Data for Good: clustering countries using unsupervised Machine Learning.

Sebastiano Denegri

March 25th, 2022

It is a capital mistake to theorize before one has data.
Arthur Conan Doyle

Unsupervised Machine Learning – IBM / Coursera.

Table of contents.

1. Introduction: the project.

Data for good means using Data Science and Machine Learning tools outside of the for-profit sector to help Non-profits and NGOs leverage the power of data for social good and humanitarian causes.

HELP International is an international humanitarian NGO that is committed to fighting poverty and promoting socially responsible and sustainable change. HELP International specializes in four core competencies:

- public health
- education
- entrepreneurship/business
- infrastructure development projects.

The scope of this project is to cluster world countries, based on socio-economic and health factors, to determine the overall development status of a country to assist HELP International, as well as other NGOs, use their funds strategically and effectively, in order to support those nations that are at the bottom of the development curve.

Considering the scope of the project, there is no predefined number of clusters set; however, the purpose of this analysis is to add some insights into the data, therefore I reckoned that clustering countries in only 2 categories (such as “More Developed” vs “Less Developed”, for instance) wouldn't really help NGOs make strategic decisions on fund spending.

Therefore, besides clustering data points correctly, this analysis aims to recommend a clustering algorithm able to categorize the data into **the highest number of distinct, accurate, non-overlapping clusters.**

2. Methodology.

In order to meet the project requirements, I’ve followed a **descriptive analytic approach** aimed to correctly classify countries into the highest number of meaningful non-overlapping categories.
To deliver reliable results, I’ve followed the CRoss-Industry Standard Process for Data Mining (CRISP-DM), which consists of the following steps:

1. **Business Understanding** (see Introduction section).
2. **Data Understanding:** data cleaning and exploratory data analysis.
3. **Data Preparation**: transform data into a usable dataset for modeling.
4. **Modeling**:
    - In order to choose the best clustering model, I’ve trained K-Means algorithm on 3 different feature sets and compared model performance; feature sets were as follows:
        i) Trained K-Means using all data features.
        ii) Performed Principal Component Analysis to combine linearly correlated variables, if any, and trained K-Means on a fewer number of components.
        iii) Performed feature selection based on boxplot analysis of each feature in relation to the found clusters:
            - Used all variables to train K-Means
            - For every feature, tested if there was a significant difference between the clusters
            - Deleted the variables by which there was no significant difference
            - Re-trained K-Means on the left features only.
    - After choosing the best feature set, I trained the following algorithms and compared performance:
        i) Hierarchical Agglomerative Clustering
        ii) DBSCAN
5. **Evaluation**: clustering models were tested using a mix of metrics such as Inertia, Distortion, Silhouette Coefficient, as well as visually inspecting the model results (pairplots, boxplots...).

2.1. Data Understanding and Preparation.

Dataset source: www.kaggle.com.

The dataset includes 9 features with information on 167 countries. Features are as follows:

A. **child_mort**: death of children under 5 years of age per 1,000 live births
B. **exports**: exports of goods and services, expressed as % of the GDP per capita.
C. **health**: total health-related spending, expressed as % of GDP per capita.
D. **imports**: imports of goods and services, expressed as % of the GDP per capita.
E. **income:** net annual income per person in USD.
F. **Inflation:** rate of increase in prices and fall in the purchasing value of money.
G. **life_expec:** the average number of years a new born child would live if the current mortality patterns are to remain the same.
H. **total_fer:** average number of children born per woman
I. **gdpp:** the Gross Domestic Product per capita in USD (GDP divided by the total population).

All features are numeric type: 7 floats and 2 integers. After checking the data central tendency, I could confirm the dataset contained valid data:

- Average life expectancy: 70.5 years
- Average child mortality rate (per 1,000 live births): 38.3 children
- Median fertility rate: 2.4 children
- Average income: 17,145 USD

2.1.1. Data Cleaning

Neither duplicates nor missing values were found. I performed boxplots analysis checking for outliers. Results are as follows:

- Child_mort - 2.4% of outliers: Central African Republic, Chad, Haiti, Sierra Leone.
- Exports - 2.99% of outliers: Ireland, Luxembourg, Malta, Seychelles, Singapore.
- Health - 1.2% of outliers: Micronesia, United States.
- Imports - 2.4% of outliers: Luxembourg, Malta, Seychelles, Singapore.
- Income - 4.79% of outliers: Brunei, Kuwait, Luxembourg, Norway, Qatar, Singapore, Switzerland, United Arab Emirates.
- Inflation - 2.99% of outliers: Equatorial Guinea, Mongolia, Nigeria, Timor-Leste, Venezuela.
- Life_expec - 1.8% of outliers: Central African Republic, Haiti, Lesotho.
- Total_fer - 0.6% of outliers: Niger
- GDPP - 14.97% of outliers: Australia, Austria, Belgium, Brunei, Canada, Denmark, Finland, France, Germany, Iceland, Ireland, Italy, Japan, Kuwait, Luxembourg, Netherlands, New Zealand, Norway, Qatar, Singapore, Sweden, Switzerland, United Arab Emirates, United Kingdom, United States.

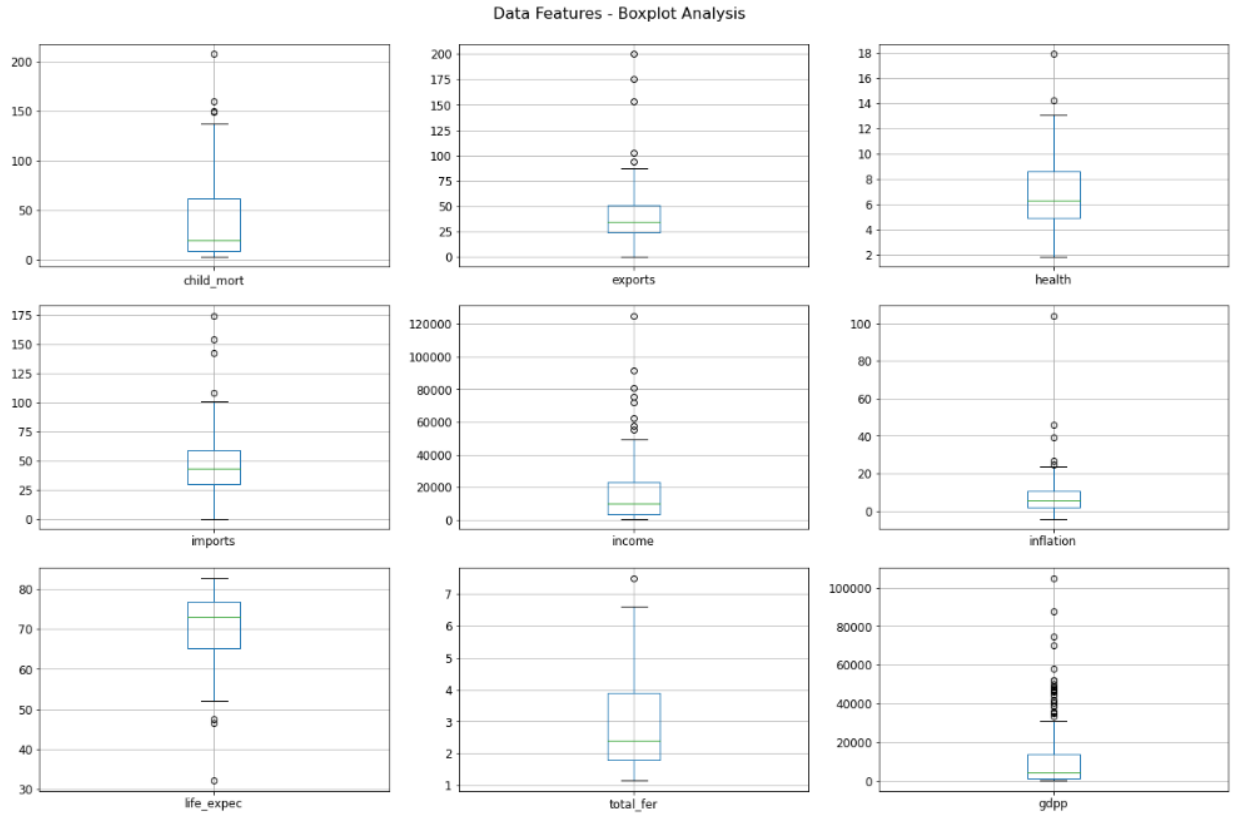

Figure 1: Outlier analysis with boxplots

All features have outliers on the right side of data distribution, with the exception of Life_Expec. After cross-checking countries' data on the Internet, I deemed the outliers as correct observations, therefore I left them untouched in the dataset.

2.1.2. Exploratory Data Analysis (EDA).

In the EDA stage, I analyzed data distribution and skewness as well as features' correlation.

Results are as below:

```
"child_mort" skew: 1.45. The variable is NOT normally distributed.
"exports" skew: 2.45. The variable is NOT normally distributed.
"health" skew: 0.71. The Variable is normally distributed
"imports" skew: 1.91. The variable is NOT normally distributed.
"income" skew: 2.23. The variable is NOT normally distributed.
"inflation" skew: 5.15. The variable is NOT normally distributed.
"life_expec" skew: -0.97. The variable is NOT normally distributed.
"total_fer" skew: 0.97. The variable is NOT normally distributed.
"gdpp" skew: 2.22. The variable is NOT normally distributed.
```

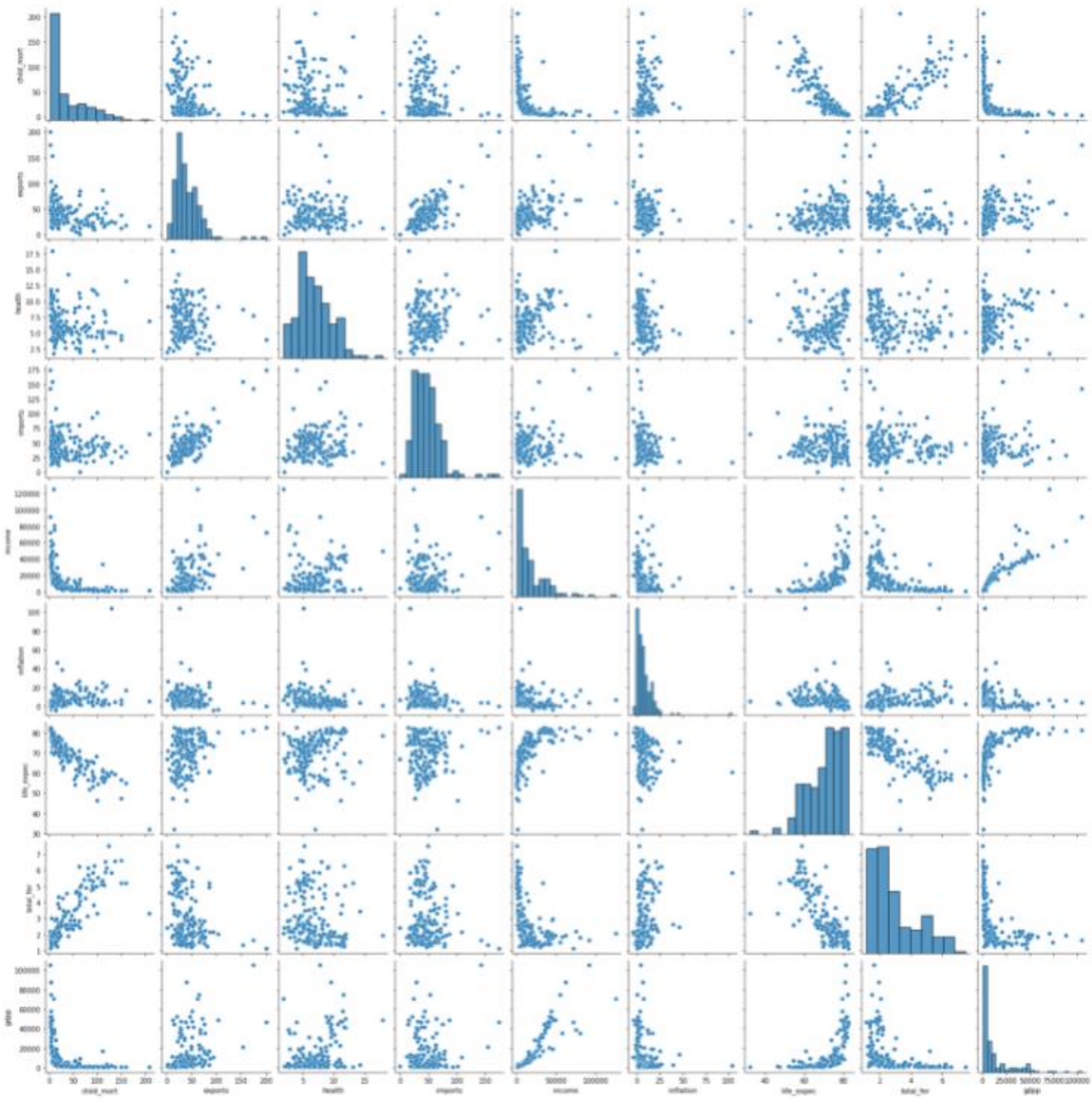

Figure 2: Pairplots - Variable distribution and pairwise correlation.

Only the “Health” variable was normally distributed. From the Pairplots above we can see there are variables that have a linear relationships:

- Child Mortality and Total Fertility have a positive linear relationship
- Child Mortality and Life Expectancy have a negative linear relationship
- Life Expectancy and Total Fertility have a negative linear relationship
- Exports and Imports have a positive linear relationship
- Income and GDPP have a positive linear relationship

Correlation Matrix (showing only Pearson Correlation Coefficients whose absolute value is greater than 0.7):

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort							-0.886676	0.848478	
exports				0.737381					
health									
imports		0.737381							
income									0.895571
inflation									
life_expec	-0.886676							-0.760875	
total_fer	0.848478						-0.760875		
gdpp					0.895571				

Figure 3: Correlation Matrix (showing only Pearson Correlation Coefficients whose absolute value is greater than 0.7)

Normality is not a requirement for Clustering algorithms, however a normal distribution of the data can help achieve better results since outliers might skew the cluster centroids. In consideration of this, I applied following transformations to enforce Normality on the data distribution (and strengthen variable correlations as well):

- Logarithmic Transformation
- Square Root Transformation
- BoxCox Transformation

2.1.3. Feature Engineering & Variable Transformations

A. Logarithmic Transformation

```
"child_mort" skew: 0.24. The Variable is normally distributed
"exports" skew: -0.36. The Variable is normally distributed
"health" skew: 0.71. The Variable is normally distributed
"imports" skew: -0.65. The Variable is normally distributed
"income" skew: -0.23. The Variable is normally distributed
"inflation" skew: -5.28. The variable is NOT normally distributed.
"life_expec" skew: -1.58. The variable is NOT normally distributed.
"total_fer" skew: 0.73. The Variable is normally distributed
"gdpp" skew: 0.01. The Variable is normally distributed
```

