CUSTOMER PERSONALITY ANALYSIS

description

It's all about understanding who your customers are, what they like, what they need, and how they behave.

businesses can tweak their products or services to better fit what their customers want.

businesses can figure out which group of customers is most likely to want their new product. Then, they can focus their efforts on marketing specifically to that group.

target

train a predictive model which allows the company to maximize the profit of the next marketing campaign

IMPORTING OUR LIBs , DATASET

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
from sklearn_extra.cluster import KMedoids
from mpl_toolkits.mplot3d import Axes3D
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
```

```
df = pd.read_csv('D:\\projects\\marketing_campaign_commas.csv')
```

CLEANING OF OUR DATASET

```
df.isna().sum().to_frame()
```

1 - We found that only one column has missing values about 24 values so we drop this rows away by :

	0
ID	0
Year_Birth	0
Education	0
Marital_Status	0
Income	24
Kidhome	0
Teenhome	0
Dt_Customer	0
Recency	0
MntWines	0
MntFruits	0
MntMeatProducts	0
MntFishProducts	0
MntSweetProducts	0
MntGoldProds	0
NumDealsPurchases	0

```
df = df.dropna()
```

```
df.isna().sum().sum()
```

0

Now is clean from missing values

It is the turn of dropping duplicates

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

0

We concluded that there is No duplicates

Detecting outliers in age of customers

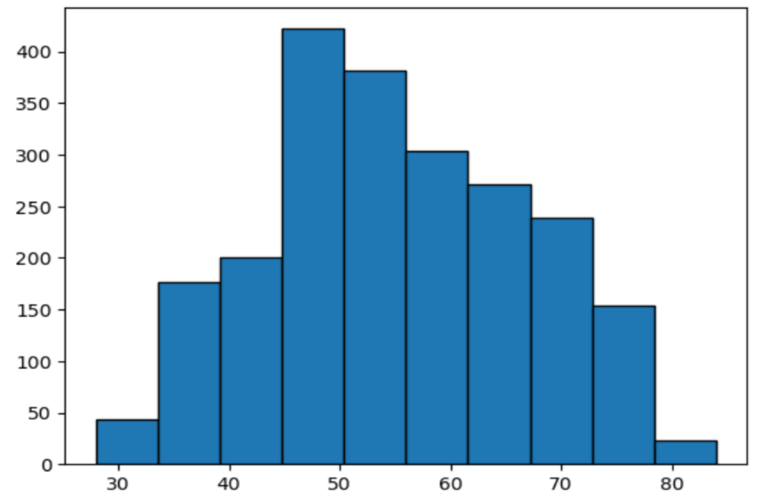
```
df = df[df['age']<90]  #removing outliers
```

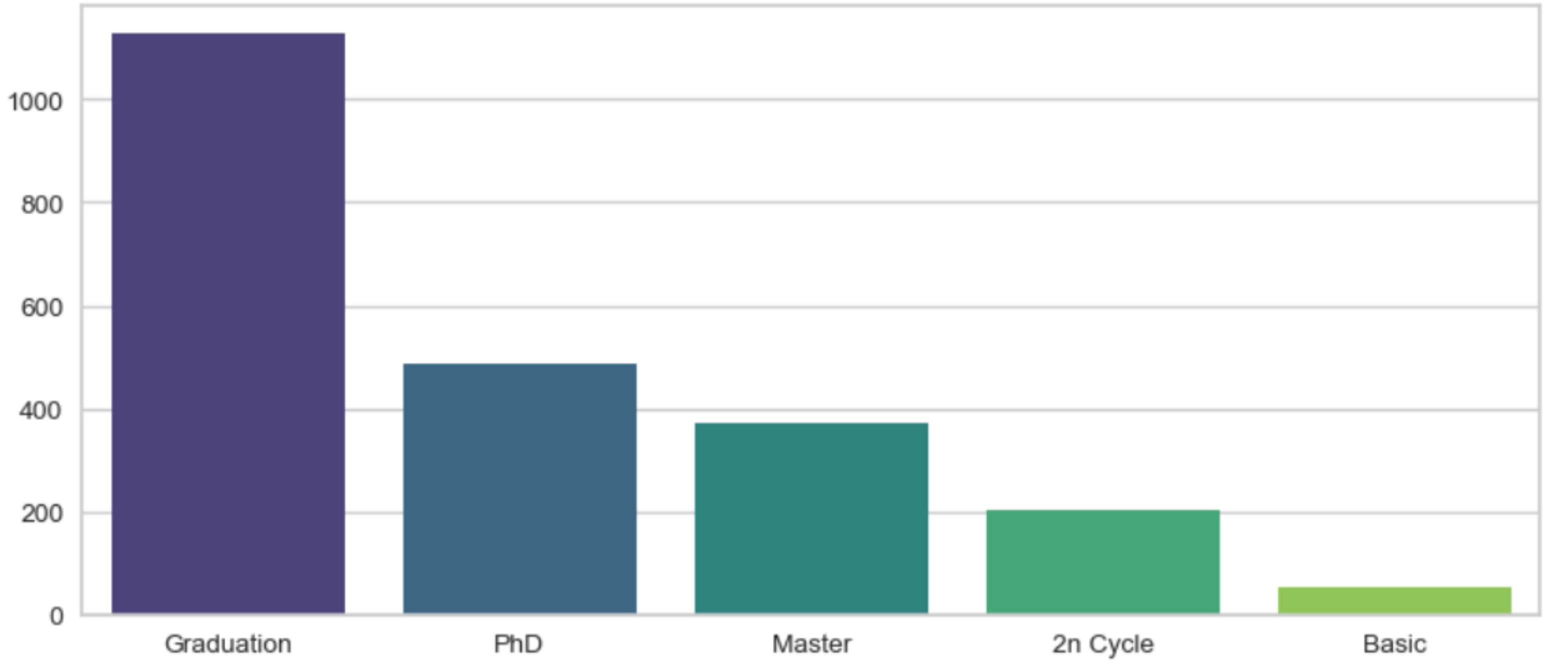
Exploratory Data Analysis

```
df['age'] = 2024 - df['Year_Birth'] # age of customers
df.drop('Year_Birth', axis=1, inplace=True)
```

```
df['years_joined'] = 2024 - df['Dt_Customer'].dt.year
df.drop('Dt_Customer', axis=1, inplace=True)
```

