Rakamin Academy

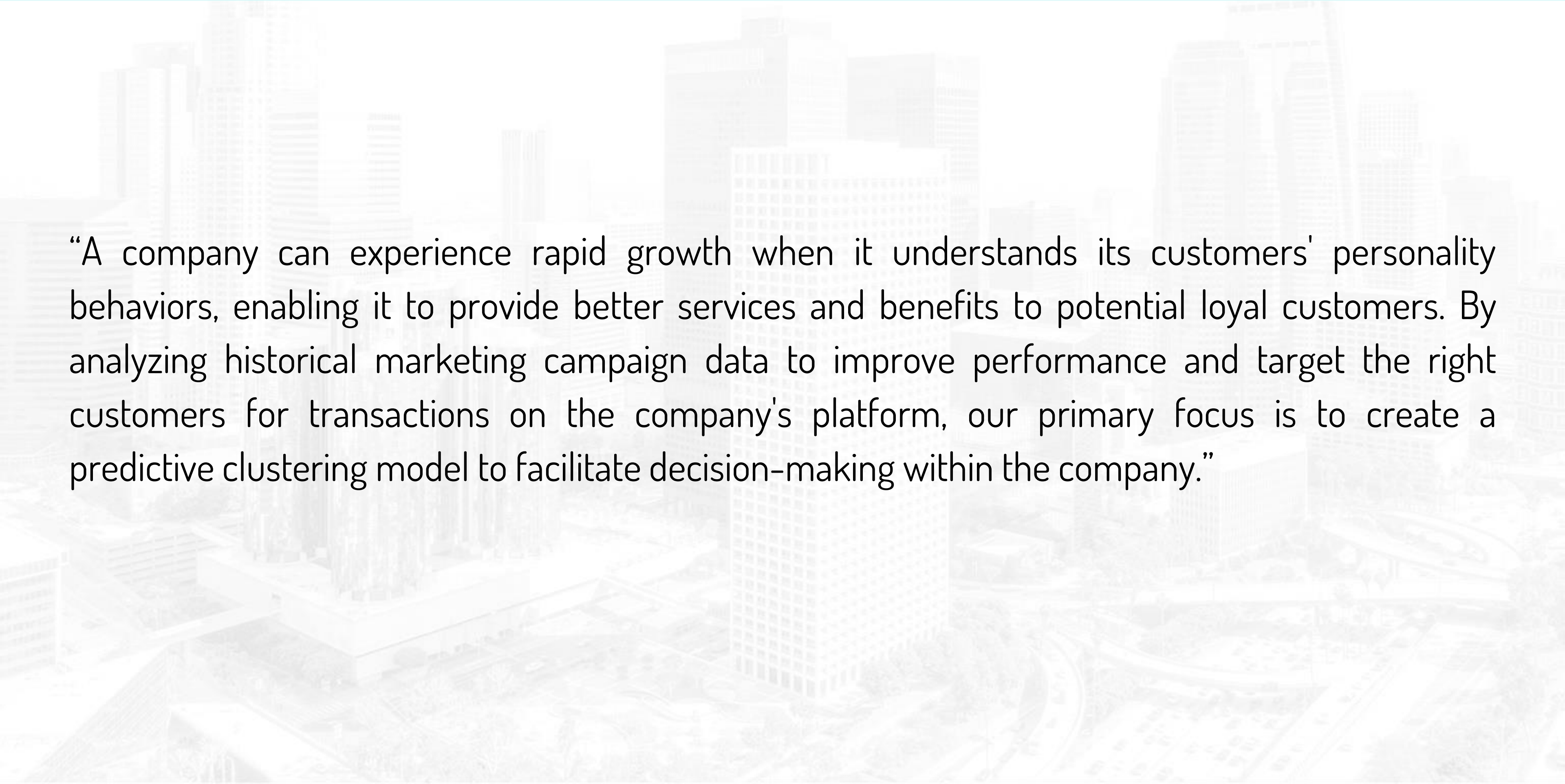
# Predict Customer Personality to boost marketing campaign by using Machine Learning

**Supported by:**
**Rakamin Academy**
Career Acceleration School
www.rakamin.com

**Created by:**
**Anik Maulia Tri Handayani**
anikmaulia263@gmail.com
linkedin.com/in/anikmauliatrihandayani

I possess foundational knowledge in data analysis and a proven ability to design and develop Machine Learning solutions. I have applied it to an predict customer personality to boost marketing campaign project. I strongly believe in the pivotal role data plays in enhancing business performance. I hold relevant certifications and practical experience, which have prepared me to be a dedicated data professional. I am also adept at collaborating within multidisciplinary teams and am ready to advance my career in data analysis across various industries, including telecommunications, commerce, retail, FMCG, finance, and banking.

# Overview



“A company can experience rapid growth when it understands its customers' personality behaviors, enabling it to provide better services and benefits to potential loyal customers. By analyzing historical marketing campaign data to improve performance and target the right customers for transactions on the company's platform, our primary focus is to create a predictive clustering model to facilitate decision-making within the company.”

# Exploratory Data Analysis (EDA)

Rakamin Academy

## Univariate Analysis using **Bar Chart, Boxplot** and **Distplot**

The following insights emerged:

- The distribution of each data point.
- Key trends and patterns within individual data sets.
- An overview of the central tendencies and variability within each dataset.

## Bivariate Analysis using **Scatter Plot**

The following insights emerged:

- Relationship between all columns in the dataset and **Conversion_Rate** column

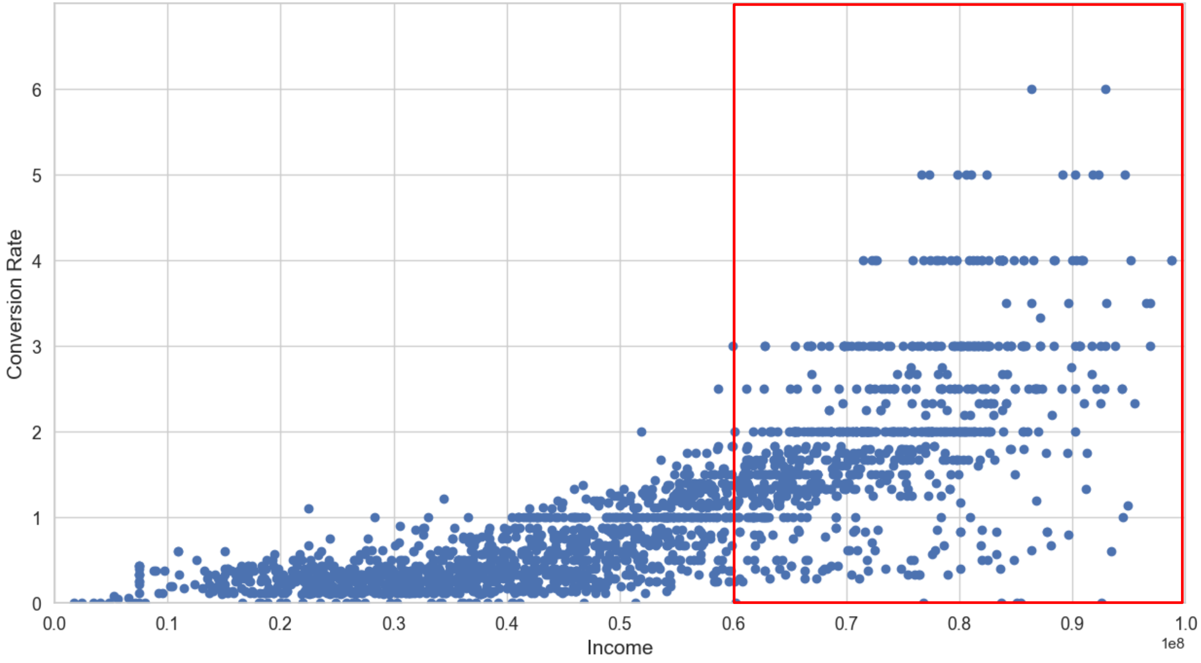
## Multivariate Analysis using **Heatmap**

The following insights emerged:

- Relationship and correlations between features.