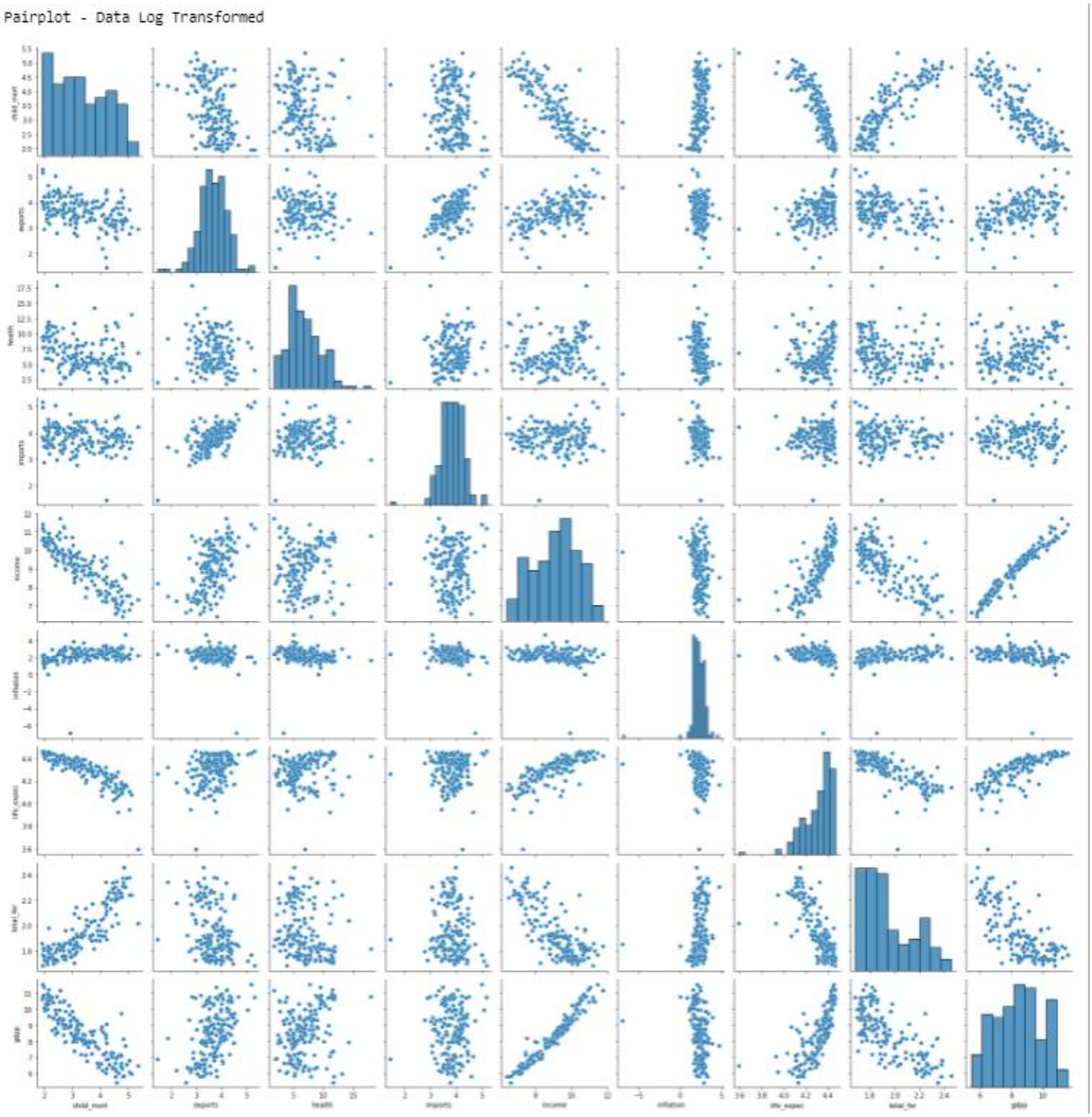

Figure 4: Pairplots - Variable distribution and pairwise correlation after Logarithmic Transformation

Correlation Matrix - Data Log Transformed

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort					-0.859524		-0.863493	0.869741	-0.868332
exports									
health									
imports									
income	-0.859524						0.773093	-0.776331	0.971986
inflation									
life_expec	-0.863493				0.773093			-0.736778	0.759246
total_fer	0.869741				-0.776331		-0.736778		-0.735979
gdpp	-0.868332				0.971986		0.759246	-0.735979	

Figure 5: Correlation Matrix after Logarithmic Transformation (showing only Correlation Coefficients greater than 0.7)

B. Square Root Transformation

"child_mort" skew: 0.81. The variable is NOT normally distributed.
"exports" skew: 0.97. The variable is NOT normally distributed.
"health" skew: 0.71. The Variable is normally distributed
"imports" skew: 0.73. The Variable is normally distributed
"income" skew: 0.86. The variable is NOT normally distributed.
"inflation" skew: 1.78. The variable is NOT normally distributed.
"life_expec" skew: -1.23. The variable is NOT normally distributed.
"total_fer" skew: 0.84. The variable is NOT normally distributed.
"gdpp" skew: 1.14. The variable is NOT normally distributed.

Correlation Matrix - Data after Square Root Transformation

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort					-0.744578		-0.899472	0.877578	-0.712024
exports									
health									
imports									
income	-0.744578						0.718478		0.947983
inflation									
life_expec	-0.899472				0.718478			-0.752539	
total_fer	0.877578						-0.752539		
gdpp	-0.712024				0.947983				

Figure 6: Correlation Matrix after Square Root Transformation (showing only Correlation Coefficients greater than 0.7)

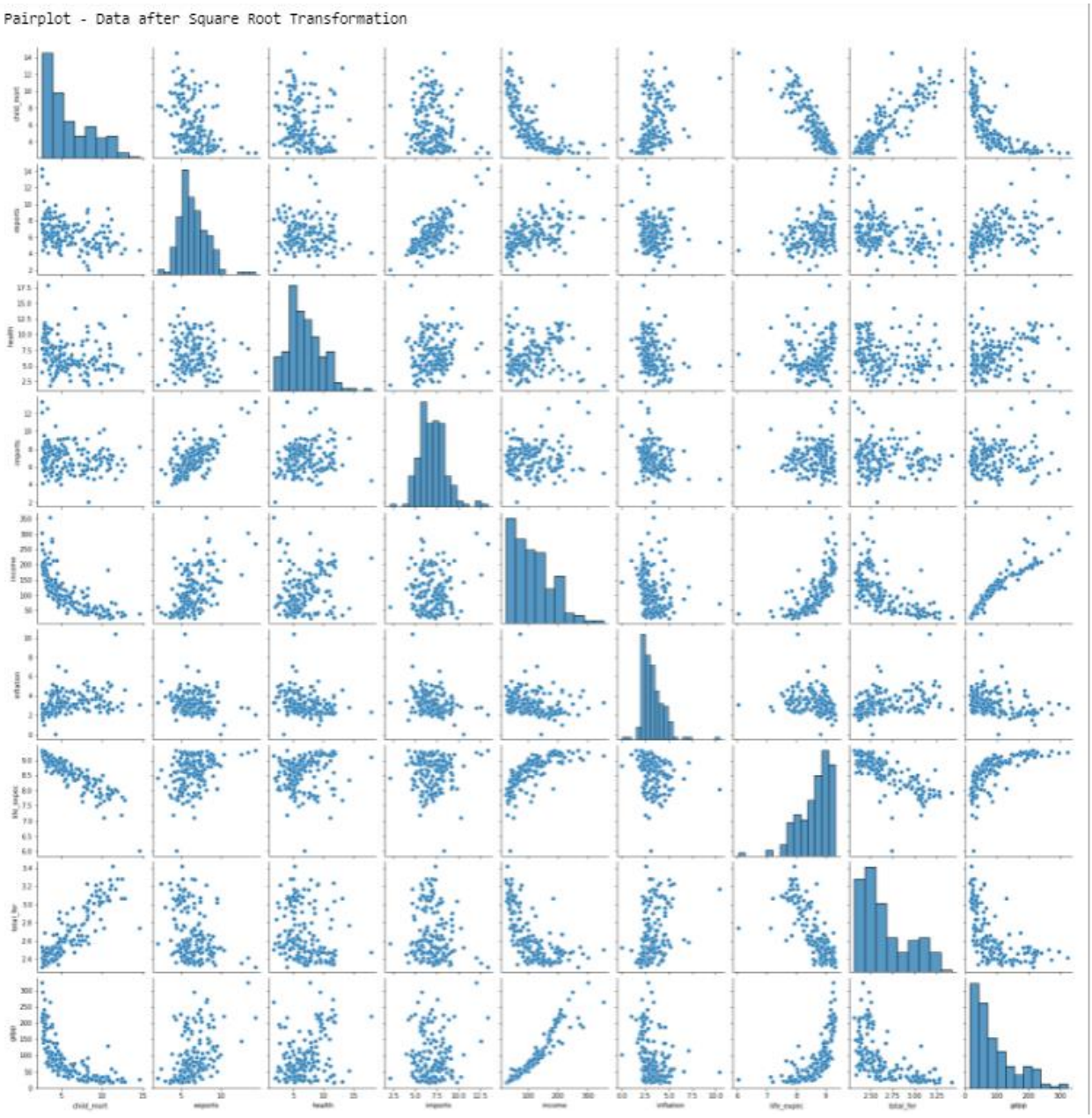

Figure 7: Pairplots - Variable distribution and pairwise correlation after Square Root Transformation

C. BoxCox Transformation

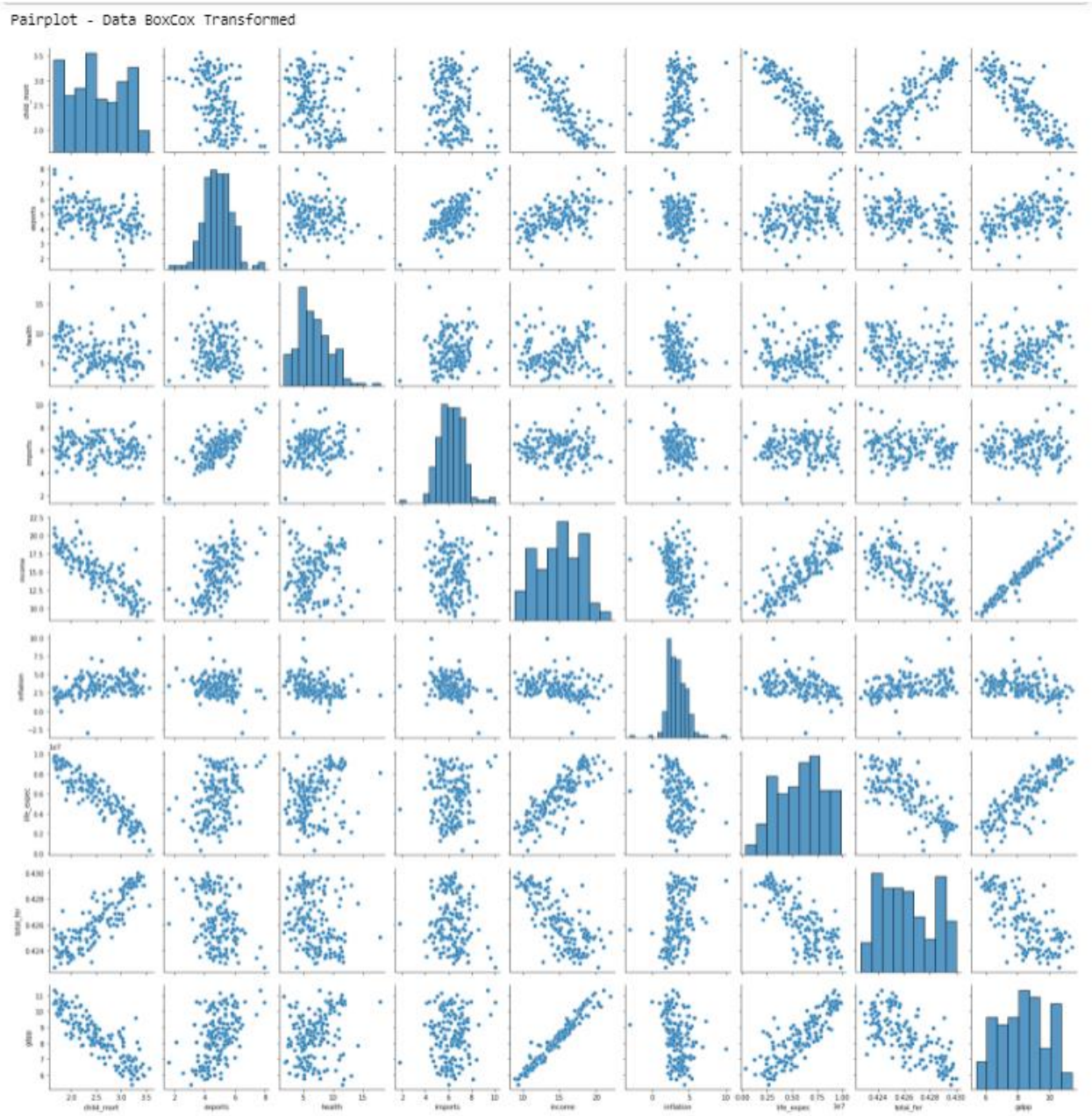

Figure 8: Pairplots - Variable distribution and pairwise correlation after BoxCox Transformation

```
"child_mort" skew: 0.06. The Variable is normally distributed
"exports" skew: 0.03. The Variable is normally distributed
"health" skew: 0.71. The Variable is normally distributed
"imports" skew: 0.07. The Variable is normally distributed
"income" skew: -0.04. The Variable is normally distributed
"inflation" skew: 0.42. The Variable is normally distributed
"life_expec" skew: -0.18. The Variable is normally distributed
"total_fer" skew: 0.2. The Variable is normally distributed
"gdpp" skew: 0.0. The Variable is normally distributed
```

Correlation Matrix - Data after BoxCox Transformation

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort					-0.858853		-0.913547	0.87042	-0.875005
exports									
health									
imports									
income	-0.858853						0.838438	-0.750676	0.974396
inflation									
life_expec	-0.913547				0.838438			-0.786441	0.849259
total_fer	0.87042				-0.750676		-0.786441		-0.733234
gdpp	-0.875005				0.974396		0.849259	-0.733234	

Figure 9: Correlation Matrix after BoxCox Transformation (showing only Correlation Coefficients greater than 0.7)

D. Results

The BoxCox transformation proved to be the most effective:

- All features are normally distributed.
- BoxCox transformation has the strongest impact in strengthening linear correlations between the features.

I transformed the Data using BoxCox and, since features were on different scales, I normalized the data using Standard Scaler (Z-Score normalization) from Scikit-Learn library.

2.2. Model Development

2.2.1. Choosing the feature set with K-Means.

I fitted K-Means algorithm on a number of clusters that goes from 2 to 10, using k-means++ as the initialization method, comparing performance from 3 different feature sets.

To choose the optimum number of clusters, per each feature set, I plotted the number of clusters vs the error, and applied the elbow rule to locate the number of clusters where the gain, in terms of error decrease, stops to be significant.

As error metric I used either Inertia or Distortion:

- Inertia: the sum of the intra-cluster distances where an intra-cluster distance is the sum of squared distances from each point to its cluster centroid.
- Distortion: the average of the mean squared distance from each point to the centroid of the respective clusters.

