clustering algorithms — assignment 2

Styliani Mertzani, Konstantinos Konstantinidis

December 18, 2024

"Mathematics is the art of giving the same name to different things"
— Henri Poincaré

Contents

1 Feeling the data **3**

2 Feature selection **7**

3 Selection of the clustering algorithm **8**
3.1 k-means clustering algorithm 9
3.2 k-medians clustering algorithm 9
3.3 k-medoids clustering algorithm 9
3.4 possibilistic clustering algorithm 9
3.5 Selecting of the clustering algorithm 10

4 Execution of the clustering algorithm **12**

5 Characterization of the clusters **14**

6 MATLAB code **17**

1 Feeling the data

The dataset consists of 167 countries with 9 different features for each country. The features included are:

1. Child Mortality (death of children under 5 years of age, per 1000 live births)
2. Exports (exports of goods and services per capita, given as %age of the GDP per capita)
3. Health (total health spending per capita, given as %age of GDP per capita)
4. Imports (imports of goods and services per capita, given as a %age of teh GDP per capita)
5. Income (net income per person)
6. Inflation (the measurement of the annual growth rate of the Total GDP)
7. Life Expectancy (the average number of years a new born could live if the current mortality patterns are to remain the same)
8. Total Fertility (the number of children that would be born to each woman if the current age-fertility rates remain the same)
9. GDPP (calculated as the total GDP divided by the total population)

For each feature, its range of values (minimum and maximum value), mean and standard deviation are given in Table 1.

Feature	**Minimum**	**Maximum**	**Mean**	**Standard Deviation**
Child Mortality	2.6	208	38.2701	40.3289
Exports	0.109	200	41.1090	27.412
Health	1.81	17.9	6.8157	2.7468
Imports	0.0659	174	46.8902	24.2096
Inflation	-4.21	104	7.7818	10.5707
Life Expectancy	32.1	82.8	70.5557	8.8932
Total Fertility	1.15	7.49	2.948	1.5138
GDPP	231	105000	1.2964e+04	1.8329e+04

Table 1: General statistics of the dataset

One might argue that some of the features are discrete-valued, but for the sake of simplicity, for the rest of this exercise, all features will be treated as continuous values.

The linear correlation of each pair of features was calculated and plotted and a scatter-plot matrix. The results are shown in Figure 1. Then, we performed the standard score normalization and the minmax feature scaling normalization on each feature. Upon performing these tasks, we computed the linear dependence of each feature with all the others, for each case. The results are shown in the figures below (Figure 2 and Figure 3). Proper normalization, of any kind, involves adjusting the values of the data to a common scale, but it doesn't change the relationships between the variables, which are what the correlation coefficients capture. This is protrayed by the calculated coefficients and the figures shown below.

For the rest of the exercise, we decided to work with the standard score normalization, to have the data in a common scale. Next, in order to visualize the actual distributions of the features,

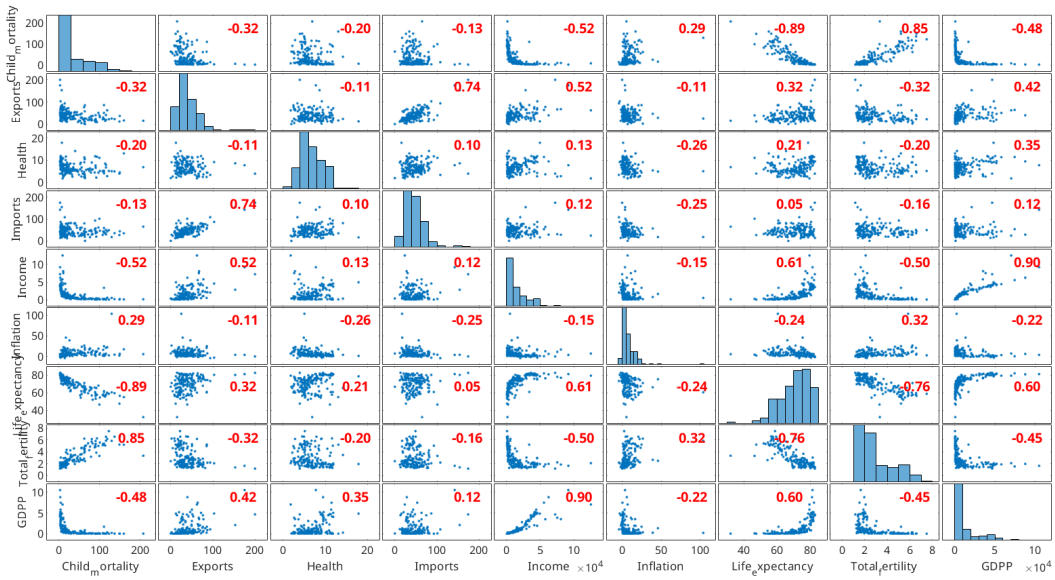

Figure 1: Matrix plot of all features with the corresponding Pearson's correlation coefficients.

