FINAL PROJECT:

UNSUPERVISED MACHINE LEARNING

Antonio Jimenez

IBM ON COURSERA. March, 2023

Table of Content

Introduction

Data segmentation is crucial for business to launch marketing strategies to target specific group of customers. This is the aim of the current work, to divide customers into a few number of clusters, so several marketing strategies could be applied targeting each one. The company asking for the study is a supermarket chain and the data comes from their customers through membership cards. This study is a first step in an expansion campaign that the company is planning.

Description of the Data

The data chosen was downloaded from Kaggle, its name is "Customer Clustering", uploaded by Dev Sharma, with 2000 observations. The attributes of are the following (Dev):

ID: Numerical. Integer shows a unique identification of a customer.

Sex: Categorical {0,1}. Biological sex (gender) of a customer. In this dataset there are only 2 different options: 0 = male, 1 = female.

Marital status: Categorical {0,1}. Marital status of a customer. 0 = single, 1 = non-single (divorced / separated / married / widowed)

Age: Numerical, Integer. The age of the customer in years, calculated as current year minus the year of birth of the customer at the time of creation of the dataset. 18 is the minimum value (the lowest age observed in the dataset). 76 is the maximum value (the highest age observed in the dataset).

Education: Categorical {0,1,2,3}. Level of education of the customer : 0 = other / unknown, 1 = high school, 2 = university, 3 = graduate school.

Income: Numerical, Real. Self-reported annual income in US dollars of the customer.

Occupation Categorical {0,1,2}. Category of occupation of the customer. 0 = unemployed / unskilled, 1 = skilled employee / official, 2 = management / self-employed / highly qualified employee / officer.

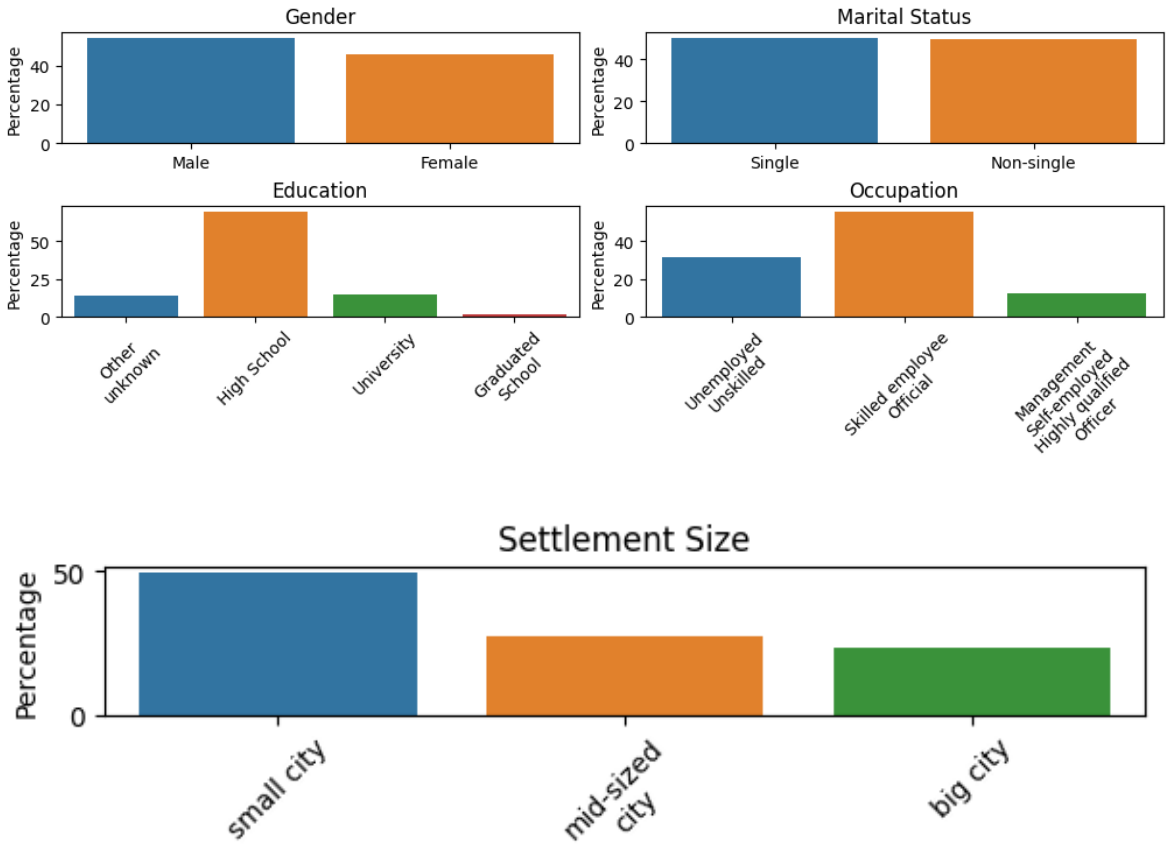
Settlement size: Categorical {0,1,2}. The size of the city that the customer lives in. 0 = small city, 1 = mid-sized city, 2 = big city.