```
fig, ax = plt.subplots()
ax.hist(df['age'],edgecolor='black' );
```



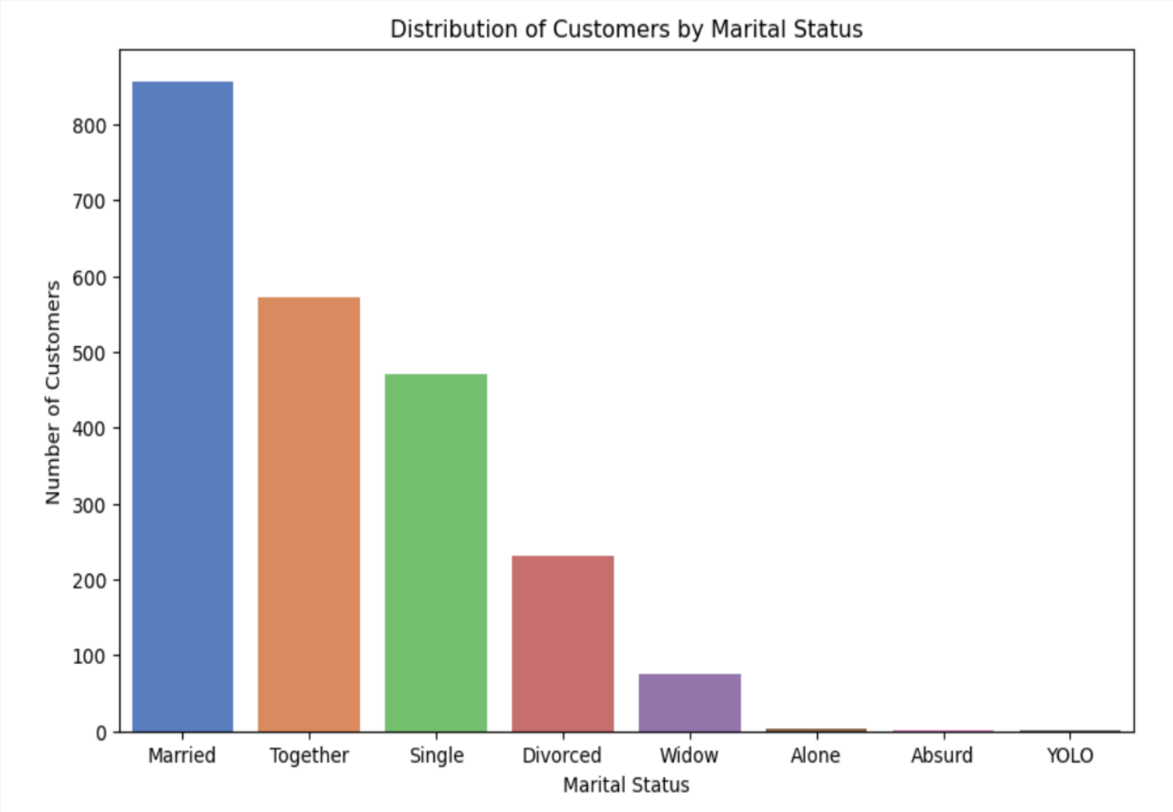


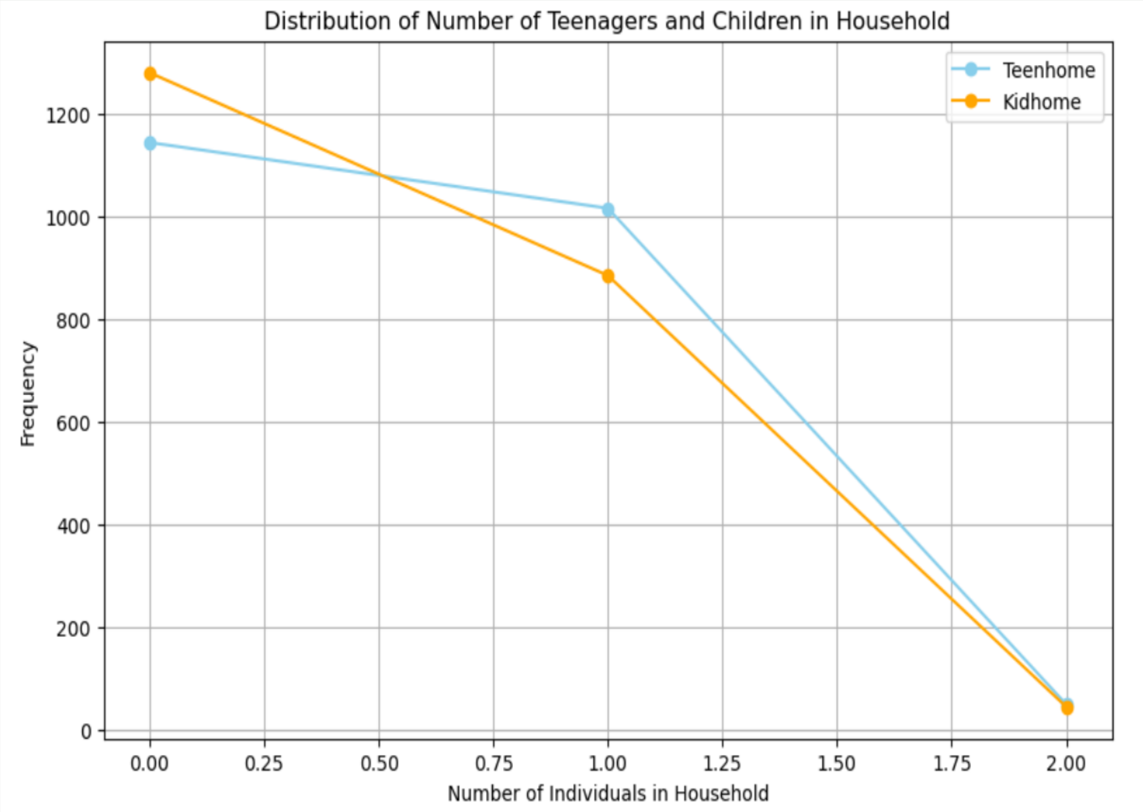
We can conclude :

1- avg age is 55

2 – most of the customers are have graduated from schools
(PhD – master – 2n cycle)

```
education_counts = df['Education'].value_counts()


plt.figure(figsize=(10, 6))
sns.barplot(x=education_counts.index, y=education_counts.values, palette='viridis')
plt.xlabel('Education Level')
plt.ylabel('Number of Customers')
plt.title('Distribution of Customers by Education Level')
plt.show()
```

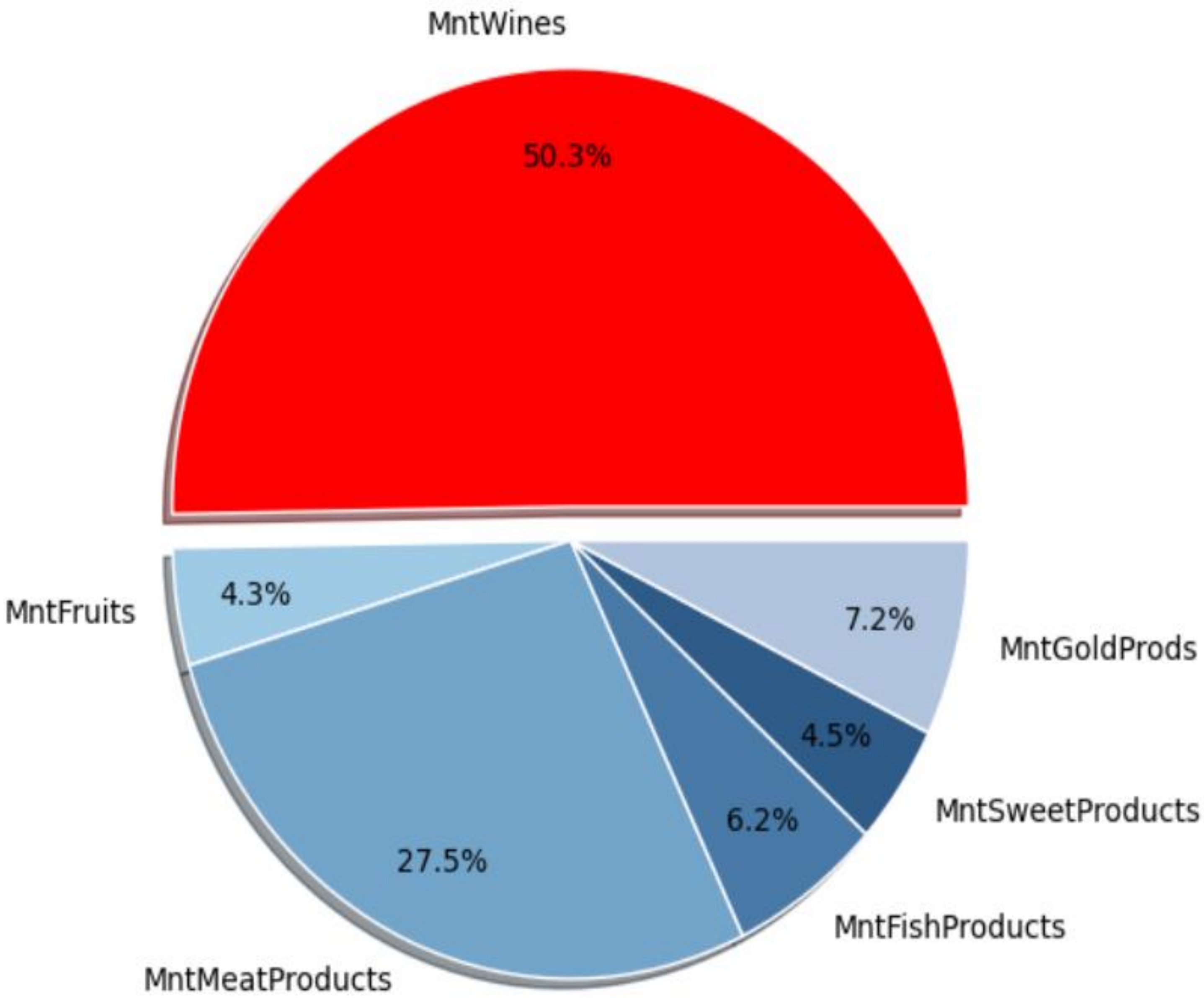




Most customers are married, followed by those who are in a relationship, and then those who are single.

Most customers have no children or one child

```
teen_counts = df['Teenhome'].value_counts()
kid_counts = df['Kidhome'].value_counts()
```




