Customer Segmentation

GUDJIME ADINSI Ahouahounko

October 2023

What is customer Segementation ?

Customer segmentation is a critical process in marketing and business strategy that involves dividing a customer base into distinct groups or segments based on specific characteristics or behaviors. The primary goal of customer segmentation is to understand and target different customer groups more effectively, thereby improving marketing strategies, product development, and customer service.

- Customer segmentation provides insight into the landscape of the market ;
- Revealing customer characteristics that can be used to group customers into segments that have something in common ;
- This process is also known as clustering ;
- The techniques used to develop these models are called **clustering algorithms** ;

Customer Segmentation's objectives

- Segment the customer base into smaller groups : this will help in tailoring services and products offered to each group ;
- Generate a new index to be used in other models as a predictive variable : For instance a segment number or label to be used in other models or as predictive variable for example when we use the variable segment number the value equivalent to the student segment will always have young age and low income.

Data to conduct Customer Segmentation ?

- Demographic information such as age gender marital, status, income, ...
- Transactional information such as the products purchased, the dollar volume purchased, number of items purchased, time of purchase
- Geographic information(Location, region, climate,...)
- And many more !

Set a Goal ?

Before go through a segmentation, It is important to set a goal, a reason why make the segementation.

- Identify the best group of customers to sell ;
- Create a successful marketing campaign ;

— Optimize the sales-channel mix.

1 Why Machine learning ?

Machine learning models can process customer data and discover patterns difficult to spot through intuition and manual examination of data.

1.1 Clustering Algorithms

A clustering machine learning algorithm is an unsupervised machine learning algorithm. It's used for discovering natural groupings or patterns in the dataset. It's worth noting that clustering algorithms just interpret the input data and find natural clusters in it.

To perform it, one can use clustering algoritms like :
— K-means
— Affinity Propagation Clustering
— Hierarchical Clustering
— Density-Based Spatial Clustering(DBSCAN)
— Expectation-Maximization (EM) Clustering, ...

2 Modelisation(Let's jump !)

To go through this project, we will go step by step like this :
— Create business case ;
— Prepare the case ;
— Data analysis and exploration ;
— Clustering Analysis ;
— Choosing optimal hyperparameters ;
— Visualization and interpretation.

2.1 Dataset

The data used is from survey of Consumer Finances which is conducted by the Federal Reserve Board. The data source is here
The survey data we're using includes responses from 10,000+ individuals in 2007 (precrisis) and 2009 (postcrisis). There are over 500 features. Since the data has many variables, we will first reduce the number of variables and select the most intuitive features directly linked to an investor's ability and willingness to take risk.

For our purpose, there are 12 attributes for each of the individuals. These attributes can be categorized as **demographic**, **financial** and **behavioral** attributes. Here are the used variables :

AGE : There are 6 age categories, where 1 represents age less than 35 and 6 represents age more than 75.

EDUC : There are 4 education categories, where 1 represents no high school and 4 represents college degree.

MARRIED : It represents marital status. There are two categories where 1 represents married and 2 represents unmarried.

KIDS : It represents number of kids.

LIFECL : This is a lifecycle variable, used to approximate a person's ability to take on risk.There are six categories in increasing level of ability to take risk. A value of 1 represents "age under 55, not married, and no kids," and a value of 6 represents "age over 55 and not working."

OCCAT : It represents occupation category. 1 represents managerial category and 4 represents unemployed.

RISK : It represents the willingness to take risk on a scale of 1 to 4, where 1 represents highest level of willingness to take risk.

HHOUSES : This is a flag indicating whether the individual is a homeowner. A value of 1 (0) implies the individual does (does not) own a home.

SPENDMOR : This represents higher spending preference if assets appreciated on a scale of 1 to 5

NWCAT : It represents net worth category. There are 5 categories, where 1 net worth less than 25 percentile and 5 represents net worth more than 90th percentile.

INCCL : It represents income category. There are 5 categories, where 1 income less than 10,000 and 5 represents net worth more than 100,000.

2.2 Problem definition

The goal of this case study is to build a machine learning model to cluster individuals/investors based on the parameters related to the ability and willingness to take risk. We will focus on using common **demographic** and **financial** characteristics to accomplish this.

2.3 Getting Started

2.3.1 Loading the python packages

```
# Load libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pandas import read_csv, set_option
from pandas.plotting import scatter_matrix
import seaborn as sns
from sklearn.preprocessing import StandardScaler
import datetime

#Import Model Packages
from sklearn.cluster import KMeans, AgglomerativeClustering,
    AffinityPropagation
from sklearn.metrics import adjusted_mutual_info_score, silhouette_score
from sklearn import cluster, covariance, manifold


```

```
#Other Helper Packages and functions
import matplotlib.ticker as ticker
from itertools import cycle

#The warnings
import warnings
warnings.filterwarnings('ignore')

set_option('display.width', 100)
```

2.3.2 Loading the Data

```
    dataset = pd.read_excel('ProcessedData.xlsx')
```

2.4 Exploratory Data Analysis

```
dataset.shape
dataset.head(5)
```

Output :