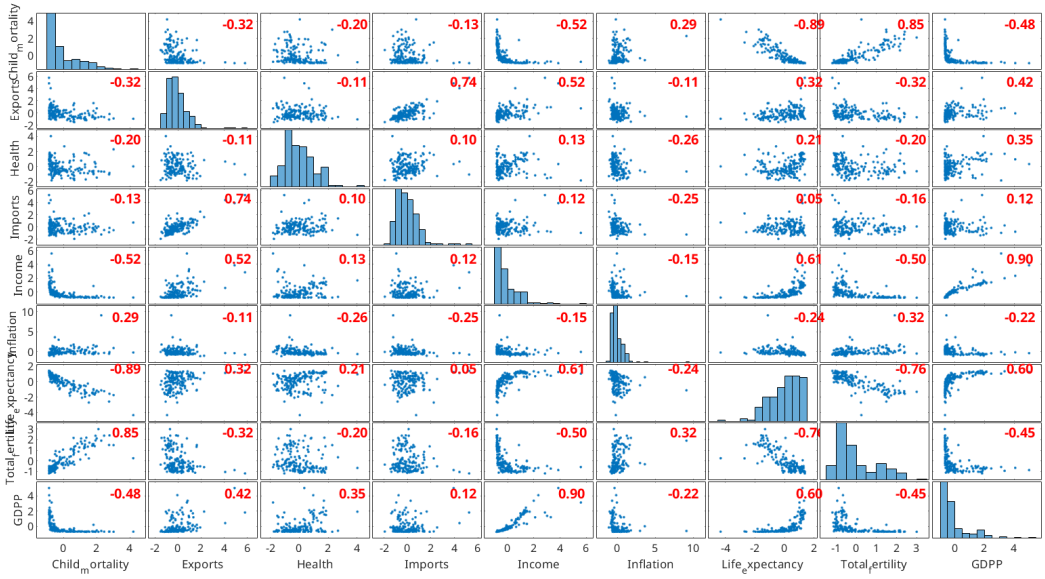

Figure 2: Matrix plot of all features with the corresponding Pearson's correlation coefficients, after standard score normalization.

we estimated the probability density functions of each feature in a non-parametric fashion, using the kernel density estimation (KDE) method (Figure 4), or on a single plot, with all features superimposed (Figure 5), highlighting their distributional differences. We also emplyed boxplots to further visualize the distributions of all nine features (Figure 6)

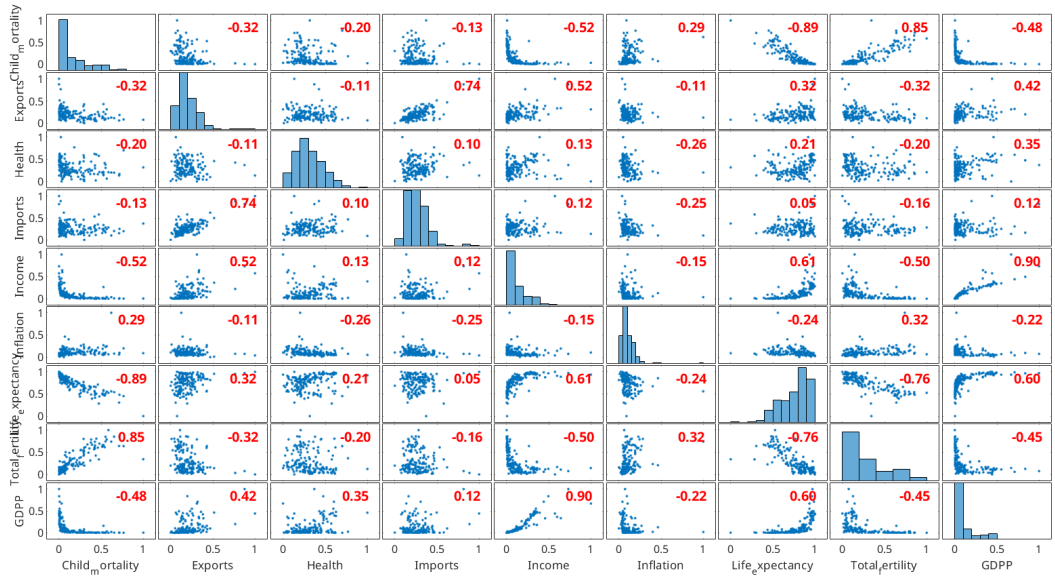

Figure 3: Matrix plot of all features with the corresponding Pearson's correlation coefficients, after minmax scaling.

