World Bank Open Data Analysis

Michał Grela, Mateusz Kocot, Maciej Trątnowiecki

May 2022

Contents

1 Introduction **2**

2 Dataset description **2**

3 Autoencoders **6**
3.1 Data preprocessing 6
3.2 Experiments 6
3.3 Final autoencoder model 8
3.4 Analysis of single time series types 10
3.5 Analysis of time series groups 11

4 Hierarchical clustering **13**
4.1 Analysis of separated indicators 14
4.2 Analysis of groups of indicators 20

5 Similarity metrics **30**
5.1 Metrics of interest 30
5.2 Methodology 31
5.3 Analysis results 33

6 Summary **39**

Appendix A Architecture of the autoencoder used for feature extraction **40**

Appendix B Time series previewed in on the World Bank Open Data website on which we base the analysis in Section 3 **41**

References **42**

1 Introduction

We present the analysis of selected time series from World Bank Open Data. The World Bank processes and generates large amounts of time series data. These data are public due to the Open Data Initiative. They can be browsed on the World Bank website [1] and downloaded in various formats. However, the most convenient way to analyze them offline is using the V2 Indicators API.

In our project we used the Indicators API to download time series from selected categories: agriculture, health and economy. We describe our dataset in-depth in section 2. We employed various methods to analyze the data. At first, we used an autoencoder neural network to extract time-independent features. This approach is described in 3. Then, we used the Ward's method to perform the hierarchical clustering, which is analyzed in section 4. Finally, in section 5 we describe using similarity matrices to compare the data. We conclude the project in section 6.

In each of the sections centered on a particular method, apart from defining the method, we also analyze obtained data by plotting them in a 2D space using manifold learning algorithms like t-SNE or UMAP.

2 Dataset description

We divided the downloaded time series into four groups of indicators as follows:

1. Selected indicators
    - Population growth (annual %)
    - Inflation, consumer prices (annual %)
    - Life expectancy at birth, total (years)
    - Exports of goods and services (% of GDP)
    - GDP growth (annual %)
    - Unemployment, total (% of total labor force) (modeled ILO estimate)
    - Agriculture, forestry, and fishing, value added (% of GDP)
    - Access to electricity (% of population)
    - Forest area (% of land area)
    - Mortality rate, under-5 (per 1,000 live births)
    - Total natural resources rents (% of GDP)
    - Fertility rate, total (births per woman)
    - Population in the largest city (% of urban population)
    - Merchandise trade (% of GDP)
    - Military expenditure (% of GDP)
2. Agriculture indicators
    - Arable land (% of land area)
    - Cereal yield (kg per hectare)

- Employment in agriculture, female (% of female employment) (modeled ILO estimate)
- Fertilizer consumption (kilograms per hectare of arable land)
- Forest area (% of land area)
- Livestock production index (2014-2016 = 100)
- Agricultural land (% of land area)
- Agriculture, forestry, and fishing, value added (% of GDP)
- Arable land (hectares per person)
- Crop production index (2014-2016 = 100)
- Employment in agriculture, male (% of male employment) (modeled ILO estimate)
- Food production index (2014-2016 = 100)
- Permanent cropland (% of land area)
- Rural population (% of total population)

3. Health indicators

    - Life expectancy at birth, total (years)
    - Mortality rate, under-5 (per 1,000 live births)
    - Fertility rate, total (births per woman)
    - Prevalence of undernourishment (% of population)
    - Immunization, DPT (% of children ages 12-23 months)
    - Population growth (annual
    - Age dependency ration (% of working-age population)
    - Incidence of tuberculosis (per 100,000 people)
    - Immunization, measles (% of children ages 12-23 months)
    - Adolescent fertility rate (births per 1,000 women ages 15-19)
    - Death rate, crude (per 1,000 people)
    - Birth rate, crude (per 1,000 people)

4. Economy indicators

    - GNI per capita, Atlas method (current US$)
    - Gross capital formation (% of GDP)
    - Imports of goods and services (% of GDP)
    - Gross savings (% of GDP)
    - Industry (including construction), value added (% of GDP)
    - Inflation, GDP deflator (annual %)
    - Inflation, consumer prices (annual %)
    - Medium and high-tech manufacturing value added (% manufacturing value added)
    - Agriculture, forestry, and fishing, value added (% of GDP)
    - Exports of goods and services (% of GDP)
    - GDP per capita (current US$)
    - GDP growth (annual %)

The indicators and the analyzed period were selected in such a way that there were as few missing data as possible. Time series were analyzed in the period from 2000 to 2018 inclusive.