The Goal of the analysis is to cluster customers into few groups.

Exploratory Data Analysis

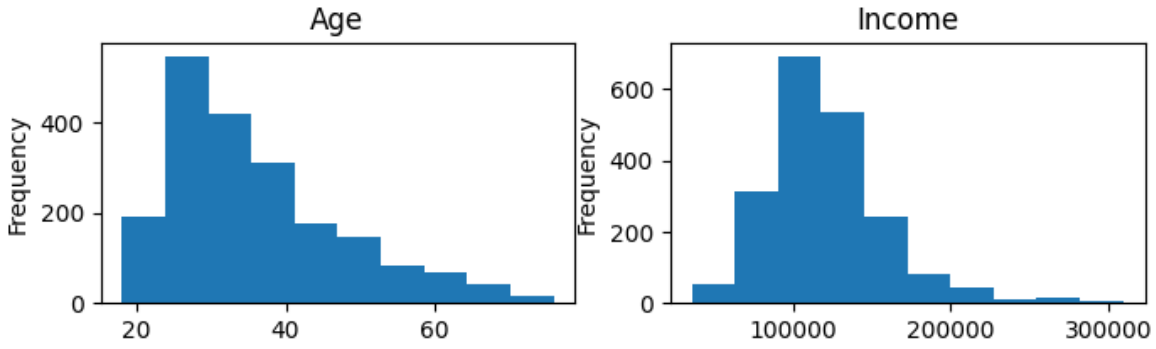
The first step was to load the csv file into a Pandas Dataframe. The next step was to understand the data types of each features and deal with missing values. Luckily, the there wasn't any missing value. Then, the unique values of categorical data were analyzed. These variables were already encoded with integer values, as explained before. For some of them, however, the ordinal order is not appropriated and they need to be encoded using one hot encoding.

The next step was to visualize the distribution of independent variables. First the categorical variables:



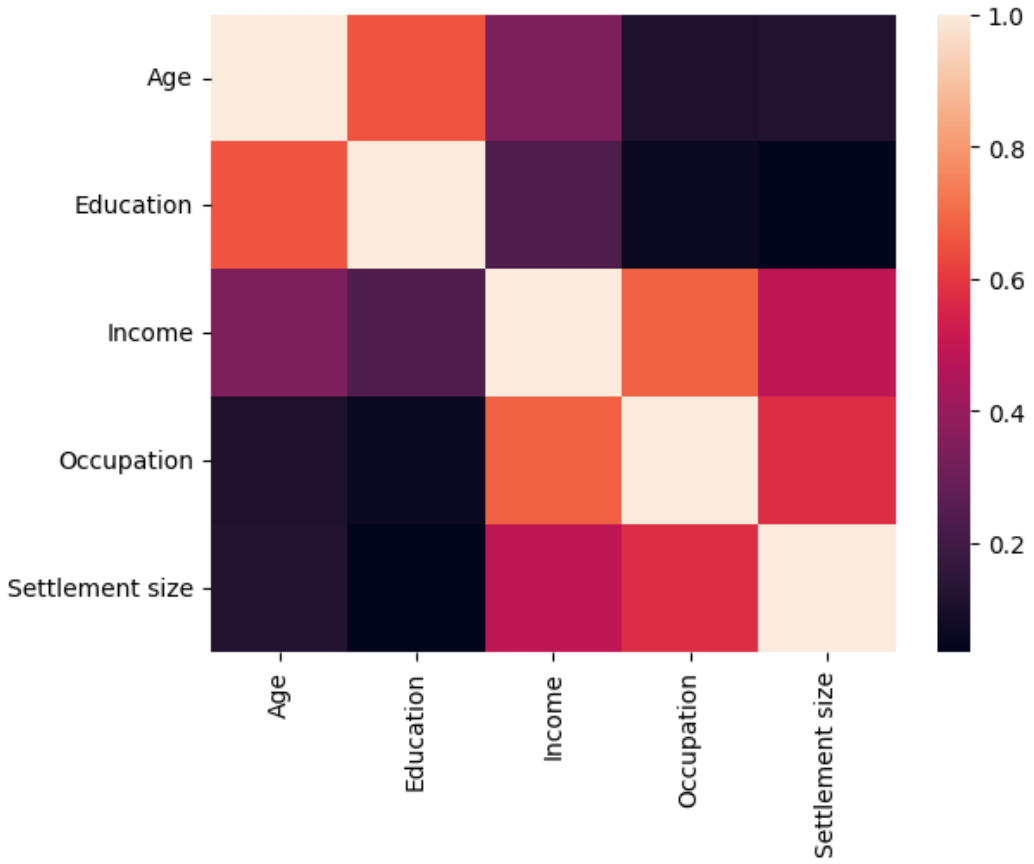
It can be seen that for gender and marital status, customers are evenly distributed between the categories. Also, most customers have high school education and are skilled employees. Also, almost half of them come from a small city, probably due to the supermarket locations.