	ID	AGE	EDUC	MARRIED	KIDS	LIFECL	OCCAT	RISK	HHOUSES	WSAVED	SPENDMOR	NWCAT	INCCL
0	1	3	2	1	0	2	1	3	1	1	5	3	4
1	2	4	4	1	2	5	2	3	0	2	5	5	5
2	3	3	1	1	2	3	2	2	1	2	4	4	4
3	4	3	1	1	2	3	2	2	1	2	4	3	4
4	5	4	3	1	1	5	1	2	1	3	3	5	5

```
# describe data
dataset.dtypes
dataset.describe()
dataset.drop(['ID'], axis = 1, inplace = True)
```

2.5 Data Visualization

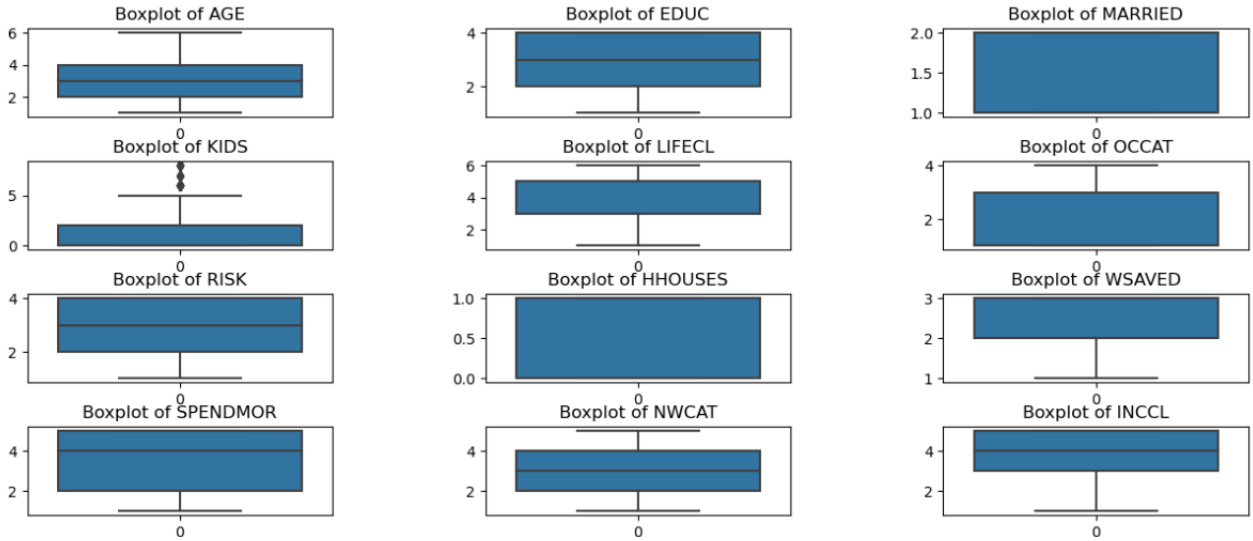
2.5.1 Variables distributions

```

plt.figure(1, figsize=(15,6))
n= 0
for x in ['AGE', 'EDUC', 'MARRIED', 'KIDS', 'LIFECL', 'OCCAT', 'RISK', '
    HHOUSES', 'WSAVED','SPENDMOR', 'NWCAT', 'INCCL']:
  n +=1
  plt.subplot(4, 3, n )
```

```
  plt.subplots_adjust(hspace = 0.5, wspace = 0.5)
  sns.boxplot(dataset[x])
  plt.title('Boxplot of {}'.format(x))
plt.show()
```

Output :



We can easily notice that the variable *KIDS* have some outliers. Let's remove those outliers from the dataset because **K-means** is sensitive to them.

```

Q1 = dataset['KIDS'].quantile(0.25)
Q3 = dataset['KIDS'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

dataset = dataset[(dataset['KIDS'] >= lower_bound) & (dataset['KIDS'] <=
    upper_bound)]
```

Then , from **3866 rows**, we end up to **3843** for our new datasets.

