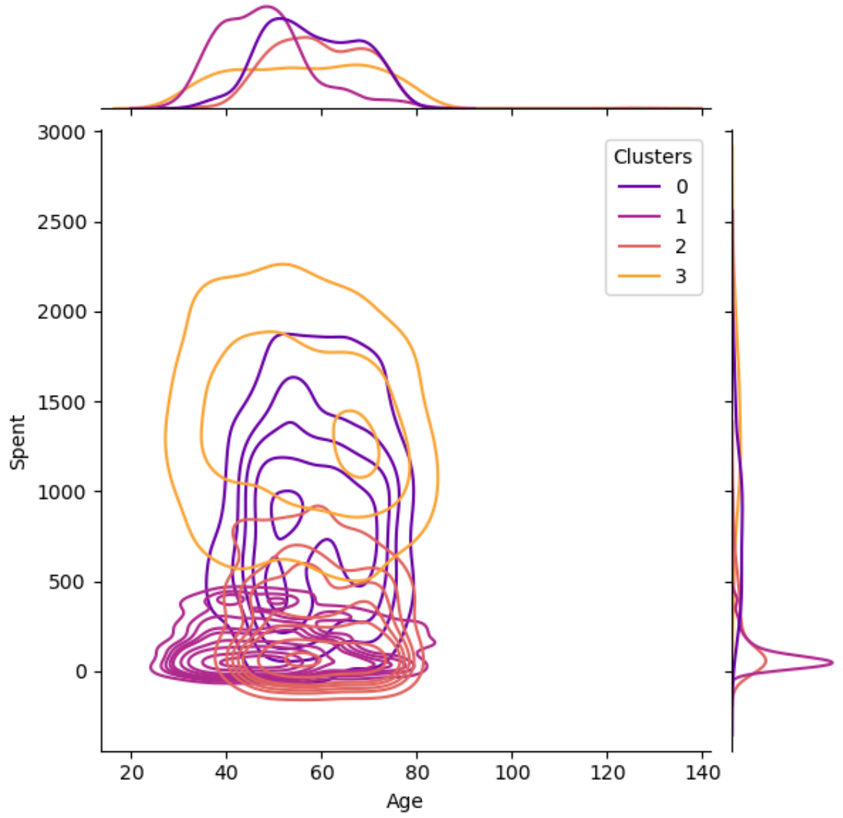
CUSTOMER LOYALTY

- Group 0 majoritively have been long-time customers
- Group 1 and 3 have no significant patterns
- Group 2 majoritively have been new customers