A. Using all data features

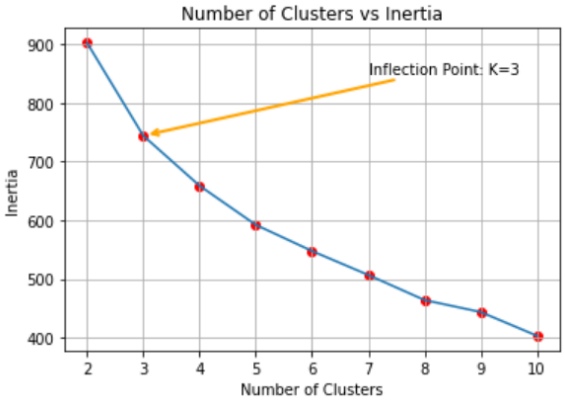

Figure 10: Analyzing the rate of Inertia decrease

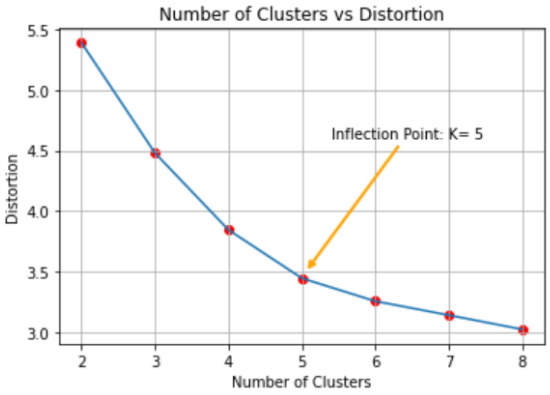

Figure 11: Analyzing the rate of Distortion decrease

Although it wasn't easy to locate an elbow point from figure 10 and 11, I can see kind of an elbow when K=5 (Number of clusters vs Distortion chart), or when K=3 (Number of clusters vs Inertia chart).

Based on the above results, I trained K-Means again with K=5, K=4, and K=3, and analyzed the quality of clusters using Boxplots.

Results are as follows:

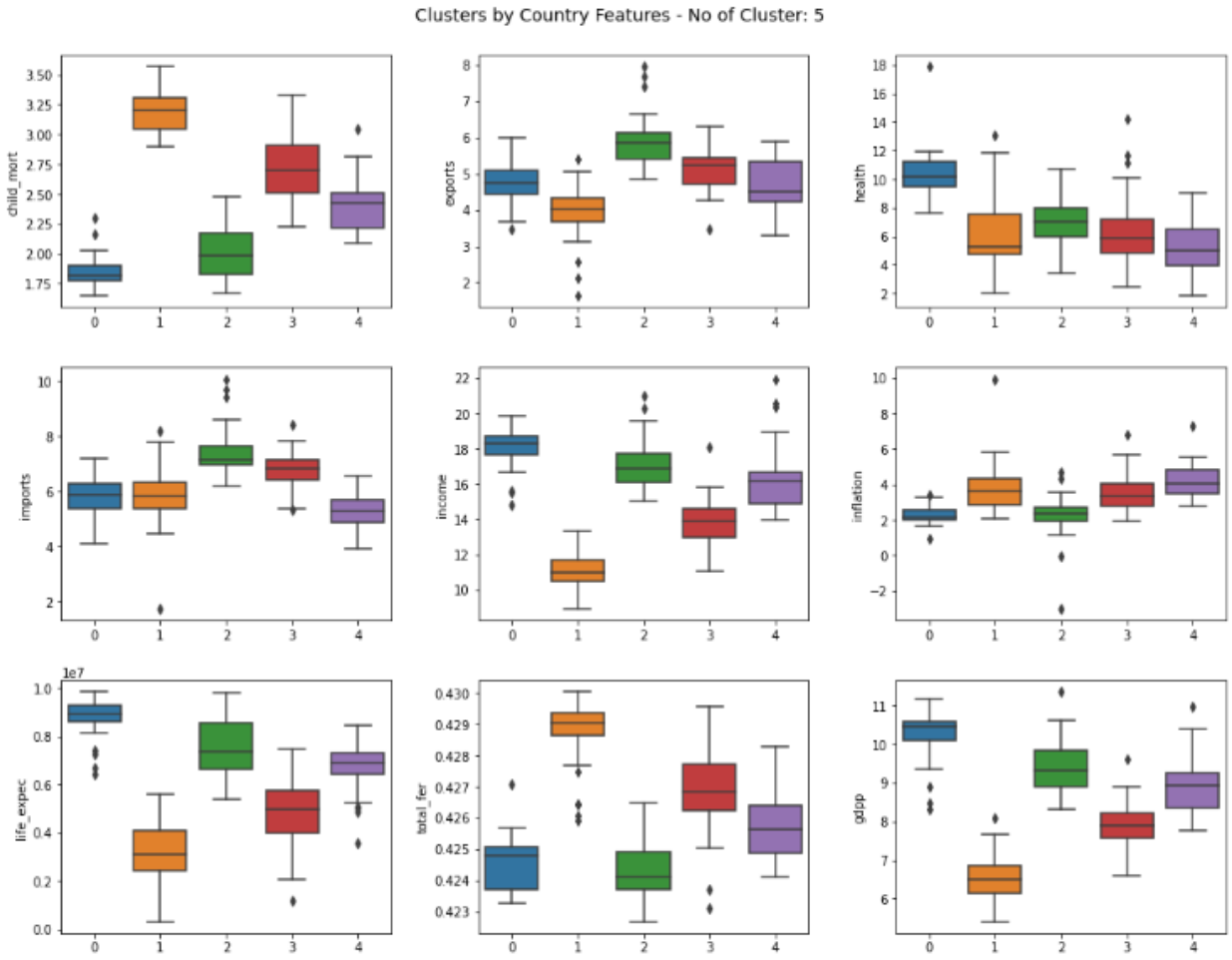

Figure 12: Boxplots – Analyzing clusters by features (number of clusters= 5).

When the number of clusters is 5, there's significant overlap between clusters 0 and 2 in Child_Mort, Income, Inflation, Life_Expec, Total_Fer, and GDPP. Clusters 2 and 4 overlap for features like Income, Life_Expec, and GDPP. In Features like Health, Inflation, Imports, and Exports, several clusters significantly overlap.

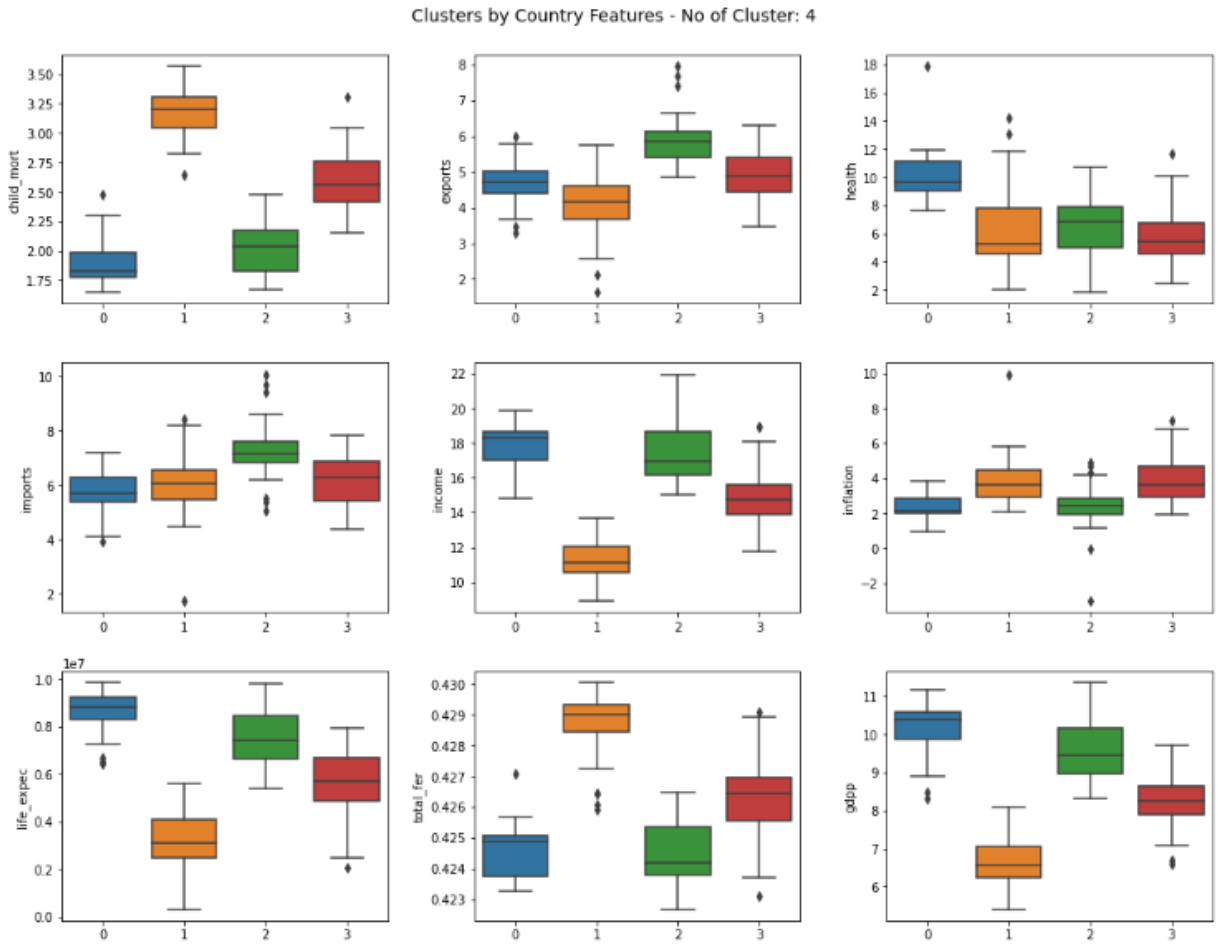

Figure 13: Boxplots – Analyzing clusters by features (number of clusters= 4).

When the number of clusters is 4, clusters 0 and 2 overlap in several features like Child_Mort, Income, Inflation, Life_Expec, Total_Fer, and GDPP. In Features like Health, Inflation, Imports, and Exports, several clusters significantly overlap.

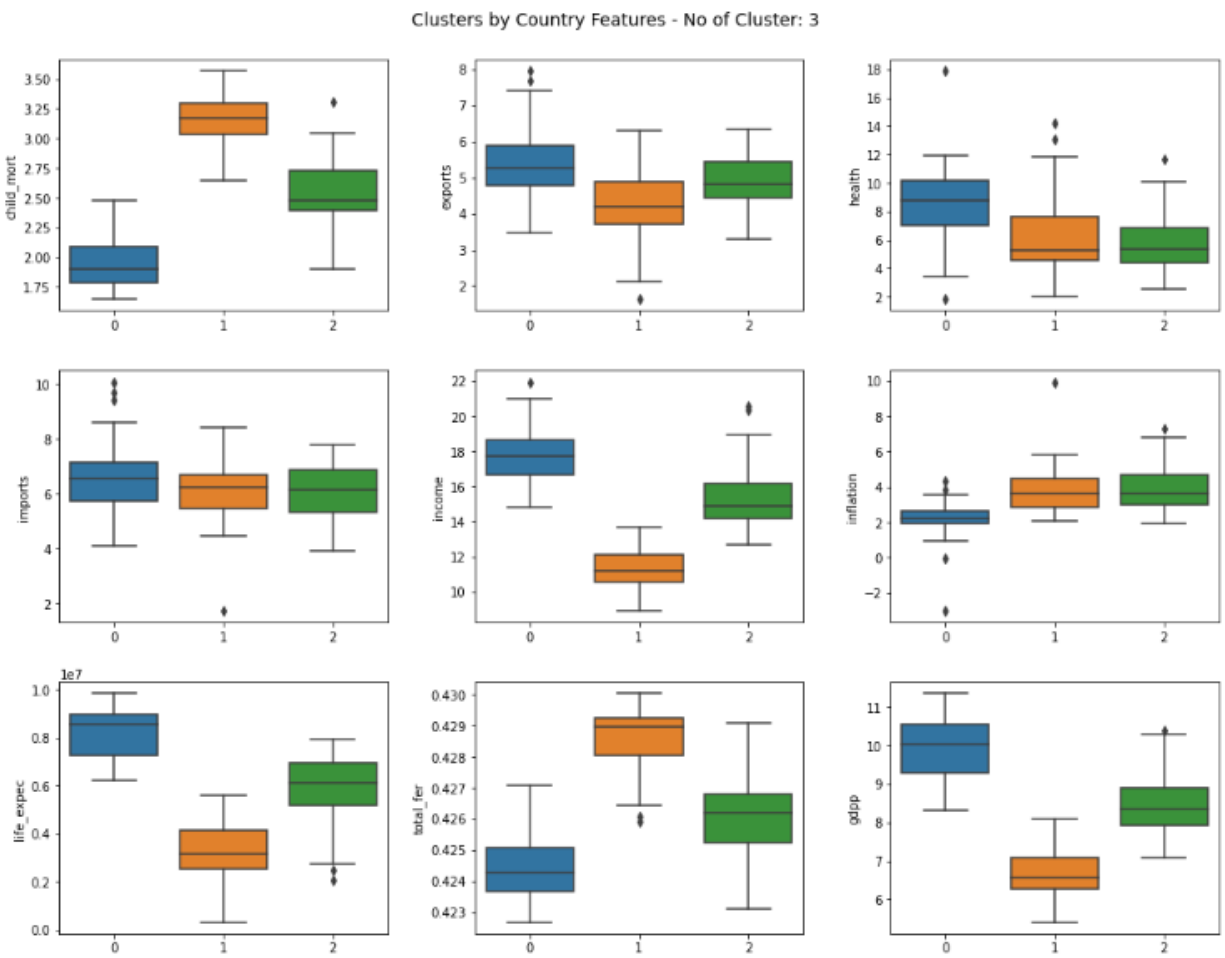

Figure 14: Boxplots – Analyzing clusters by features (number of clusters= 3).

When the number of clusters is 3, although the cluster overlap is still evident in features like Exports, Health, Imports, and Inflation, we can see distinct non overlapping clusters in features like Child_Mort, Income, Life_Expec, Total_fer, and GDPP.

3 is the greatest number of distinct meaningful clusters I managed to achieve with K-means, using all data features.

The Pairplots below confirm the results of the Boxplot analysis: there are 3 distinct clusters in features like Child_Mort, Income, Life_Expec, Total_fer, and GDPP, whilst in features like Exports, Health, Imports, and Inflation, clusters overlap.

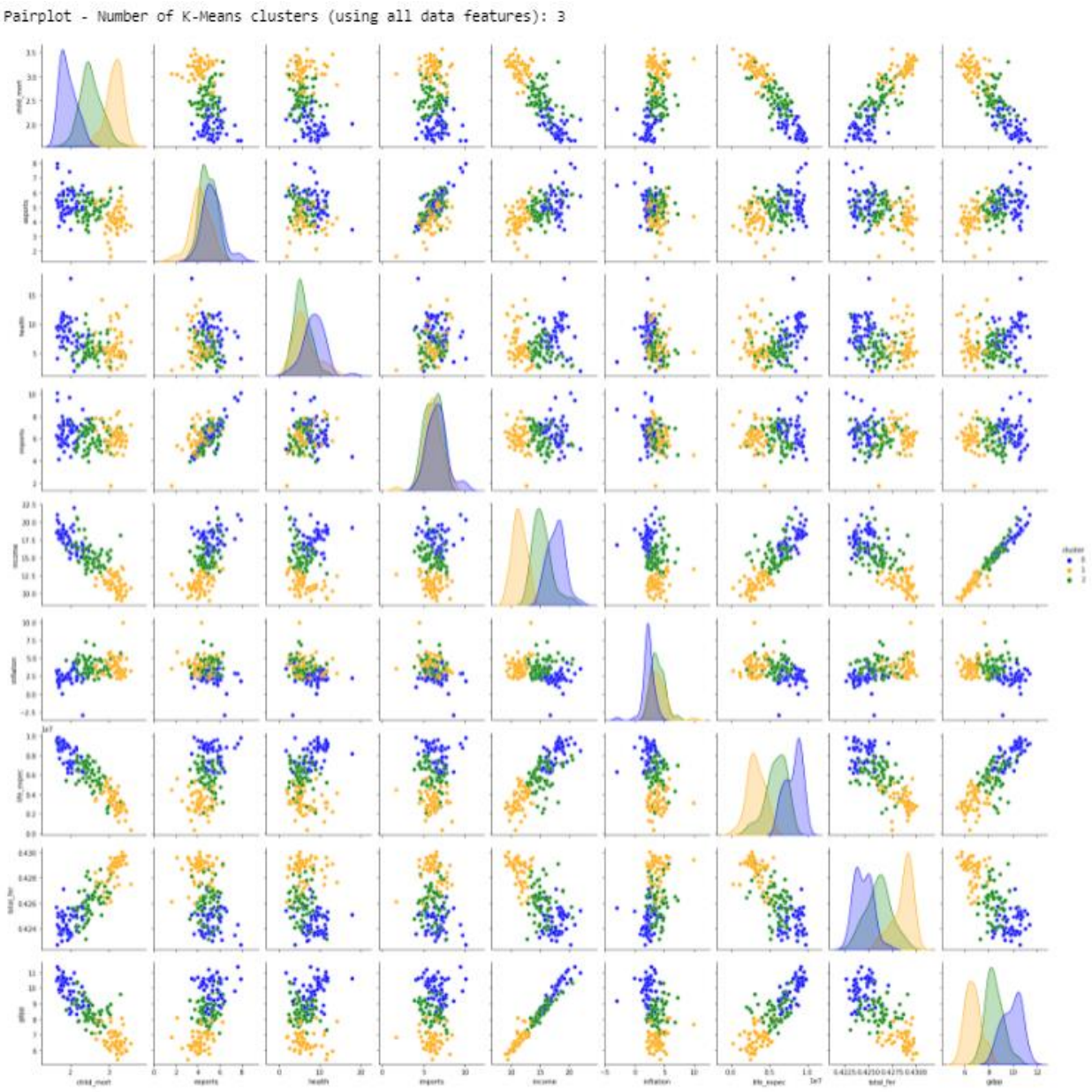

Figure 15: Pairplots – Analyzing variable distribution and pairwise correlation after K-Means clustering (K=3).