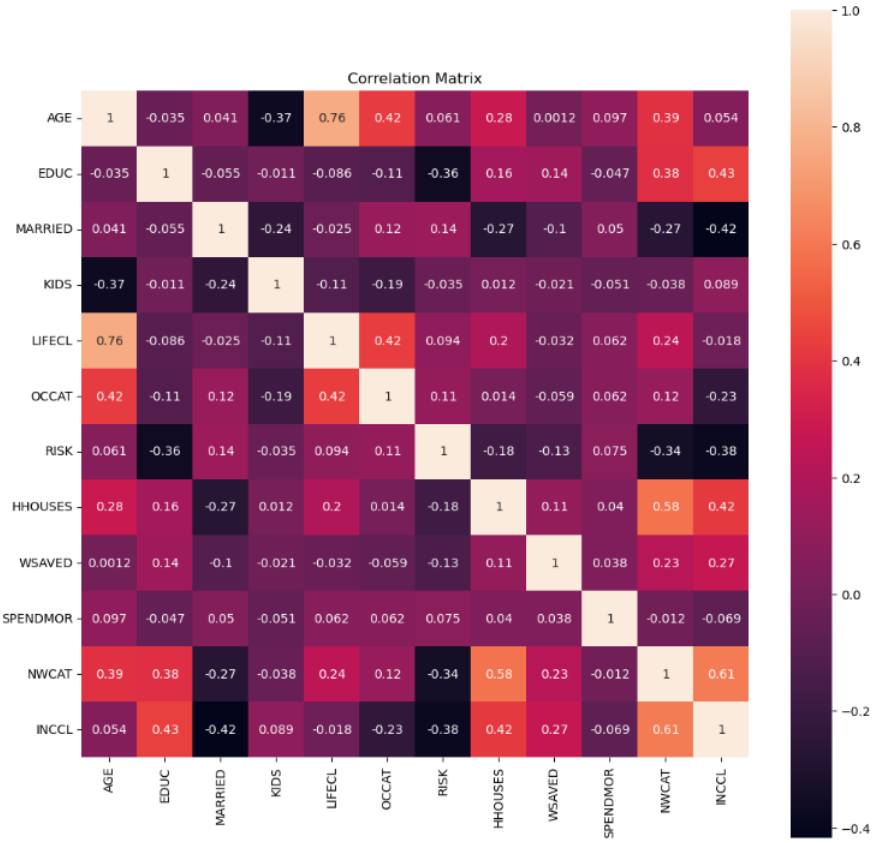
2.5.2 Correlation

Let us look at the correlation. We will take a detailed look into the visualization post clustering.

```

correlation = dataset.corr()
plt.figure(figsize=(12,12))
plt.title('Correlation Matrix')
sns.heatmap(correlation, vmax=1, square=True,annot=True,cmap='rocket')
```

Output :



As it can be seen by the picture above there is a significant positive correlation between some attributes.

2.6 Data Preparation

2.6.1 Data Cleaning

Let us check for the NAs in the rows, either drop them or fill them with the mean of the column.

```
dataset.isna().sum()
```

Fortunately, there isn't any missing data and the data is already in the categorical format no further data cleaning was performed.

2.7 Evaluate Algorithms and Models

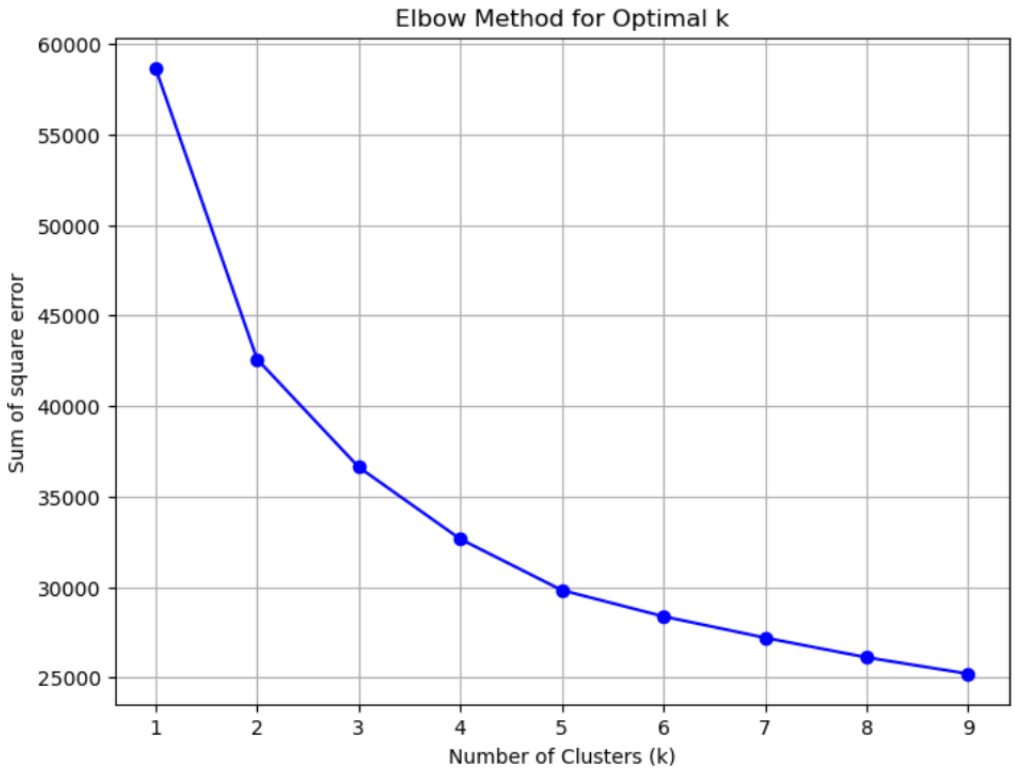
In this step, we will look at the following models and perform further analysis and visualization. For that reason, we will stick ourself to **K-means** and **Affinity Propagation**.