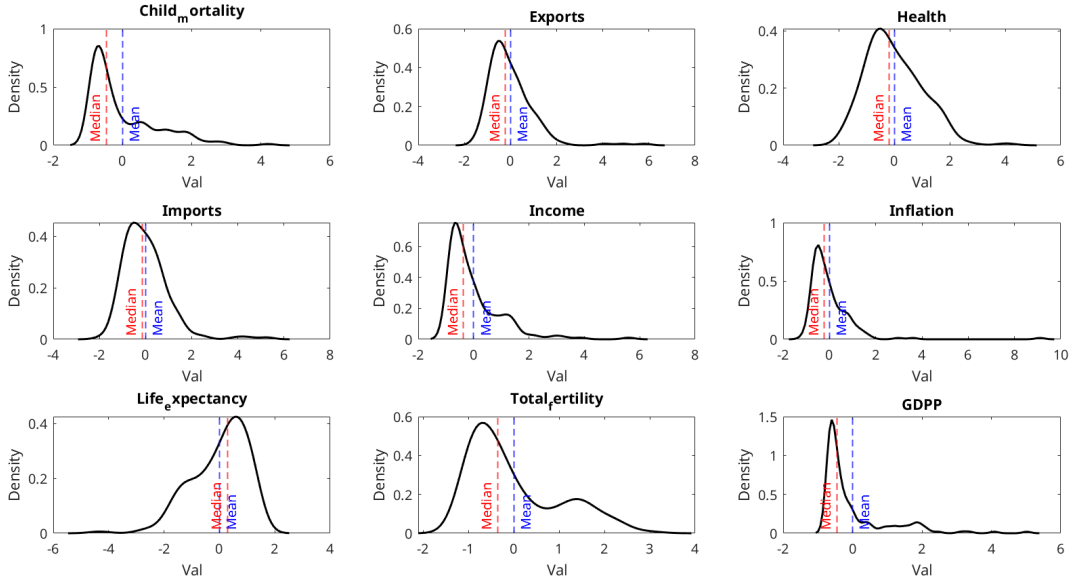

Figure 4: Kernel Density Estimation of each feature, with the mean (blue dotted line) and median (red dotted line) also visualized.

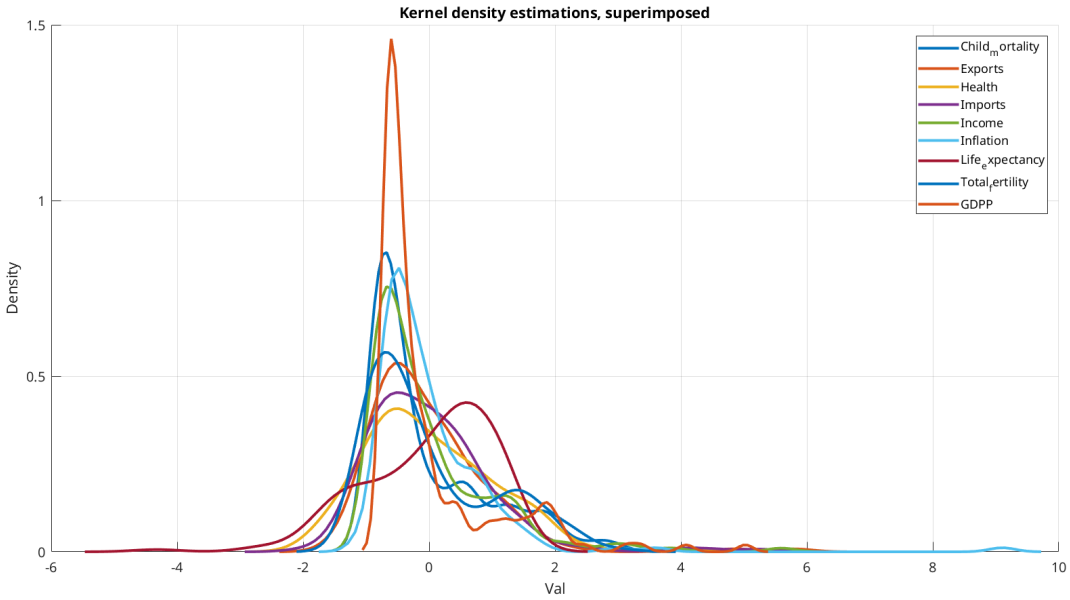

Figure 5: Kernel Density Estimation of all features superimposed on the same figure, highlighting their distributional differences.

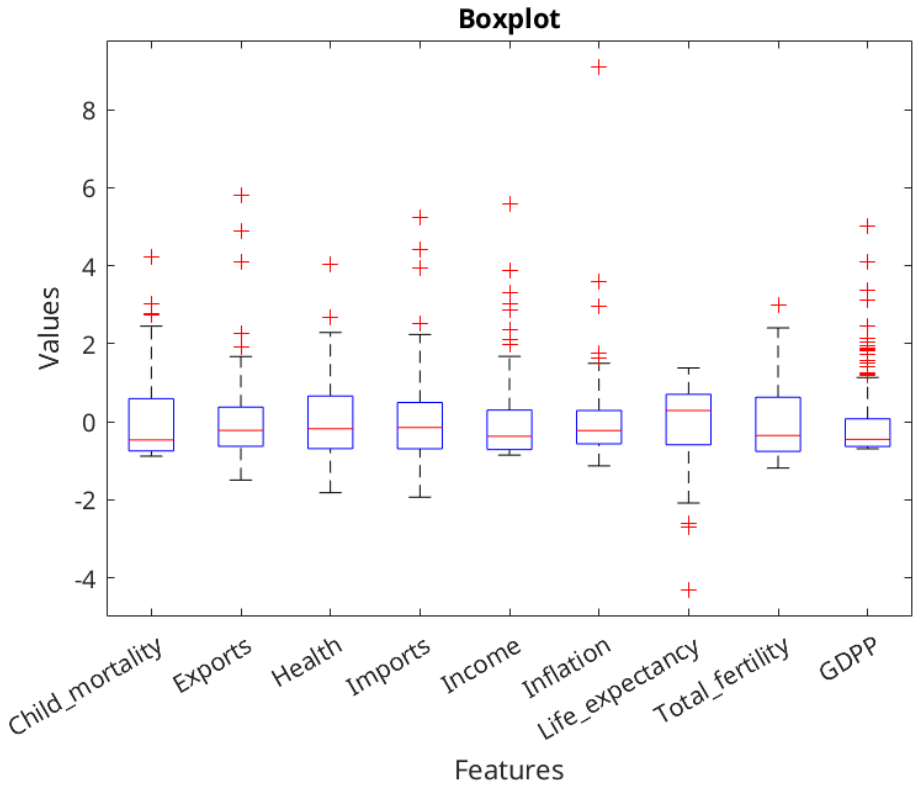

Figure 6: Boxplots of all nine features.