B. Principal Component Analysis

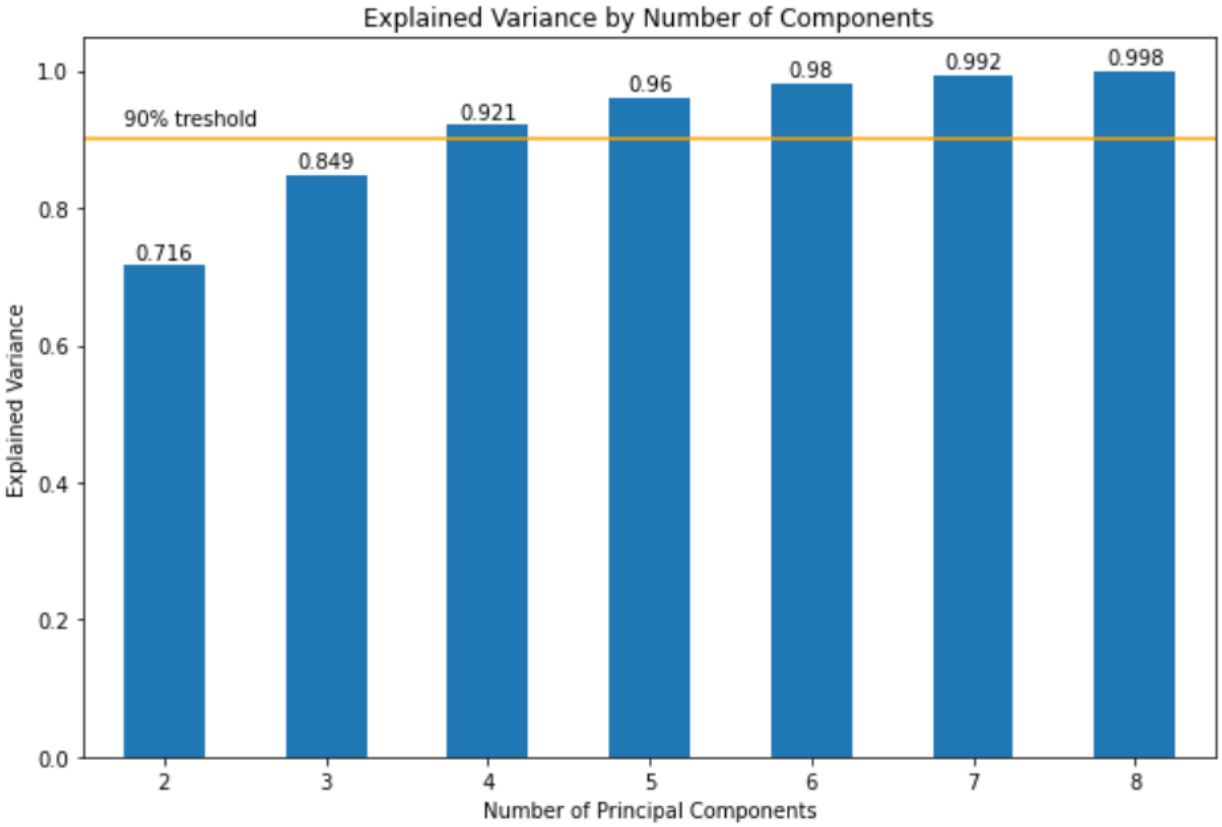

Figure 16: Barchart – Explained Variance by number of new principal components using PCA.

Reducing the data dimensionality, using PCA technique, from 9 features to 4 components, allowed me to maintain more than 90% of the original variance.

I retrained K-Means algorithm on the dimension-reduced dataset (4 components).

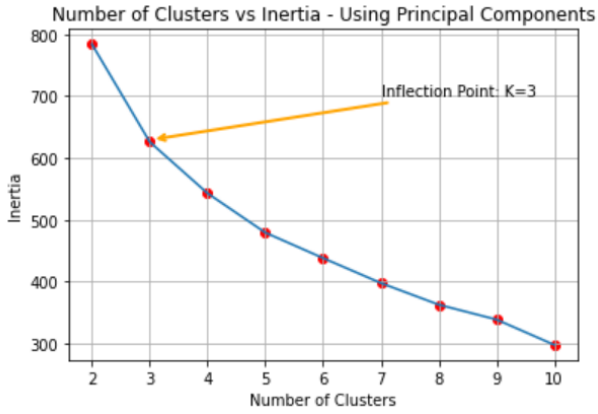

Figure 17: Analyzing the rate of Inertia decrease (using PCA)

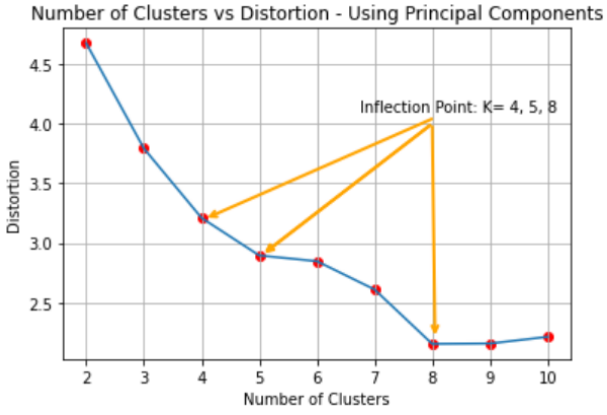

Figure 18: Analyzing the rate of Distortion decrease (using PCA)

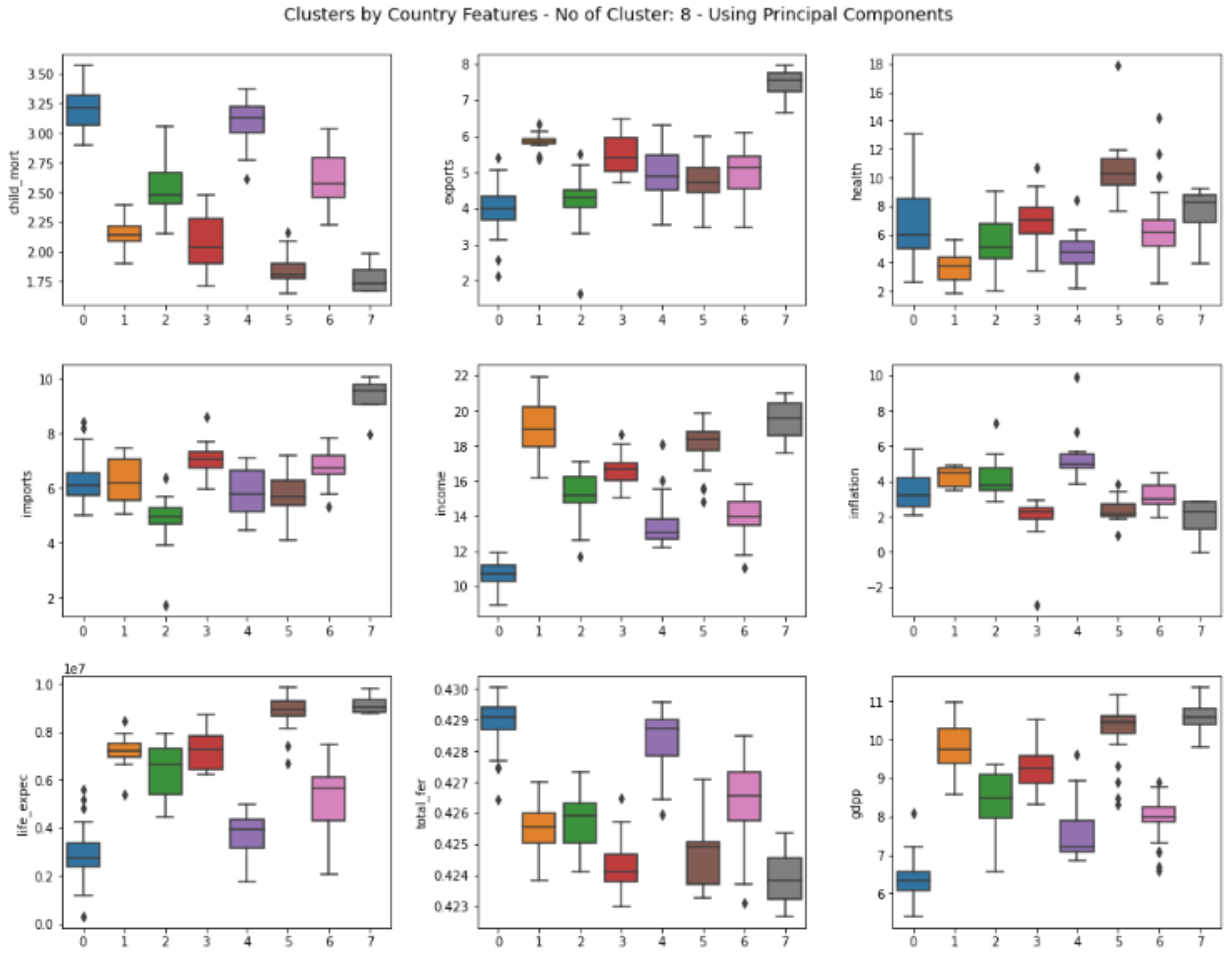

Figure 19: Boxplots – Analyzing clusters by features using 4 Principal Components (number of clusters= 8).

Clusters significantly overlap.

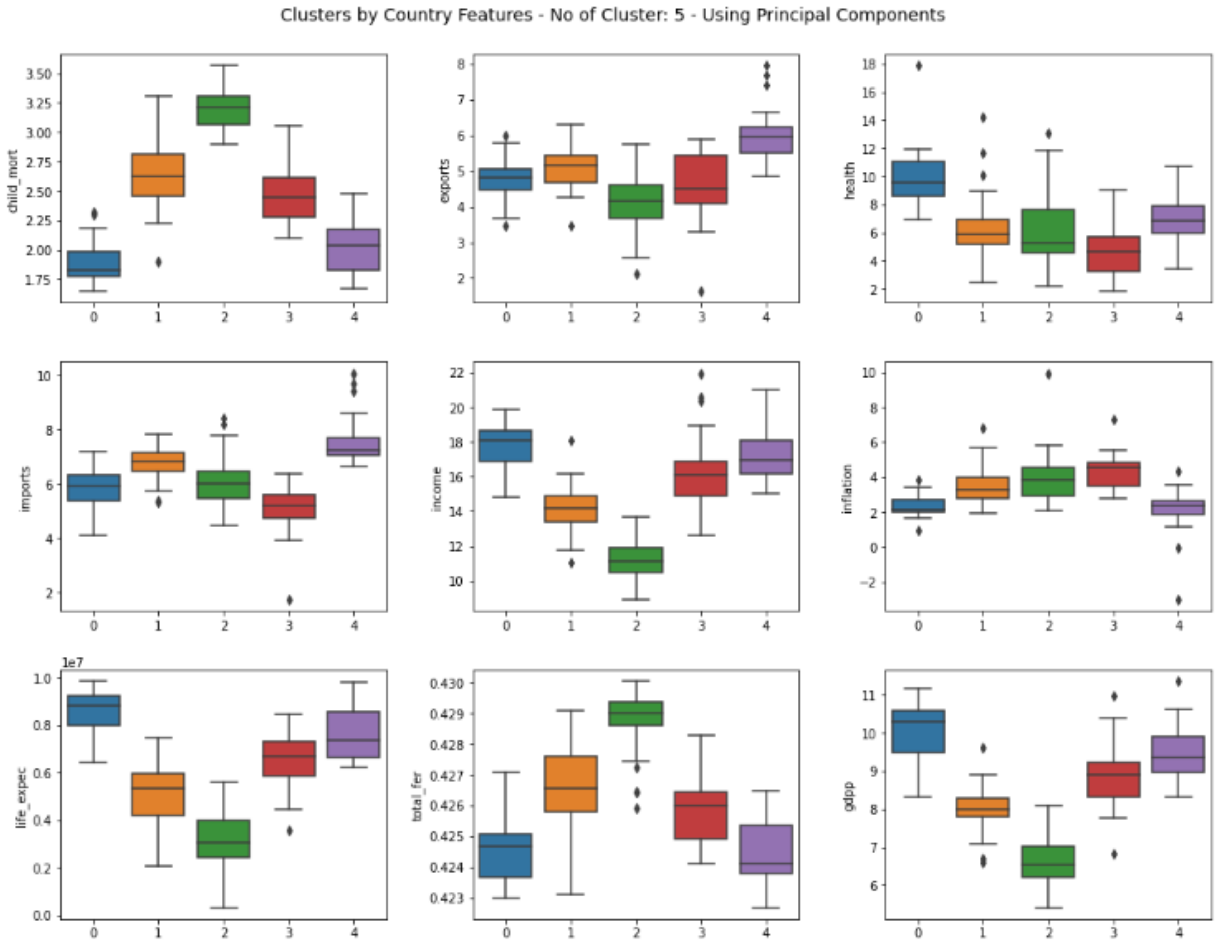

Figure 20: Boxplots – Analyzing clusters by features using 4 Principal Components (number of clusters= 5).

Clusters 0 and 4, and 1 and 3 significantly overlap.