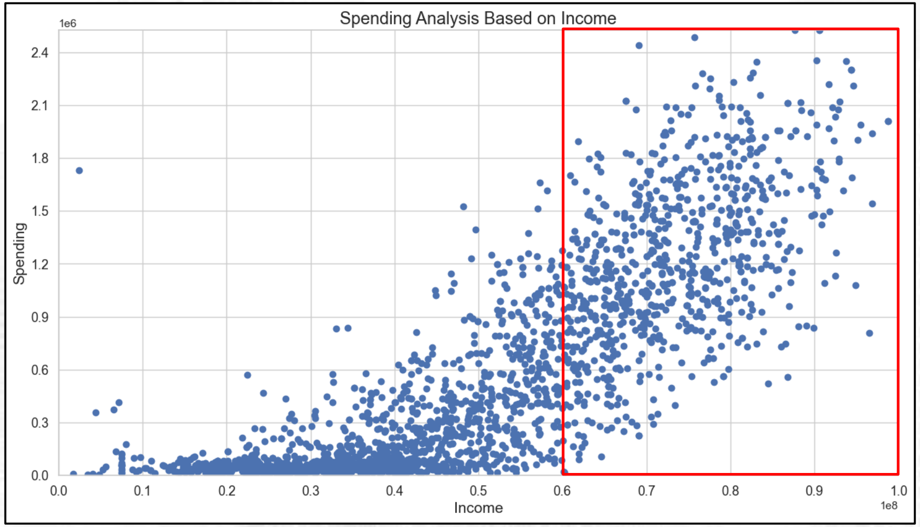
For more details, you can refer to the Jupyter notebook here

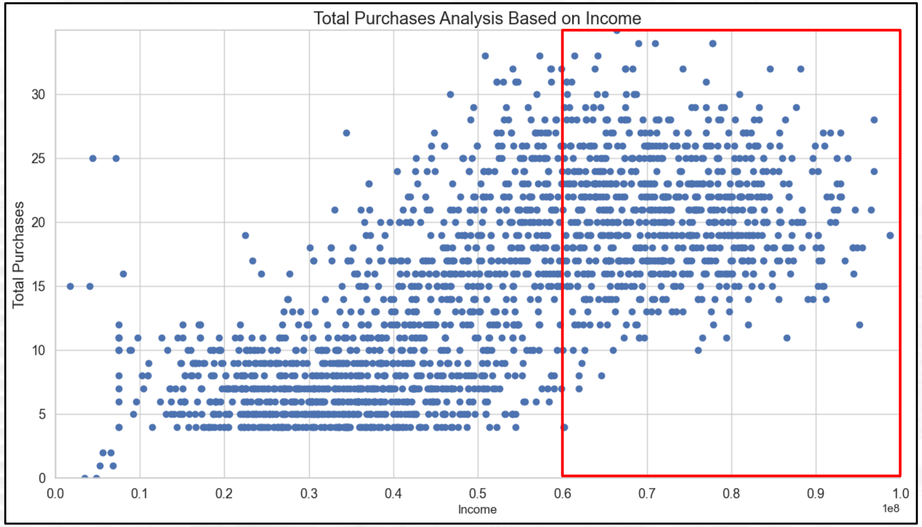
# Conversion Rate Analysis Based on Income



The larger a customer's total income, the more it can potentially boost the conversion rate, where the increase in conversion rate becomes noticeable when their monthly income exceeds 60.000.000 IDR.

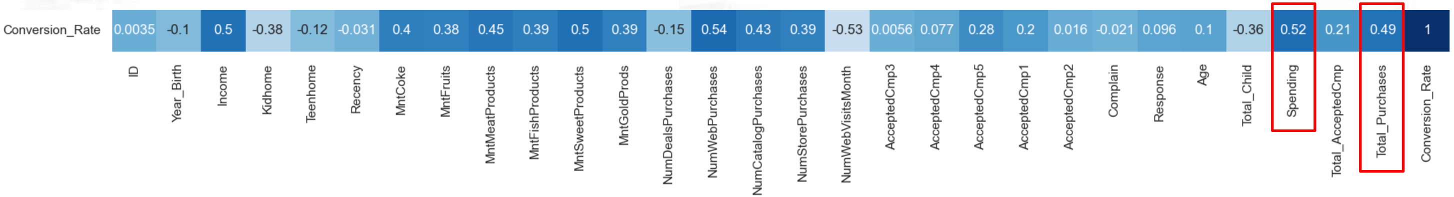
# Spending and Total Purchases Analysis Based on Income



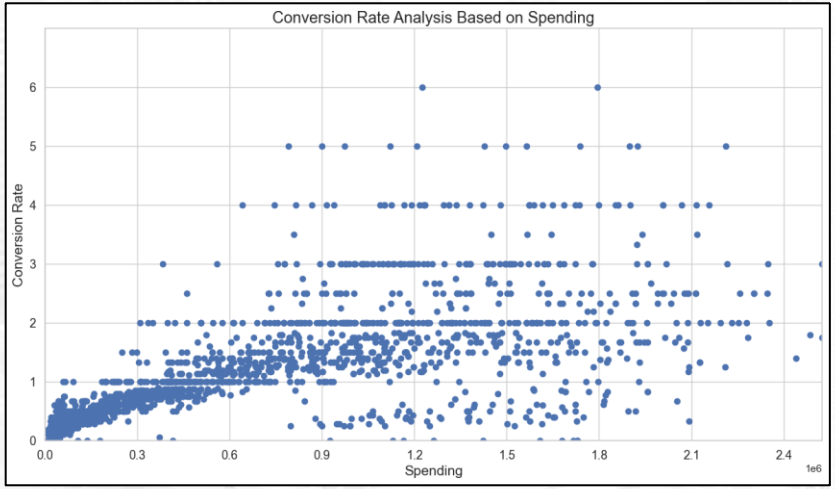


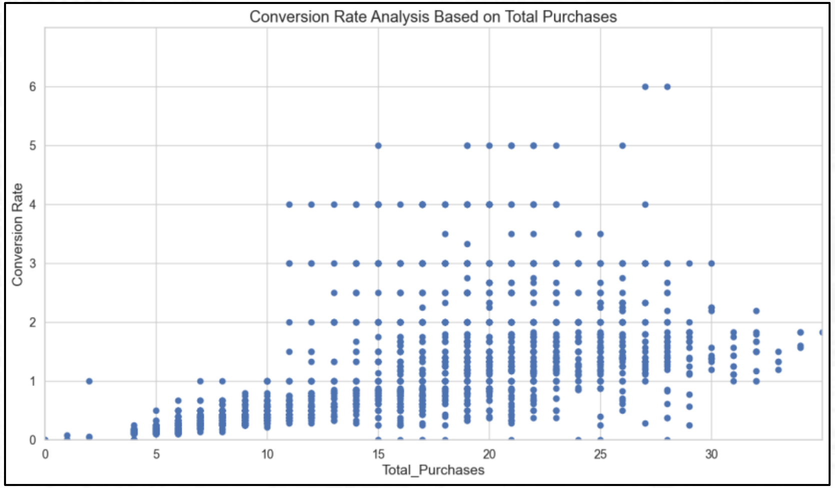
Supported by other insights, the higher income tends to increase both spending and total purchases made by customers, especially when the income exceeds 60.000.000 IDR.