2.7.1 K-Means Clustering

Here we look at the following metrices to get the optimum number of clusters : **Sum of square error**s (SSE) within clusters and **Silhouette score**.

```
1
2  inertia = []
3
4  k_range = range(1, 10)
5
6  for k in k_range:
7      kmeans = KMeans(n_clusters=k, random_state=42)
8      kmeans.fit(dataset)
9      inertia.append(kmeans.inertia_)
10
11 plt.figure(figsize=(8, 6))
12 plt.plot(k_range, inertia, marker='o', linestyle='-', color='b')
13 plt.xlabel('Number of Clusters (k)')
14 plt.ylabel('Sum of square error')
15 plt.title('Elbow Method for Optimal k')
16 plt.grid(True)
17 plt.show()
```

Output :



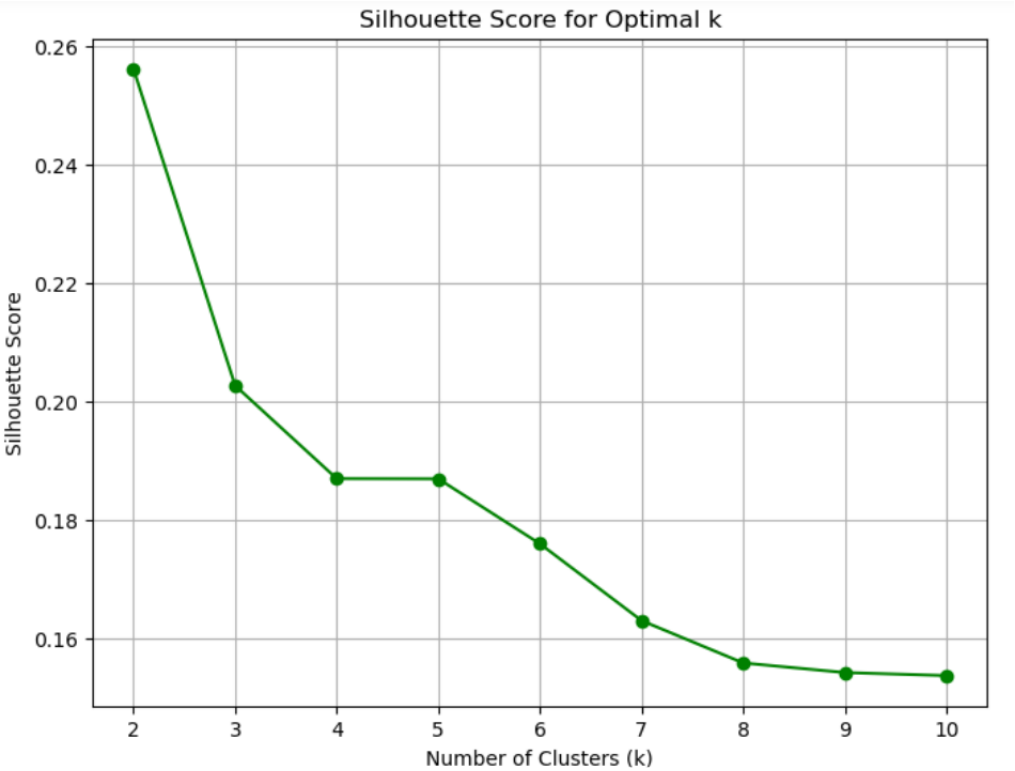
2.7.2 Silhouette score

```
1  silhouette_scores = []
2
3  k_range = range(2, 11)
4
5  for k in k_range:
```

```
    kmeans = KMeans(n_clusters=k, random_state=42)
    cluster_labels = kmeans.fit_predict(dataset)
    silhouette_avg = silhouette_score(dataset, cluster_labels)
    silhouette_scores.append(silhouette_avg)

plt.figure(figsize=(8, 6))
plt.plot(k_range, silhouette_scores, marker='o', linestyle='-', color='g')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score for Optimal k')
plt.grid(True)
plt.show()
```

Output :



Looking at both the charts above, the optimum number of clusters seem to be around 7. We can see that as the number of clusters increase pass 6, the sum of square of errors within clusters plateaus off. From the second graph, we can see that there are various parts of the graph where a kink can be seen. Since there is not much a difference in SSE after 7 clusters, we would use 7 clusters in the k-means model below.

2.7.3 Clustering and Visualisation

```
nclust=7

k_means = cluster.KMeans(n_clusters=nclust)
k_means.fit(dataset)

```

```
target_labels = k_means.predict(dataset)
```

2.8 Affinity Propagation

```
ap = AffinityPropagation(damping = 0.5, max_iter = 250, affinity = '
    euclidean')
ap.fit(dataset)
clust_labels2 = ap.predict(dataset)

cluster_centers_indices = ap.cluster_centers_indices_
labels = ap.labels_
n_clusters_ = len(cluster_centers_indices)
print('Estimated number of clusters: %d' % n_clusters_)
```

Output :
Estimated number of clusters : 159

2.9 Cluster Evaluation

We evaluate the clusters using Silhouette Coefficient (*sklearn.metrics.silhouette_score*). Higher Silhouette Coefficient score means a model with better defined clusters.

```
from sklearn import metrics
print("kmeans", metrics.silhouette_score(dataset, k_means.labels_, metric=
    'euclidean'))
print("Affinity Propagation", metrics.silhouette_score(dataset, ap.labels_
    , metric='euclidean'))
```

kmeans : 0.16258401400119774
Affinity : Propagation 0.09619746241020978

2.10 Cluster Intuition

In the next step, we will check each cluster and understand the intuition behind the clusters.

```
cluster_output= pd.concat([pd.DataFrame(dataset), pd.DataFrame(k_means.
    labels_, columns = ['cluster'])],axis = 1)
output=cluster_output.groupby('cluster').mean()
output
```