2 Feature selection

Although some features show a high linear correlation (e.g., GDPP and Income with a Pearson's coefficient of 0.9), we decided to retain all 9 features. The tradeoff of discarding one or two features out of a total of nine is minimal, especially when compared to scenarios where the dimensionality of the dataset is cut by half or more.

Also, although the correlated features are generally conveying similar information –like in the case of GDPP and Income– one should not so hastily decide to discard one in favour of the other, as even that small degree of variance between them could provide valuable insights. Or in the case of life expectancy and total fertility, even though they share a Pearson's coefficient of -0.76, not only do they have estimated probability density functions of different shapes, but they also convey information that is not strictly identical or dependent.

3 Selection of the clustering algorithm

For this stage of the analysis we chose to cluster the data, using the k-means, k-medians, k-medoids and possibilistic clustering algorithms in order to determine which one results in clusters that represent better the data and to also find the best initial parameters.

For this part, we used the standardized data to do the clustering, since features were on very different scales and since we did not want to prioritize any of the features. Standardization ensures all features contribute equally and produce more meaningful clusters due to the improved distance calculations. Also, as an additional note, when we clustered the non-normalized data, the cost was many orders of magnitute bigger, and the clustering algorithms took several more loops to cluster the data (these results are not show here). We plotted the Cost Functions value (J) for the different number of clusters for the k-means, k-medians, and k-medoids algorithms in the same plot, in order to compare the number of clusters, for which the relative gain is optimized. The results are shown in Figure 7.

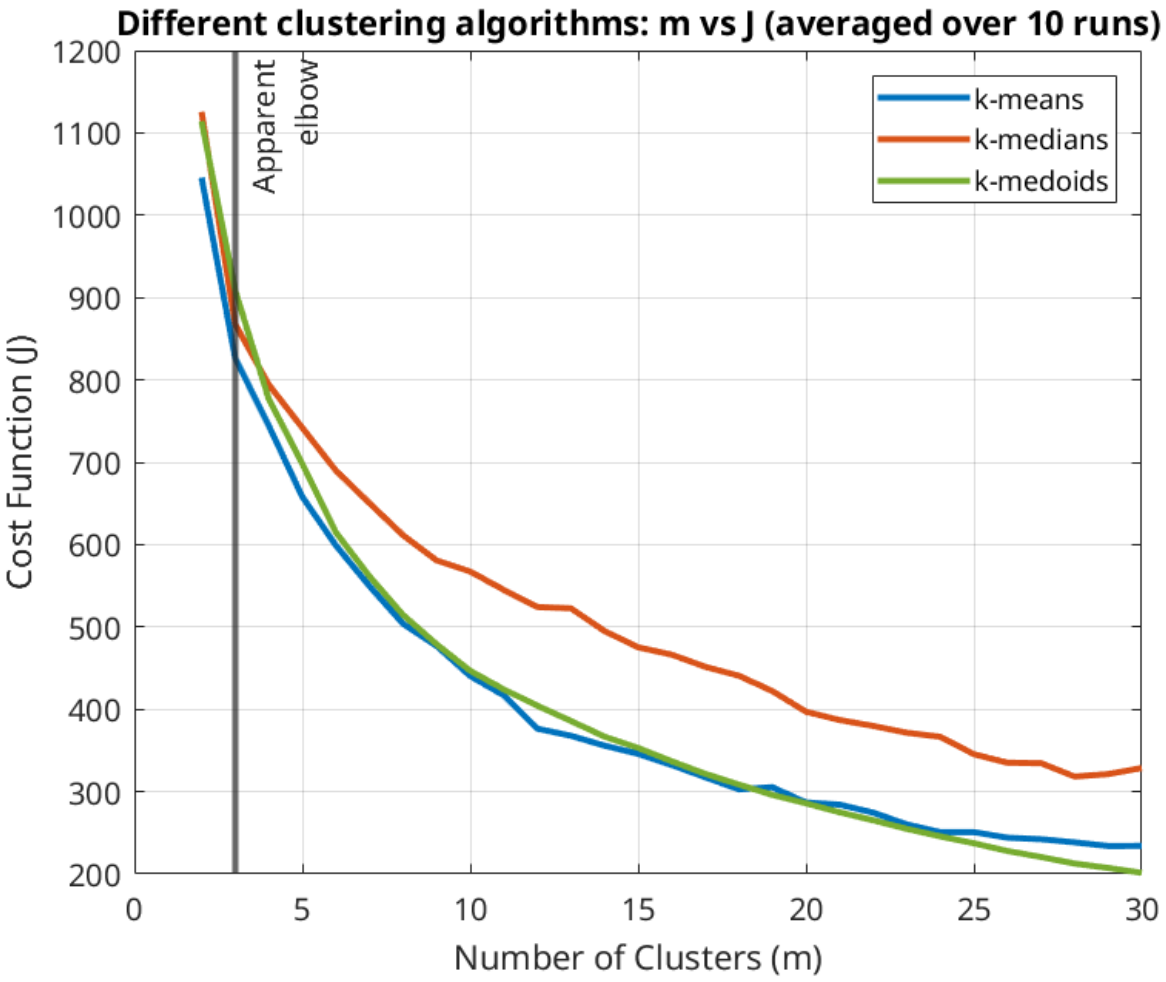

Figure 7: Cost function J over the number of clusters, for the k-means (blue line), k-medians (red line), and k-medoids algorithms (green line). The apparent elbow is represented by a grey line. Presented are the results of 10 runs, avaraged.