```
AmountSpentLastTWoYEAR =[df['MntWines'].sum(),
                df['MntFruits'].sum(),
                df['MntMeatProducts'].sum(),
                df['MntFishProducts'].sum(),
                df['MntSweetProducts'].sum(),
                df['MntGoldProds'].sum()]
```

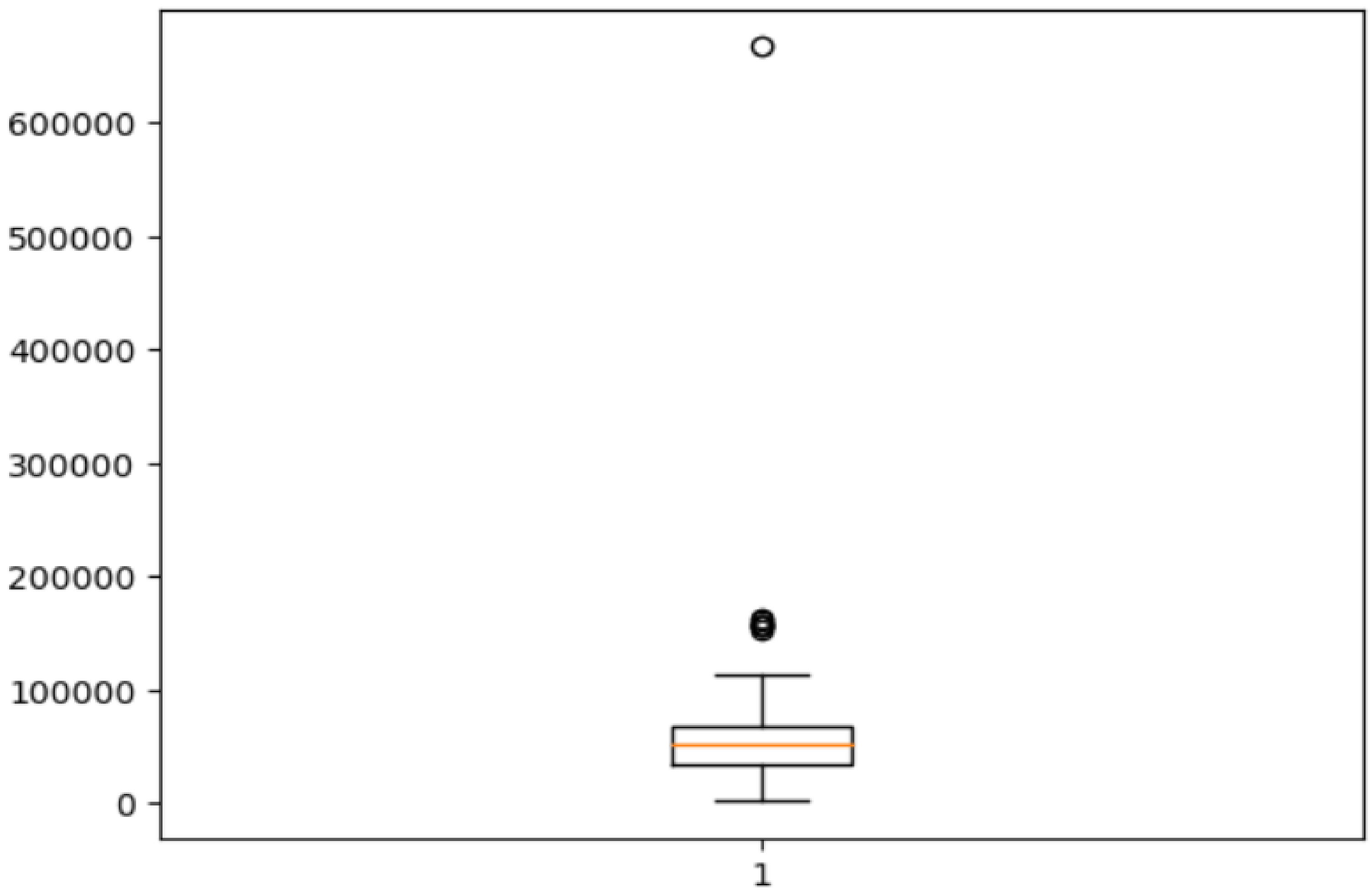
```
Labels = ['MntWines','MntFruits','MntMeatProducts','MntFishProducts','MntSweetProducts' , 'MntGoldProds' ]
Color = ["red", "#9FCAE6", "#73A4CA", "#497AA7", "#2E5B88" ,"#B0C4DE"]
fig ,ax = plt.subplots()
ax.pie(AmountSpentLastTWoYEAR ,labels = Labels ,radius = 1.3 ,colors = Color ,
    shadow = True , autopct = '%1.1f%%' , pctdistance = 0.8 ,
    explode = [0.1,0,0,0,0,0] ,
    wedgeprops ={"linewidth": 1, "edgecolor": "white"});
```

We can see that the amount spent on wine products is the largest amount spent overall products in the last 2 years

Customer’s yearly household income

75k $ AVG

```
fig, ax = plt.subplots()
ax.boxplot(df['Income']); #Customer’s yearly household income
```



```python
Q1 = df['Income'].quantile(0.25)
Q3 = df['Income'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df = df[(df['Income'] >= lower_bound) & (df['Income'] <= upper_bound)]
```

```python
df['total_purchases'] = df[['NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth']].sum(axis=1)
```

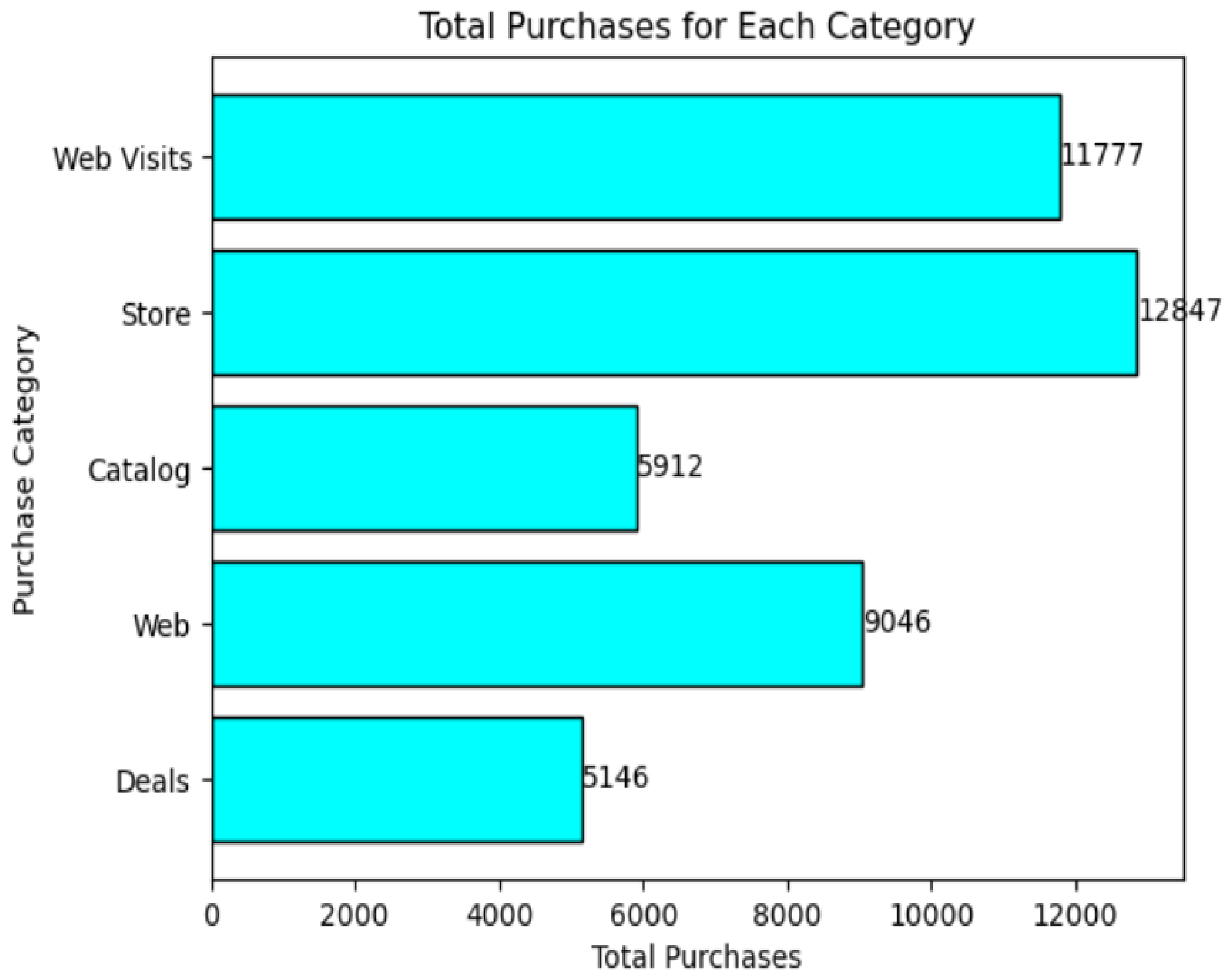
```python
#segmentation of Customer Incomes
income_bins = [0, 20000, 40000, 60000, 80000, 100000, 120000]
income_labels = ['0-20000', '20001-40000', '40001-60000', '60001-80000', '80001-100000', '100001-120000']

df['Income_Category'] = pd.cut(df['Income'], bins=income_bins, labels=income_labels)

income_purchase_mean = df.groupby('Income_Category')['total_purchases'].mean()
```

Detecting Outliers and filter data from them

As income rises, total purchases also increase

Speaking engagement metrics

Impact factor	Measurement	Target	Achieved
Audience interaction	Percentage (%)	85	88
Knowledge retention	Percentage (%)	75	80
Post-presentation surveys	Average rating	4.2	4.5
Referral rate	Percentage (%)	10	12
Collaboration opportunities	# of opportunities	8	10