We had to cut them after 2018 since many time series lacked recent data. We accepted the countries where the number of missing observations for each of the analyzed time series in a given group was maximum two. Often, when one of the observations was missing, it was so that the first, last or both of theme were missing. So to get rid of the missing values both back fill and front fill were performed. Due to our adopted method, in each of the analyzed groups of indicators, there was a different number of countries. In this way, we obtained data for 95 countries in the selected indicators group, 139 countries in the agriculture indicators group, 154 countries in the health indicators group, 104 countries in the economy indicators group and 79 countries in the group containing all indicators. Some of the countries appeared in each of the group. We also removed country groups such as: Arab World, Low income, World from dataset, as they were not subject to our analysis.

The examined dataset contains time series data of different types and shapes. Most of the values are described by percentages, which usually are skewed into lower or higher values, which can be shown on a mean values histogram. Another group of indicators popular among our dataset describe incremental values. Examples of different kinds of time series data, along with a simple statistical analysis, are shown in Figure 1.

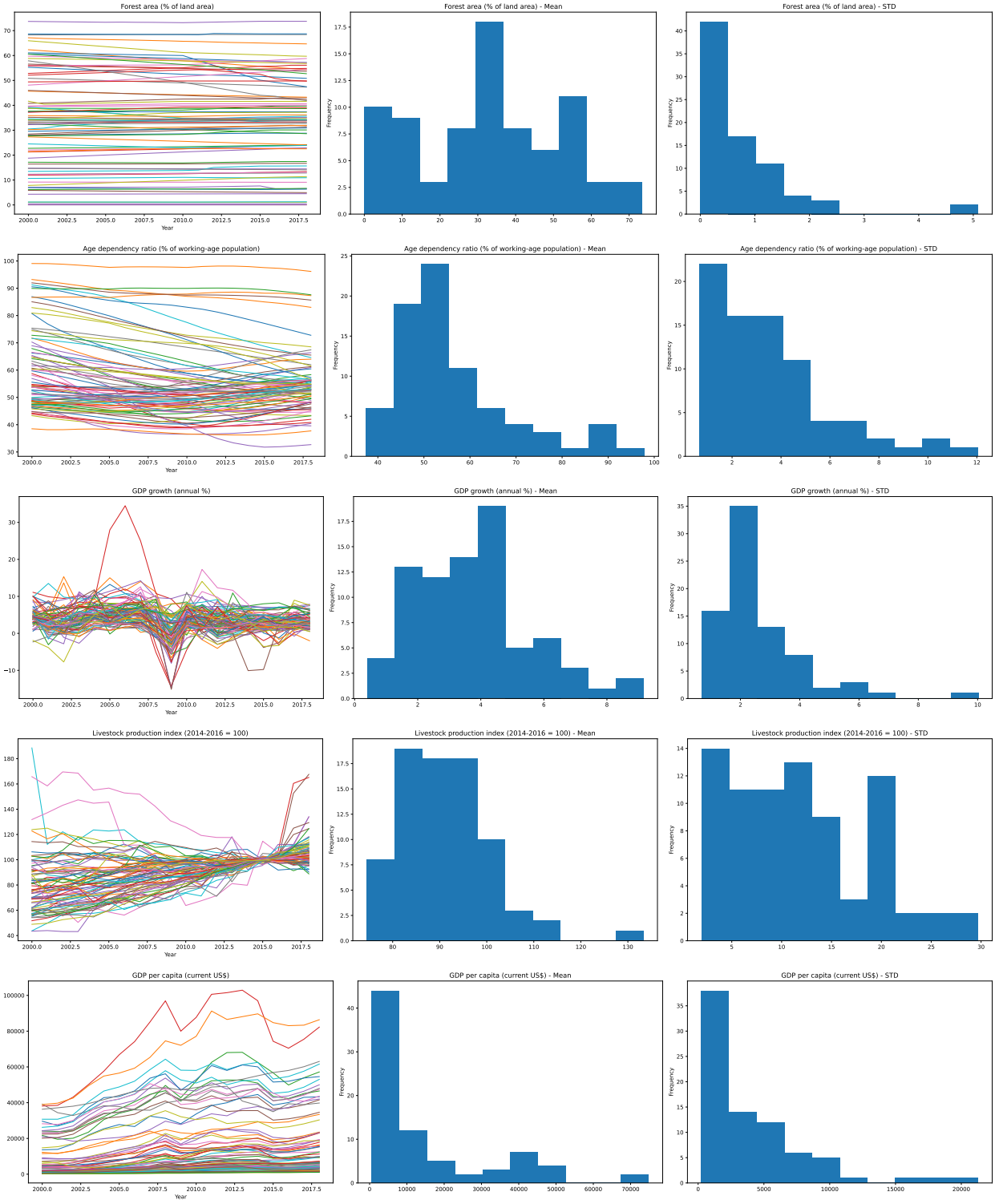


Figure 1: Statistics of selected time series types (values, mean histograms and std histograms)

3 Autoencoders

Autoencoders are neural networks consisting of two modules: encoder and decoder. An encoder encodes an input vector into a latent (embedding) space. Then, a decoder decodes this vector back to the original dimensionality so that it resembles the input vector with the least possible error. After training an autoencoder, the encoder module can be used for feature extraction. We used this approach to extract time-independent features from our time series.

3.1 Data preprocessing

Before playing with neural networks, we had to preprocess the data. We could not simply standardize the data, column by column, since it would have ruined the neighborhood relationship between samples. Therefore, we normalized the data so that their norms would be equal to 1. We treated their default norms as one of the final features, apart from the features retrieved from an autoencoder. Therefore, if an autoencoder has 4 neurons in the bottleneck, the final size of the latent space is 5.

