Customer Personality Analysis using Clustering

BY GRISHMA VEMIREDDY

Introduction

Problem statement:

Perform customer segmentation to understand different groups of customers.

Using clustering algorithm like K-means and Hierarchical, try to identify meaningful patterns.

Purpose:

Helps businesses target marketing efforts more effectively.

Identifies different types of consumer groups

```
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 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
```

Dataset overview

Source: Kaggle (Customer Segmentation : Clustering)

Data description:

2,240 total observations

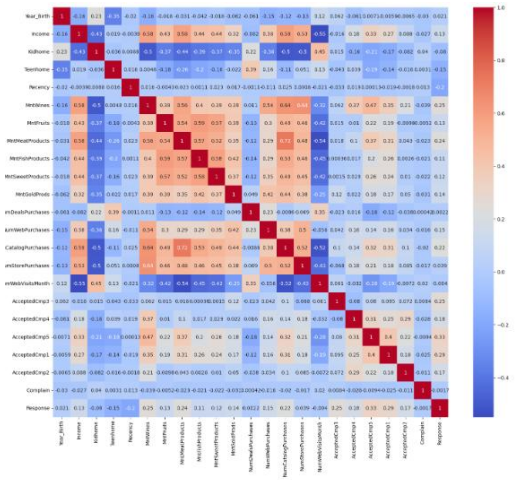
29 features

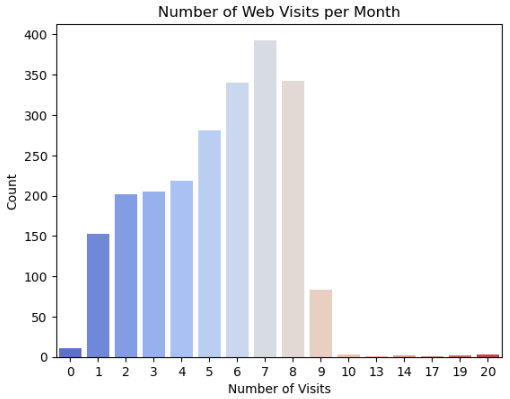
Some missing values performed Imputation (using mean)

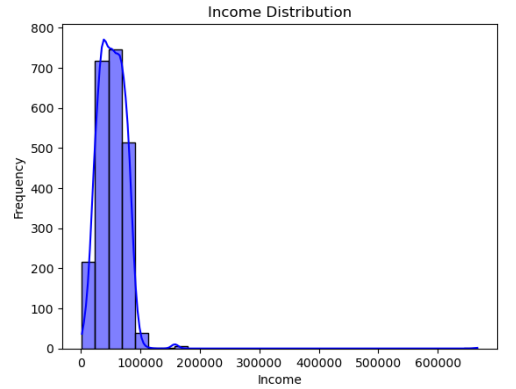
```
data_df = pd.read_csv("customer_segmentation.csv")
data_df.head()
```

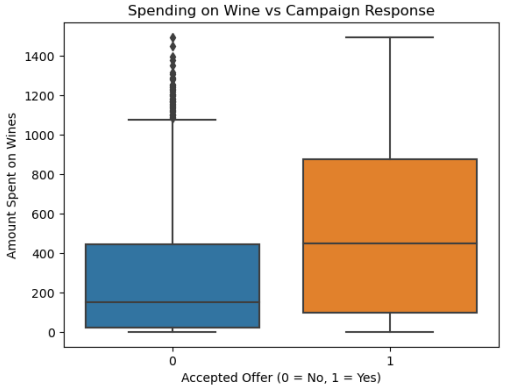
	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	R
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	

5 rows × 29 columns









Methodology

Data cleaning and preparation:

Handled missing data by imputation

Encoded categorical variables

Dropped some irreverent features

Scaled numerical features (ScandardScaler)

Data exploration:

Plotted distribution of Income, spending on Wine and other spending features to identify and patterns in spending.

The heatmap shows some correlation between some features.