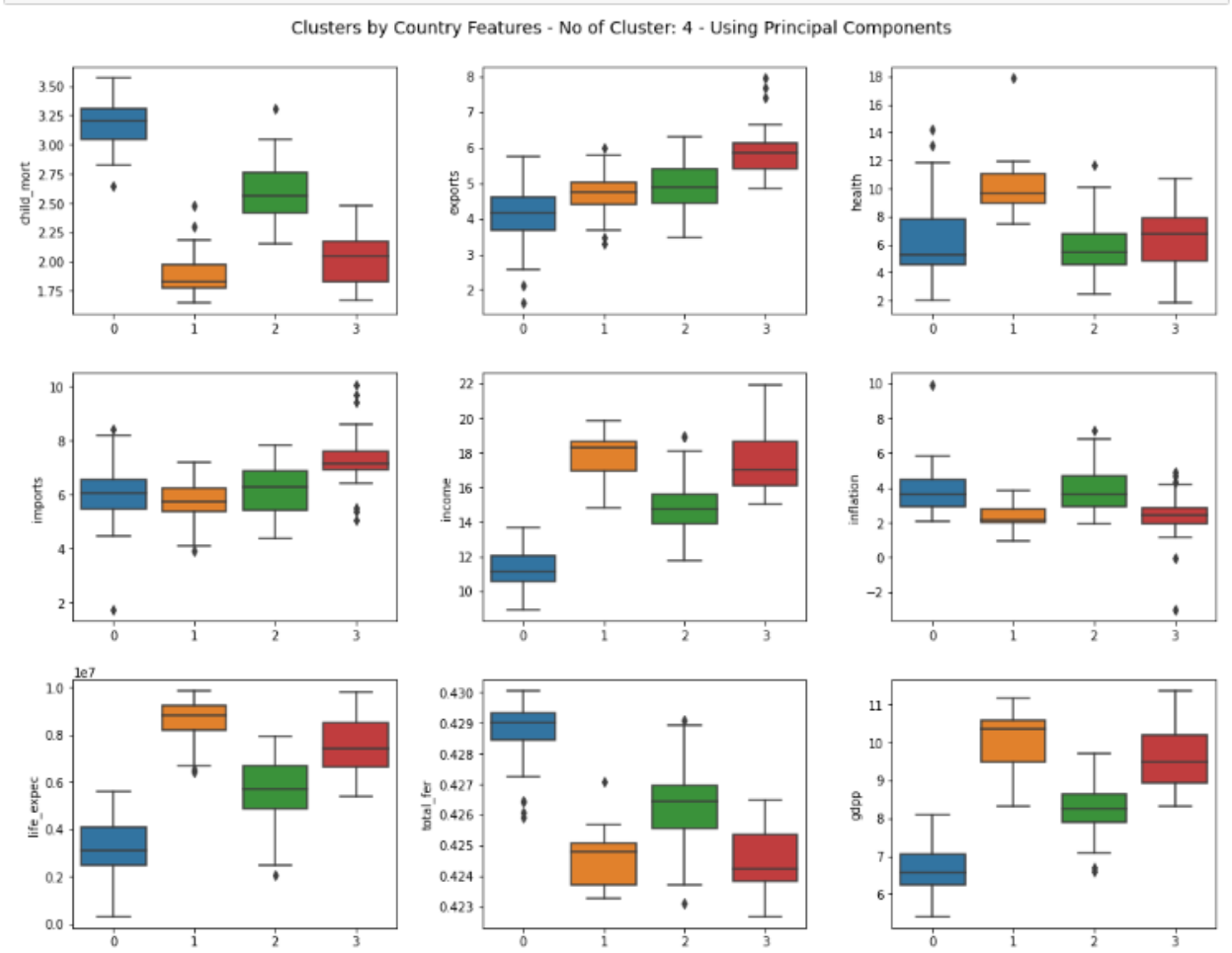

Figure 21: Boxplots – Analyzing clusters by features using 4 Principal Components (number of clusters= 4).

Clusters 1 and 3 significantly overlap.

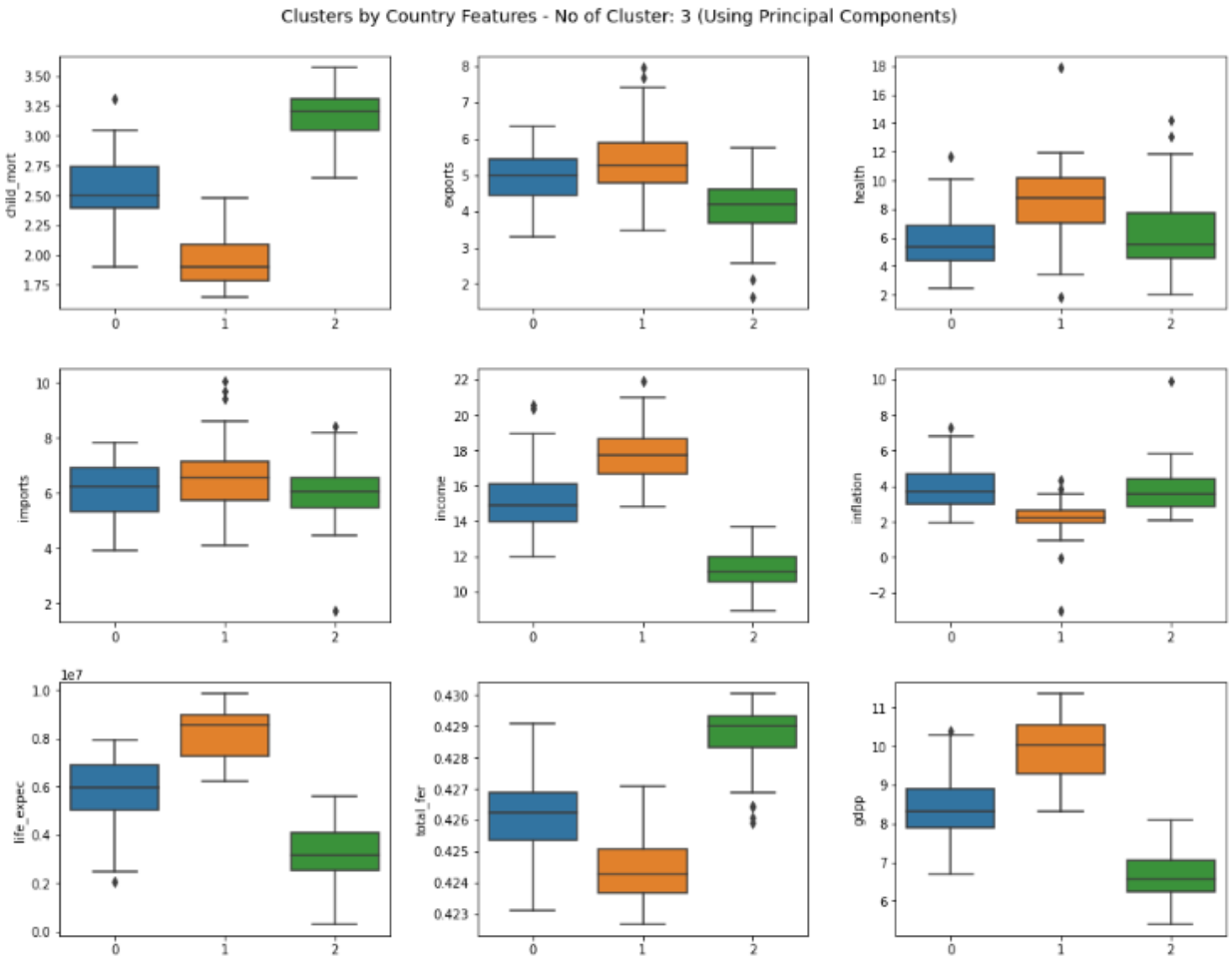

Figure 22: Boxplots – Analyzing clusters by features using 4 Principal Components (number of clusters= 3).

Even when using Principal Component Analysis technique, the greatest number of non-overlapping clusters is 3.

C. Feature Selection

In the 3rd Feature set, I didn't include the features where clusters more significantly overlap: Health, Imports, Inflation, and Exports.

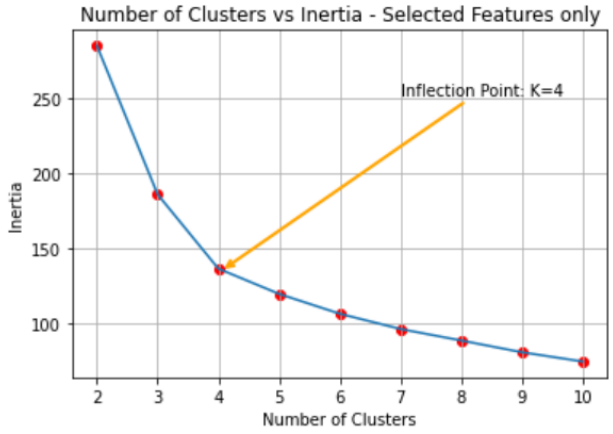

Figure 23: K vs Inertia (with selected features only)

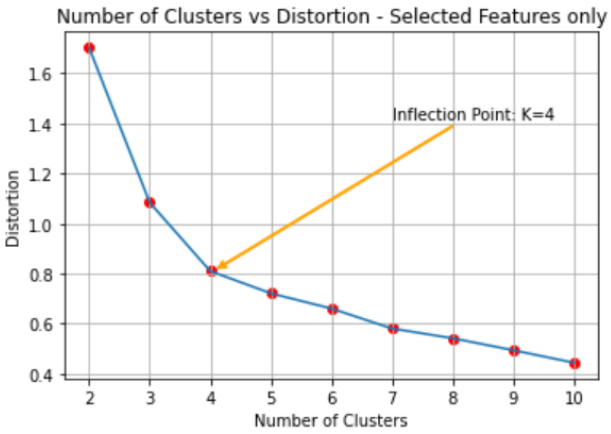

Figure 24: K vs Distortion (with selected features only)

With this feature set, the elbow point is much clearer: 4 clusters. I trained K-Means using only the selected features (child_mortality, income, life_expectancy, total_fertility, GDPP), and setting K = 4. Results are as follows:

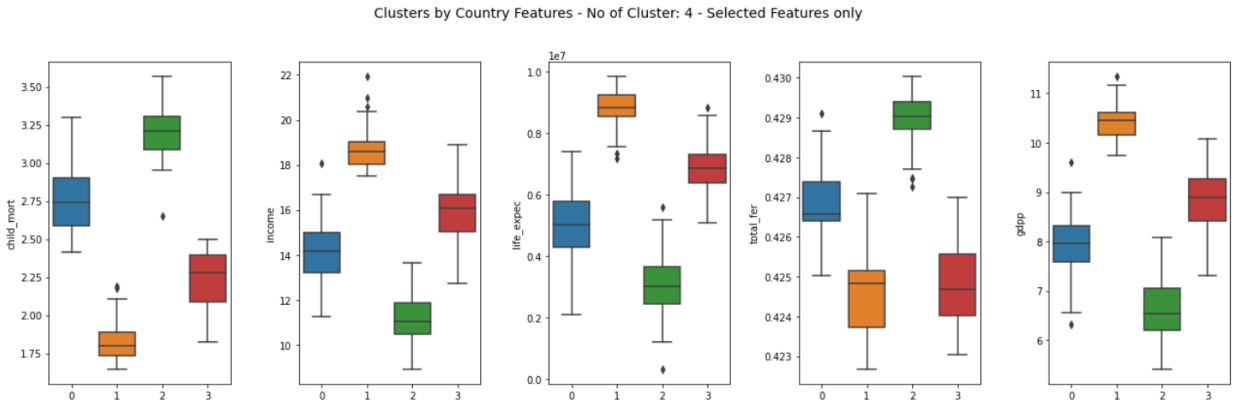

Figure 25: Boxplots – Analyzing clusters by features using only selected features (number of clusters= 4).

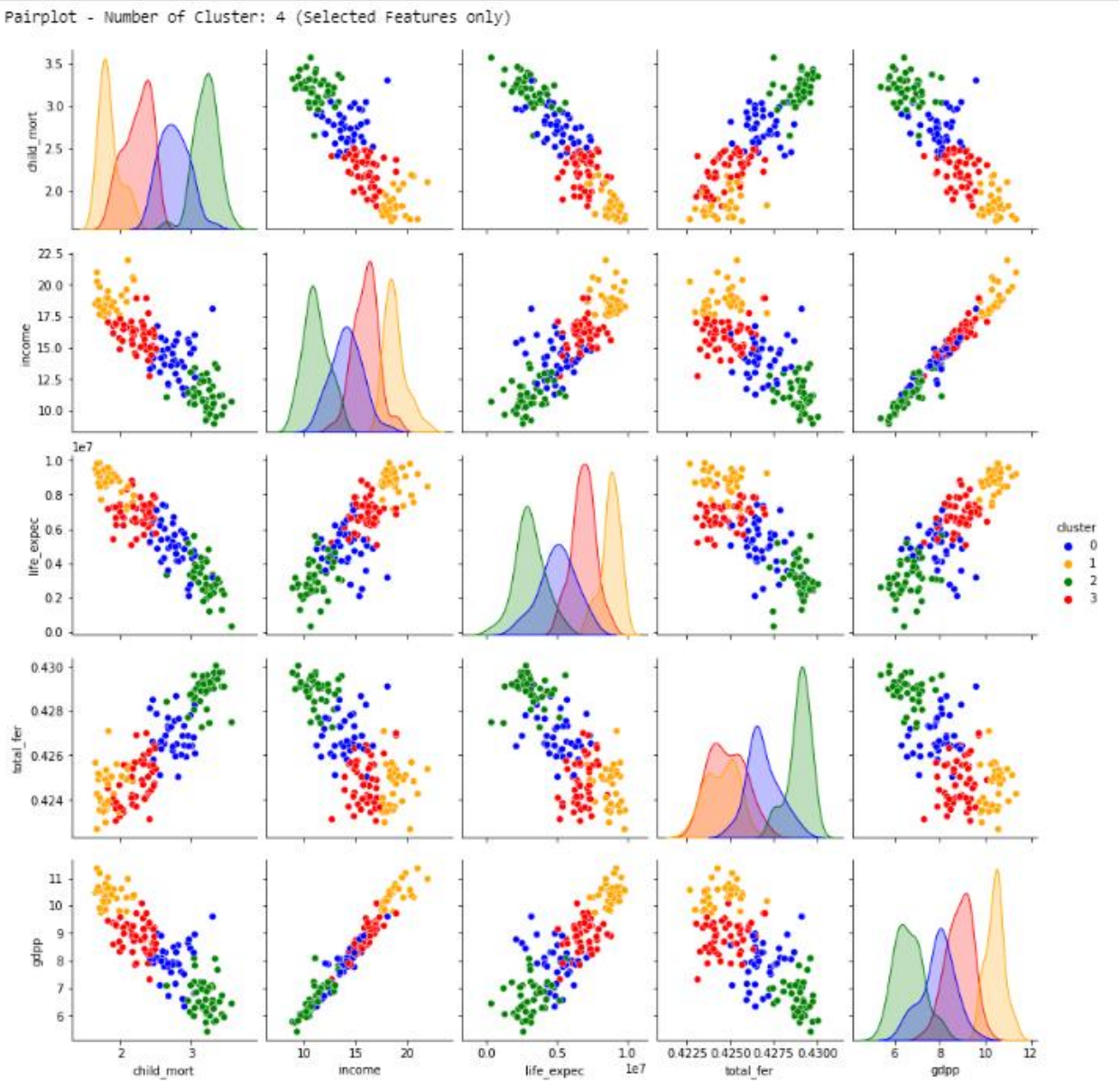

Figure 26: Pairplots – Analyzing variable distribution and pairwise correlation using only selected features to train K-Means.

Using only the selected features, **I managed to increase the number of distinct clusters from 3 to 4:** cluster 1 and 3 still overlap in total_fertility, but 4 clusters are now clearly distinguishable in all other features.

2.2.2. Hierarchical Agglomerative Clustering

I trained Hierarchical Agglomerative Clustering (HAC) on the same dataset (using only the selected features) to see if I could obtain better results, that is 4 or more than 4 non-overlapping clusters. To choose the ideal number of clusters, I analyzed the dendrogram charts looking for the greatest distance gap between clusters: the greater the distance, the more

distinct/meaningful the clusters. I trained HAC algorithm using different distance metrics as well as linkage options.