# Conversion Rate Analysis Based on Spending and Total Purchases



| | ID | Year_Birth | Income | Kidhome | Teenhome | Recency | MntCoke | MntFruits | MntMeatProducts | MntFishProducts | MntSweetProducts | MntGoldProds | NumDealsPurchases | NumWebPurchases | NumCatalogPurchases | NumStorePurchases | NumWebVisitsMonth | AcceptedCmp3 | AcceptedCmp4 | AcceptedCmp5 | AcceptedCmp1 | AcceptedCmp2 | Complain | Response | Age | Total_Child | Spending | Total_AcceptedCmp | Total_Purchases | Conversion_Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Conversion_Rate | 0.0035 | -0.1 | 0.5 | -0.38 | -0.12 | -0.031 | 0.4 | 0.38 | 0.45 | 0.39 | 0.5 | 0.39 | -0.15 | 0.54 | 0.43 | 0.39 | -0.53 | 0.0056 | 0.077 | 0.28 | 0.2 | 0.016 | -0.021 | 0.096 | 0.1 | -0.36 | 0.52 | 0.21 | 0.49 | 1 |





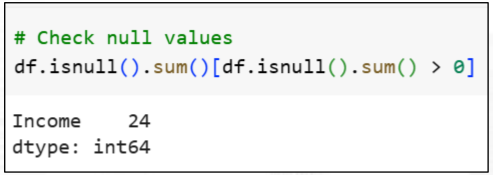
Spending and total purchases positively correlate with the conversion rate. However, there is no specific spending or total purchases threshold that directly indicates a higher conversion rate.

# Recommendation

- Taking into consideration the insights we've gathered, it may be worthwhile to target customers with an income exceeding 60.000.000 IDR per month for a more specific approach in our marketing campaigns. Although it's possible that among them, some may have a relatively lower conversion rate.
- In view of the recommendations, we do not yet have additional suggestions for customers with an income below 60.000.000 IDR. To generate such recommendations, we can explore clustering further to provide more tailored suggestions for these customers, with the aim of enhancing their potential for transactions on our platform.

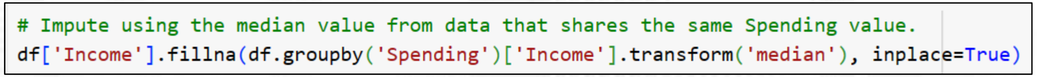
# Data Cleaning & Preprocessing

## Missing Values

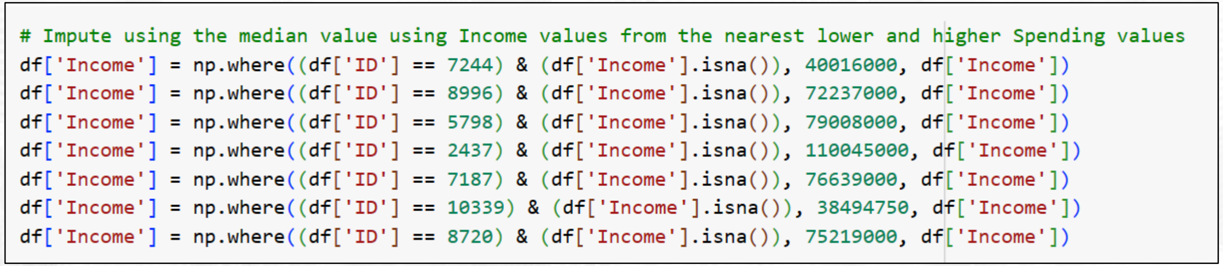


```
# Check null values
df.isnull().sum()[df.isnull().sum() > 0]
```

```
Income    24
dtype: int64
```



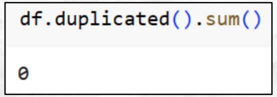
```
# Impute using the median value from data that shares the same Spending value.
df['Income'].fillna(df.groupby('Spending')['Income'].transform('median'), inplace=True)
```



```
# Impute using the median value using Income values from the nearest lower and higher Spending values
df['Income'] = np.where((df['ID'] == 7244) & (df['Income'].isna()), 40016000, df['Income'])
df['Income'] = np.where((df['ID'] == 8996) & (df['Income'].isna()), 72237000, df['Income'])
df['Income'] = np.where((df['ID'] == 5798) & (df['Income'].isna()), 79008000, df['Income'])
df['Income'] = np.where((df['ID'] == 2437) & (df['Income'].isna()), 110045000, df['Income'])
df['Income'] = np.where((df['ID'] == 7187) & (df['Income'].isna()), 76639000, df['Income'])
df['Income'] = np.where((df['ID'] == 10339) & (df['Income'].isna()), 38494750, df['Income'])
df['Income'] = np.where((df['ID'] == 8720) & (df['Income'].isna()), 75219000, df['Income'])
```

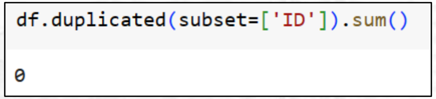
There are missing values in the **Income** column (24 data points), and the missing values will be handled through imputation based on the median value within a subset of the data.

## Duplicated Data



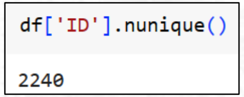
```
df.duplicated().sum()
```

```
0
```



```
df.duplicated(subset=['ID']).sum()
```

```
0
```



```
df['ID'].nunique()
```

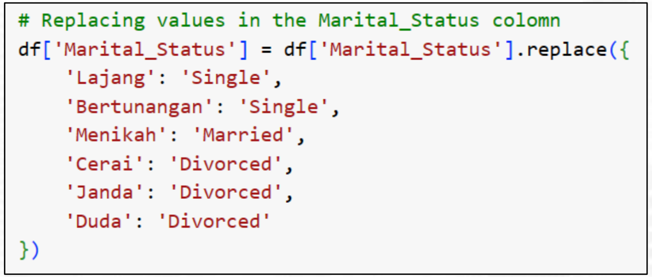
```
2240
```

**Dataset contains no duplicates**

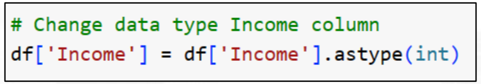
For more details, you can refer to the Jupyter notebook here