3.2 Experiments

At first, we used a single time series type ('Population growth (%)') to perform experiments and choose the best autoencoder architecture.

We tested many different architectures with the numbers of the neurons in the bottleneck, that is the embedding dimensionality, from 1 to 8 (later referred to as `n_bottleneck`). We describe the architectures briefly below. The best architecture will be described in-depth in the next section.

The first three models are regular networks with different numbers of dense-connected layers and the ReLU activations:

- `Autoencoder v1` with one layer (`n_bottleneck` neurons) in the encoder and the decoder,
- `Autoencoder v2` with two layers (20, `n_bottleneck` neurons) in the encoder and the decoder,
- `Autoencoder v3` with four layers (80, 40, 20, `n_bottleneck` neurons) in the encoder and the decoder.

The other three models are convolutional networks composed of convolutional layers with increasing numbers of filters, the ReLU activations and, respectively, max pooling or upsampling layers to decrease or increase the length of a time series. In the bottleneck we used two dense-connected layers without an activation to transform the features created by the convolutional layers into the latent space and then back to the input for the decoder. Again, the models have different numbers of convolution blocks (convolution, activation and max pooling or upsampling):

- `Autoencoder v4` with one convolution block (32 filters) in the encoder and the decoder,
- `Autoencoder v5` with two convolution blocks (32, 64 filters) in the encoder and the decoder,
- `Autoencoder v6` with three convolution blocks (32, 64, 128 filters) in the encoder and the decoder,

Before training the models, we investigated the complexity of the data using the Principal Component Analysis (PCA). We show the explained variance ratio plot in Figure 2. We also applied the inverse PCA transformation for different numbers of first PCA components and compared the original time series with the reconstructed ones by measuring the mean absolute error (MAE). The MAE values are presented in Figure 3. Note that the MAE values are multiplied by 10,000 for clarity. Since we use the MAE metric as a loss function for autoencoders, these values can serve as a reference in order to determine how much we gain from employing autoencoders with respect to the simple, PCA approach.