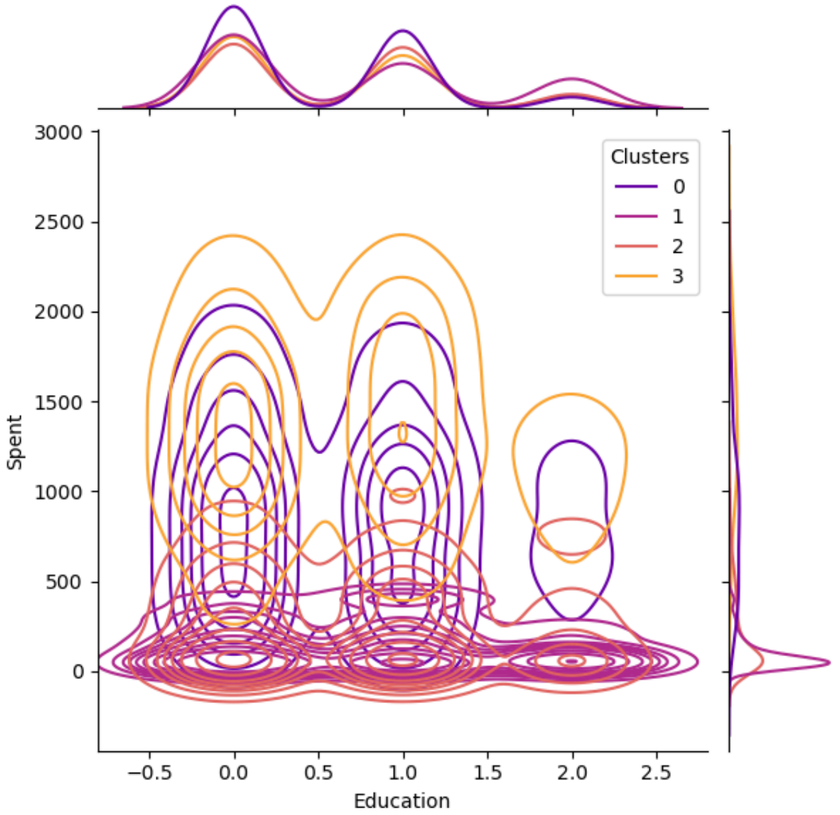
AGE

- Group 0 and 2 are older (majority above 45) customers
- Group 1 are middle-aged customers (majority between 30-45)
- Group 3 have no significant patterns



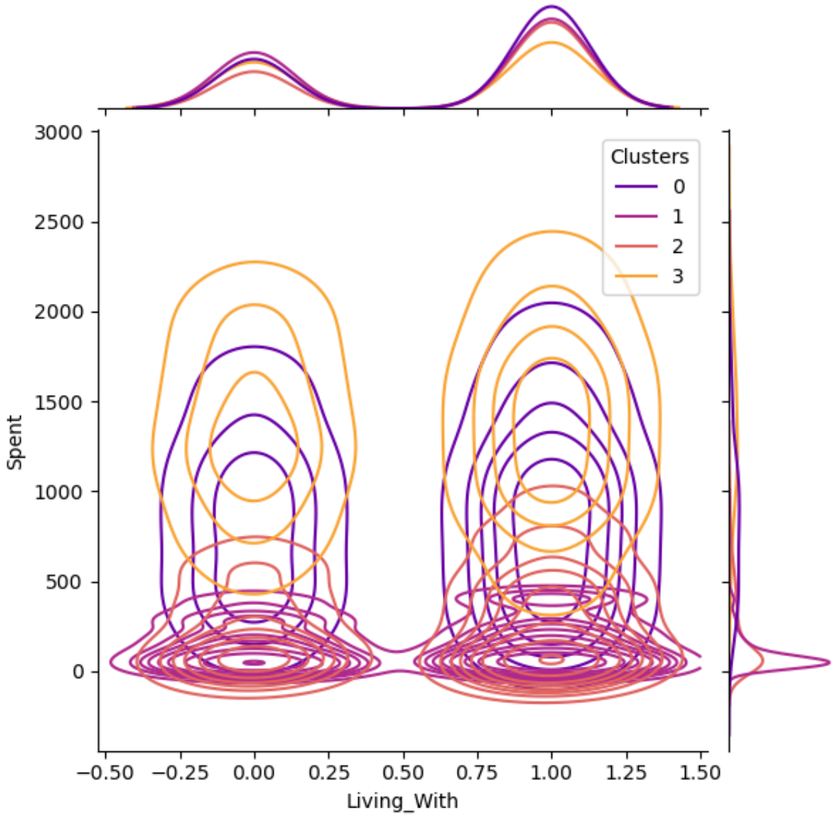
EDUCATION LEVEL

- No significant differences are observed, there is an expected downward trend from Undergraduate to Graduate to Postgraduate for all clusters