3.1 k-means clustering algorithm

The first algorithm we implemented was the k-means algorithm, where we chose to use randomly selected initial thetas, from a range around the mean value for each feature. The k-means algorithm adopts the squared Euclidean distance to measure the dissimilarity between vectors $\boldsymbol{x}$ (countries) and cluster representatives $\boldsymbol{\theta}$. One major advantage of the k-means algorithm is its computational simplicity, making it eligible for processing large data sets. Given the fact that most of the features in our data set have continuous values, k-means is really suitable algorithm for processing this data set. K-means benefits from the use of standardized data, since it is sensitive to feature scales. Although this algorithm is fast and computationally efficient for large data sets, it is sensitive to outliers, which is considered to be an issue in this case since our data set has features that are scaled very differently.

3.2 k-medians clustering algorithm

The difficulty k-means faces when dealing with outliers does not hold the k-medians algorithm, due to the fact that this algorithm employs the median instead of the mean and the medians are less affected by extreme values. The k-medians algorithm adopts the Manhattan distance and it aims to minimize the sum of absolute distances between vectors $\boldsymbol{x}$ (countries) and cluster representatives $\boldsymbol{\theta}$. This algorithm assumes more compact clusters.

3.3 k-medoids clustering algorithm

The next algorithm we implemented was the k-medoids, that is similar to the k-medians algorithm but uses actual points of the data set as representatives for the clusters. One major advantage of the k-medoids over the k-means algorithm, is that the k-medoids algorithm is more suitable for data sets that consist of both discrete and continuous values, whereas the k-means algorithm is mainly used for data sets originating from continuous domains. Also, the k-medoids algorithm is less sensitive to outliers than the k-means, although that comes with more computationally demanding estimation of the dissimilarity measures.

3.4 possibilistic clustering algorithm

We also ran the possibilistic algorithm, which follows a soft clustering scheme. Compared to the algorithms above, it does not assign points to a cluster, but rather assigns a membership degree to them for the clusters. It handles noise and outliers really well and results in overlapping clusters, since its relaxed from constraints and different points can belong to different clusters. In this case, q and η are introduced. Because of the intensive computation of the cost function and the difficulty in parameter tuning, the possibilistic algorithm is not considered suitable for our case, especially since our data is not noisy.

Nonetheless, we performed a test run of the possibilistic algorithm, checking for different values of q to check which one would be optimal, in case it was used. The number of clusters was determined by the previous explorations and was set to 3 and η was calculated based on equation 1.

$$\eta_j = \eta = \frac{\beta}{q\sqrt{m}}$$
$$\beta = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{x} - \overline{\boldsymbol{x_i}}||^2 \tag{1}$$
$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x_i}$$

The resulting cost was plotted against different initial q values and is presented in Figure 8.

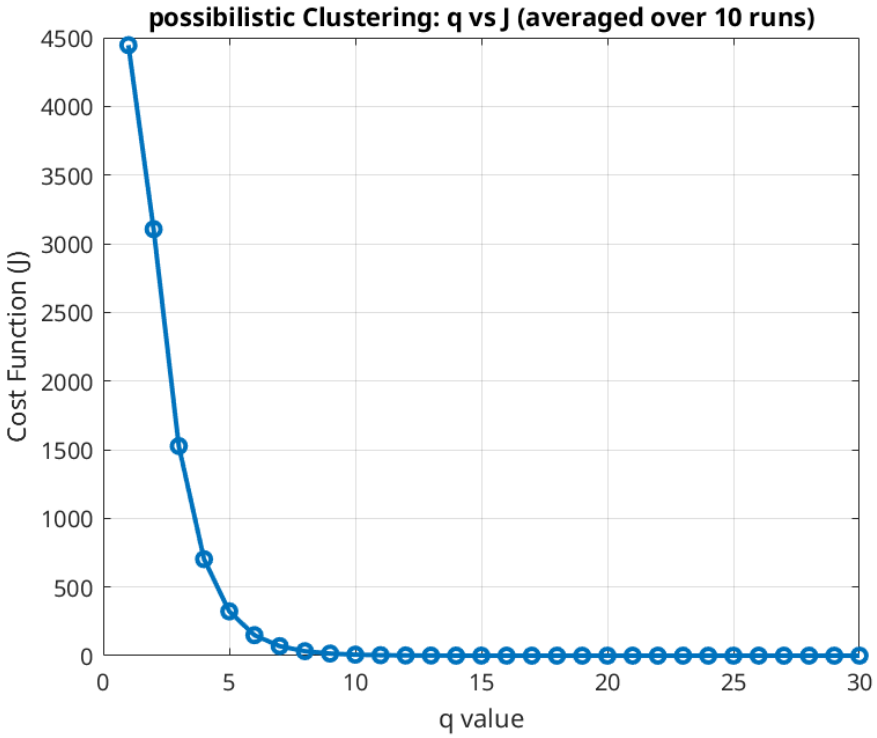

Figure 8: Cost function J over different initial q values. Presented are the results of 10 different runs, averaged.