```python
pur = [
    df['NumDealsPurchases'].sum(),
    df['NumWebPurchases'].sum(),
    df['NumCatalogPurchases'].sum(),
    df['NumStorePurchases'].sum(),
    df['NumWebVisitsMonth'].sum()
]

labels = ['Deals', 'Web', 'Catalog', 'Store', 'Web Visits']

plt.barh(y=range(len(pur)), width=pur, color="cyan", edgecolor='black',capsize=10)
plt.yticks(ticks=range(len(pur)), labels=labels)

for i, val in enumerate(pur):
    plt.text(val, i, str(val), ha='left', va='center')

plt.xlabel('Total Purchases')
plt.ylabel('Purchase Category')
plt.title('Total Purchases for Each Category')
plt.show()
```

We found something interesting :
Web visits in the last month was massive compared to the transactions (web) as the avg age of the dataset is 56

After conducting a thorough analysis :
it's evident that the majority of purchases, totaling 12,841, are made in-store, making it the highest channel for transactions.
Following closely behind are purchases through web at 9,042 transactions
deals and catalogs represent the lowest volume of transactions.

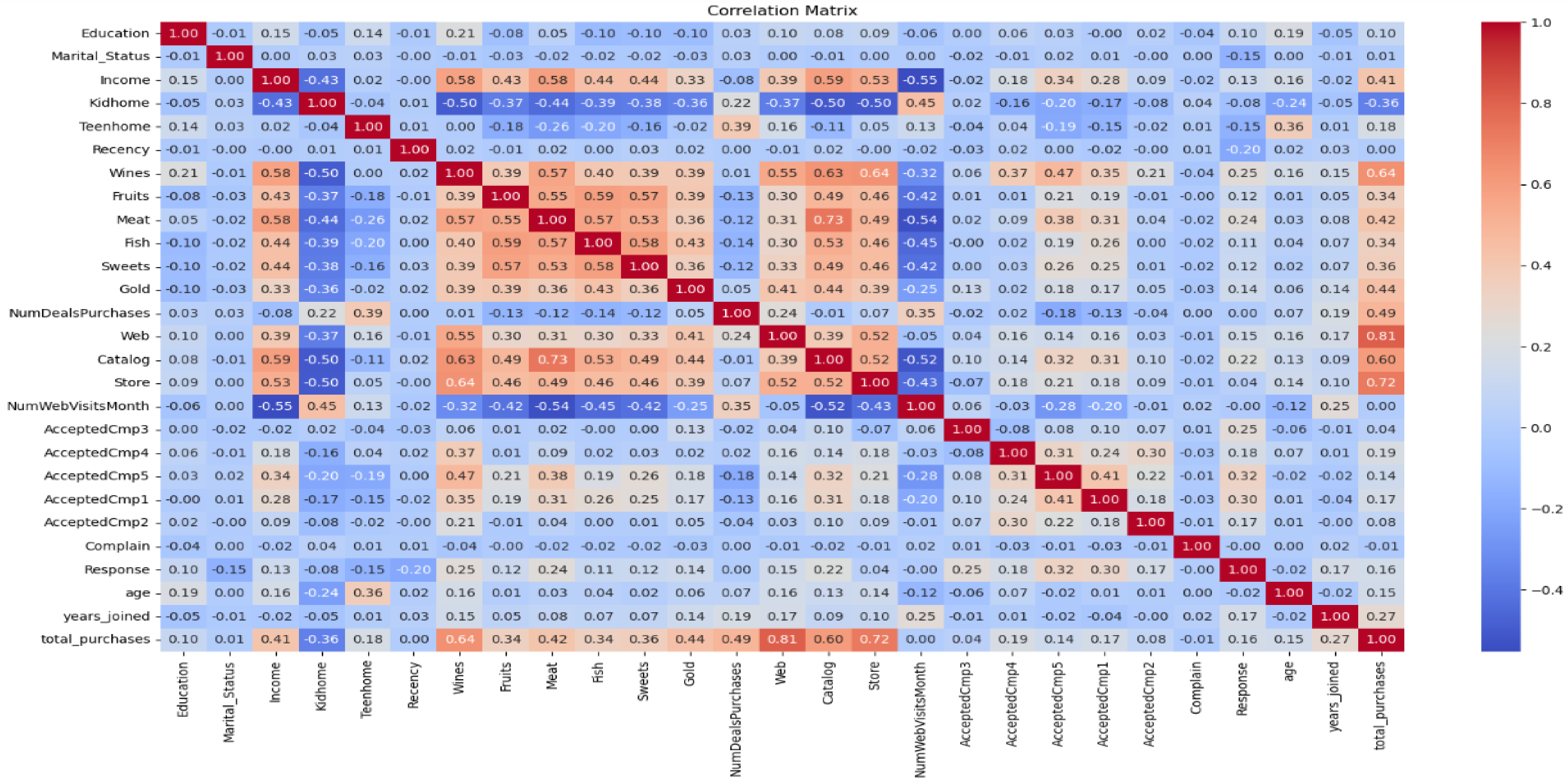
preprocessing before Clustering

```
# limit values
data = df.drop(['ID','Z_CostContact', 'Z_Revenue', 'Income_Category'], axis=1)
data['Marital_Status']=data['Marital_Status'].replace({'Divorced':'Alone','Single':'Alone','Married':'couple','Together':'couple','Absurd':'Alone','Widow':'Alone','YOLO
# data['Education']=data['Education'].replace({'Basic':'Undergraduate','2n Cycle':'Undergraduate','Graduation':'Postgraduate','Master':'Postgraduate','PhD':'Postgra
data=data.rename(columns={'MntWines': "Wines",'MntFruits':'Fruits','MntMeatProducts':'Meat','MntFishProducts':'Fish','MntSweetProducts':'Sweets','MntGoldPro
data=data.rename(columns={'NumWebPurchases': "Web",'NumCatalogPurchases':'Catalog','NumStorePurchases':'Store'})
data.columns
```