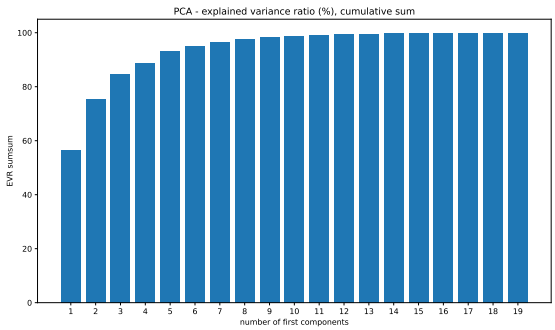


Figure 2: PCA explained variance ratio cumulative sum of first components

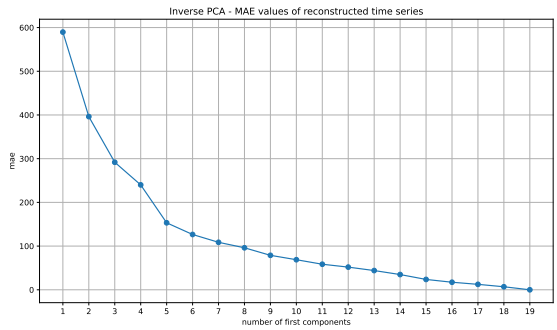


Figure 3: MAE values for the time series reconstructed by inverse PCA

We used the Adam optimizer. We trained each of the models with a dynamic learning rate that was configured to decrease on a loss plateau, that is when the loss is not improving for a selected number of training epochs. The models were trained for the numbers of epochs corresponding to their complexity. Note that we did not split the dataset into training and testing set. We did not need the models to generalize so the split was not necessary. We present the comparison of achieved loss values in Figure 4.

The loss values look reasonable – the more complex the architecture and the more neurons in the bottleneck, the smaller the loss is. We also see higher performance of the convolutional approach in this problem. Therefore, we decided to use one of the convolutional models for feature extraction. The efficiencies of `Autoencoder v5` and `Autoencoder v6` are very similar, so it would seem more efficient to use the smaller model. However, the ultimate goal of the model was to embed multiple time series types, so we decided to select the one with a higher capacity, that is `Autoencoder v6`.

Now, the losses of the autoencoders can also be compared with the MAE values of the inverse PCA reconstructions. While the loss curves of the first two models and PCA are similar, superiority of more complex architectures is clearly visible. The ultimate `Autoencoder v6`

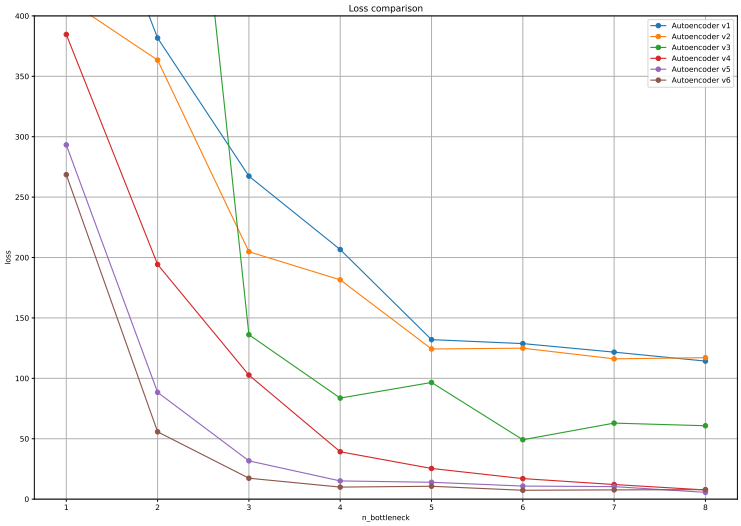


Figure 4: Comparison of the loss values achieved by the autoencoders

reaches the MAE value of around 20 for 3 neurons in the bottleneck while PCA requires about 14 dimensions in the latent space to achieve similar performance.

In the end, we compared the methods visually too. Figure 5 shows the difference of reconstruction quality between PCA and the best model. For this particular time series, autoencoder with `n_bottleneck=2` has a smaller MAE value than inverse PCA using 8 first components.

All training details can be found in the project repository [2].