3.5 Selecting of the clustering algorithm

From the plots shown in Figure 7, we observe that the k-means, k-medians and k-medoids algorithms result in similar costs values, which gives the largest gain when the number of clusters is 3. Taken into account that this result is supported from all the algorithms, we take it to be the best approximation to the physical clustering.

In order to select the best algorithm to continue using in our analysis, we took into account all the aforementioned characteristics of each algorithm, as well as some factors such as simplicity, efficiency and time into consideration. The possibilistic algorithm was rejected from the decision pool, because of its ambiguous resulting clustering and the many parameters that had to be defined. From the three remaining choices, the k-medoids was considered a poor one compared to the other two, due to the fact that it is more ideal for smaller data sets, it is computationally

expensive and is really sensitive to the initial medoid choice. We concluded that the best algorithm to use for this data set was the **k-medians**. Both k-means and k-medians algorithms are more suitable for largest data sets and k-means is faster, but k-medians is more robust and less sensitive to outliers, which we found to be more important.

4 Execution of the clustering algorithm

Based on our observations above, we proceeded with the **k-medians** algorithm with 3 clusters. The algorithm terminated after a few seconds, the cost function (J) had a value of around 850. The resulting clustered countries are displayed on the following table (Table 2).

Cluster 1	**Cluster 2**	**Cluster 3**
Albania	Afghanistan	Australia
Algeria	Angola	Austria
Antigua and Barbuda	Benin	Bahamas
Argentina	Burkina Faso	Belgium
Armenia	Burundi	Brunei
Azerbaijan	Cameroon	Canada
Bahrain	Central African Republic	Cyprus
Bangladesh	Chad	Czech Republic
Barbados	Comoros	Denmark
Belarus	Congo Dem Rep	Finland
Belize	Congo Rep	France
Bhutan	Cote d'Ivoire	Germany
Bolivia	Equatorial Guinea	Greece
Bosnia and Herzegovina	Eritrea	Iceland
Botswana	Gabon	Ireland
Brazil	Gambia	Israel
Bulgaria	Ghana	Italy
Cambodia	Guinea	Japan
Cape Verde	Guinea-Bissau	Kuwait
Chile	Haiti	Luxembourg
China	Iraq	Malta
Colombia	Kenya	Netherlands
Costa Rica	Kiribati	New Zealand
Croatia	Lao	Norway
Dominican	Republic Lesotho	Portugal
Ecuador	Liberia	Qatar
Egypt	Madagascar	Singapore
El Salvador	Malawi	Slovenia
Estonia	Mali	South Korea
Fiji	Mauritania	Spain
FYROM	Mozambique	Sweden
Georgia	Myanmar	Switzerland
Grenada	Namibia	United Arab Emirates
Guatemala	Niger	United Kingdom
Guyana	Nigeria	United States
Hungary	Pakistan	
India	Rwanda	
Indonesia	Senegal	
Iran	Sierra Leone	
Jamaica	South Africa	
Jordan	Sudan	

Kazakhstan	Tanzania
Kyrgyz Republic	Timor-Leste
Latvia	Togo
Lebanon	Uganda
Libya	Yemen
Lithuania	Zambia
Malaysia	
Maldives	
Mauritius	
Micronesia Fed Sts	
Moldova	
Mongolia	
Montenegro	
Morocco	
Nepal	
Oman	
Panama	
Paraguay	
Peru	
Philippines	
Poland	
Romania	
Russia	
Samoa	
Saudi Arabia	
Serbia	
Seychelles	
Slovak Republic	
Solomon Islands	
Sri Lanka	
St. Vincent and the Grenadines	
Suriname	
Tajikistan	
Thailand	
Tonga	
Tunisia	
Turkey	
Turkmenistan	
Ukraine	
Uruguay	
Uzbekistan	
Vanuatu	
Venezuela	
Vietnam	

Table 2: Final clusters

5 Characterization of the clusters

At a first glance we can distinguish that the countries that have been clustered together are what are referred to as the developed, developing and underdeveloped countries, corresponding to clusters 3, 1, and 2 respectively (Figure 9). Cluster 1 contains countries from South America, Asia, and Northern Africa. Cluster 2 is almost entirely comprised of African countries, with a few middle Eastern and Asian countries. Cluster 3 is made up of European and Northern American countries, along with a few others.