```
category_order = ['Basic', '2n Cycle', 'Graduation', 'Master', 'PhD']
data['Education'] = pd.Categorical(data['Education'], categories = category_order, ordered = True)
data['Education'] = data['Education'].cat.codes
mapping = {'Alone': 1, 'couple': 2}
data['Marital_Status'] = data['Marital_Status'].map(mapping)
data.head()
```

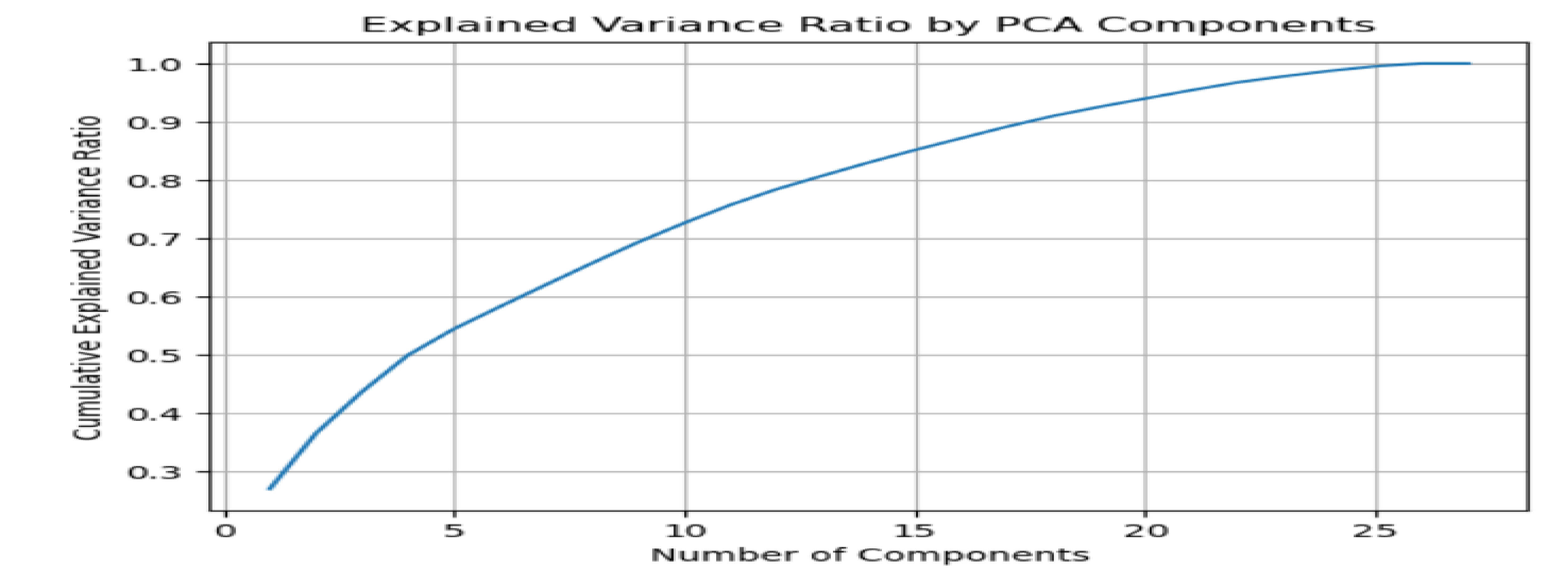
```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
scaled_data = pd.DataFrame(scaled_data, columns=data.columns)
scaled_data.head()
```

Correlation Matrix



there are correlated features, and the dimensionality is high, so we do PCA

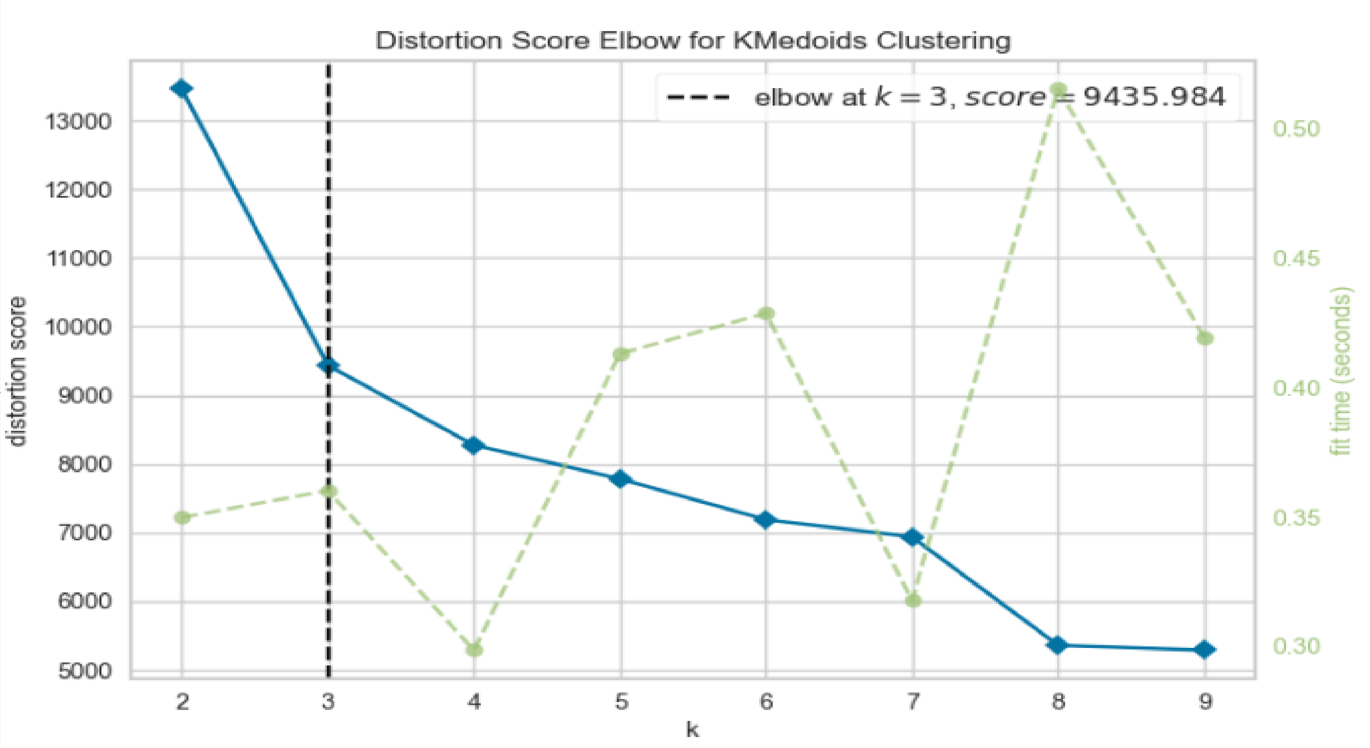
```
pca = PCA()
pca.fit(scaled_data)
cumsum = np.cumsum(pca.explained_variance_ratio_)