3.3 Final autoencoder model

The final autoencoder model takes time series with 19 samples as an input. It uses convolutional and max pooling layers to scale it to the length 10, 5 and 3 at the end. Number of convolution channels is increasing from 32, through 64, to 128. In the bottleneck, it uses a dense layer to encode the data in the latent space (`n_bottleneck` neurons) and another dense layer to go back to `3 * 128 = 384` numbers. Now, convolutional and upsampling layers are used to decode the data to the length 24 and 32 channels. The length is different than 19, since 19 cannot be reached from 3 only by multiplying by 2. Therefore, we added a dense layer at the end with 19 output neurons. The summary of the final model (with 4 neurons in the bottleneck) is printed in Appendix A.

We trained the model with all the time series we had. Later, we copied the weights and fine-tuned the autoencoder for each of the time series types. We performed this procedure for models with 2, 4 and 8 neurons in the bottleneck.

For instance, the network with 4 neurons in the bottleneck achieved MAE equal to 27.2 (Figure 6, left). Then, we fine-tuned it for each type of the time series. 'Inflation, GDP deflator (annual %)' turned out to be the easiest type, as the network reached MAE very close to 0 (Figure Figure 6, middle). On the other hand, 'Immunization, DPT (% of children

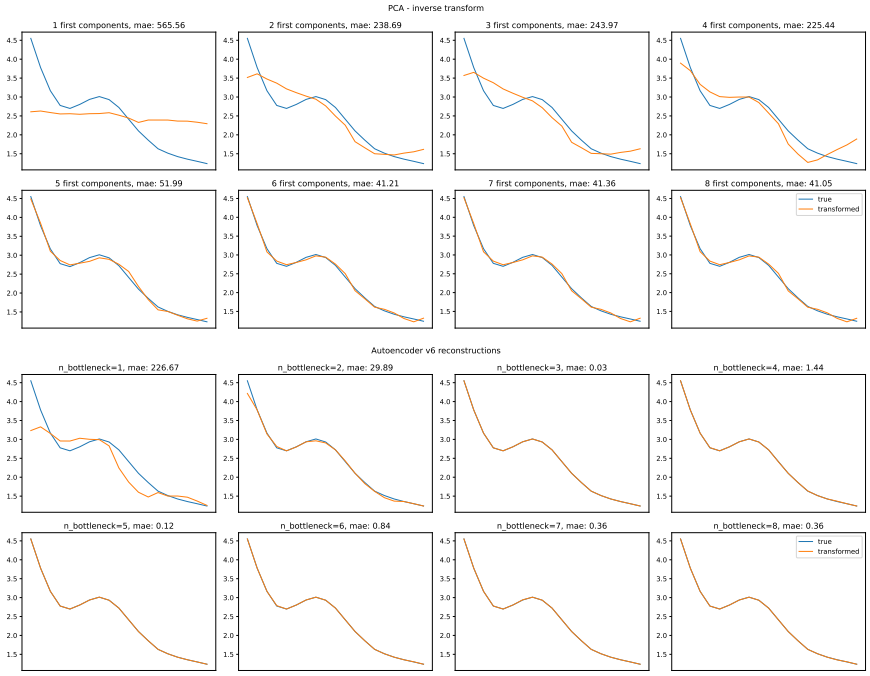


Figure 5: Comparison between reconstructions made by inverse PCA with different numbers of first components used and autoencoder (v6) reconstructions for different numbers of the neurons in the bottleneck

ages 12-23 months)', with MAE equal to 6.5 was the most difficult one (Figure 6, right).

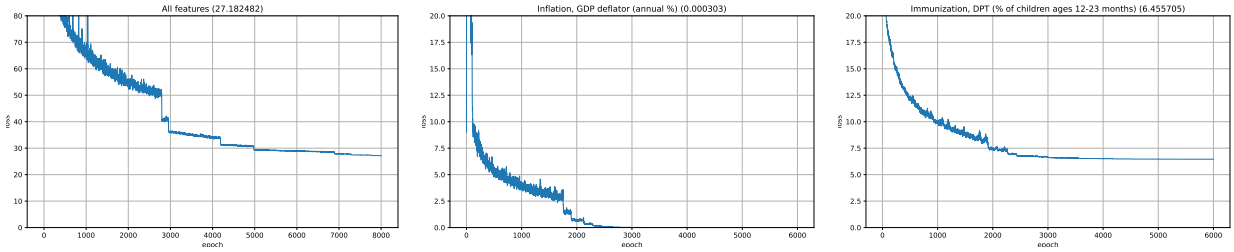


Figure 6: Loss curves for the autoencoder with 4 neurons in the bottleneck. Left: training on all features, middle: fine-tuning for the easiest time series type, right: fine-tuning for the most difficult time series type. The values of the loss functions are printed in parentheses. Note that the learning rate is automatically decreased during training, and thus the curves have a staircase shape.