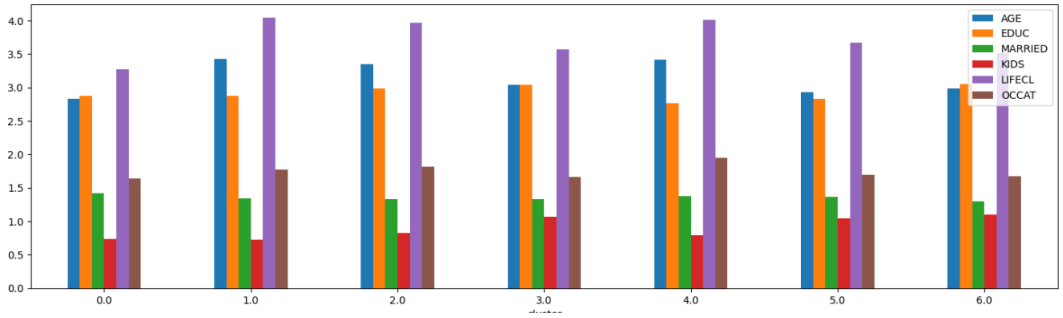
Output :

cluster	AGE	EDUC	MARRIED	KIDS	LIFECL	OCCAT	RISK	HHOUSES	WSAVED	SPENDMOR	NWCAT	INCCL
0.0	2.827303	2.876645	1.414474	0.735197	3.269737	1.634868	3.057566	0.643092	2.373355	3.569079	2.662829	3.500000
1.0	3.426997	2.878788	1.338843	0.730028	4.044077	1.776860	2.969697	0.749311	2.438017	3.363636	3.165289	3.716253
2.0	3.353846	2.982692	1.332692	0.821154	3.971154	1.811538	3.025000	0.767308	2.571154	3.698077	3.250000	3.819231
3.0	3.041308	3.044750	1.330465	1.065404	3.571429	1.659208	2.936317	0.777969	2.500861	3.784854	3.166954	3.838210
4.0	3.413969	2.768313	1.371380	0.795571	4.017036	1.945486	3.172061	0.705281	2.405451	3.667802	2.873935	3.505963
5.0	2.927298	2.831276	1.366255	1.042524	3.674897	1.698217	3.120713	0.659808	2.436214	3.517147	2.780521	3.547325
6.0	2.990762	3.053118	1.293303	1.096998	3.512702	1.674365	2.921478	0.771363	2.415704	3.224018	3.207852	3.935335

2.10.1 Demographics Features

```
output[['AGE','EDUC','MARRIED','KIDS','LIFECL','OCCAT']].plot.bar(rot=0,
    figsize=(18,5));
```

Output :

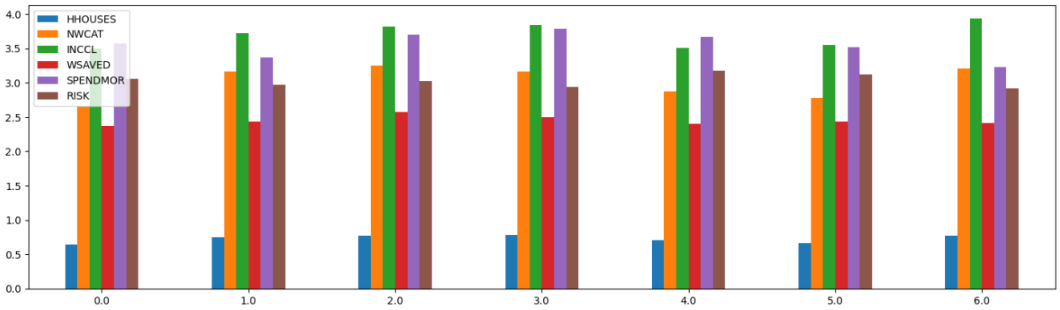


The plot here shows the average value of the attributes for each of the clusters. For example, comparing cluster 0 and cluster 1, cluster 0 has lower average age, yet higher average education. In terms of marriage and number of children, these two clusters are similar. So, the individuals in cluster 0 will on an average have higher risk tolerance as compared to the individuals in cluster 1, based on the demographic attributes.

2.10.2 Financial Features and Features related to willingness to take risk

```
output[['HHOUSES','NWCAT','INCCL','WSAVED','SPENDMOR','RISK']].plot.bar(
    rot=0, figsize=(18,5));
```

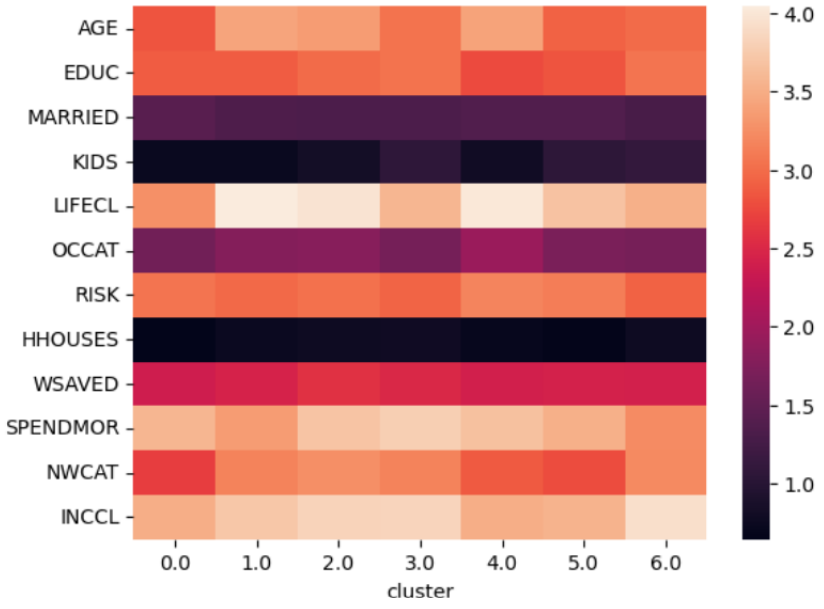
Output :