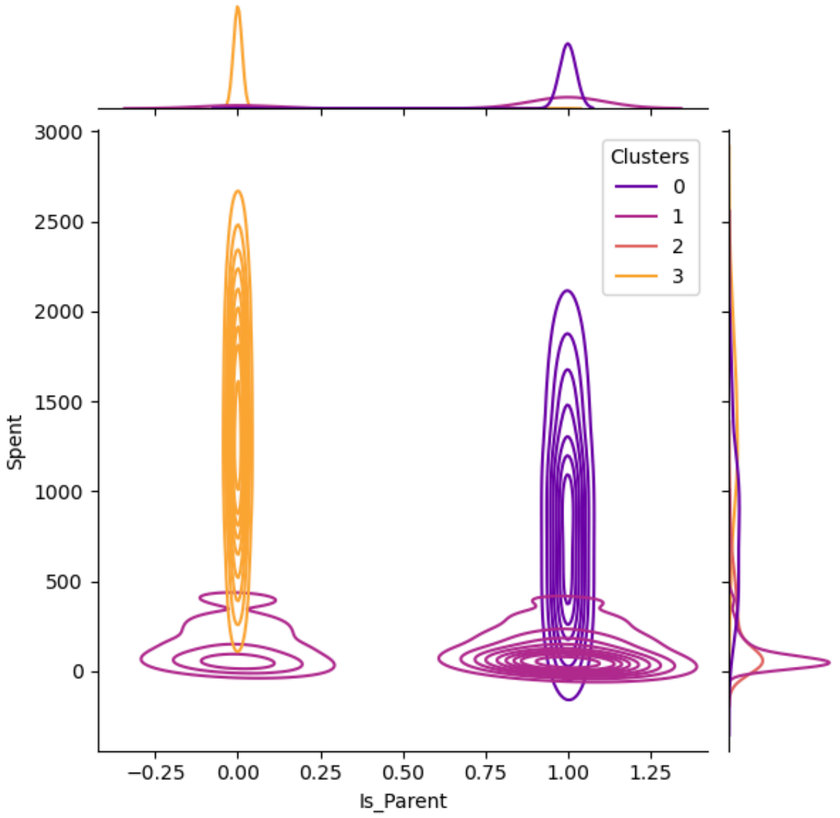
PARTNER STATUS

- Again, no significant differences are observed, there is a slight majority of “Partner” for all clusters



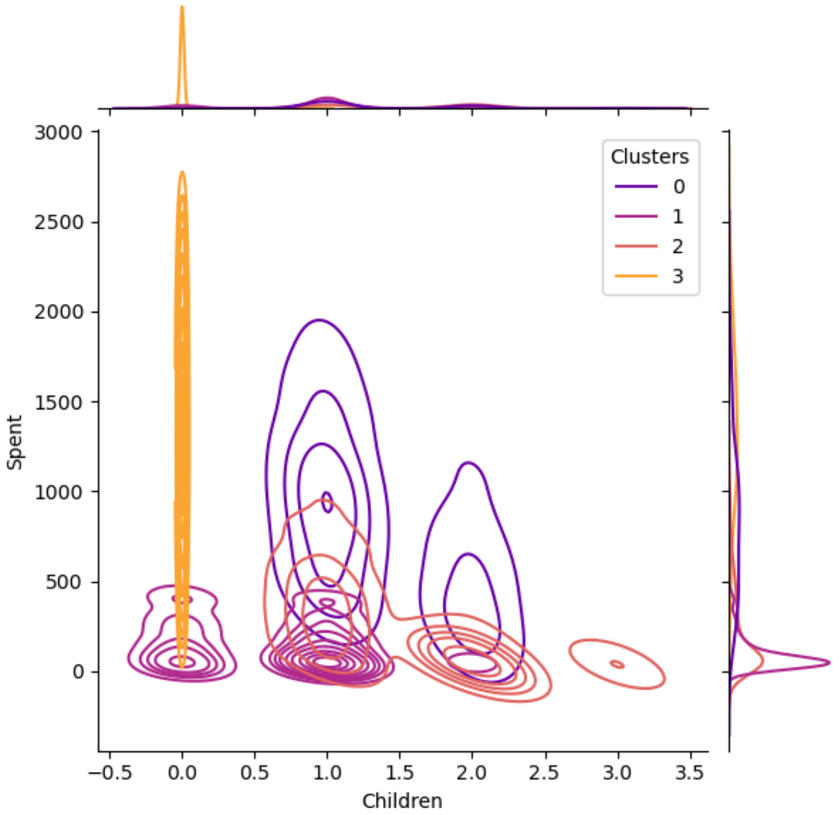
PARENT OR NOT

- Group 0 and 2 are definitely parents
- Group 1 are majoritively parents
- Group 3 are definitely not parents