In Figure 4 we can observe so-called elbows for 2, 3 and 4 neurons, and after 4 there is a plateau, therefore we use only the model model with 4 neurons in the bottleneck during the analysis.

All training details can be found in the project repository [3].

3.4 Analysis of single time series types

At first, we analyze each time series type independently. Every time series is encoded into 5 values: norm and 4 values retrieved from the autoencoder. Each country is thus represented as a 5-number vector. These vectors are passed to the t-SNE and UMAP algorithms which are configured adequately for this type of data. Figure 7 shows some of more interesting results. A brief analysis is included below. The specific graphs from World Bank Open Data on which we base the analysis can be found in the appendix B. All the plots can be found in the project repository [4].

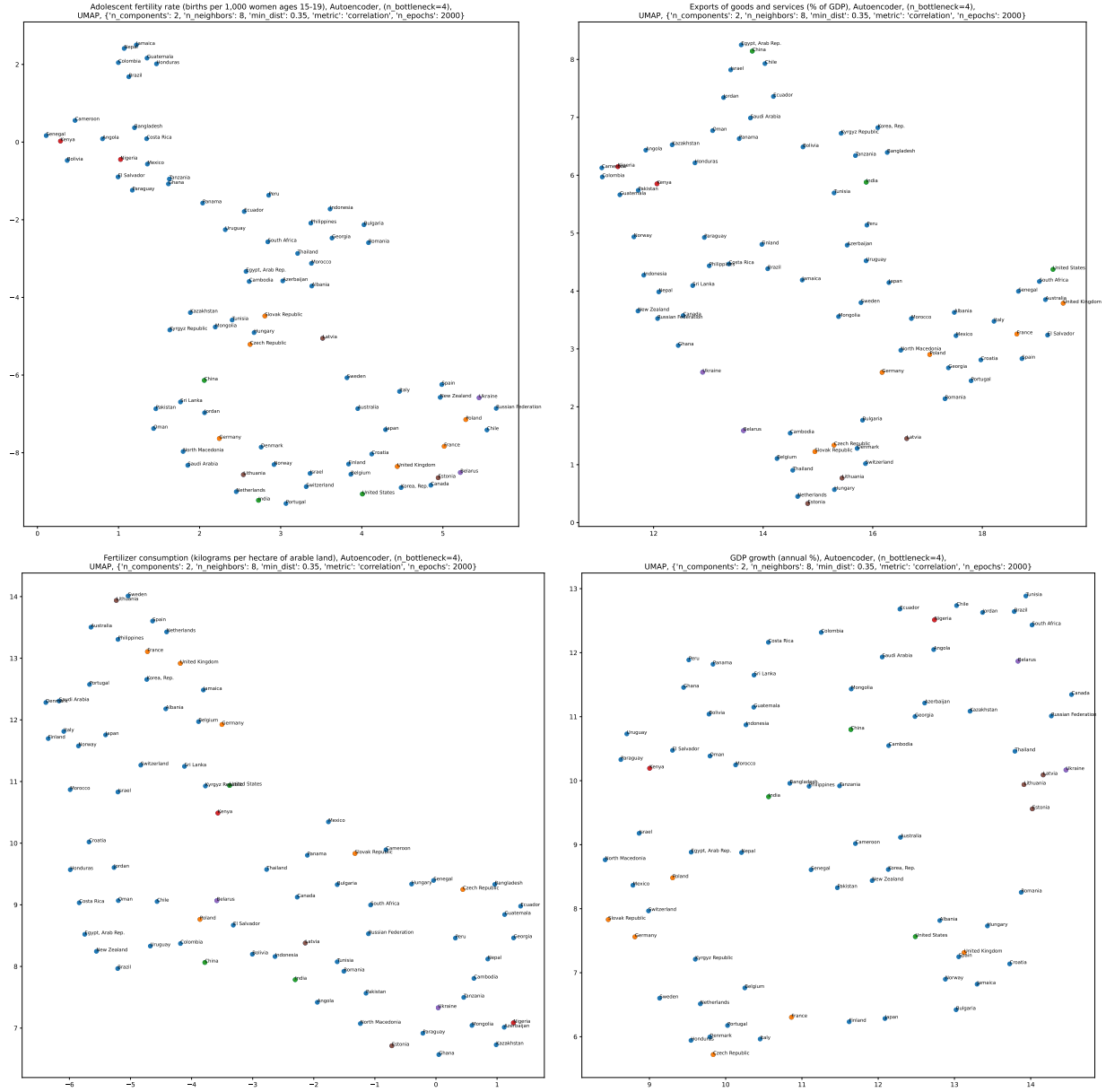


Figure 7: Some of more interesting visualisations of single time series types. Some countries are marked with different colours to increase clarity.

- Top left – Adolescent fertility rate. Poland is next to countries which do not seem to have much in common, e.g. Chile, Russia, Ukraine, New Zealand, France or Estonia. However, in all of these countries we can observe a decrease in the value of adolescent fertility rate with a more dynamic trend at the end. The magnitude is different (Chile – 40 in 2018, Poland – 10 in 2018), but the trend is similar. New Zealand, Ukraine and Russia form a smaller cluster of countries that had a small increase between 2002 and 2007. There is also an interesting cluster in in the top left corner. These countries share a huge decrease of the indicator. Again, the magnitude is a little different, but the plots are parallel, e.g. Honduras goes down from 116 in 2000 to 70 in 2018, and Jamaica – from 88 to 50.
- Top right – Exports of goods and services. We could expect that Poland would be next to countries like Czech Republic and Slovak Republic, but this is not the case. Poland, North Macedonia and Georgia that are next to each other share almost the same curves, they start from about 30% in 2000, and reach about 50% in 2018. On the other hand, the curves of Czech Republic and Slovak Republic are higher, while the curve of Bulgaria is, as expected, between the Polish and Czech groups.