Modeling

K-means clustering:

Partition data into specific numb of cluster, K need to be defined so used elbow method and Silhouetted score analysis.

This method is simple, fast, and scalable on large datasets

```
# K-means clustering

#find optimal K
from sklearn.cluster import KMeans

# where we'll store all of the wcss values for plotting later.
wcss = []
silhouette_scores = []

for i in range(2, 11):
    # random_state just to ensure we get the same values in the end.
    kmeans = KMeans(n_clusters = i, random_state = 42)
    kmeans.fit(scaled_data)
    # inertia method returns wcss for that model.
    wcss.append(kmeans.inertia_)
    silhouette_avg = silhouette_score(scaled_data, kmeans.labels_)
    silhouette_scores.append(silhouette_avg)
    print(f"Clusters: {i}, Silhouette Score: {silhouette_avg:.4f}")
```

Hierarchical clustering:

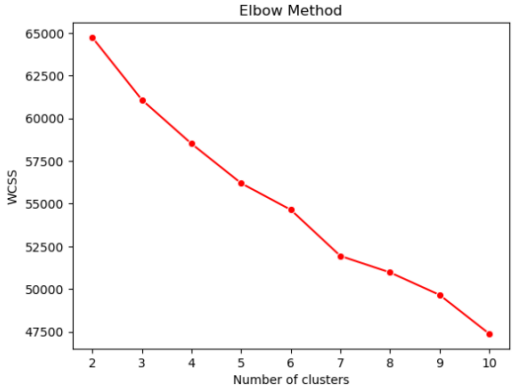
Agglomerative Hierarchical Clustering starts with each point as a cluster and mergers as they go up. This method is good at discovering nested or hierarchical relationships. No need to specify k, works well with complex data.

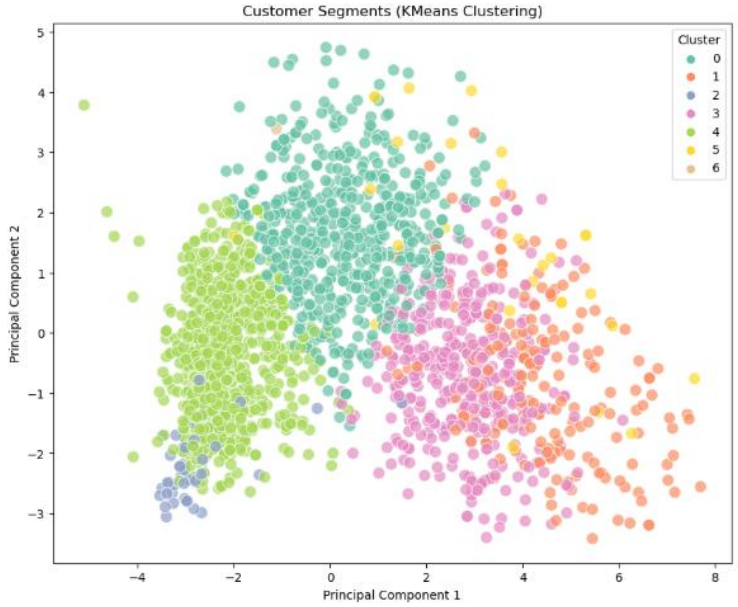
Performed PCA as there are so many features.