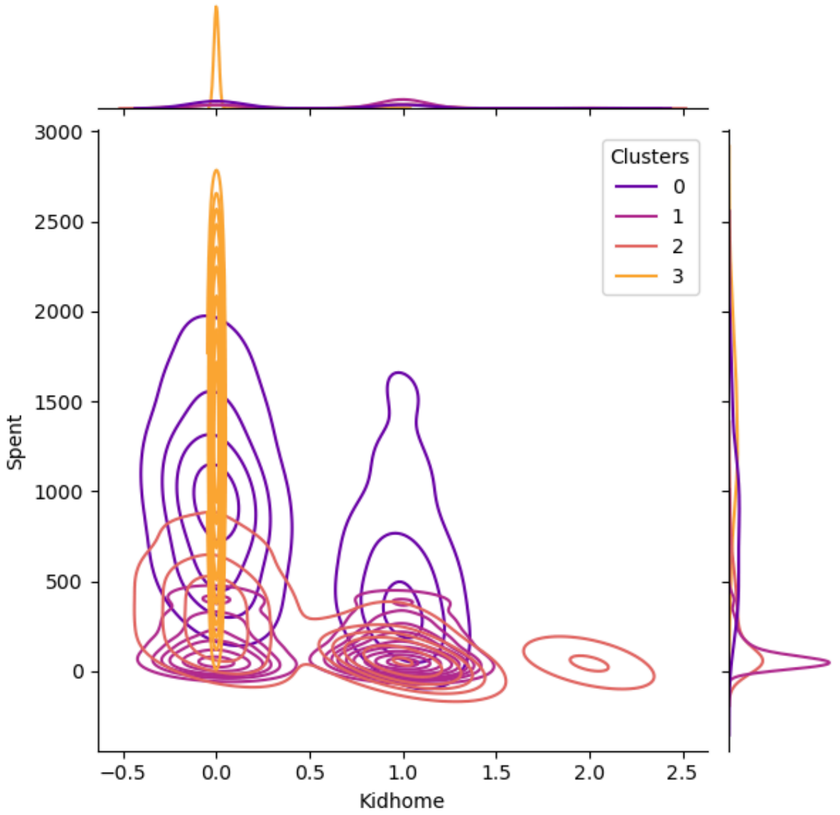
TOTAL CHILDREN

- Group 0 all have children, at most 2 children
- Group 1 majoritively have 1 child
- Group 2 all have children
- Group 3 all have no children



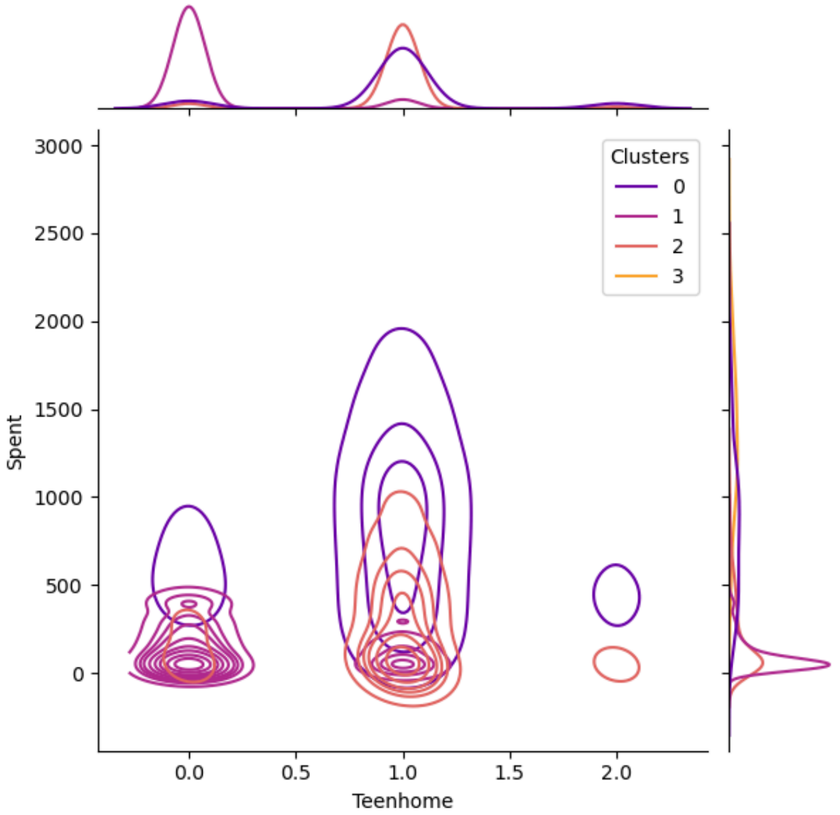
NUMBER OF KIDS

- Group 0 majoritively have no kids, and at most have 1 kid
- Group 1 majoritively and at most have 1 kid
- Group 2 majoritively have 1 kid
- Group 3 have no kids at home