The plot here shows the average value of the attributes for each of the cluster on the financial and behavioral attributes. For example, comparing cluster 0 and cluster 1, cluster 0 has higher average house ownership, higher average net worth and income, and a lower willingness to take risk. In terms of saving vs. income comparison and willingness to save, the two clusters are comparable. Therefore, we can posit that the individuals in cluster 0 will, on average, have a higher ability, yet lower willingness, to take risk compared with cluster 1.

```
1 sns.heatmap(output.T)
```

Output :



Combining the information from the demographics, financial, and behavioral attributes for cluster 0 and cluster 1, the overall ability to take risk for individual cluster 0 is higher as compared to cluster 1. Performing similar analysis across all other clusters, we summarize the results in the table below. The risk tolerance column represents the subjective assessment of the risk tolerance of each of the clusters.

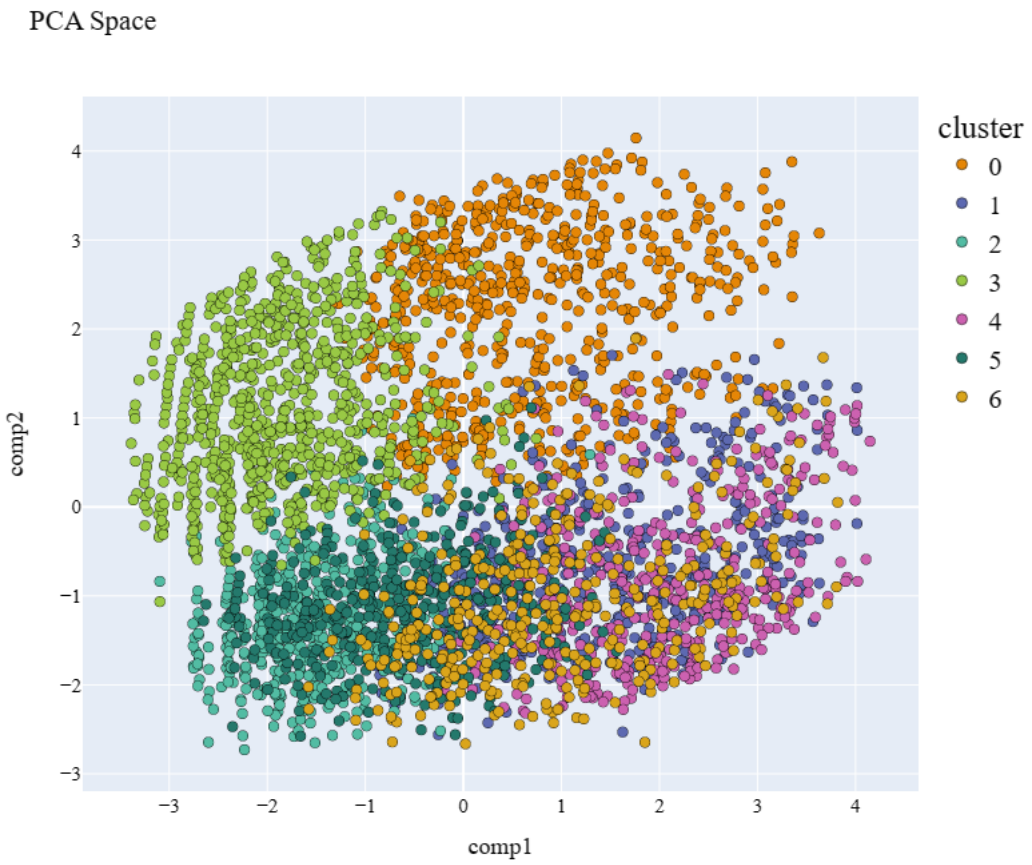
Cluster	Features	Risk Capacity
Cluster 0	Low Age, High Networth and Income, Less risky life category, willingness to spend more	High
Cluster 1	High Age, low net worth and Income, highly risky life category, Willing ness to take risk, low education	High
Cluster 2	High Age, high net worth and Income, highly risky life category, Willing ness to to take risk, own house	Medium
Cluster 3	Low age, very low income and net worth, high willingness to take risk, many kids	Low
Cluster 4	Medium age, very high income and net worth, high willingness to take risk, many kids, own house	High
Cluster 5	Low age, very low income and net worth, high willingness to take risk, no kids	Medium
Cluster 6	Low age, medium income and net worth, high willingness to take risk, many kids, own house	Low