# Data Cleaning & Preprocessing

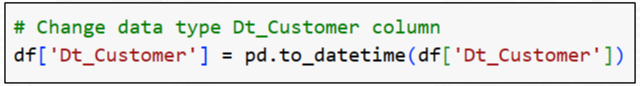
## Invalid Values



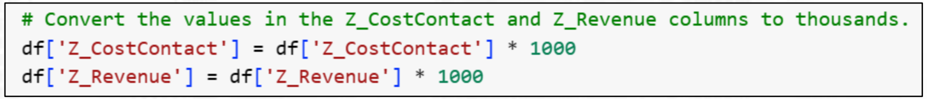
```python
# Replacing values in the Marital_Status colomn
df['Marital_Status'] = df['Marital_Status'].replace({
    'Lajang': 'Single',
    'Bertunangan': 'Single',
    'Menikah': 'Married',
    'Cerai': 'Divorced',
    'Janda': 'Divorced',
    'Duda': 'Divorced'
})
```



```python
# Change data type Income column
df['Income'] = df['Income'].astype(int)
```



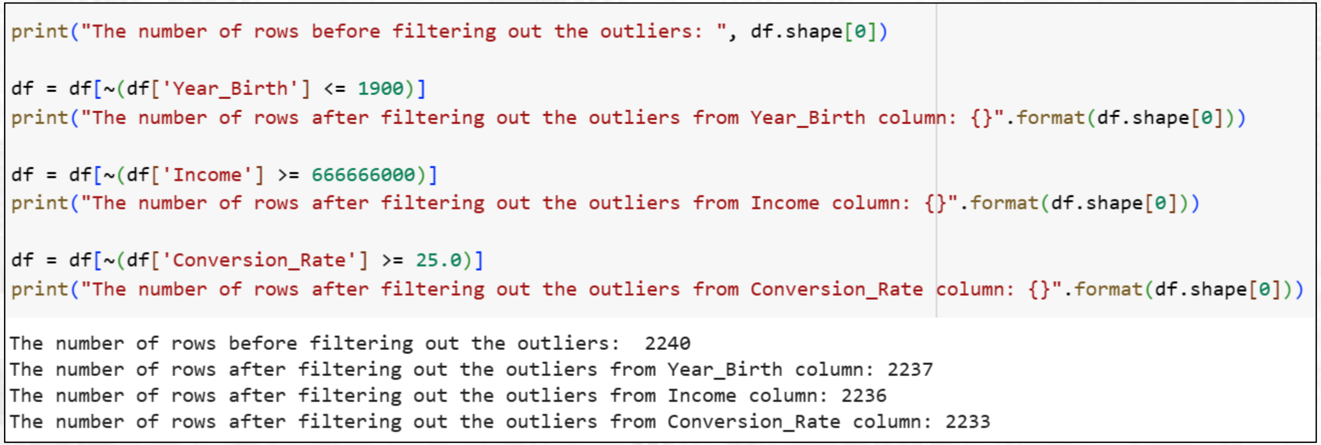
```python
# Change data type Dt_Customer column
df['Dt_Customer'] = pd.to_datetime(df['Dt_Customer'])
```



```python
# Convert the values in the Z_CostContact and Z_Revenue columns to thousands.
df['Z_CostContact'] = df['Z_CostContact'] * 1000
df['Z_Revenue'] = df['Z_Revenue'] * 1000
```

There are invalid values in the **Income**, **Dt_Customer**, **Marital_Status, Z_CostContact**, and **Z_Revenue** columns data

## Outliers



```python
print("The number of rows before filtering out the outliers: ", df.shape[0])

df = df[~(df['Year_Birth'] <= 1900)]
print("The number of rows after filtering out the outliers from Year_Birth column: {}".format(df.shape[0]))

df = df[~(df['Income'] >= 666666000)]
print("The number of rows after filtering out the outliers from Income column: {}".format(df.shape[0]))

df = df[~(df['Conversion_Rate'] >= 25.0)]
print("The number of rows after filtering out the outliers from Conversion_Rate column: {}".format(df.shape[0]))
```

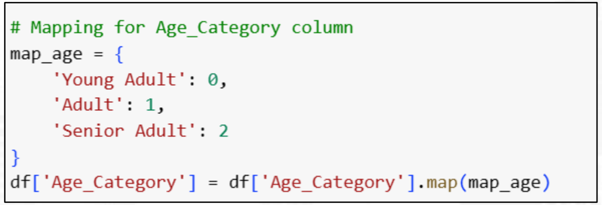
```
The number of rows before filtering out the outliers:  2240
The number of rows after filtering out the outliers from Year_Birth column: 2237
The number of rows after filtering out the outliers from Income column: 2236
The number of rows after filtering out the outliers from Conversion_Rate column: 2233
```

There are extreme values in the **Year_Birth**, **Income**, and **Conversion_Rate** columns, and the extreme values will be handled by **manual trimming**.

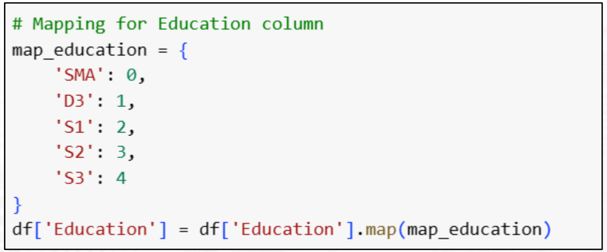
# Data Cleaning & Preprocessing

Rakamin Academy

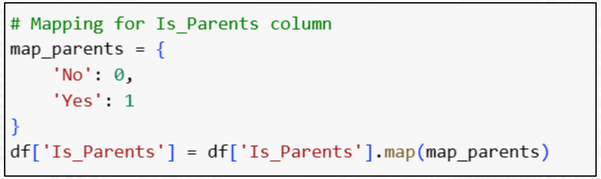
## Feature Encoding



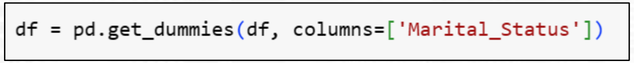
```python
# Mapping for Age_Category column
map_age = {
    'Young Adult': 0,
    'Adult': 1,
    'Senior Adult': 2
}
df['Age_Category'] = df['Age_Category'].map(map_age)
```



```python
# Mapping for Education column
map_education = {
    'SMA': 0,
    'D3': 1,
    'S1': 2,
    'S2': 3,
    'S3': 4
}
df['Education'] = df['Education'].map(map_education)
```



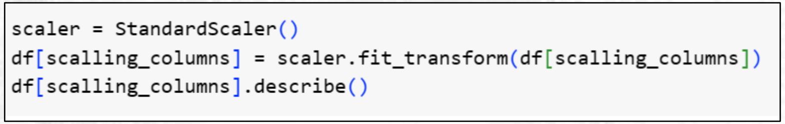
```python
# Mapping for Is_Parents column
map_parents = {
    'No': 0,
    'Yes': 1
}
df['Is_Parents'] = df['Is_Parents'].map(map_parents)
```



```python
df = pd.get_dummies(df, columns=['Marital_Status'])
```

Two techniques used in feature encoding are: Label encoding for **Education, Age_Category** and **Is_Parents** columns; One-Hot Encoding for the **Marital_Status** column.