plt.plot(range(1, len(cumsum) + 1), cumsum)  # Plot component number vs. cumulative explained variance ratio
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Explained Variance Ratio')
plt.title('Explained Variance Ratio by PCA Components')
plt.grid(True)
plt.show()
```



K-MEDOID

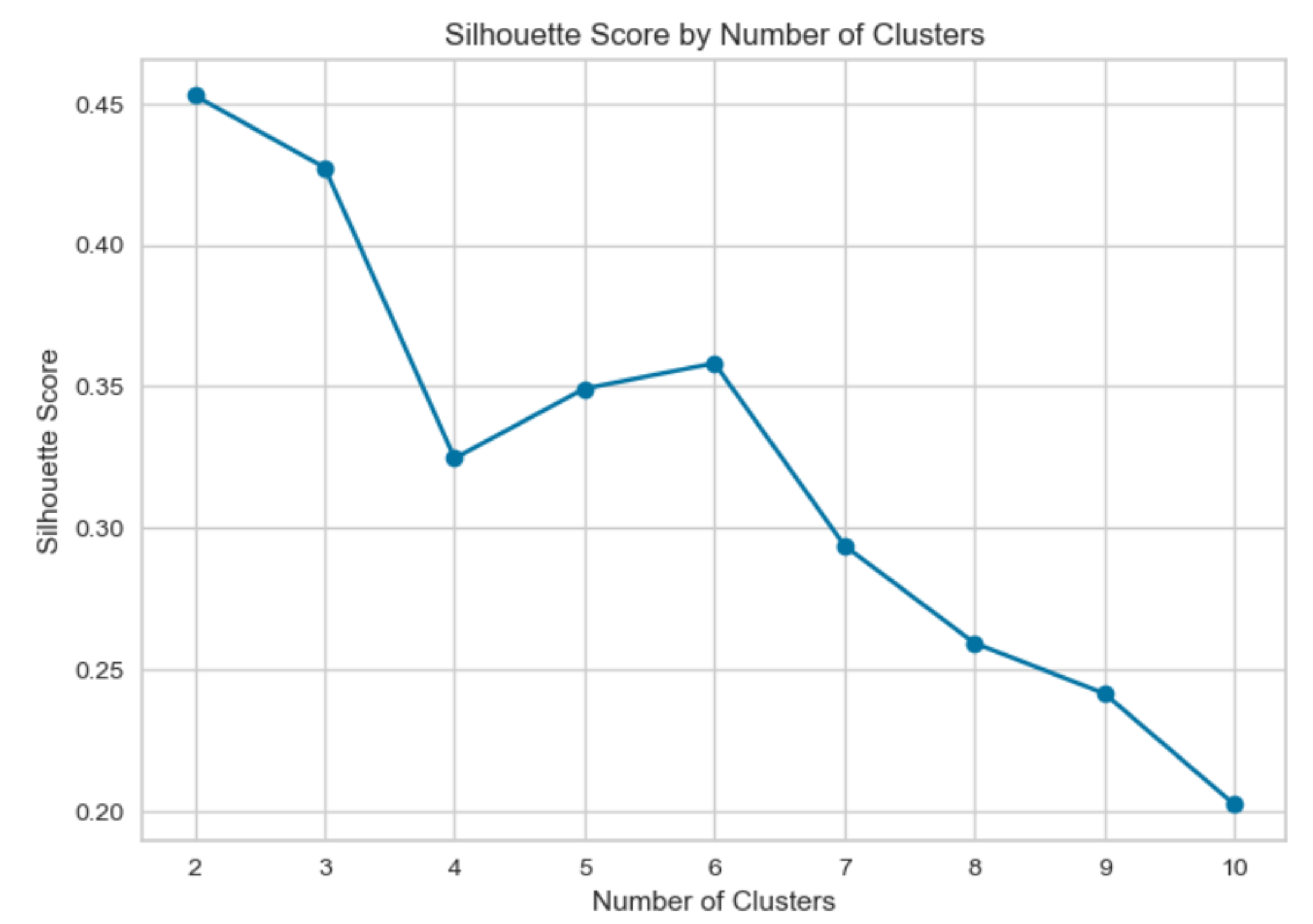
WE MUST DECIDE HOW MANY CLUSTERS BEFORE BEGAIN WITH K-MEDOID
SO WE CAN APPLY **Elbow For K-Medoids**

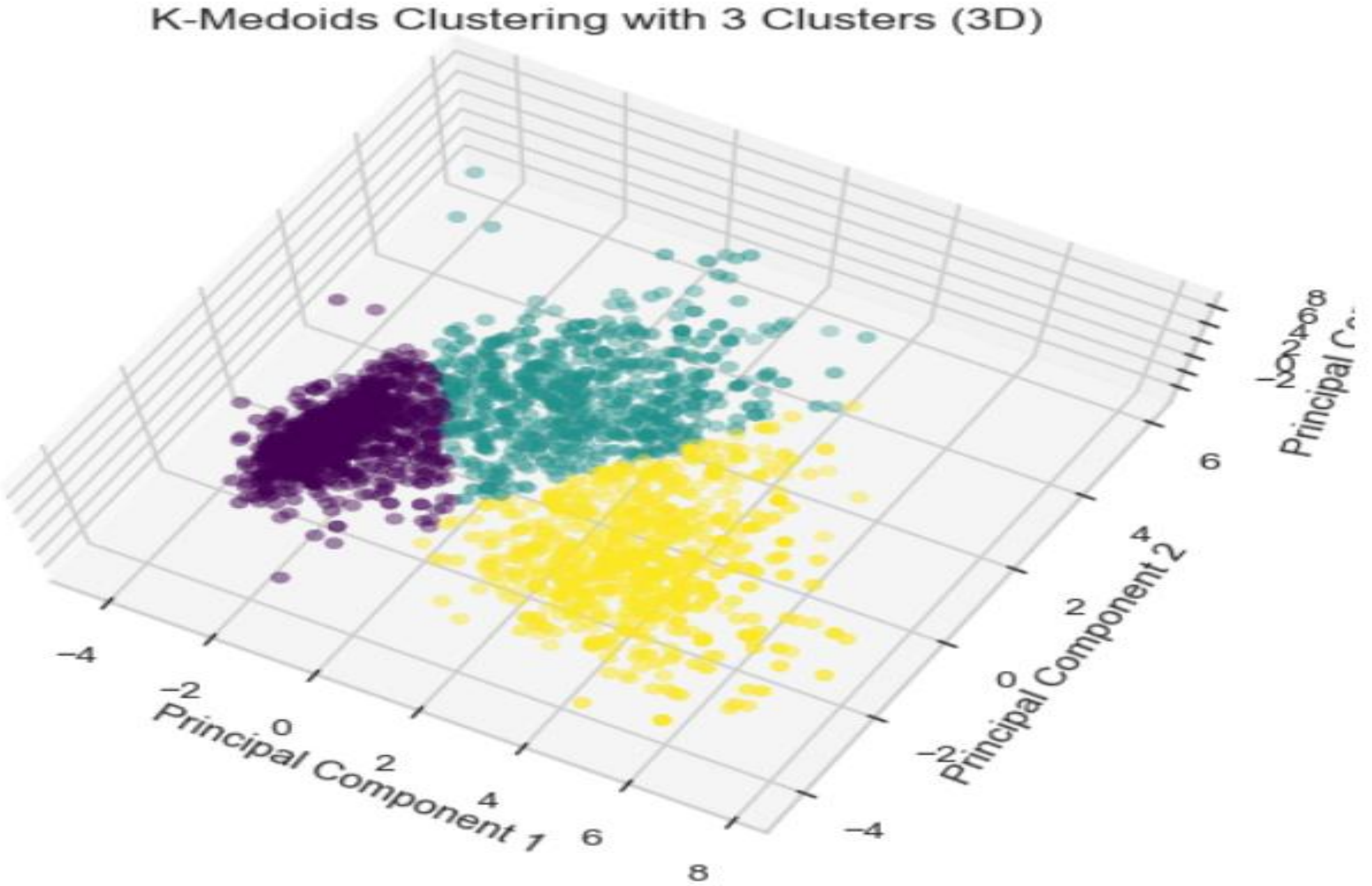
```
from yellowbrick.cluster import KElbowVisualizer
from sklearn_extra.cluster import KMedoids
model = KMedoids()
visualizer = KElbowVisualizer (model, k=(2,10))
visualizer.fit(pca_data)
visualizer.show()
```



After Elbow , We can see that the appropriate number of clusters is 3