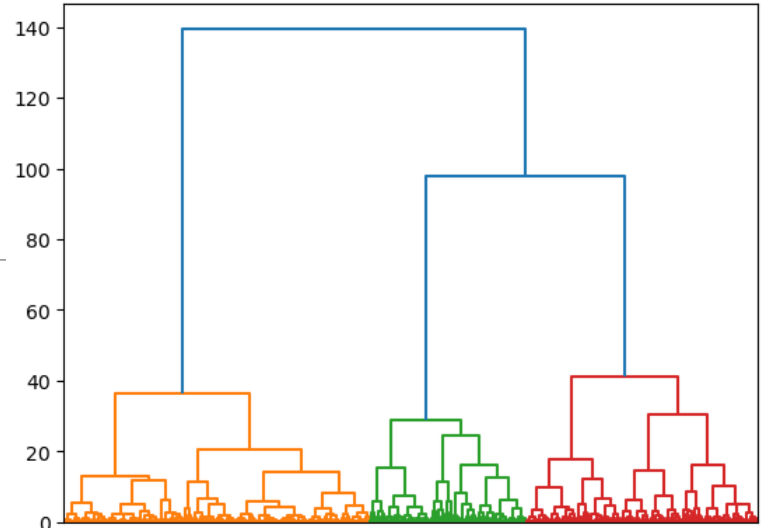
```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(scaled_data)

pca_df = pd.DataFrame(data=X_pca, columns=['PC1', 'PC2'])
```

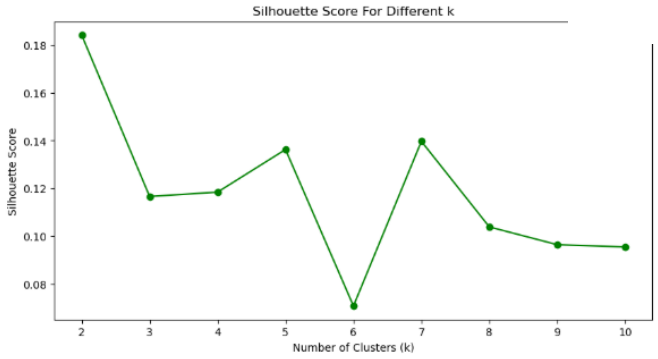
```
for n_clusters in range(2, 11):
    agglom = AgglomerativeClustering(n_clusters=n_clusters)
    cluster_labels = agglom.fit_predict(scaled_data)
```