The distributions of the numerical variables are the following:



It can be seen that most of the customers are between 25 and 40 years with an average income of 120 000 $.

A visualization of the Pearson correlations between the variables is depicted below (the variables has already being encoded with integers, see the previous section for details).



We can see that there are higher correlations between variables close to the diagonal. Income, occupation and age are highly correlated, as well as education and age.

Finally, a summary statistic is displayed below (categorical variables should be interpreted using the encoding given in the data set).

	Age	Education	Income	Occupation	Settlement size
count	2000.000000	2000.00000	2000.000000	2000.000000	2000.000000
mean	35.909000	1.03800	120954.419000	0.810500	0.739000
std	11.719402	0.59978	38108.824679	0.638587	0.812533
min	18.000000	0.00000	35832.000000	0.000000	0.000000
25%	27.000000	1.00000	97663.250000	0.000000	0.000000
50%	33.000000	1.00000	115548.500000	1.000000	1.000000
75%	42.000000	1.00000	138072.250000	1.000000	1.000000
max	76.000000	3.00000	309364.000000	2.000000	2.000000

The 'mean' of categorical variables gives a sense of where lies the category for most customers.

Feature engineering steps were taken. Categorical columns were encoded using one hot encoding, and numerical variables were scaled using StandardScaler. Also, ordinal columns were given codes starting from 0 up to the number of categories. Below is the classification for each variable.

Numerical columns: Age, Income.

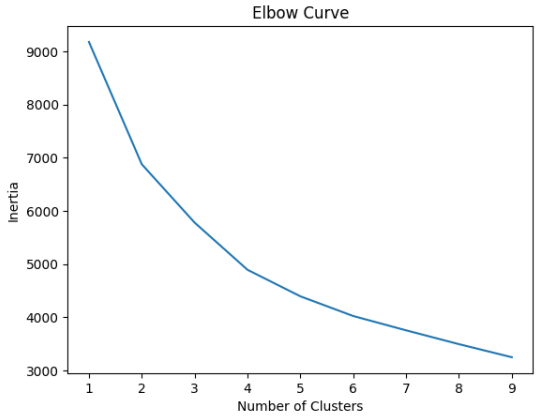
Ordinal columns: Education, Settlement size