```python
# k-mediods clustering
silhouette_scores = []
for k in range(2, 11):
    kmedoids = KMedoids(n_clusters=k, random_state=42)
    kmedoids.fit(pca_data)
    silhouette_scores.append(silhouette_score(pca_data, kmedoids.labels_))

plt.plot(range(2, 11), silhouette_scores, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score by Number of Clusters')
plt.grid(True)
plt.show()
```




K-Medoids Clustering with 3 Clusters (3D)
Principal Component 1
−4
−2
0
2
4
6
8
Principal Component 2
−4
−2
0
2
4
6

group by clusters

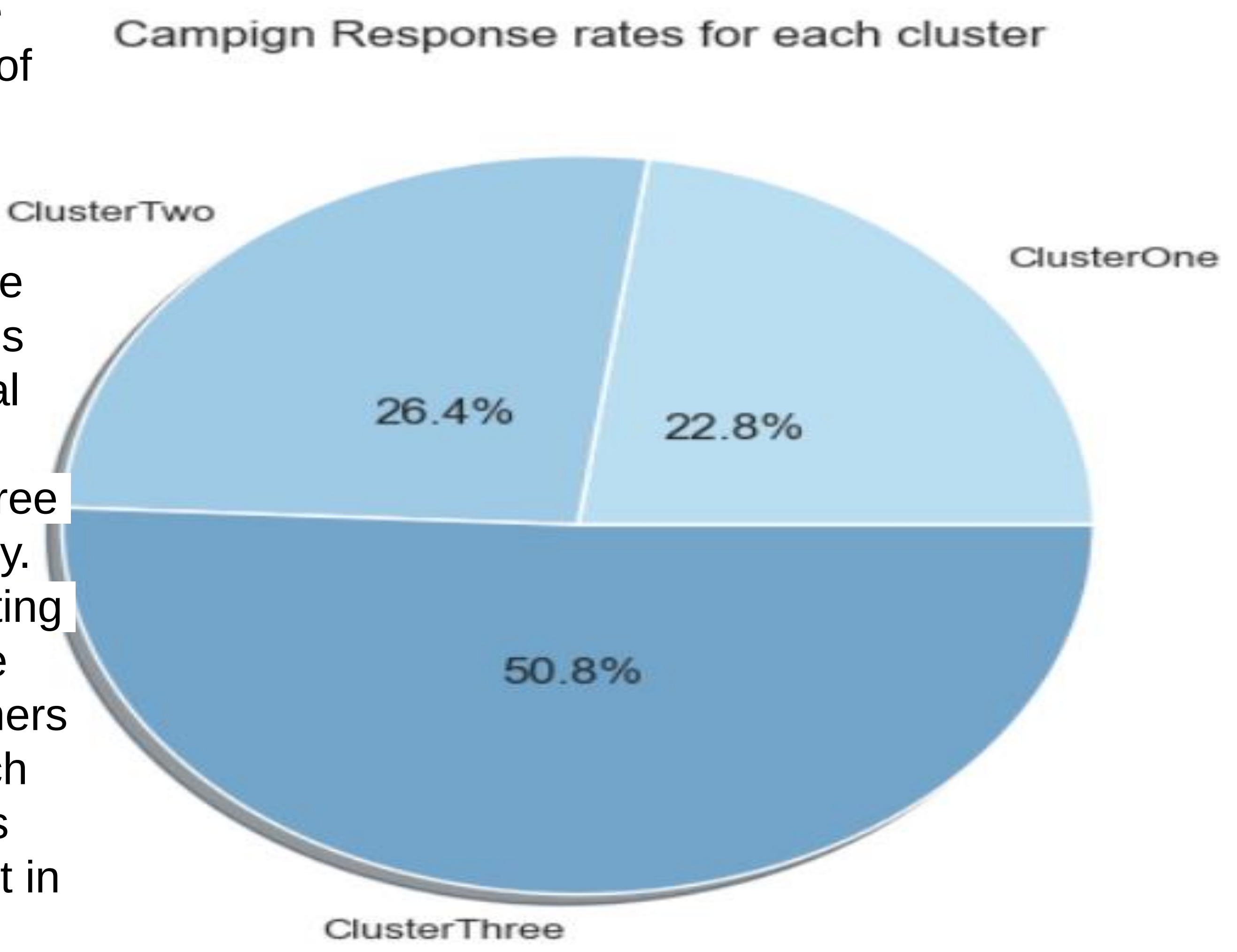
```
cluster_groups_mean = data.groupby(['cluster']).agg({
    'Income': 'mean',
    'age': 'mean',
    'years_joined': 'mean',
    'total_purchases': 'mean',
    'Wines': 'mean',
    'Fruits': 'mean',
    'Meat': 'mean',
    'Fish': 'mean',
    'Sweets': 'mean',
    'Gold': 'mean',
    'NumDealsPurchases': 'mean',
    'Web': 'mean',
    'Catalog': 'mean',
    'Store': 'mean',
    'NumWebVisitsMonth': 'mean',
    'AcceptedCmp3': 'sum',
    'AcceptedCmp4': 'sum',
    'AcceptedCmp5': 'sum',
    'AcceptedCmp1': 'sum',
    'AcceptedCmp2': 'sum',
    'Complain': 'sum',
    'Response': 'sum'
})
cluster_groups_sum = data.groupby(['cluster']).sum()
```

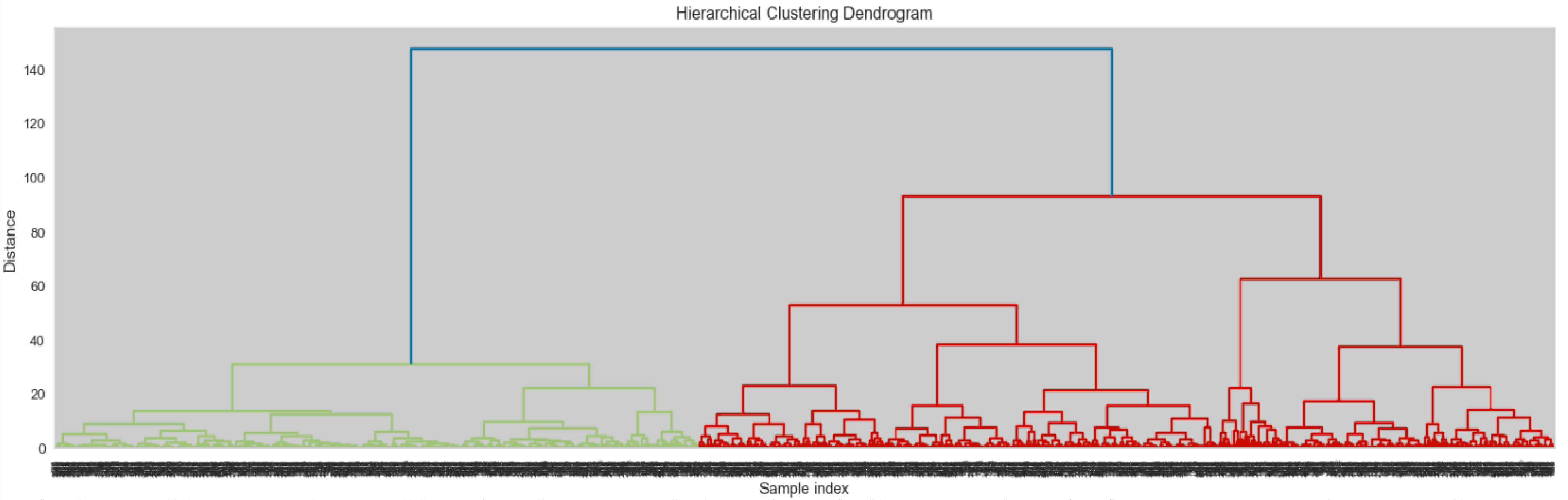
```
#Separate data by each cluster among three clusters
c1 = data[data['cluster'] == 0]
c2 = data[data['cluster'] == 1]
c3 = data[data['cluster'] == 2]

# Calculate response rates for each cluster
cluster_groups_sum['Response Rate'] = (cluster_groups_sum['Response'] / cluster_groups_sum['Response'].sum()) * 100
```

Cluster three exhibits the highest campaign response rate among all clusters. This indicates that a larger proportion of customers in cluster three responded positively to the marketing campaigns compared to customers in other clusters.

Based on the analysis, it's clear that people in cluster three respond better to marketing campaigns. This means stores have a chance to make the most of this by creating special marketing plans just for them. By doing this, stores can make sure they're offering things that people in cluster three like, which can help them sell more and make more money.

In addition to having the most positive response to marketing campaigns, cluster three stands out with its higher income level and an average age of around 55. Moreover, customers in this cluster tend to spend more on various products such as meat, wine, gold, fish, sweets, and fruits. This indicates that they have both the means and the inclination to invest in higher-quality products. Stores can use this information to make their marketing plans better. They can focus more on selling high-quality products and giving special deals to people in cluster three. By doing this, stores can make customers in this group happier and sell more, which means they'll make more money in the end.





plt.figure(figsize=(20, 5)) dendrogram(Z) plt.title('Hierarchical ClusteringDendrogram')
plt.xlabel('Sample index')
plt.ylabel('Distance')
plt.show()

Hierarchical clustering

```python
# agglomerative
agglo = AgglomerativeClustering(n_clusters=2)
agglo.fit_predict(pca_data)
data['agglo_cluster'] = agglo.labels_
```

✓ 0.2s