## Feature Scaling

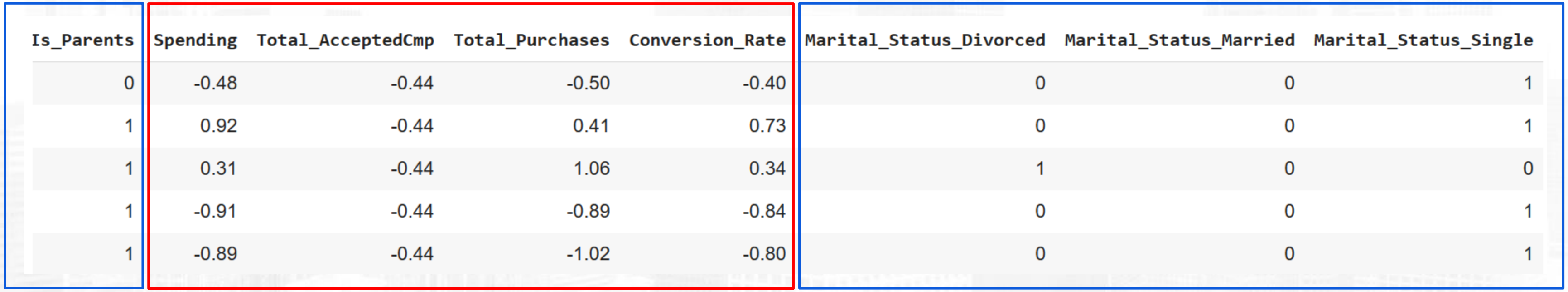


```python
scaler = StandardScaler()
df[scalling_columns] = scaler.fit_transform(df[scalling_columns])
df[scalling_columns].describe()
```

Feature scaling is performed using **StandardScaler** for numerical columns that are not binary and are not a result of feature encoding.

# Data Cleaning & Preprocessing

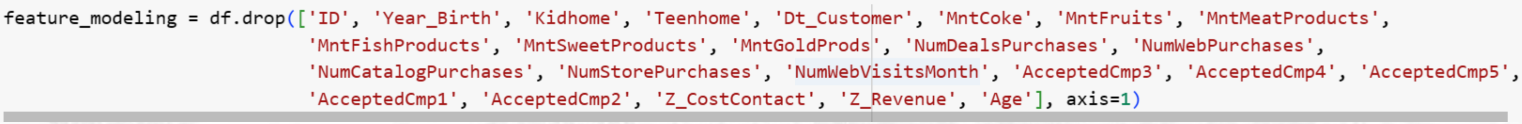
Here is the new dataset after data cleaning and processing:



| Is_Parents | Spending | Total_AcceptedCmp | Total_Purchases | Conversion_Rate | Marital_Status_Divorced | Marital_Status_Married | Marital_Status_Single |
|---|---|---|---|---|---|---|---|
| 0 | -0.48 | -0.44 | -0.50 | -0.40 | 0 | 0 | 1 |
| 1 | 0.92 | -0.44 | 0.41 | 0.73 | 0 | 0 | 1 |
| 1 | 0.31 | -0.44 | 1.06 | 0.34 | 1 | 0 | 0 |
| 1 | -0.91 | -0.44 | -0.89 | -0.84 | 0 | 0 | 1 |
| 1 | -0.89 | -0.44 | -1.02 | -0.80 | 0 | 0 | 1 |

🟦 Examples of columns after feature encoding 🟥 Examples of columns after feature scaling

## Feature Selection

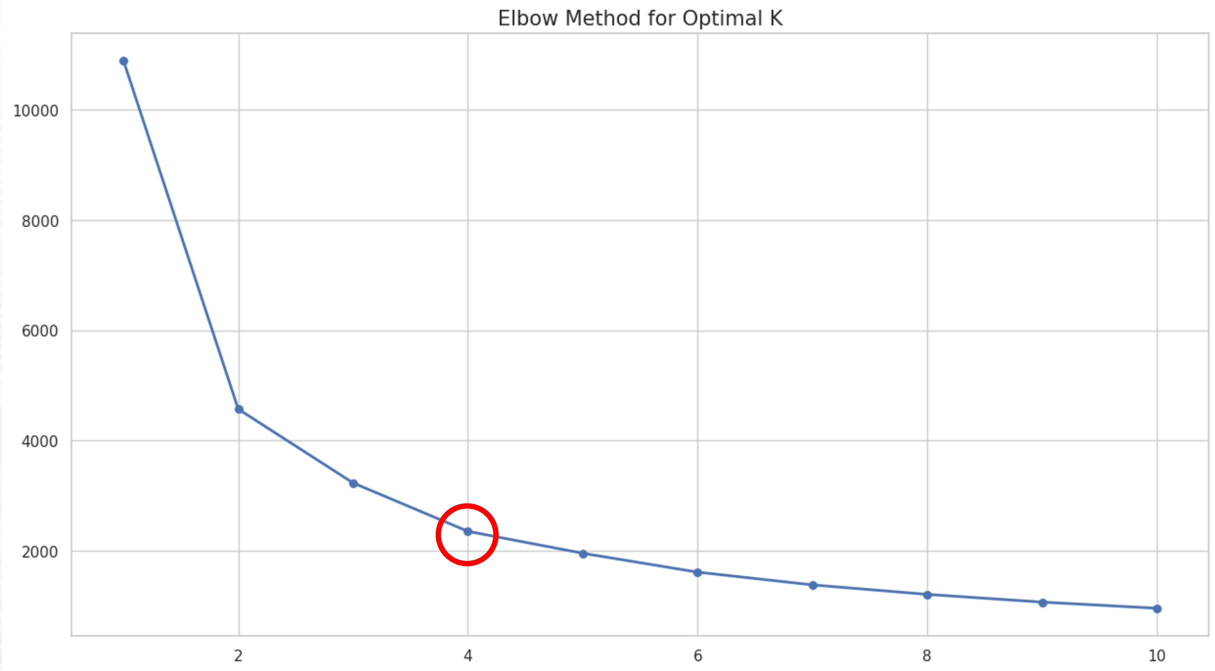


```
feature_modeling = df.drop(['ID', 'Year_Birth', 'Kidhome', 'Teenhome', 'Dt_Customer', 'MntCoke', 'MntFruits', 'MntMeatProducts',
                            'MntFishProducts', 'MntSweetProducts', 'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
                            'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth', 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5',
                            'AcceptedCmp1', 'AcceptedCmp2', 'Z_CostContact', 'Z_Revenue', 'Age'], axis=1)
```

The features to be used for creating a k-means clustering machine learning model include the following columns: **Income, Recency, Complain, Response, Age_Category, Total_Child, Is_Parents, Spending, Total_AcceptedCmp, Total_Purchases, Conversion_Rate, Marital_Status_Divorced, Marital_Status_Married,** and **Marital_Status_Single**.