A. Distance Metric: Euclidean

Linkage: Ward - minimizing the intra-cluster inertia:

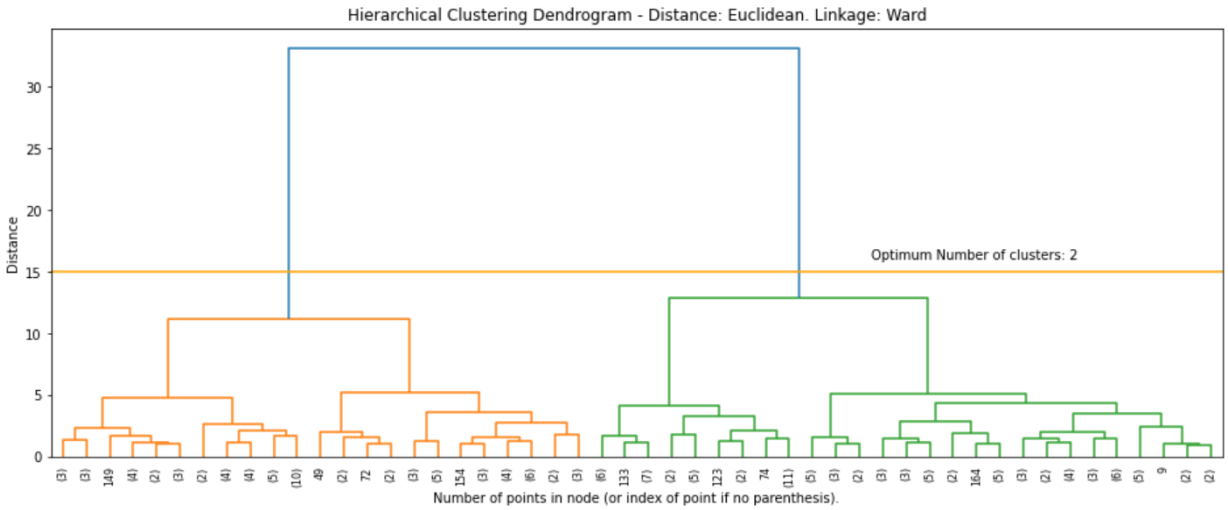

Figure 27: Dendrogram – Distance: Euclidean; Ward: Linkage. The greatest distance jump is when the number of clusters is 2.

Linkage: Single – insuring clear boundaries between clusters:

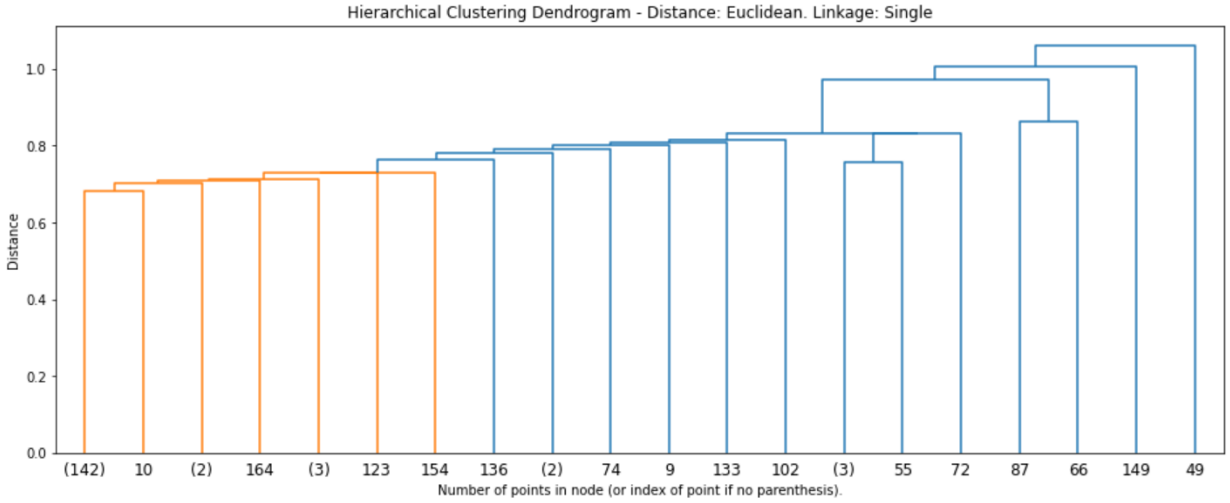

Figure 28: Dendrogram – Distance: Euclidean; Ward: Single. Very poor performance.

Linkage: Complete – reducing the impact of outliers:

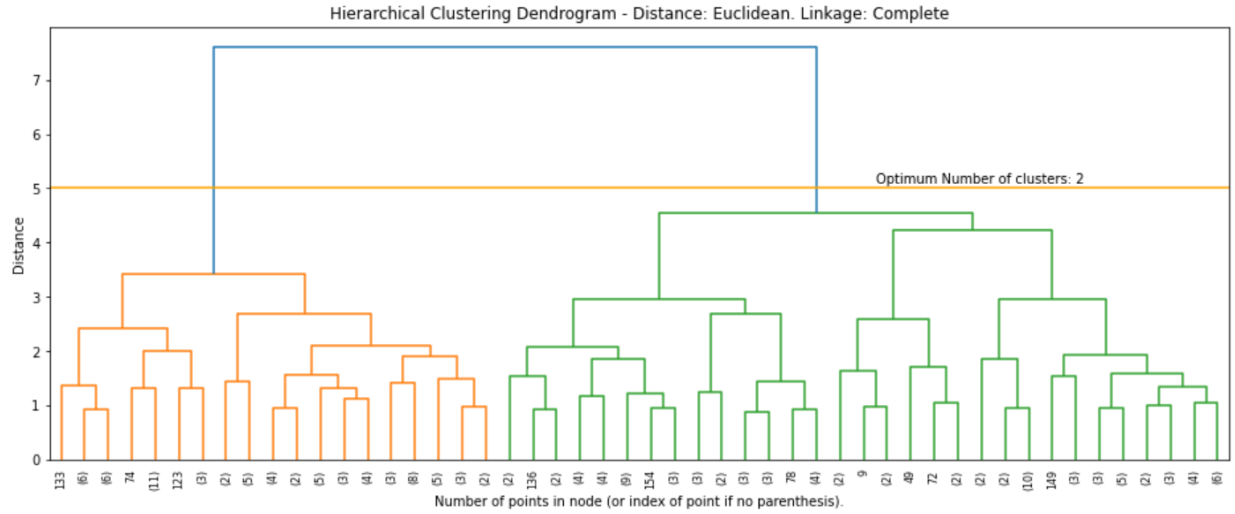

Figure 29: Dendrogram – Distance: Euclidean; Ward: Complete. The greatest distance jump is when the number of clusters is 2.

Linkage: Average – reducing the average distance between clusters:

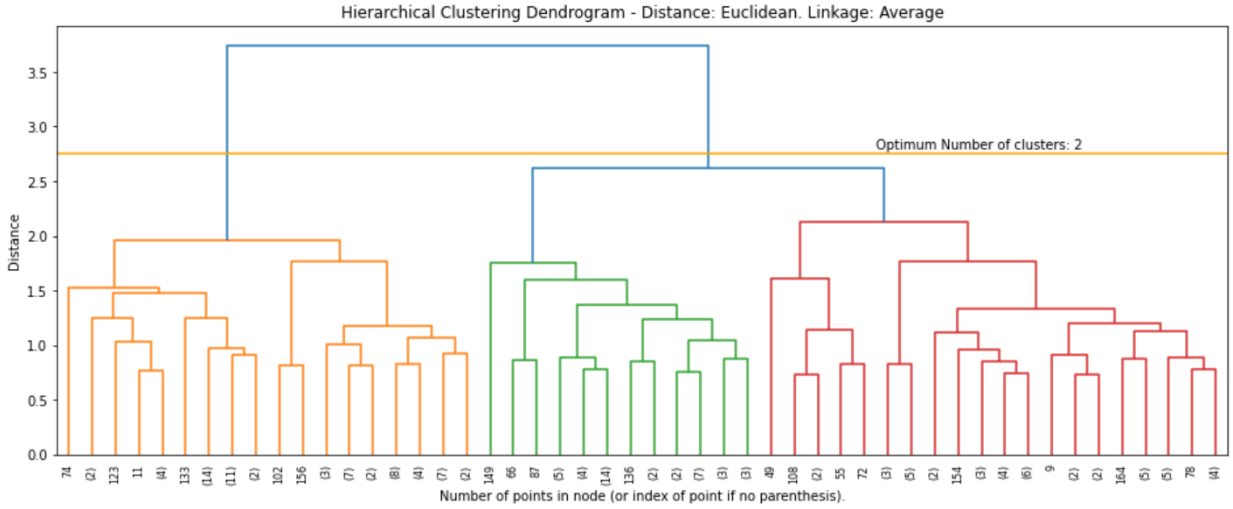

Figure 30: Dendrogram – Distance: Euclidean; Ward: Average. The greatest distance jump is when the number of clusters is 2.

B. Distance Metric: Manhattan

Linkage: single - insuring clear boundaries between clusters:

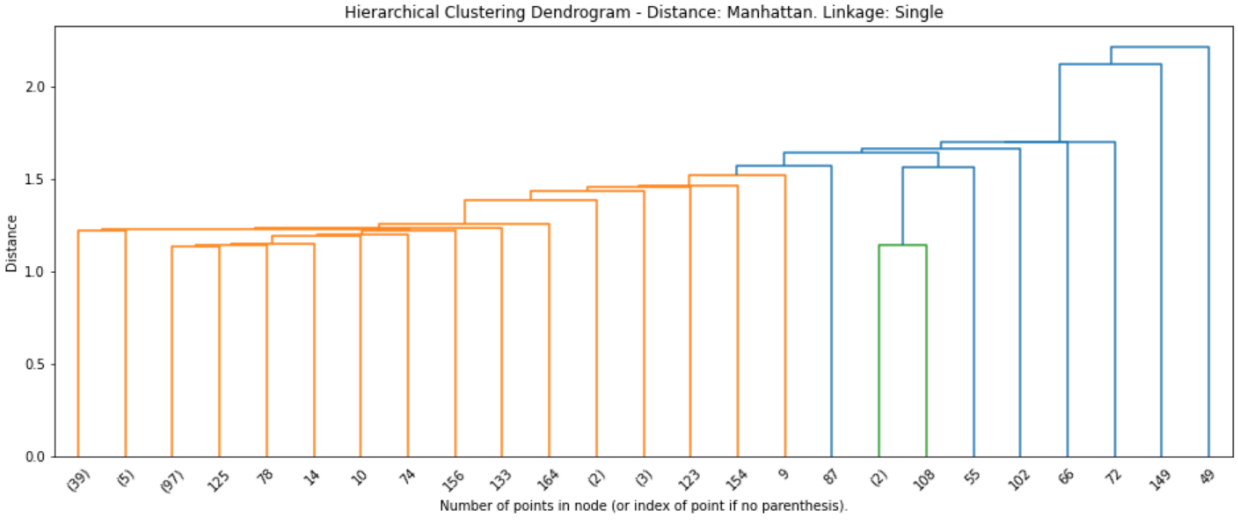

Figure 31: Dendrogram – Distance: Manhattan; Ward: Single. Very poor performance

Linkage: complete - reducing the impact of outliers:

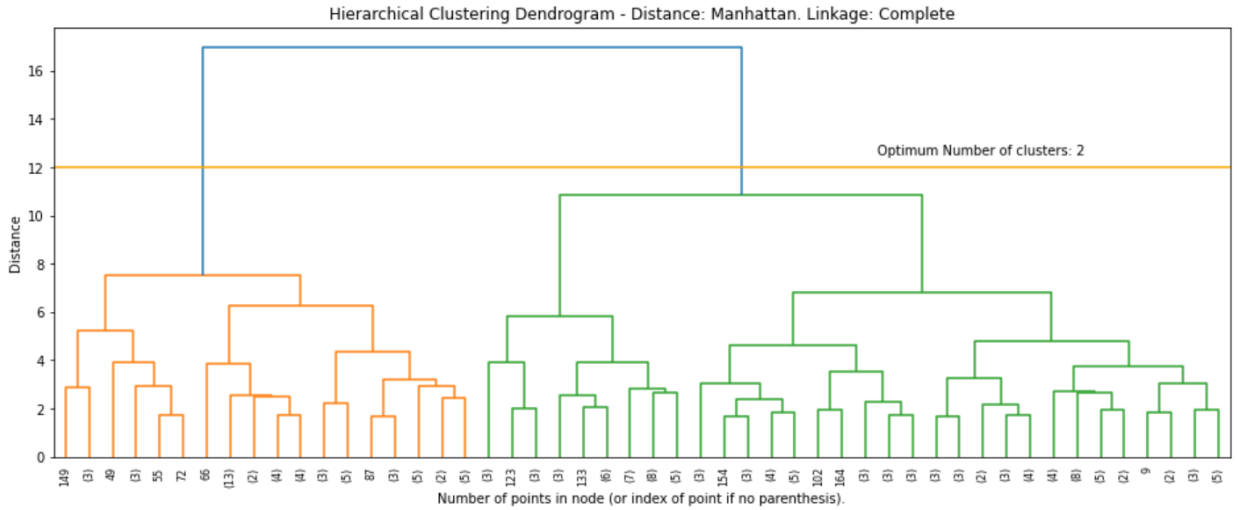

Figure 32: Dendrogram – Distance: Manhattan; Ward: Complete. The greatest distance jump is between 2 clusters.

Linkage: average - reducing the average distance between clusters

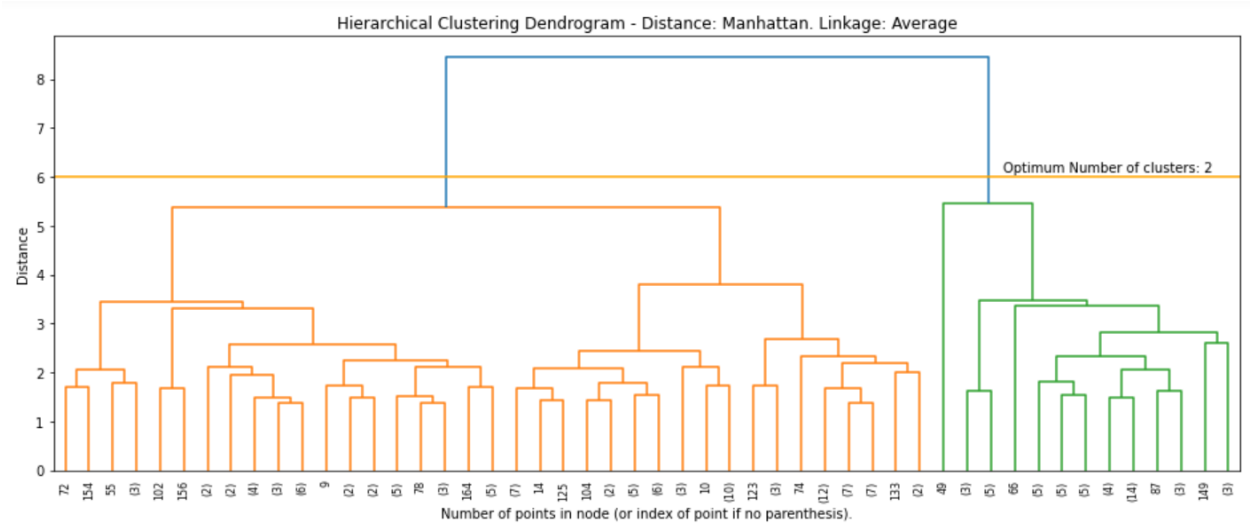

Figure 32: Dendrogram – Distance: Manhattan; Ward: Average. The greatest distance jump is when the number of clusters is 2.