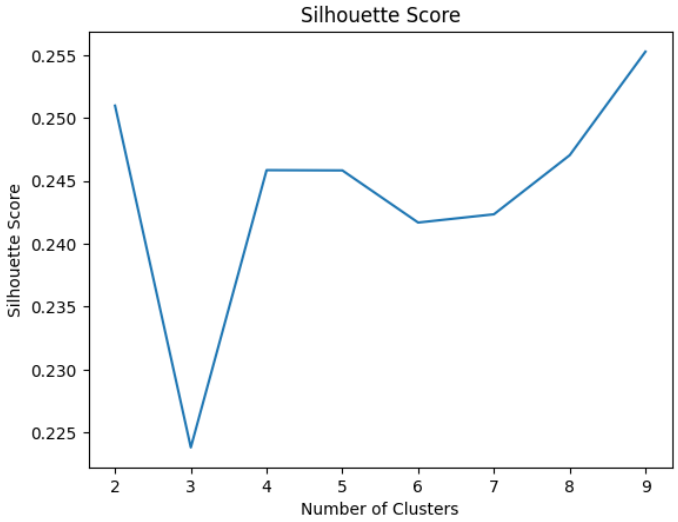
Categorical columns: Sex, Marital status, Occupation.

Modeling

Three different clustering algorithms were implemented: K-means, HDBSCAN, and Gaussian Mixture Model. For each one, steps to determine the best parameters were taken, and a comparison using the silhouette score was made to choose the optimal one. Scikit learn was used for K-means and GMM, but for HDBSCAN, the hdbscan library was used.

For K-means, the first step was to determine the right number of clusters. Two approaches were taken, the elbow method and a comparison of silhouette scores over different K's. Due to the non-deterministic nature of the algorithm, ten runs for each K were performed and then the result averaged for each approach. The results are the following:



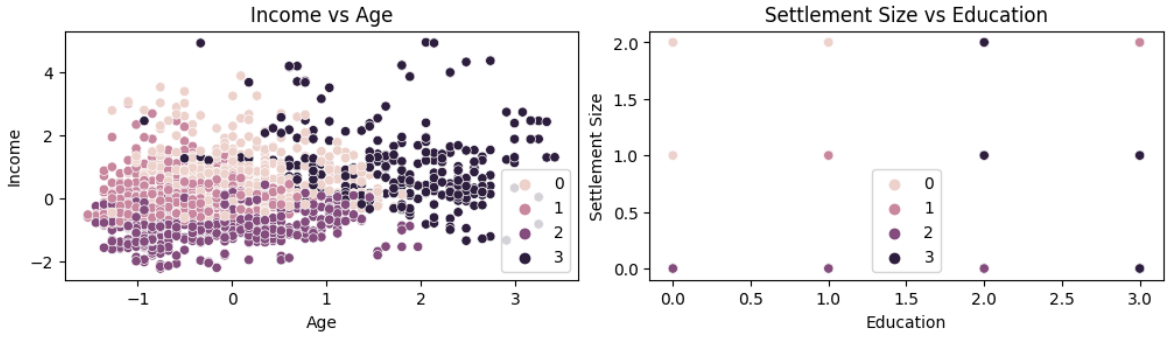


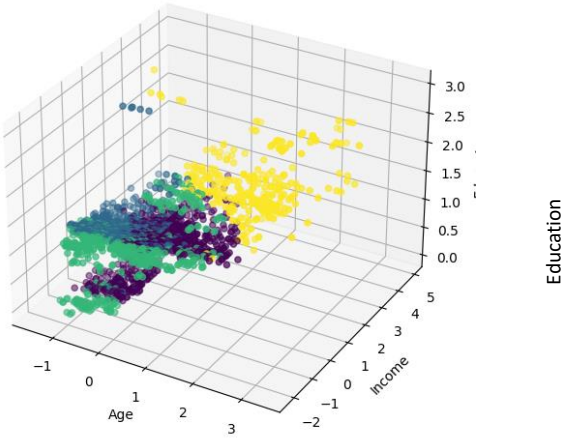
From the elbow curve, it is not so clear were the “elbow” lies. But, from the silhouette score, K equals four seem to be the optimal. So, K equals four was chosen.

Then, the algorithm was trained using this K, and its average silhouette score was also calculated.

```
KMeans
KMeans(n_clusters=4)
```

Below, there are visualizations of how the algorithm labeled data points.





The second algorithm evaluated was HDBSCAN, this was made using the library “hdbscan” (McInnes). The first step was to find the optimal values for *min_cluster_size* and *min_samples.* These are the most relevant hyperparameters for this algorithm (as a note, there is no need to find the number of clusters since it is automatically provided for the algorithm). To determine their values, the Silhouette score was calculated over many different combinations of them, and the most relevant ones were picked for consideration.