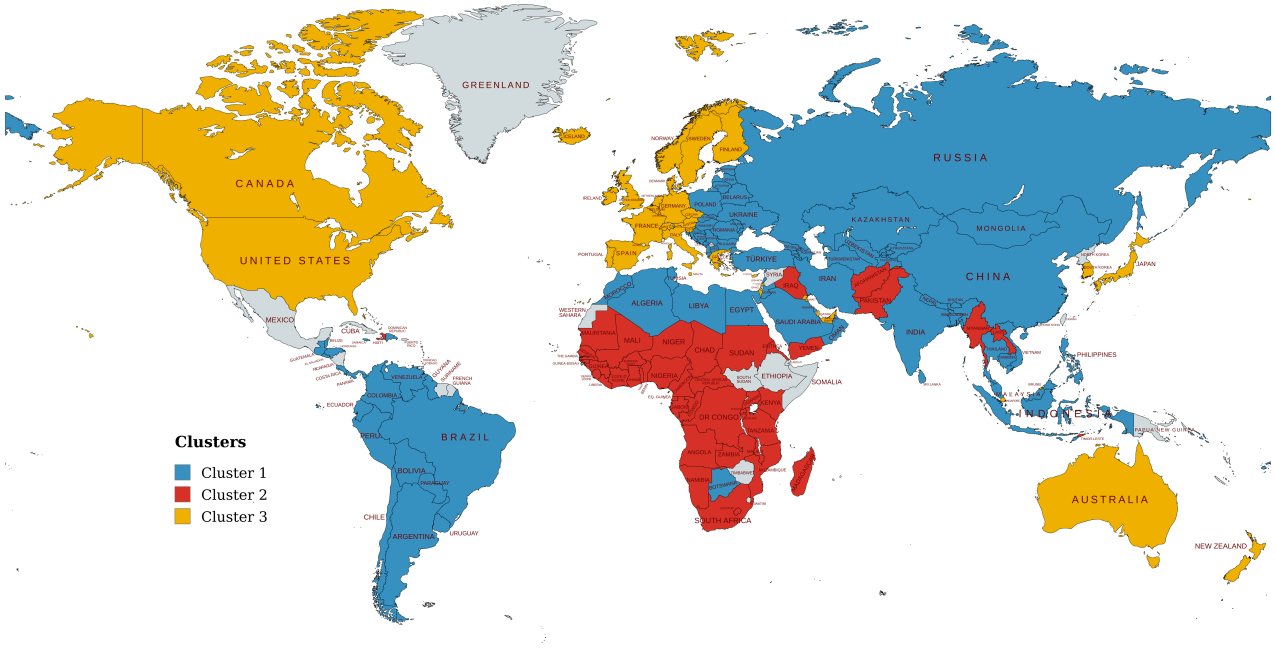

Figure 9: World map with countries colored based on their cluster. Countries of cluster 1, 2, and 3 are colored in blue, red and yellow respectively, while countries not in the dataset are in grey. (This map was created using MapChart, `https://www.mapchart.net/`)

Using several dimensionality reduction methods we can visualize the clustering results in a two-dimensional space. By performing Principal Component Analysis (PCA) on our dataset and plotting the first and second principal components, coloring each country based on its cluster, we gain insight to the success of our clustering (Figure 10). From the figure, it is evident that each cluster occupies distinct regions in the component space, indicating that our clustering effectively separates the dataset. We can also employ other methods like the t-Distributed Stochastic Neighbor Embedding (t-SNE) and Multidimensional Scaling (MDS) (Figure 11, Figure 12) and like in the case of PCA, the results show a well separated space between the three clusters.

By calculating the feature means for each cluster, we can deduce how each cluster's features differ from each other (Table 3). We can see that countries in the third cluster have a bigger mean GDPP from others, smaller inflation, higher income, export more goods, have way less child mortality and have better provided health. Countries in the second cluster are the extreme opposite, while countries in the first cluster lie somewhere in the middle.

So we can safely say that we succeeded –as Poincaré said– in giving the same name to different things.

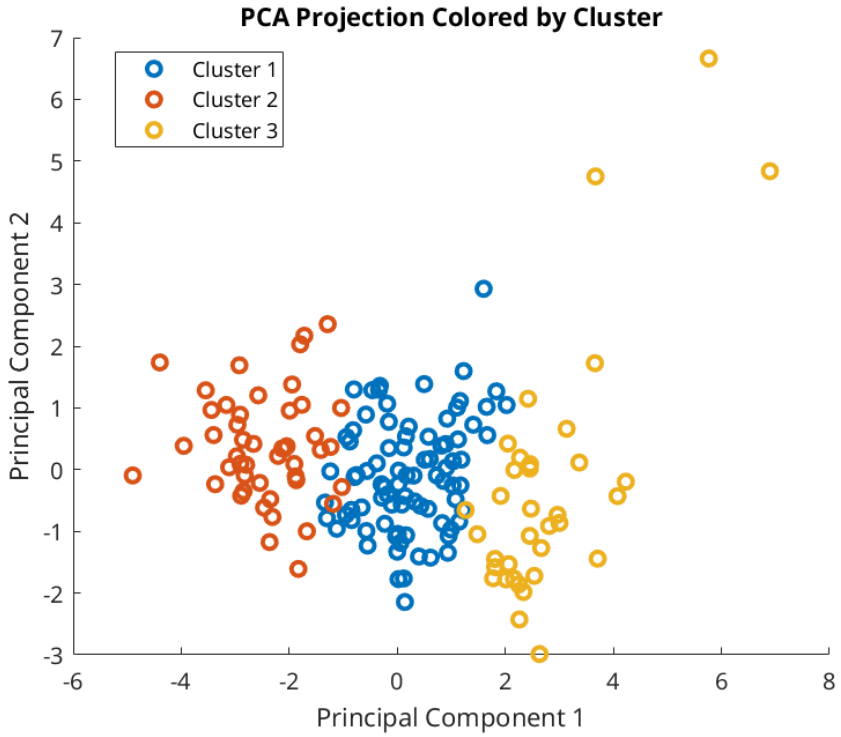

Figure 10: PCA: first and second principal components with each country colored based on its corresponding cluster.