Hierarchical Agglomerative Clustering failed to cluster the data in 4, or more than 4, clusters.

2.2.3. Density-Based Spatial Clustering of Application with Noise (DBSCAN)

I trained DBSCAN algorithm on the dataset containing the selected features only, to test if I can obtain better results: 4 or more than 4 not-overlapping clusters.

I trained DBSCAN using different hyper-parameter values, and recorded the highest Silhouette Coefficient per Number of Clusters.

N_Cluster	Silhouette_Coefficient
2	0.479489
3	0.405192
4	0.359300
5	0.270852
6	0.188485
7	0.127782
8	0.029490
9	0.054372
10	-0.093637

Figure 33: Table with the greatest Silhouette Coefficients per number of clusters obtained by DBSCAN.

The Silhouette Coefficient obtained by K-Means, with K = 4, is 0.3921752. DBSCAN got higher Silhouette Coefficients only when grouping the data points in 2 or 3 clusters, which is less than the 4 clusters obtained by K-Means.

3. Results.

K-Means performed better than either Hierarchical Agglomerative Clustering or DBSCAN, managing to cluster our data points into 4 distinct clusters, using a subset of the original data features selected by analyzing the cluster difference/overlap per each variable through a boxplot analysis.

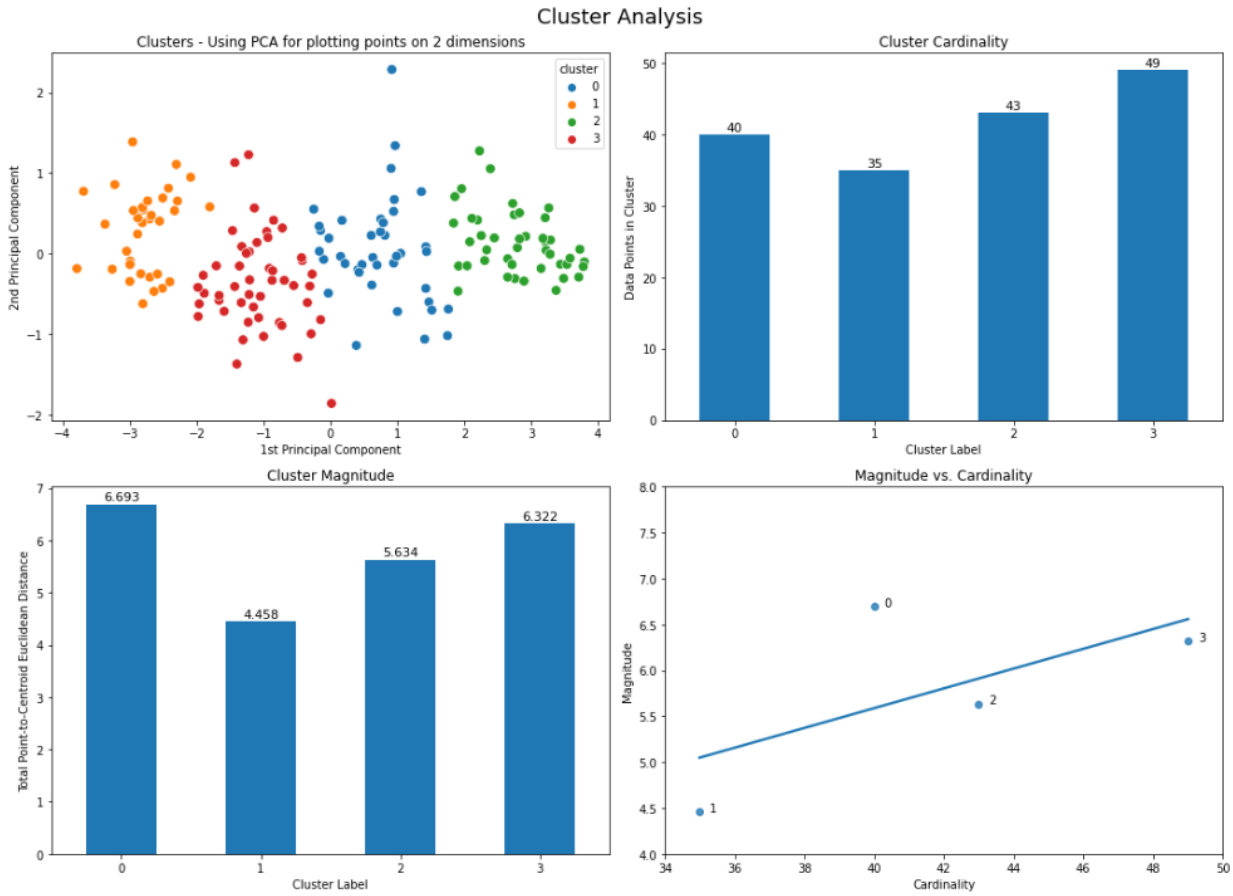

Figure 34: Analyzing the clusters by plotting data points in 2 dimensions (using PCA), and checking cluster cardinality (how many data points are in a cluster), magnitude (the sum of distances from all examples to the centroid of the cluster), and the relationships between clusters' cardinality and magnitude.

Results:

- The 4 clusters have approximately same size: there's no major outliers when it comes to the number of observations per cluster.
- Cluster 1 is the smallest in terms of both cardinality and magnitude.
- A higher cluster cardinality tends to result in a higher cluster magnitude, which intuitively makes sense: since magnitude is the sum of the distances from all examples to the cluster's

centroid, the greater the number of examples, the greater the magnitude. However, if we observe the Magnitude vs Cardinality Regression Plot, we can see that, for Cluster 0, cardinality doesn't correlate with magnitude relatively to the other clusters (it's the data point furthest away from the fitting line): Cluster 0 is the cluster with the greatest magnitude, but it's not the biggest cluster.

- Cluster 0 anomaly (in terms of Cardinality vs Magnitude) is probably due to the presence of an outlier, which increase the cluster magnitude:

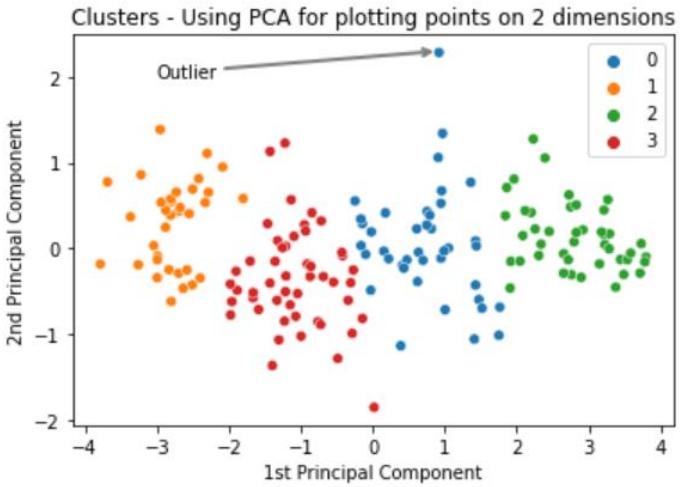

Figure 35: Analyzing the clusters by plotting data points in 2 dimensions (using PCA).

Besides Cluster 0 being slightly anomalous compared to the other clusters, we can conclude that K-Means did a good job clustering the data points in 4 distinct clusters.

4. Discussion: Clusters description

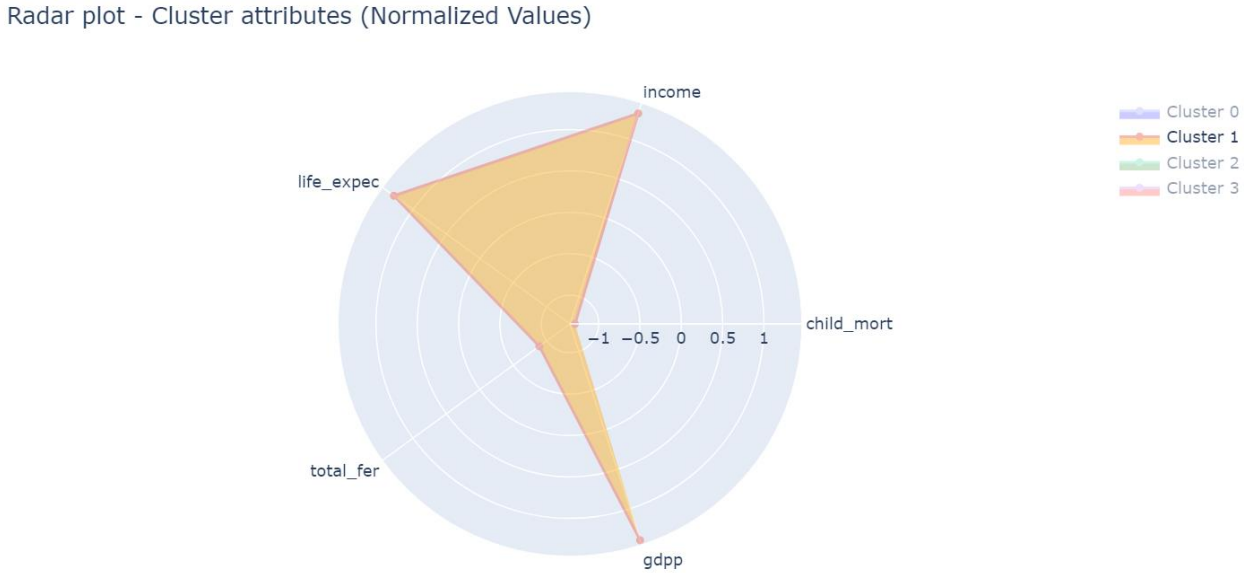

Figure 36: Radar plot - Analyzing cluster 1 characteristics.

Cluster 1: this cluster include the countries with the highest average values in Income, Life Expectancy, GDP per capita, and lowest average values in Child Mortality and Total Fertility. **Cluster 1 label: Highly Developed countries.**

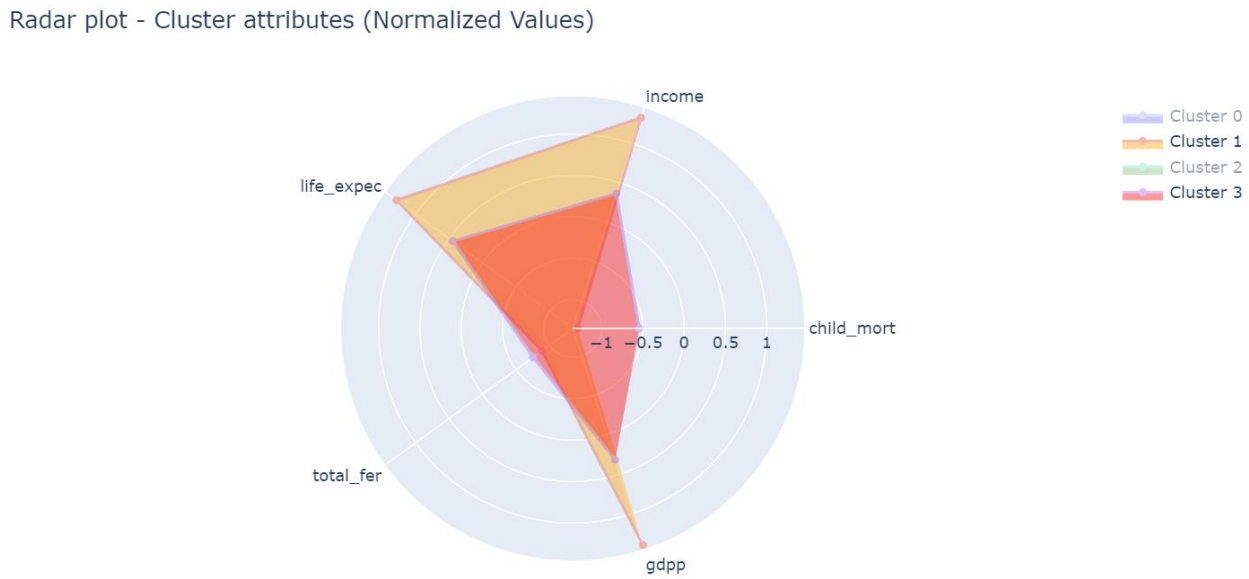

Figure 37: Radar plot - Analyzing cluster 3 characteristics.

Cluster 3: these countries have lower average values than the Highly Developed observations in Income, Life Expectancy, GDP per capita but still above the overall average. They also have higher average values than the cluster 1 countries in Child Mortality and Total Fertility, however below the dataset average. **Cluster 3 label: Upper-Middle Developed countries.**

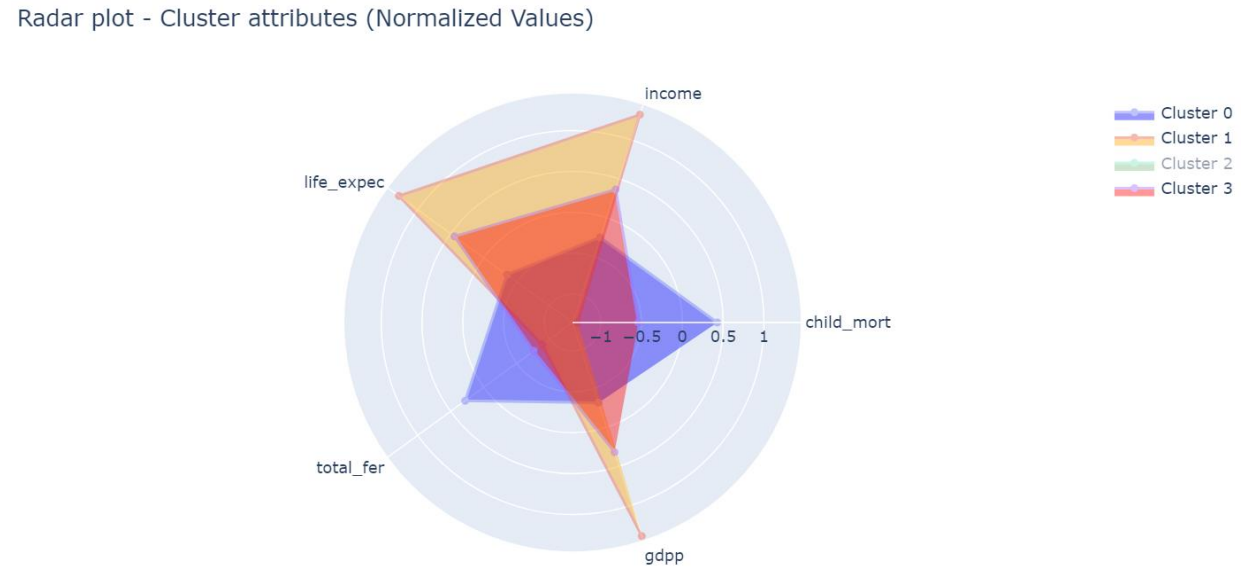

Figure 38: Radar plot - Analyzing cluster 0 characteristics.