# Cluster Determination: Inertia-Based Data Modeling Evaluation

Rakamin Academy

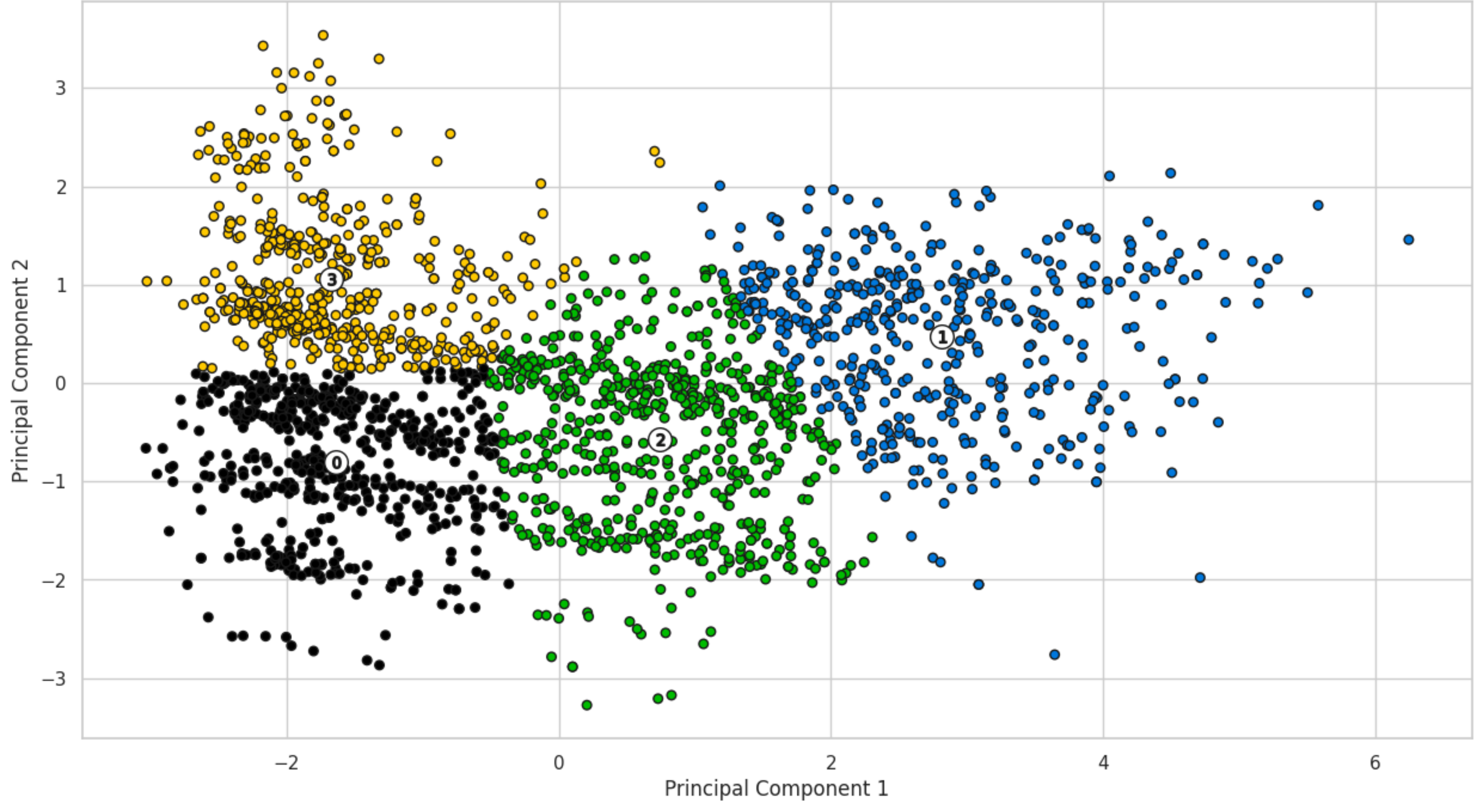


Based on the analysis using the elbow method above, it can be concluded that the value of K (number of clusters) is 4, as after reaching 4 clusters, the line graph shows a diminishing decrease in inertia.

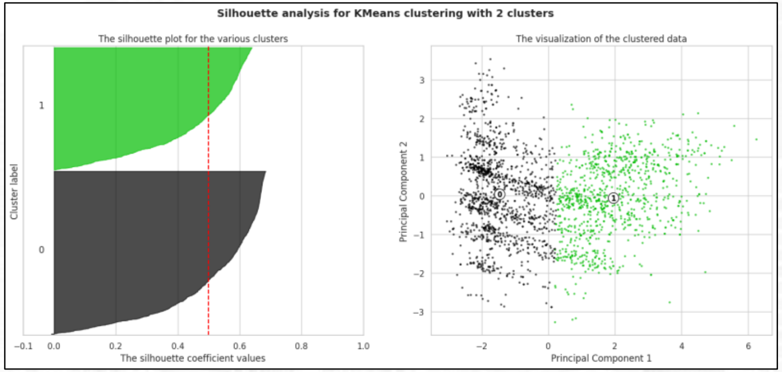
For more details, you can refer to the Jupyter notebook here

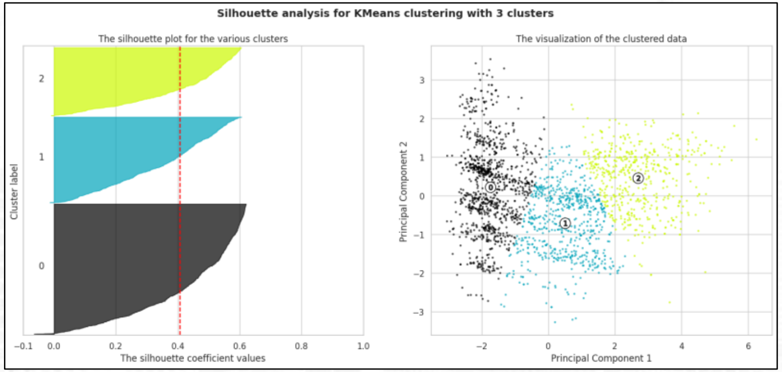
# Data Modeling: Clustering Algorithm using K-Means

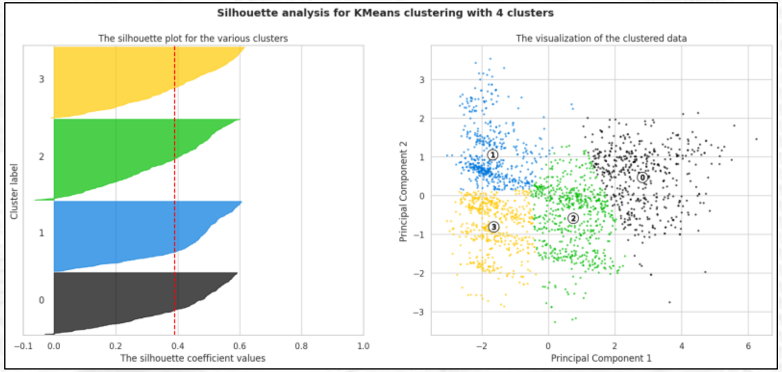
Here is the scatter plot visualization of the 4 clusters:

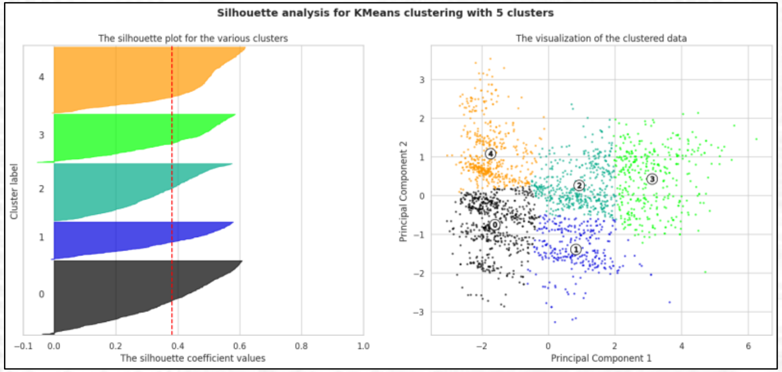


# Data Modeling: Evaluation using Silhouette Analysis


Silhouette analysis for KMeans clustering with 2 clusters

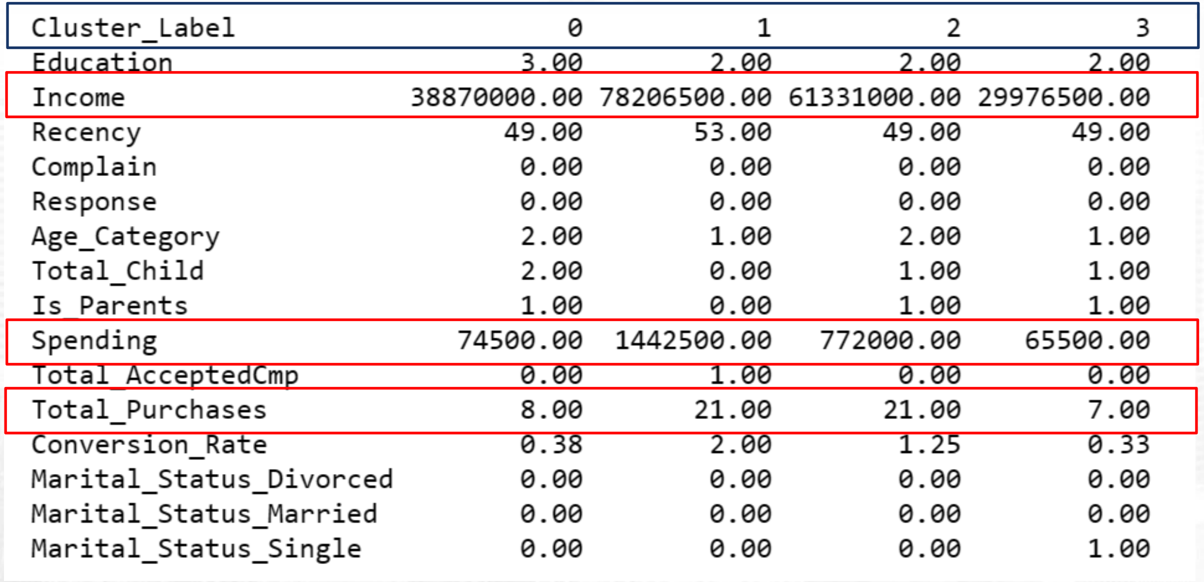

Silhouette analysis for KMeans clustering with 3 clusters


Silhouette analysis for KMeans clustering with 4 clusters


Silhouette analysis for KMeans clustering with 5 clusters

Silhouette analysis indicates that the optimal number of clusters is 4, as it exhibits a reasonably high coefficient value and balanced data distribution. However, the coefficient value is still relatively low, below 0.5, suggesting challenges in separating data due to complex dataset structure and similarities in customer personalities.

# Customer Personality Analysis for Marketing Retargeting



```
Cluster_Label                       0           1           2           3
Education                        3.00        2.00        2.00        2.00
Income                    38870000.00 78206500.00 61331000.00 29976500.00
Recency                         49.00       53.00       49.00       49.00
Complain                         0.00        0.00        0.00        0.00
Response                         0.00        0.00        0.00        0.00
Age_Category                     2.00        1.00        2.00        1.00
Total_Child                      2.00        0.00        1.00        1.00
Is_Parents                       1.00        0.00        1.00        1.00
Spending                     74500.00  1442500.00   772000.00    65500.00
Total_AcceptedCmp                0.00        1.00        0.00        0.00
Total_Purchases                  8.00       21.00       21.00        7.00
Conversion_Rate                  0.38        2.00        1.25        0.33
Marital_Status_Divorced          0.00        0.00        0.00        0.00
Marital_Status_Married           0.00        0.00        0.00        0.00
Marital_Status_Single            0.00        0.00        0.00        1.00
```

Based on the clustering results, we can conclude that there are four distinct customer groups:

**0:** Planners Customers, these are customers who tend to plan their purchases.

**1:** Prime Customers, this group represents the primary target for increasing revenue due to their superior characteristics.

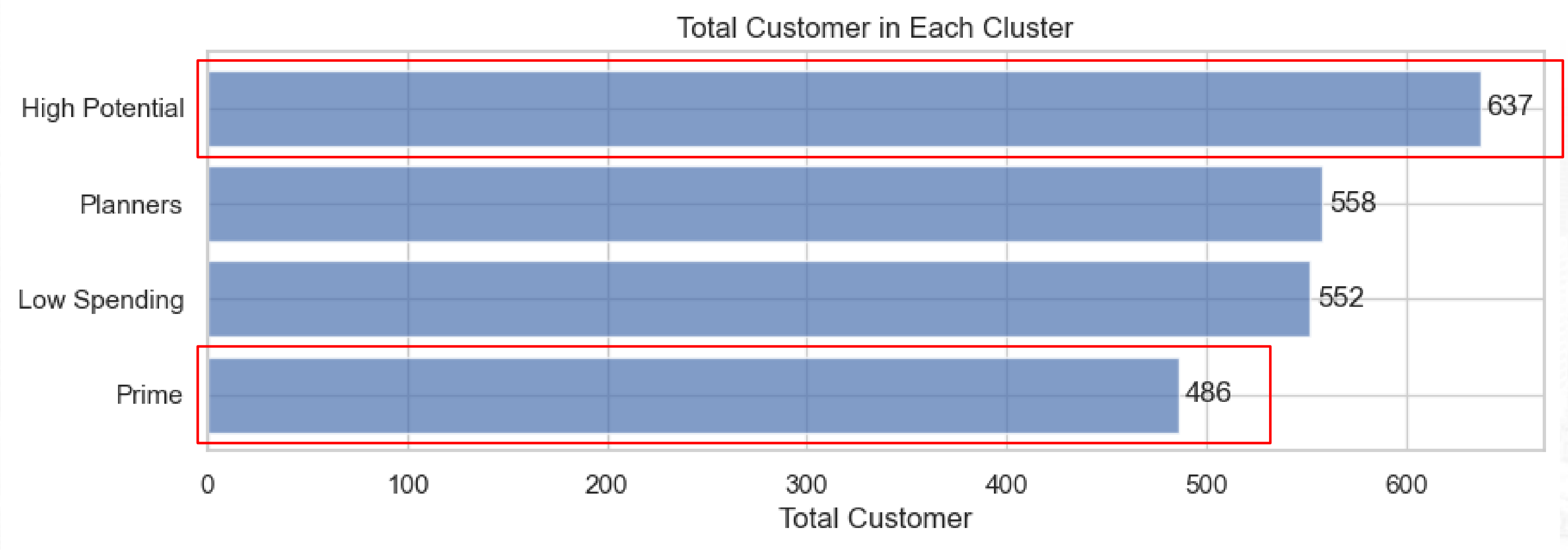
**2:** High Potential Customers, these customers exhibit high business potential and may require specialized marketing strategies.