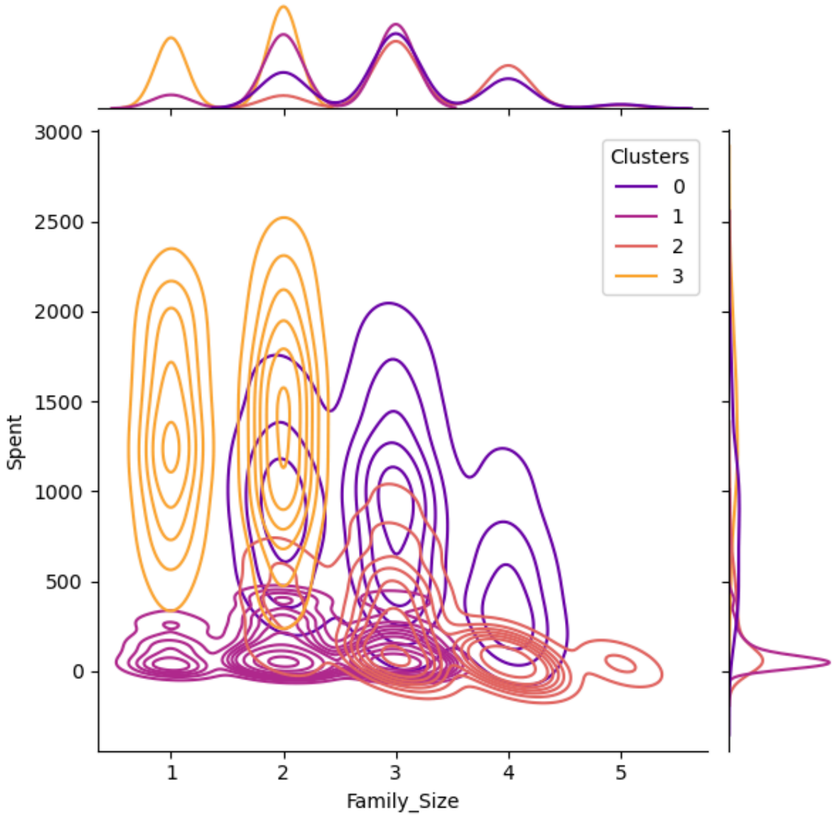
NUMBER OF TEENS

- Group 0 majoritively have 1 teenager at home
- Group 1 majoritively have no teenagers at home
- Group 2 majoritively have 1 teenager at home
- Group 3 have no teenagers at home



SIZE OF FAMILY

- Group 0 have a medium family size, from 2-4
- Group 1 have a small family size, from 1-3
- Group 2 have a big family size, from 2-5
- Group 3 are either single or couples only



CUSTOMER PROFILES

Group 0

- Definitely a parent
- Mostly have teenagers at home
- Long-time customers
- Relatively older customers (above 45)
- Medium family size of 3-4
- Medium-high income

Group 2

- Definitely a parent
- Have both kids and teenagers at home
- New customers
- Relatively older customers (above 45)
- Big family size upwards to 5
- Medium income group

Group 1

- Most likely a parent
- If parent, most likely have a kid and not a teenager
- Relative younger customers (30-45)
- Small family size upwards to 3
- Low income group

Group 3

- Definitely not a parent
- Spans all ages
- Slight majority of couples over singles
- High income group

STRATEGY RECOMMENDATIONS

Focus on the **retention of Group 0 and Group 3** customers, while **boosting spending from Group 2** to emulate Group 3

Retain Group 0 and Group 3

- For Group 3, offer **teens-focused promotions** on snacks, beverages, and after-school treats
- Introduce a **premium/gourmet section** with speciality and artisanal products to capitalise on both groups' **higher income brackets**

Boost Group 2 Spending

- Groups 2 and 3 share **many similarities in their profiles**, and Group 2's **recent exposure** to the business could account for their **lower spending**
- Offer **larger family-size product bundles** and bulk discounts on **staple items**
- Provide dedicated checkout lanes for **larger orders** to improve convenience

THANK YOU