Cluster 0: countries with Income, Life Expectancy, GDP per capita mean values below the average, and Child Mortality and Total Fertility mean values above the average. **Cluster 0 label: Lower-Middle Developed countries.**

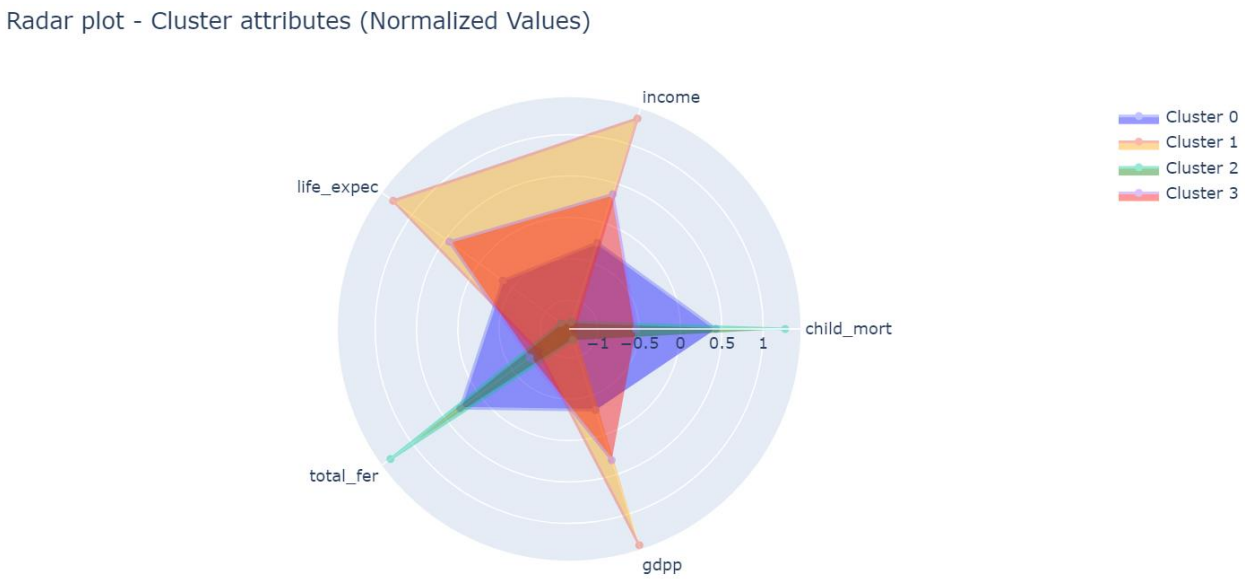

Figure 39: Radar plot - Analyzing cluster 2 characteristics.

Cluster 2: countries with, by far, the highest average values in Child Mortality and Total Fertility as well as the smallest average values in Income, Life Expectancy, and GDP per capita. **Cluster 2 label: Least Developed countries.**

cluster	0	1	2	3
label	Lower-middle developed	Highly developed	Least developed	Upper-middle developed
child_mort	37.902500	4.942857	94.788372	12.777551
exports	39.694725	58.237143	26.311860	43.014286
health	5.689750	8.808286	6.380930	6.693061
imports	45.506647	50.740000	43.667442	48.097959
income	8499.000000	46257.142857	2177.279070	16542.448980
inflation	8.976025	2.733714	11.797791	6.888571
life_expec	68.632500	80.260000	59.351163	75.026531
total_fer	2.968500	1.762000	5.121163	1.871224
gdpp	3920.350000	43234.285714	1004.883721	9220.204082

Figure 40: Table with clusters' mean values per each feature.

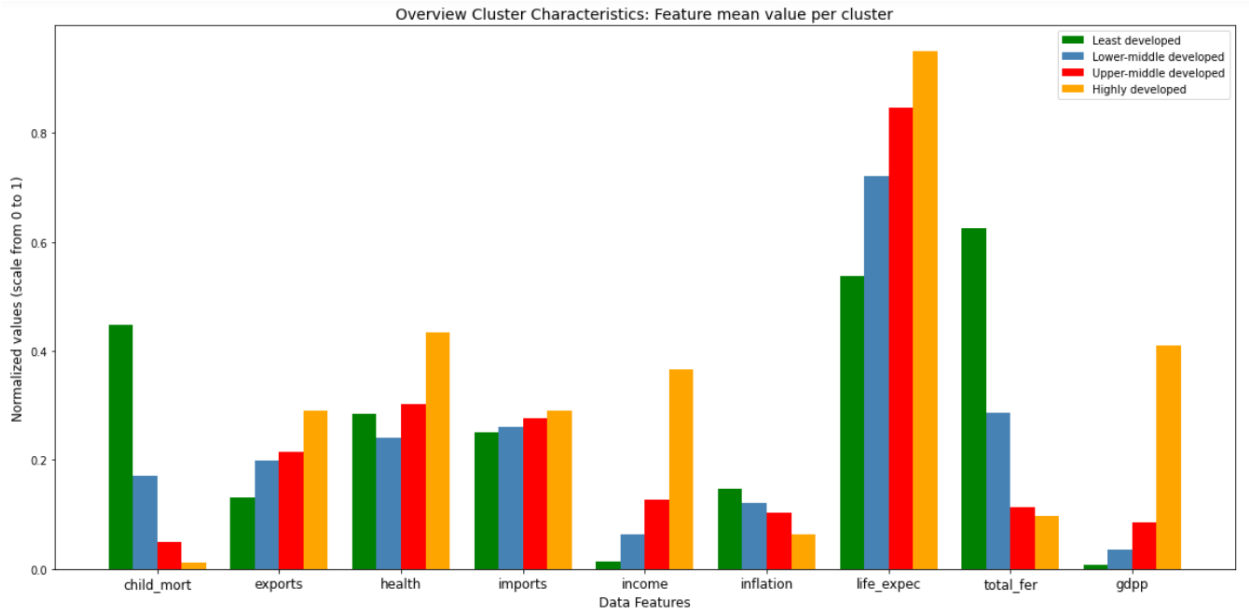

Figure 41: Clustered Barchart – analyzing clusters' average values per each feature.

Main insights:

- It's interesting to note that the found clusters help find patterns also in those features that haven't been selected to train the clustering algorithm, such as Exports, Imports, and Inflation.
- The more developed the countries, the higher the percentage of both Imports and Exports on the GDP per capita
- The Highly Developed country group is the only one with an average Export value greater than the Import value, suggesting that, besides single country differences and specific characteristics, well developed economies tend to have a positive trade balance. It'd be interesting to perform Hypothesis Testing Analysis to assess whether the difference between Highly Developed countries' trade balance and the population average is statistically significant or not.
- The more developed the countries the lesser the level of Inflation.
- Health (total health-related spending per capita, given as % of GDP per capita) is the only feature that doesn't seem to have neither an explanatory value on the clustering, nor being explained by the clusters.
- For some features, the gap between one cluster and the others is particularly pronounced.
- The Least Developed Countries have a very high level of Child Mortality, compared to all other clusters: 95 children (under 5 years of age) per 1,000 live births vs 38 children per 1,000 live births as population average, as well as a very high level for Total Fertility: 5.12 new born per woman vs 2.95 born children per woman as population average.
- The above point suggests a very important action point NGOs could undertake to help reduce the gap between the Least Developed countries and the others: reduce the level of Child Mortality will help reduce the level of Total Fertility as well (the features are highly

correlated), and decrease the distance between this cluster and the other groups on the development scale.

- Highly Developed countries' GDPP average values are much greater compared to all other clusters: 43,234 USD vs 12,964 USD (GDPP general average); Highly Developed countries' Income mean value is 46,257 USD per person vs 17,145 USD per person (population average).
- This confirms the picture of a highly polarized world, where wellness and high-standard of living are not equally distributed but still a privilege of the few. NGOs should focus their efforts helping the Least Developed countries reduce their child mortality rate, and support policies that tend to foster economic growth, which will result in an increase of the GDPP as well as Income rates:
    - reduce trade deficit
    - control the level of inflation
    - improve education system and local infrastructure
    - increase real wages (whilst keeping inflation under control).

5. Conclusion.

5.1. Project Summary.

In this project I trained 3 unsupervised Machine Learning Clustering models (K-Means, Hierarchical Agglomerative Clustering, and DBSCAN) in order to correctly classify world countries into clusters, and assess their overall development status.

Scope of the project is to help NGOs use their funds strategically and effectively, in order to support the countries at bottom of the development curve.

Project stages:

- The dataset has been cleaned and analyzed: 8 out of 9 features were not normally distributed. I applied BoxCox transformation, to enforce Normality on the data distribution, as well as strengthening the linear correlation between the features. To make it ready for modeling, after transformation, the data has been normalized using Z-Score normalization.
- I first trained K-Means algorithm, looking for the best number of clusters, and compared performance of the model on different feature sets:
    - Using all dataset features
    - Performed PCA to combine and reduce the number of variables
    - Performed feature selection based on an empirical analysis of each feature in relation to the found clusters.
- Feature Selection was the technique which performed better: K-Means managed to cluster the data points in 4 distinct meaningful groups.
- I then trained other 2 clustering algorithms, using only the selected features:

- Hierarchical Agglomerative Clustering
- DBSCAN

Modeling outputs: K-Means performed better than either Agglomerative Hierarchical Clustering or DBSCAN, managing to cluster the data points in 4 distinct clusters, using a subset of the original data features selected by analyzing the cluster difference/overlap per each variable through boxplot analysis.

5.2. Outcome of the Analysis.

K-Means clustered the 167 countries included in the dataset into the following categories:

- **Cluster 1: 35 highly developed countries:**

Australia, Austria, Bahrain, Belgium, Brunei, Canada, Cyprus, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Israel, Italy, Japan, Kuwait, Luxembourg, Malta, Netherlands, New Zealand, Norway, Portugal, Qatar, Singapore, Slovenia, South Korea, Spain, Sweden, Switzerland, United Arab Emirates, United Kingdom, United States.

- **Cluster 3: 49 upper-middle developed countries:**

Albania, Antigua and Barbuda, Argentina, Armenia, Bahamas, Barbados, Belarus, Bosnia and Herzegovina, Brazil, Bulgaria, Chile, China, Colombia, Costa Rica, Croatia, Estonia, Georgia, Grenada, Hungary, Iran, Jamaica, Latvia, Lebanon, Libya, Lithuania, Macedonia, FYR, Malaysia, Maldives, Mauritius, Moldova, Montenegro, Oman, Panama, Peru, Poland, Romania, Russia, Saudi Arabia, Serbia, Seychelles, Slovak Republic, Sri Lanka, St. Vincent and the Grenadines, Thailand, Tunisia, Turkey, Ukraine, Uruguay, Venezuela.

- **Cluster 0: 40 lower-middle developed countries:**

Algeria, Azerbaijan, Bangladesh, Belize, Bhutan, Bolivia, Botswana, Cambodia, Cape Verde, Dominican Republic, Ecuador, Egypt, El Salvador, Equatorial Guinea, Fiji, Gabon, Guatemala, Guyana, India, Indonesia, Iraq, Jordan, Kazakhstan, Kyrgyz Republic, Micronesia Fed. Sts., Mongolia, Morocco, Myanmar, Namibia, Nepal, Paraguay, Philippines, Samoa, South Africa, Suriname, Tonga, Turkmenistan, Uzbekistan, Vanuatu, Vietnam.

- **Cluster 2: 43 least developed countries (the countries at the bottom of the development curve):**

Afghanistan, Angola, Benin, Burkina Faso, Burundi, Cameroon, Central African Republic, Chad, Comoros, Congo, Dem. Rep., Congo, Rep., "Cote dIvoire", Eritrea, Gambia, Ghana, Guinea, Guinea-Bissau, Haiti, Kenya, Kiribati, Lao, Lesotho, Liberia, Madagascar, Malawi, Mali, Mauritania, Mozambique, Niger, Nigeria, Pakistan, Rwanda, Senegal, Sierra Leone, Solomon Islands, Sudan, Tajikistan, Tanzania, Timor-Leste, Togo, Uganda, Yemen, Zambia.

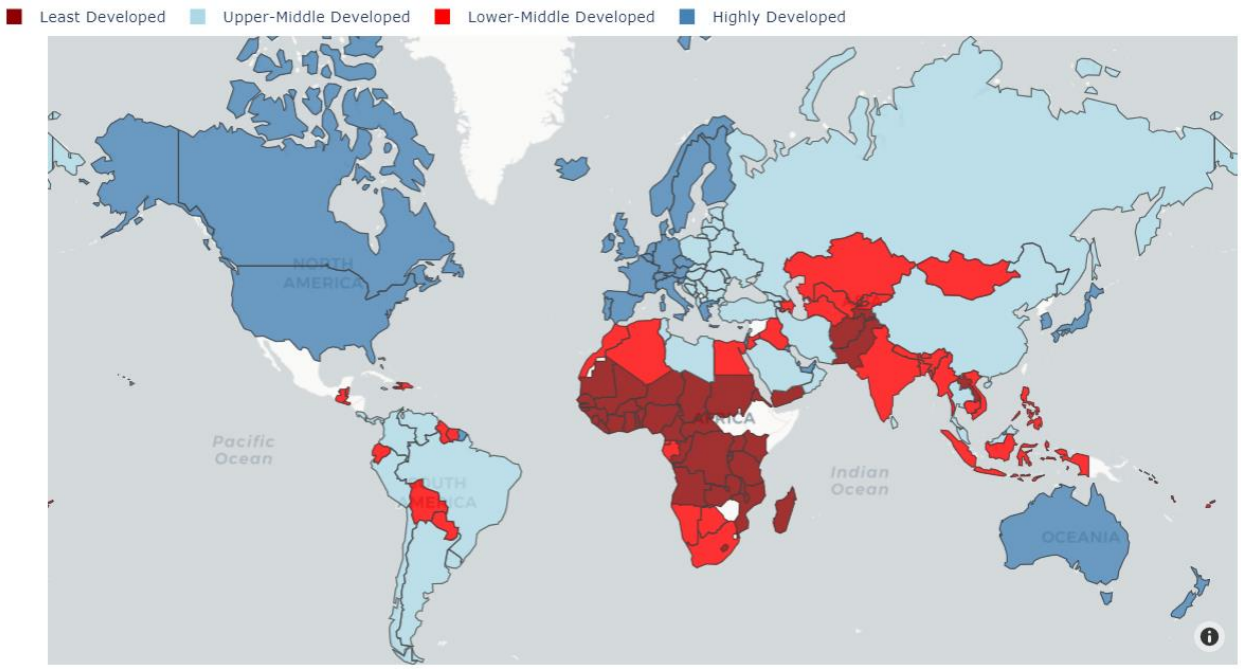

Figure 42: World map with Country Clusters.

Note: the below countries were excluded from this analysis because not part of the dataset: *Cuba, Djibouti, Ethiopia, Eswatini, Falkland Is., Gaza Strip, Greenland, Honduras, Kososvo, Mexico, Nicaragua, North Korea, Palestine, New Caledonia, Papua New Guinea, Perto Rico, Somalia, Somaliland, South Sudan, Syria, Taiwan, Trinidad and Tobago, West Bank, West Sahara, Zimbabwe.*

5.3. Potential Developments.

1. This analysis is limited to the countries included in the dataset. It'd be interesting to include in this analysis those countries that weren't included in the dataset.
2. As observed, the "Highly Developed Countries" cluster is the only group with a mean positive trade balance. Since Imports and Exports were not included in the selected features to train the model, it'd be interesting to perform Hypothesis Testing Analysis to assess whether the difference between Highly Developed countries' trade balance and the population average is statistically significant or not.
3. Include more significant features (Industrialization rate, Freedom rate, Political Stability index…) could help cluster the countries in more distinct groups, providing more insights into the data.
4. Record countries data and perform this type of analysis on a periodical basis will be very useful in order to record countries' change of status over time, and what drove that change.

6. Appendix.

Lambda values used for the BoxCox transformation:

```
child_mort   -0.163454
exports       0.142427
imports       0.232995
income        0.097441
inflation     0.288592
life_expec    3.911049
total_fer    -2.317588
gdpp         -0.003218
```

Link to the Notebook: GitHub.

Link to the Notebook: Jupiter nbviewer