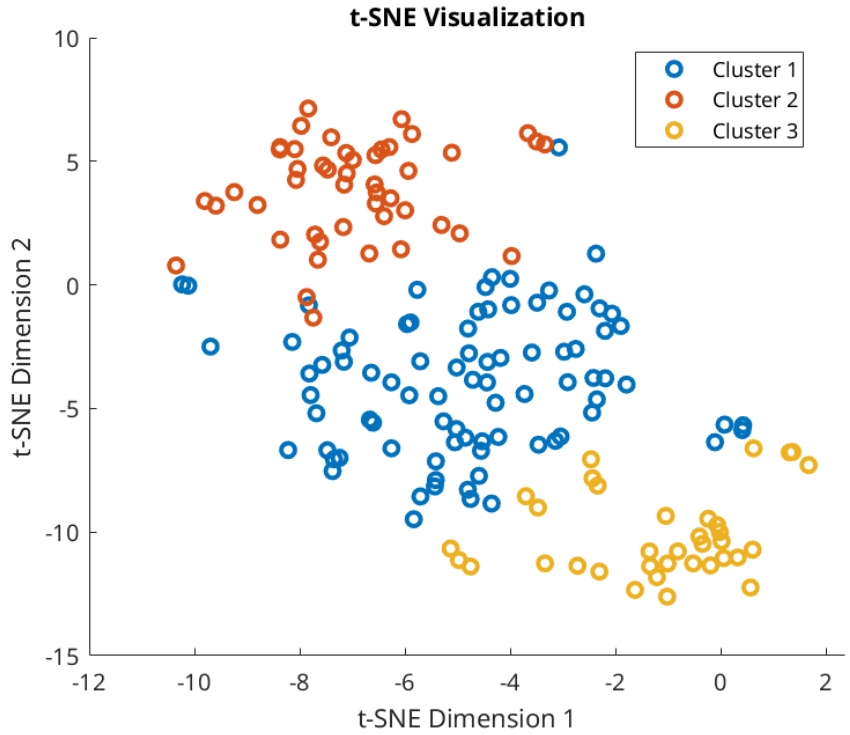

Figure 11: t-SNE: first and second dimensions with each country colored based on its corresponding cluster.

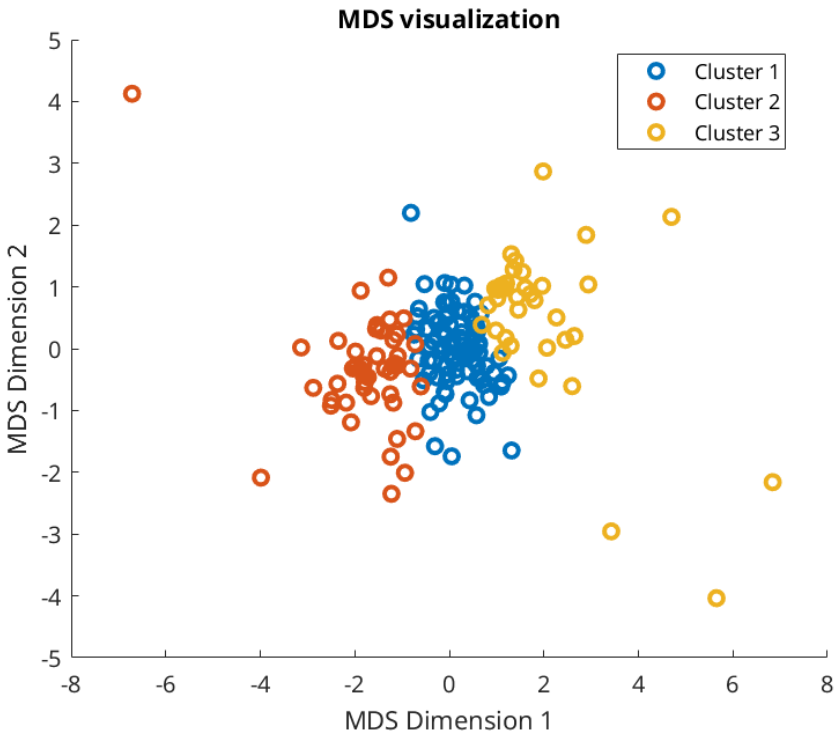

Figure 12: MDS: first and second principal MDS dimensions with each country colored based on its corresponding cluster.

Cluster	Child mortality	Exports	Health	Imports	Income	Inflation	Life expectancy	Total fertility	GDPP
Cluster 1	-0.41458	0.017389	-0.1981	0.067223	-0.22619	-0.014225	0.2476	-0.42391	0.34817
Cluster 2	1.3624	-0.46998	-0.20455	-0.23366	-0.69541	0.3712	-1.2551	1.3542	-0.60865
Cluster 3	-0.8227	0.58888	0.75579	0.15052	1.4832	-0.49873	1.0841	-0.78907	1.6629

Table 3: Means of each cluster's features

6 MATLAB code

All MATLAB scripts and functions used in our work, as well as the dataset which is clustered, are provided along with this pdf document in the same compressed folder.

The provided MATLAB functions were used as-is, without any modifications. Of those functions that were provided to us, we used the following:

1. `cost_comput.m`
2. `distan.m`
3. `distant_init.m`
4. `k_means.m`
5. `k_medians.m`
6. `k_medoids.m`
7. `possibi.m`

Other than those, we created some additional auxilarry functions which were used for calculating the η parameter for the possibilistic soft clustering and for calculating the cost of the same clustering.

1. `calc_eta.m`
2. `possibi_cost.m`

All the analyses and clustering was performed using the following scripts, the first one for "feeling the data", the second for identifying the best clustering and parameters, and the third for performing the actual clustering and visualizing the results.

1. `data_overview.m`
2. `exploratory.m`
3. `clusters.m`
4. `characterization.m`