- Bottom left – Fertilizer consumption. One of the clusters is located in the top left corner. It starts with Germany, and goes through UK, France, ending on Sweden. All of these curves are similar, but their magnitude is a little different. The countries are not sorted in this order, however. What seems to be the order on the visualisation is the size of the 'pit' between 2006 and 2010. We can also compare these curves with the Polish cluster in the middle of the plot. The curves of Poland and El Salvador rise at the beginning of the twenties, while the values top-left-corner countries stay constant or decrease. On the other hand, the cluster in the bottom-left corner is composed of the countries using very small amounts of fertilizer, especially between 2000 and 2010.
- Bottom right – GDP growth. While Poland and Switzerland could never be next to each other in a GDP per capita plot, they share the same cluster in the GDP growth plot. We can compare them with Italy or Czech Republic, which are in another, but not distant, cluster. The main difference seems to be the size of the 'pit' near 2012 – Czech Republic and Italy lost more GDP than countries like Poland, Switzerland, Israel, Mexico or Slovac Republic. Another interesting fact is a presence of Albania next to USA and UK. Their curves are very similar starting from 2011 – they did not suffer that much as the previously mentioned countries in 2012.

3.5 Analysis of time series groups

Now we analyze the time series groups which also include the group of all the time series. Again, each time series is represented as a 5-number vector. We concatenate these vectors for each country, which results in vectors of a length 5 times longer than the number of time series types in a group. As previously, we use t-SNE and UMAP for visualisation purposes. Now, UMAP has two configurations though – one supporting more dense (`min_dist = 0.1`) and the other – less dense clusters (`min_dist = 0.4`). The results are show in Figure 8. Only one visualisation per group is shown, but all the plots can be found in the project repository

[5].