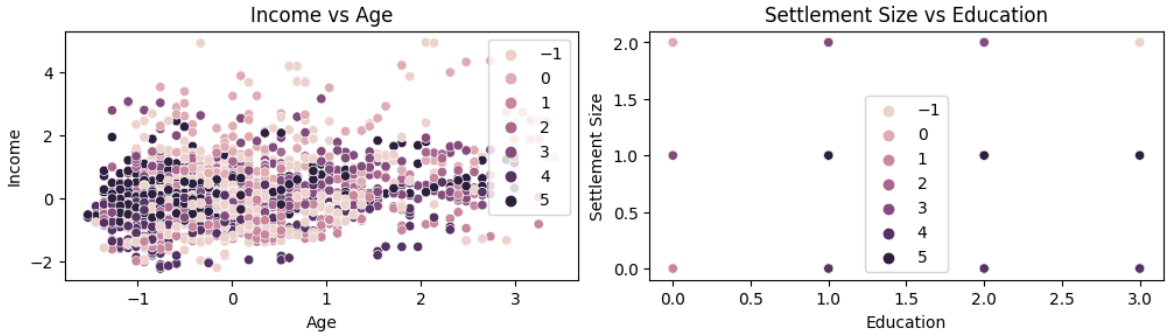
	min_cluster_size	min_samples	Silhouette score	Number of clusters
0	20	2	0.299667	28
6	140	2	0.187317	5

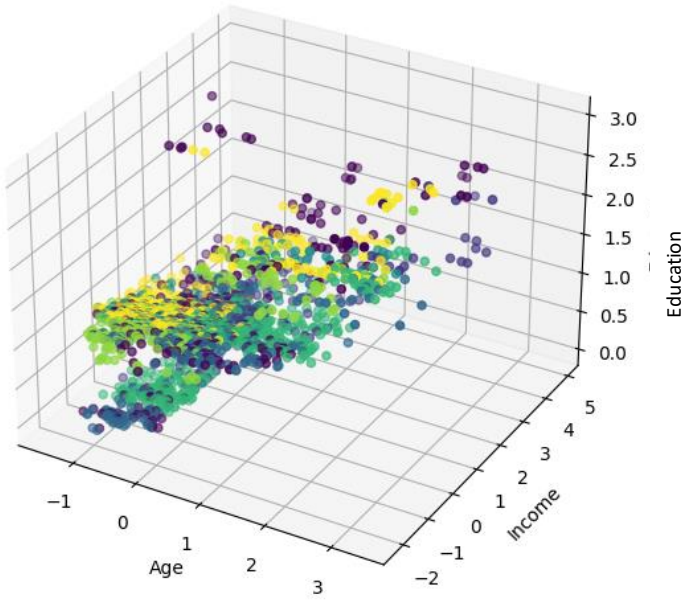
It could be seen that with *min_clustersize 20* and *min_samples = 2,* with 28 clusters, the Silhouette score is the highest. However, one of the objectives is to have a “few” number of clusters. For that reason, only the one with the highest score with “small” K is chosen, this is the one with *min_clustersize 140* and *min_samples = 2.*

The model was trained using these parameters:

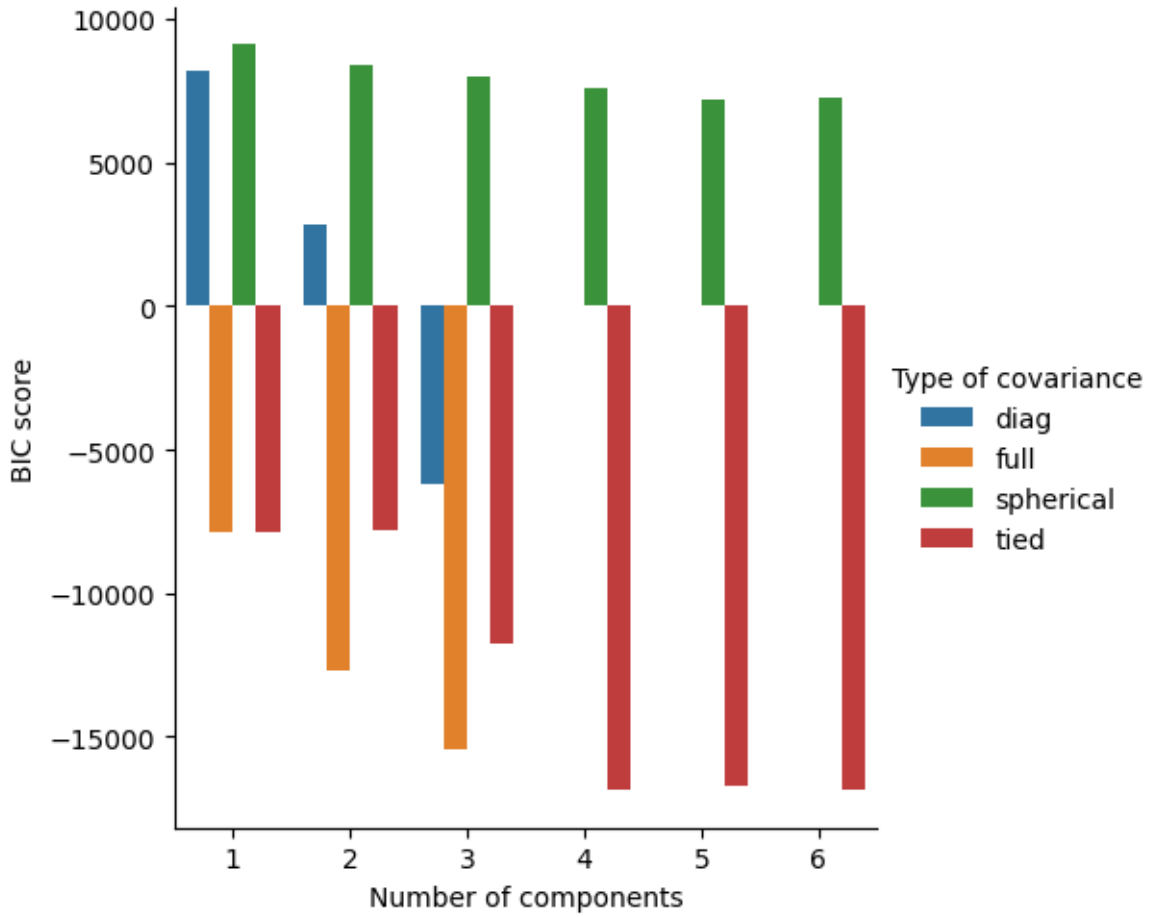
```
HDBSCAN
HDBSCAN(min_cluster_size=140, min_samples=2)
```

Below, there are visualizations of how the algorithm clustered data points.





Finally, a Gaussian Mixture Model was considered. For this, two types of parameters needed to be considered: number of clusters and the covariance Type. Two approaches were taken, grid search to calculate the best BIC score, and the calculation of the silhouette score over various combination of parameters. The results for the Bic score was:



As a note, the lower the value is, the better. It could be seen that a number of clusters equals four and with covariance type of tied was the optimal.

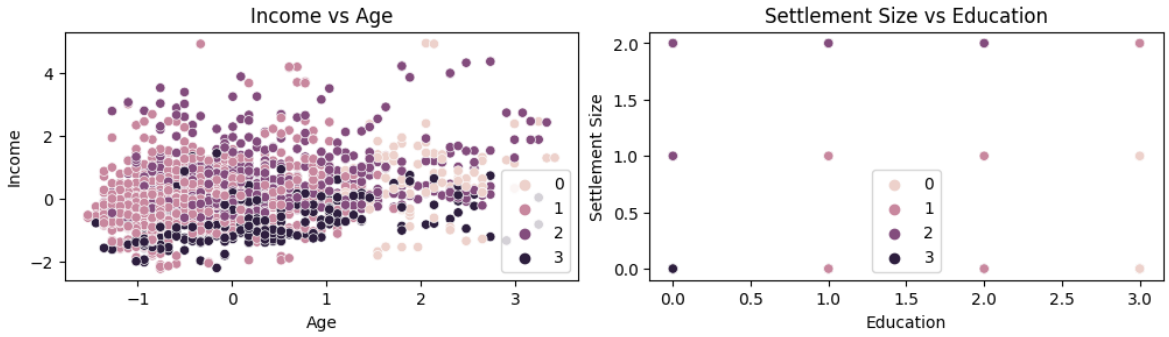
The silhouette score for this model was also calculated:

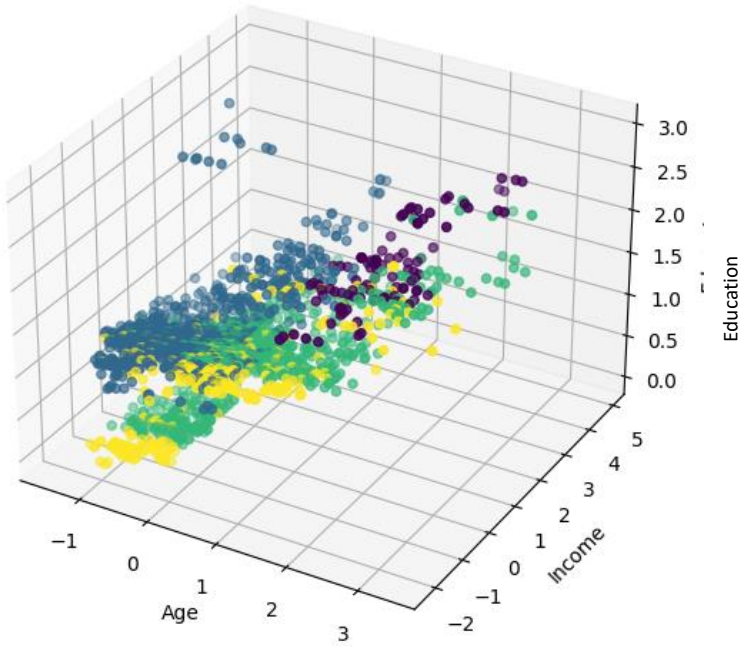
	index	covariance_type	n_components	Silhouette score	Number of clusters
0	0	spherical	2	0.240794	1
1	3	full	2	0.233361	1
8	9	tied	4	0.202006	3

Dropping the ones with only two clusters and also the ones with covariance types for K equals 4, 5 and 6 that yielded very high values for BIC score (those were suppressed from the plot above), we are left with K equals 4 and tied covariance. This was the chosen one.

```
GaussianMixture
GaussianMixture(covariance_type='tied', n_components=4)
```

The following are the visualization of the resulting model:

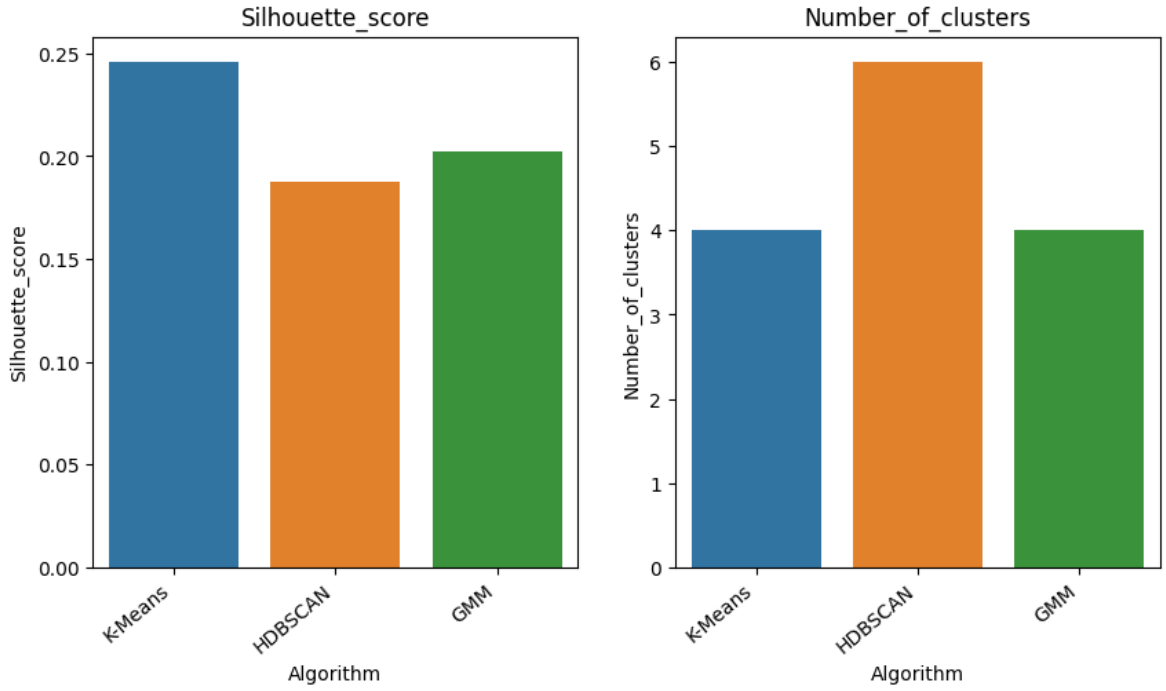




Final model

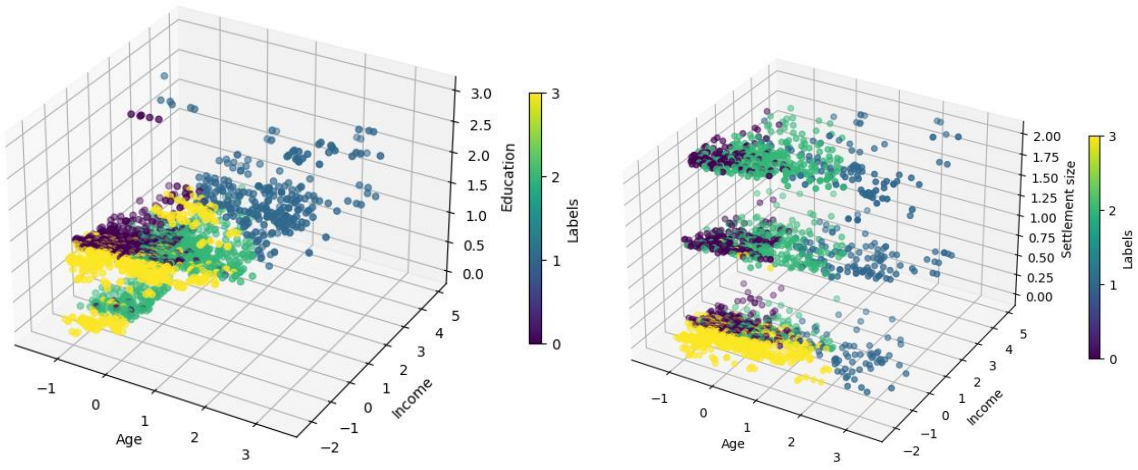
To choose the optimal model, a comparison of the silhouette scores for the different models was made. The results are summarized in the following table and a bar plot:

	Algorithm	Silhouette_score	Number_of_clusters
0	K-Means	0.245847	4
1	HDBSCAN	0.187317	6
2	GMM	0.202006	4



The best performing model was K-means with 4 clusters.

Visualization of the outputs of this model are shown below (the data is normalized).



Key findings and insights

From the exploratory data analysis, it can be seen that the majority of the data comes from people with high school education. Also, the majority of them are skilled employees and live in small cities. They are concentrated between 20 a 40 years old and their incomes ranges from 80 thousand to 150 thousand dollars per year. In addition, positive correlation is observed between age and income.

After visual inspection of plots and the data, the four clusters defined by the final model seem to be well separated following a clear pattern.

Visually, Settlement size seem to be key in defining the cluster number 3. For small settlement size the samples are categorized as label 3. With regard to label 0, income above the mean define the this one. Finally, Income and Age define label 1, the higher the income and age, the most likely a sample is going to be categorized as label 1.

Suggestions and next steps

As a suggestion, there is the possibility of creating new features or transforming existing ones to improve the performance of the clustering model. Also, it is recommendable to test other algorithms that weren't considered, such as Agglomerative Clustering or Mean Shift, to see if they perform better on the data. In addition, a further evaluation of hyperparameters to find better models is suggested.

Another way to increase model performance is to investigate and develop new evaluation metrics that may be more appropriate for the application. Also, finding new visualization techniques can help to better understand and interpret the clusters. Finally, incorporating external information such as domain knowledge or additional data could improve clustering results.

References

Dev Sharma. 2021. Customer Clustering. Kaggle. www.kaggle.com/datasets/dev0914sharma/customer-clustering.

L. McInnes, J. Healy, S. Astels, hdbscan: Hierarchical density based clustering In: Journal of Open Source Software, The Open Journal, volume 2, number 11. 2017