```python
cluster_groups_mean = data.groupby('agglo_cluster').agg({
    'Income': 'mean',
    'age': 'mean',
    'years_joined': 'mean',
    'total_purchases': 'mean',
    'Wines': 'mean',
    'Fruits': 'mean',
    'Meat': 'mean',
    'Fish': 'mean',
    'Sweets': 'mean',
    'Gold': 'mean',
    'NumDealsPurchases': 'mean',
    'Web': 'mean',
    'Catalog': 'mean',
    'Store': 'mean',
    'NumWebVisitsMonth': 'mean',
    'AcceptedCmp3': 'sum',
    'AcceptedCmp4': 'sum',
    'AcceptedCmp5': 'sum',
    'AcceptedCmp1': 'sum',
    'AcceptedCmp2': 'sum',
    'Complain': 'sum',
    'Response': 'sum'
})
cluster_groups_sum = data.groupby(['agglo_cluster']).sum()
cluster_groups_mean.T
```

Agglomerative with 2 clusters

Income	64410.098859	32727.459551
age	57.406844	51.680899
years_joined	11.063118	10.838202
total_purchases	24.606844	13.750562
Wines	492.987072	30.129213
Fruits	41.506464	4.087640
Meat	264.946768	18.098876
Fish	59.427376	5.737079
Sweets	42.698859	4.122472
Gold	64.820532	13.378652
NumDealsPurchases	2.609125	1.888764
Web	5.596958	1.889888
Catalog	4.143726	0.431461
Store	7.727757	3.010112
NumWebVisitsMonth	4.529278	6.530337
AcceptedCmp3	90.000000	73.000000
AcceptedCmp4	153.000000	11.000000
AcceptedCmp5	161.000000	0.000000
AcceptedCmp1	142.000000	0.000000
AcceptedCmp2	28.000000	2.000000
Complain	8.000000	12.000000
Response	243.000000	90.000000

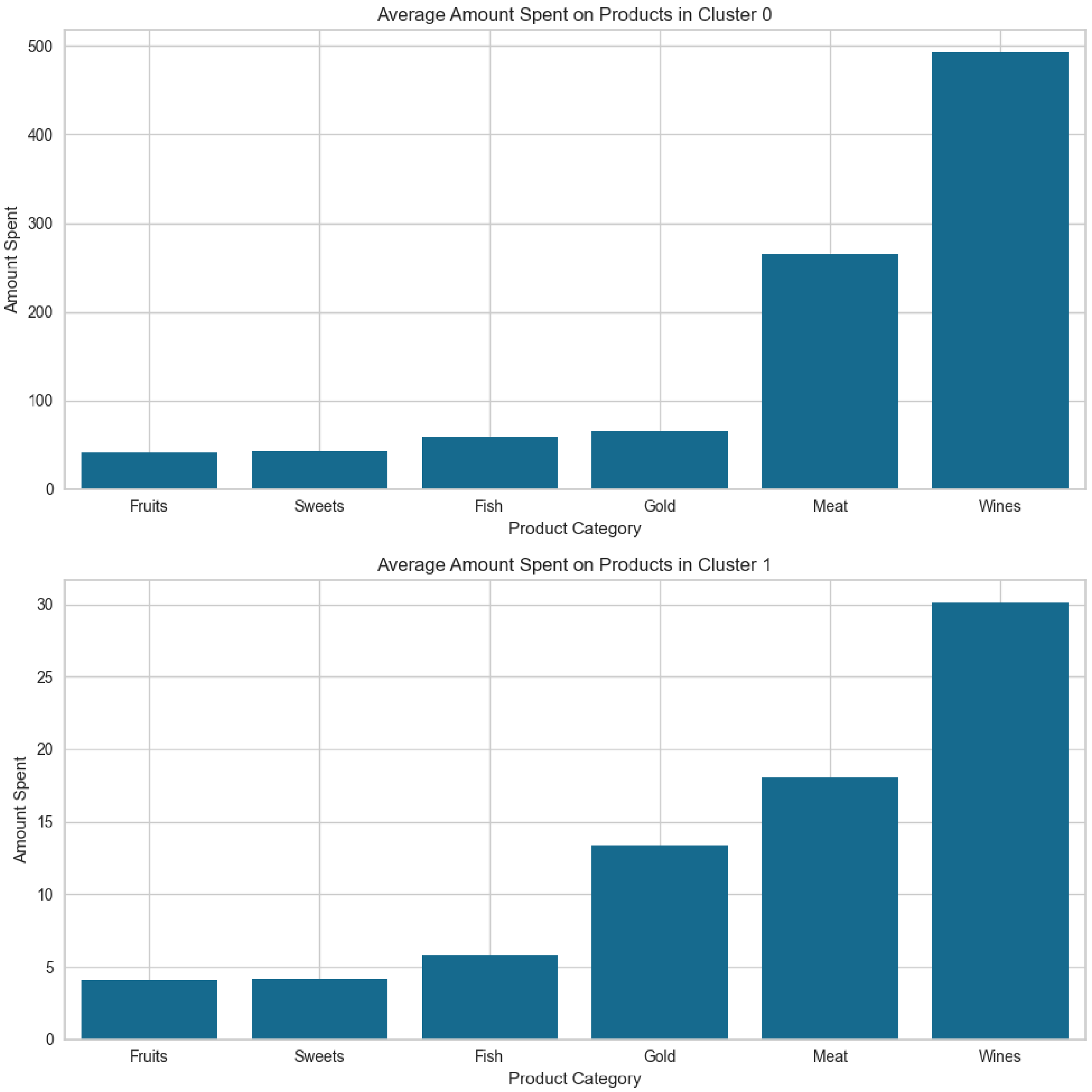
We notice the following:

- cluster 0 has high avg income while cluster 1 has low avg income
- cluster 0 spend more in avg and respond more to deals
- cluster 1 visit the website more often while cluster 0 buy from website more
- cluster 1 tend to complain more than cluster 0

Comparing clusters

Income	64410.098859	32727.459551
age	57.406844	51.680899
years_joined	11.063118	10.838202
total_purchases	24.606844	13.750562
Wines	492.987072	30.129213
Fruits	41.506464	4.087640
Meat	264.946768	18.098876
Fish	59.427376	5.737079
Sweets	42.698859	4.122472
Gold	64.820532	13.378652
NumDealsPurchases	2.609125	1.888764
Web	5.596958	1.889888
Catalog	4.143726	0.431461
Store	7.727757	3.010112
NumWebVisitsMonth	4.529278	6.530337
AcceptedCmp3	90.000000	73.000000
AcceptedCmp4	153.000000	11.000000
AcceptedCmp5	161.000000	0.000000
AcceptedCmp1	142.000000	0.000000
AcceptedCmp2	28.000000	2.000000
Complain	8.000000	12.000000
Response	243.000000	90.000000

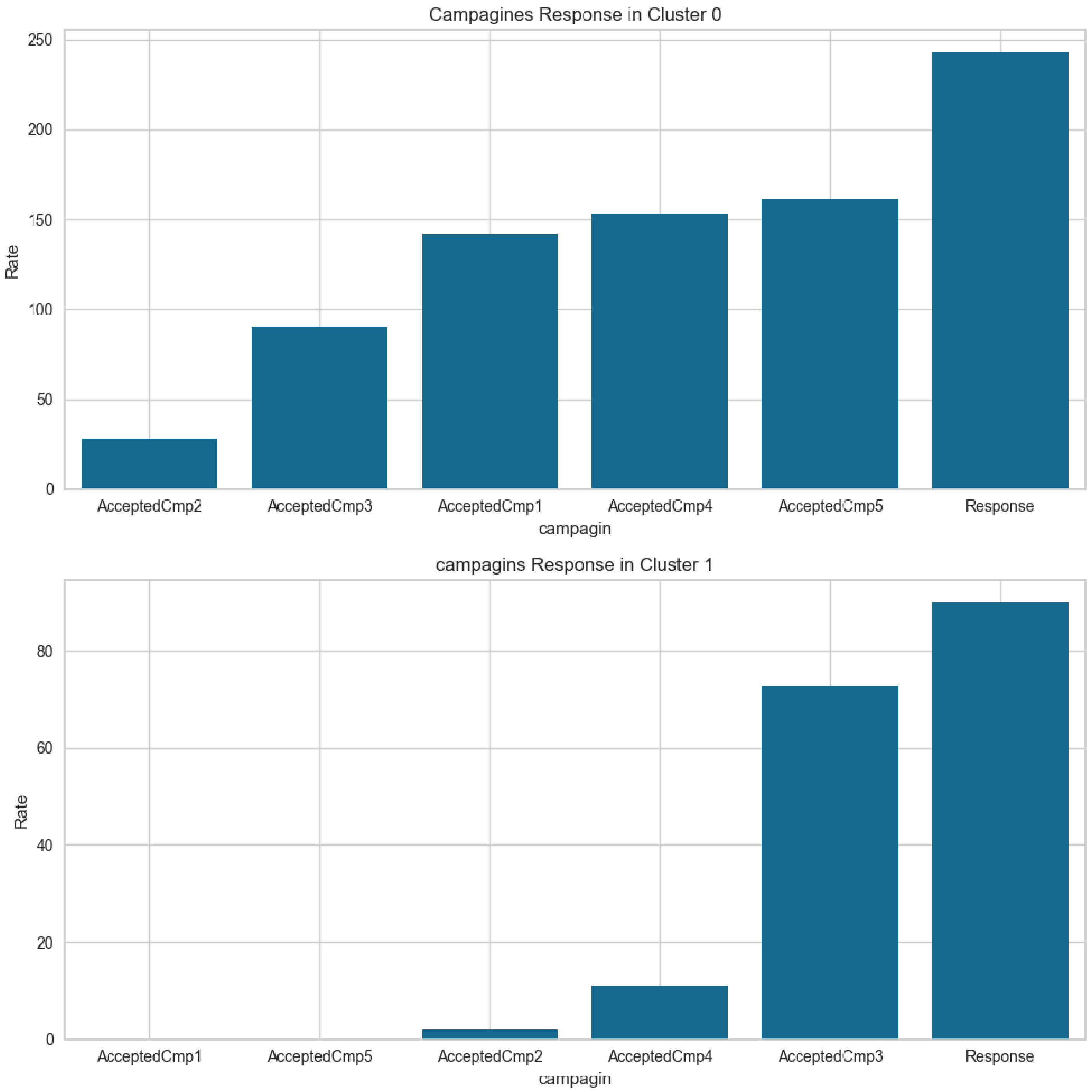
Both clusters spend the same proportion with different amounts (cluster 0 spends more)



Last campaign targeted both clusters the most

Cluster 0 :
They responded most to camp 5, 4, 1
While camp 2 was not suitable for them.

Cluster 1:
They responded most to cluster 3
Got low responses in 2, 4
And never accepted camp 1, 5 they were not targeting this cluster of customer



Evaluation

- both clustering methods gave similar silhouette score at the chosen number of clusters
- while kmediods gave best fit at 3 clusters , hierarchical clustering gave best fit at 2 clusters
- both methods provided valuable insights into customer segmentation, offering different perspectives on grouping customers and which campaigns targeted them the most which will help in targeting them in future campaigns.