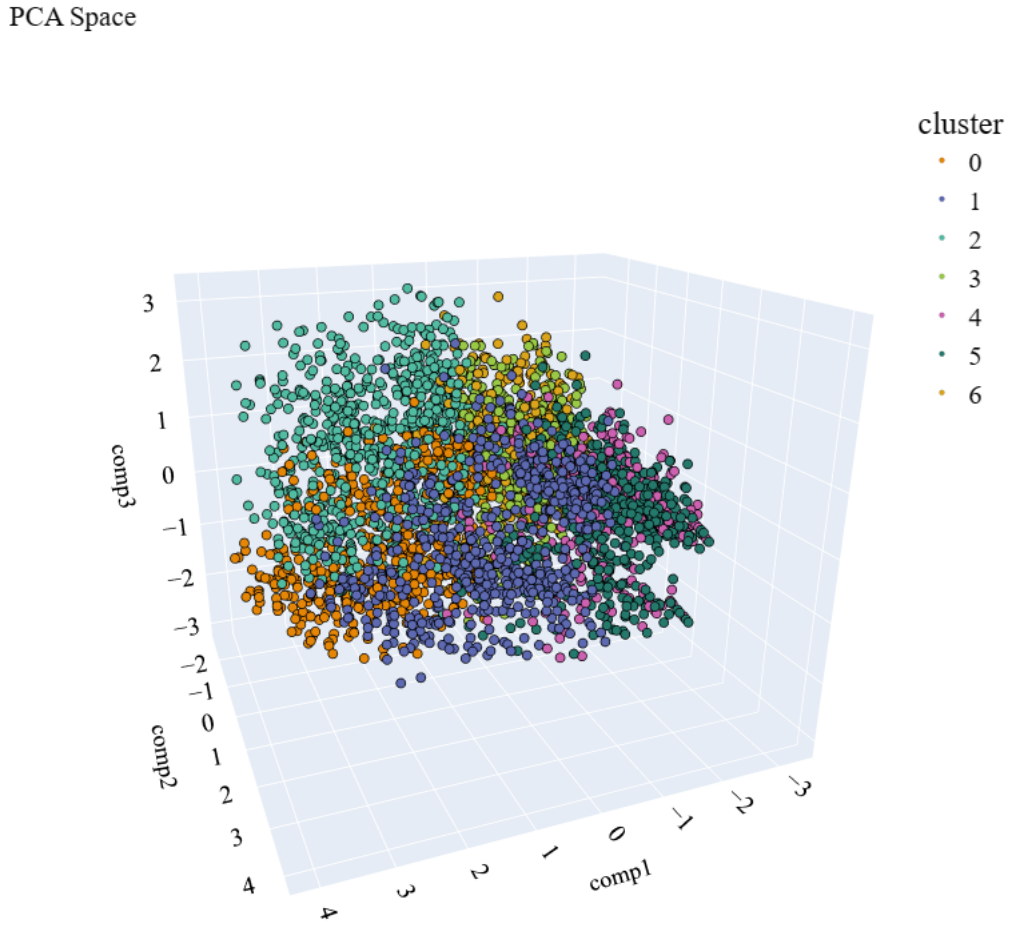
2.11 Visualization

2.11.1 PCA

For visualizations, we can use the method to reduce dimensionality, PCA. For them we are going to use the Prince library, focused on exploratory analysis and dimensionality reduction. If you prefer, you can use Sklearn's PCA, they are identical.

```
pca_2d_object, df_pca_2d = get_pca_2d(dataset, k_means.labels_)
plot_pca_2d(df_pca_2d, title = "PCA Space", opacity=1, width_line = 0.5)
```