**3:** Low Spending Customers, this group have low spending patterns.

For more details, you can refer to the Jupyter notebook here

# Customer Personality Analysis for Marketing Retargeting

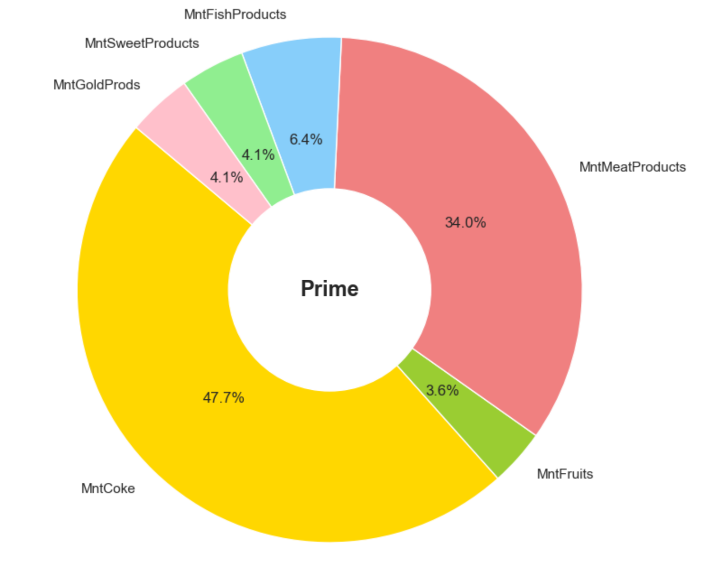
Here is the distribution of the number of customers in each cluster:

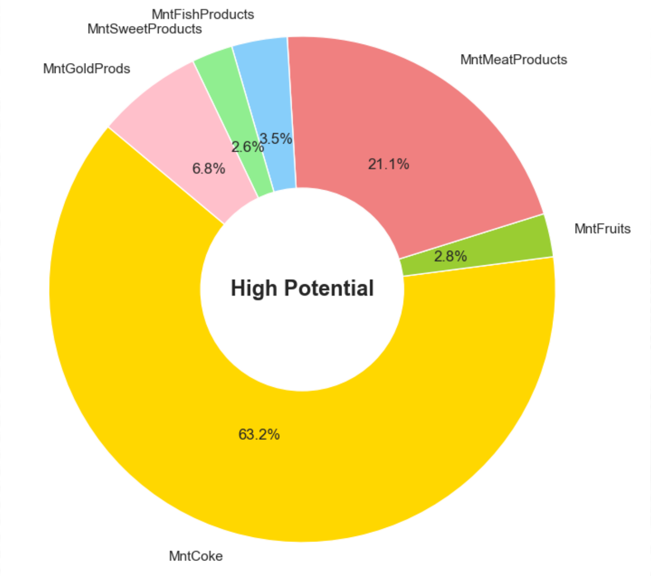


Considering the observed customer behavior, the ideal target audience would be the **High Potential** and **Prime** customers. These customers exhibit higher income, spending, and total purchases, making them more likely to enhance overall performance and boost marketing campaign effectiveness.

# Customer Personality Analysis for Marketing Retargeting

## Spending

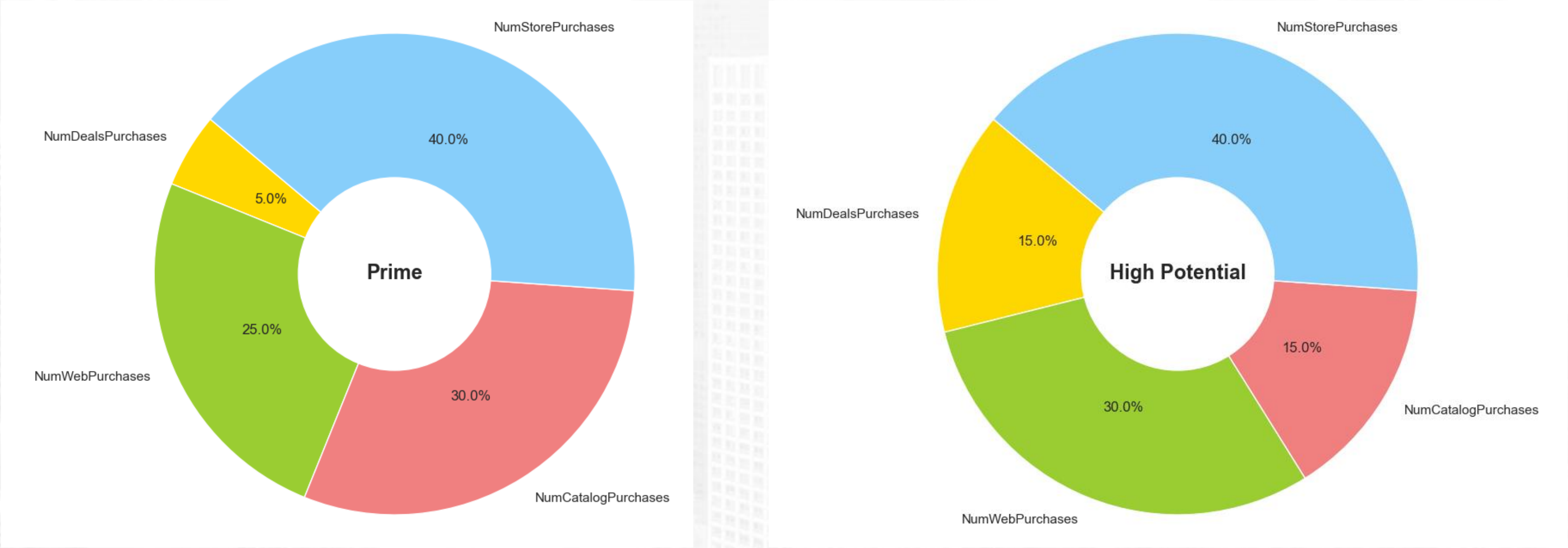




In terms of spending on the products offered, customers in both the 'Prime' and 'High Potential' clusters tend to allocate a significant portion of their spending on **Coke**, followed by **Meat** Products.

# Customer Personality Analysis for Marketing Retargeting

## Total Purchases



Based on the transactions conducted, customers in the **Prime** cluster are more likely to make purchases in the store and through catalogs. On the other hand, customers in the **High Potential** cluster tend to engage in transactions at physical stores as well as through the company's website.

# Customer Personality Analysis for Marketing Retargeting

Based on the previous analysis of customer behavior, the following recommendations can be made:

**1. For Prime Customers:**

- Offer a 10% discount or bundle deals for Coca-Cola and meat products.
- Provide special reward points or cashback for every purchase of Coca-Cola and meat products.
- Offer exclusive deals such as early access to new product offers or free delivery if they purchase on the same day the offer is presented.

**2. For High Potential Customers:**

- Implement cross-selling and upselling strategies by suggesting complementary products or services that match their preferences, such as dessert products or sauces.
- Provide an additional 5-10% discount when they purchase products online.
- Offer a subscription program that ensures regular delivery of their favorite products, such as Coca-Cola and meat.

Assuming that the recommendations are effective, the expected revenue to be obtained is Rp 3,553,000 for Cluster Prime Customers and Rp 3,894,000 for Cluster High Potential Customers. The total revenue would be Rp 7,447,000. The costs incurred would be Rp 1,458,000 for Cluster Prime Customers and Rp 1,911,000 for Cluster High Potential Customers, resulting in a total cost of Rp 3,369,000. Hence, the total profit would be Rp 7,447,000 - Rp 3,369,000 = Rp **4,078,000**.