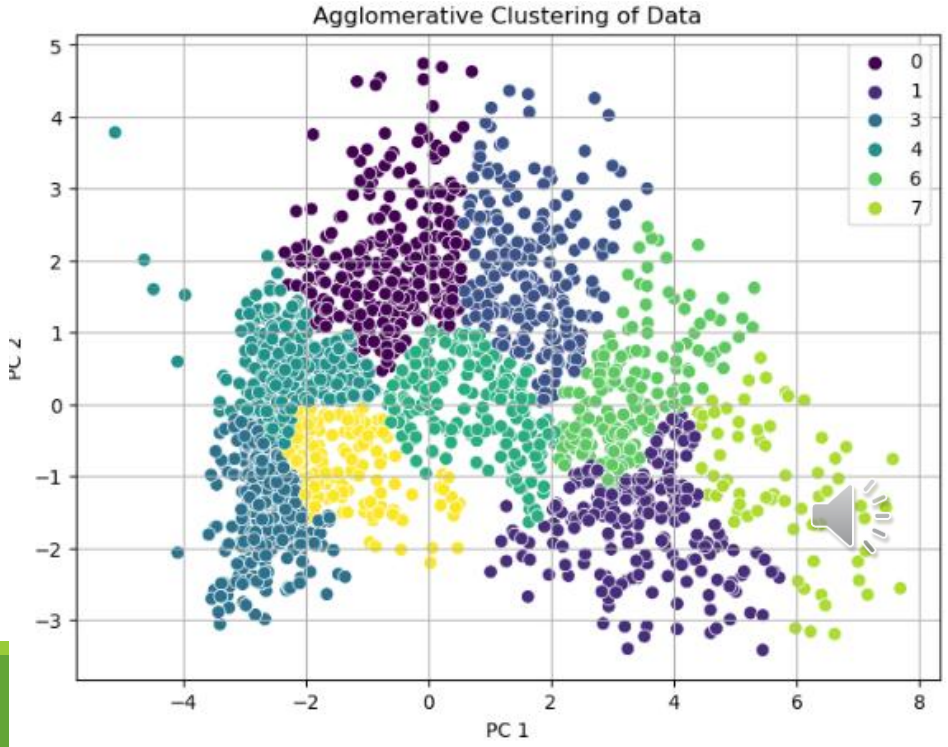
Results and Visualization

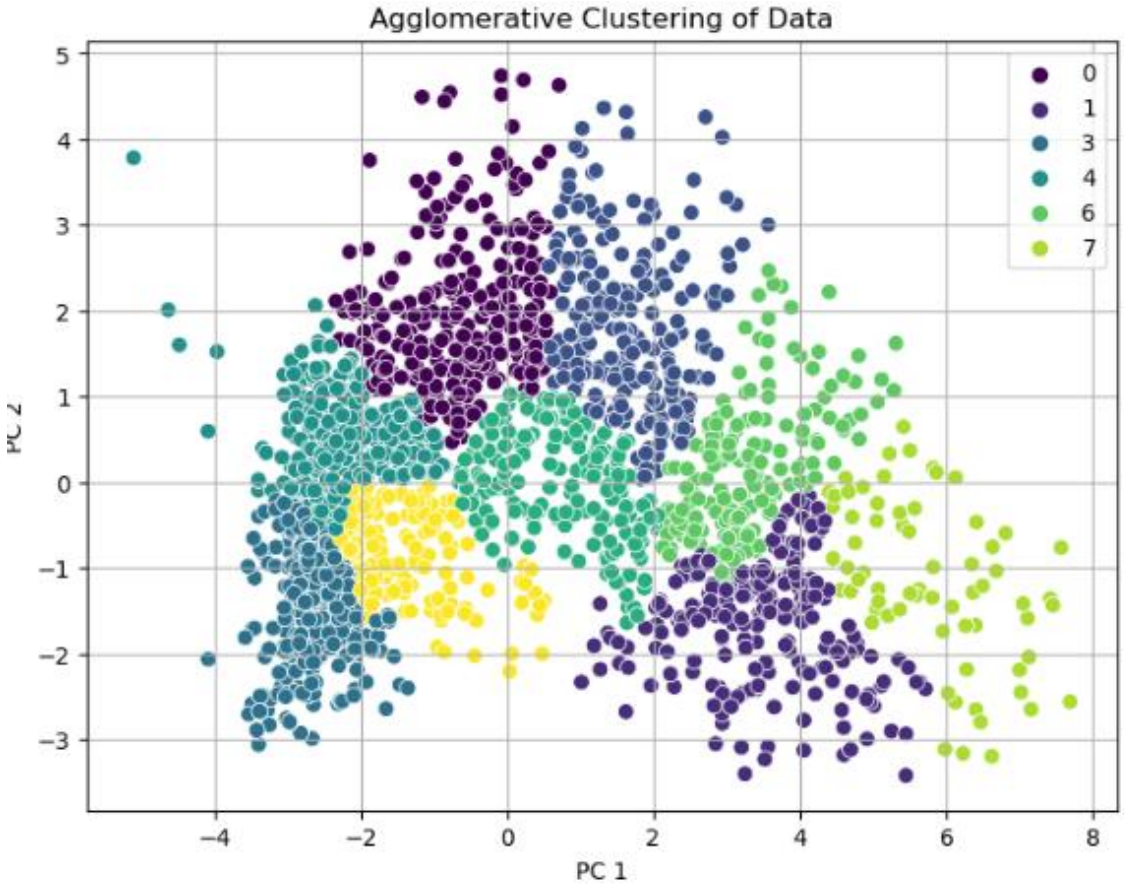












Conclusion

Based on the silhouette score of k-mean 2 cluster is the best but looking at the elbow method plot 7 clusters seem to be the mark and according to the silhouette score k = 7 has the second best score.

However, agglomerative clustering has best silhouette score for 9 cluster and the performance of this model is slightly better than K-mean. Though both models show weak silhouette scores.

Hierarchical clustering fit this data better than Kmeans based off the silhouette scores.

Thank You!