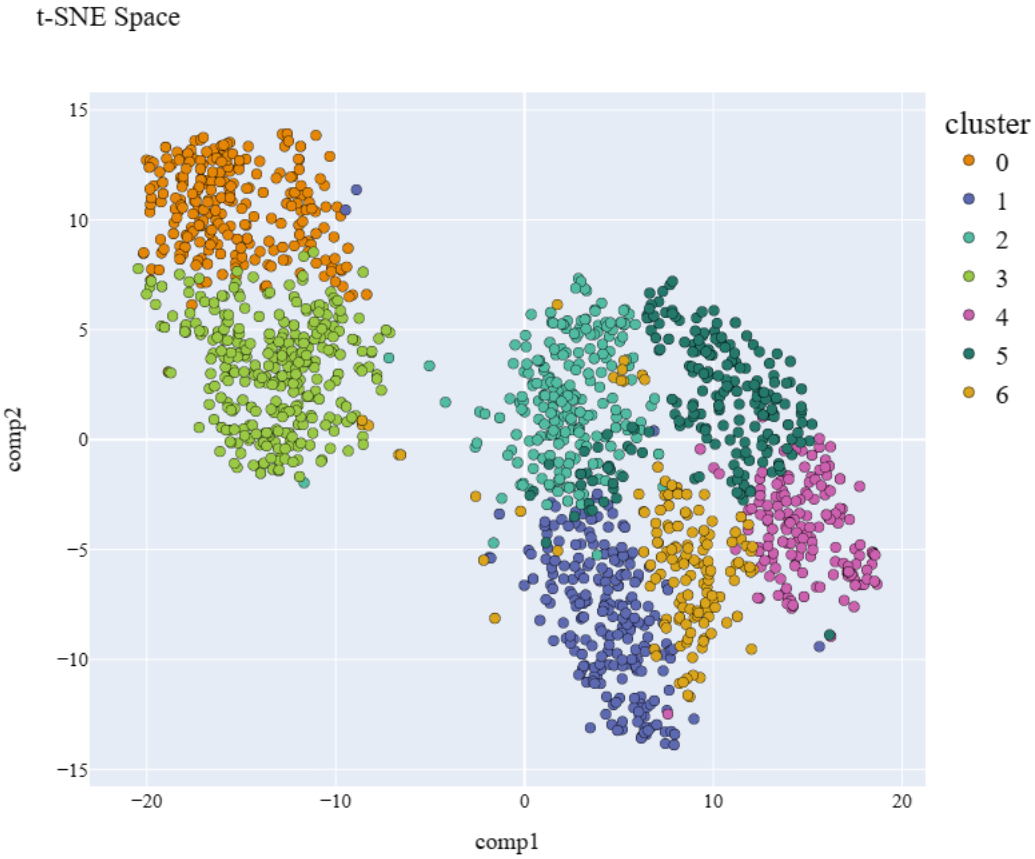


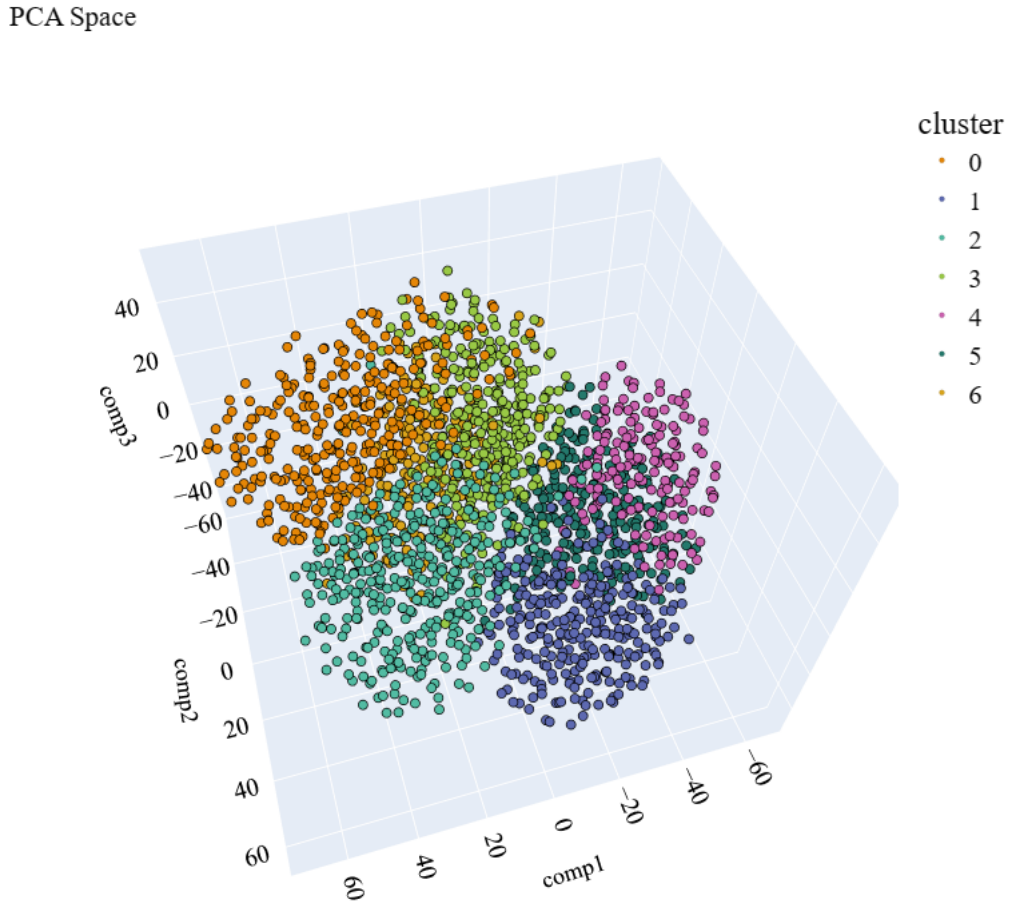
2.11.2 t-SNE

```
    df_tsne_2d = TSNE(
                  n_components=2,
                  learning_rate=500,
                  init='random',
                  perplexity=200,
                  n_iter = 5000).fit_transform(sampling_data)

df_tsne_2d = pd.DataFrame(df_tsne_2d, columns=["comp1", "comp2"])
df_tsne_2d["cluster"] = sampling_clusters

plot_pca_2d(df_tsne_2d, title = "PCA Space", opacity=1, width_line = 0.5)
```


t-SNE Space
cluster
0
1
2
3
4
5
6
comp2
comp1
15
10
5
0
−5
−10
−15
−20
−10
0
10
20



3 Conclusion

One of the key takeaways from this case study is the approach to understand the cluster intuition. We used visualization techniques to understand the expected behavior of a cluster member by qualitatively interpreting mean values of the variables in each cluster.

We demonstrate the efficiency of the clustering technique in discovering the natural intuitive groups of different investors based on their risk tolerance.

Given, the clustering algorithms can successfully group investors based on different factors, such as age, income, and risk tolerance, it can further used by portfolio managers to understand the investor's behavior and standardize the portfolio allocation and rebalancing across the clusters, making the investment management process faster and effective.