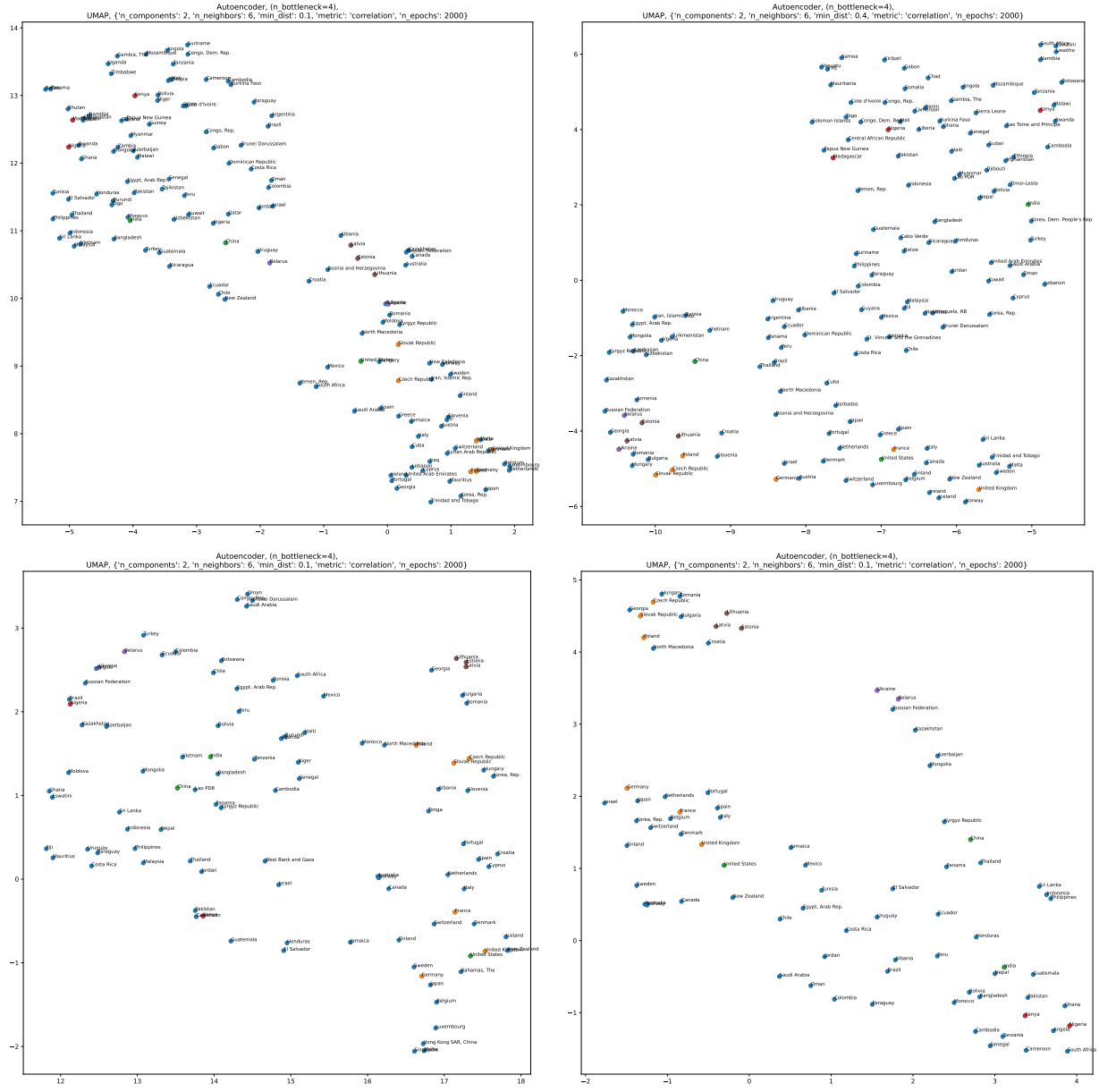


Figure 8: Time series groups visualisations. Top left: agriculture indicators, top right: health indicators, bottom left: economy indicators, bottom right: all indicators. Some countries are marked with different colours to increase clarity.

- Top left – Agricultural indicators. We can observe a lot of clusters with neighbouring countries, e.g.: Poland, Germany; Belgium, Luxembourg, Netherlands or Latvia, Estonia, Lithuania. These countries have similar agricultural conditions. There are some nonobvious relations. For instance, Iran is located between Czech Republic and Sweden.
- Top right – Health indicators. It looks like the bottom of the plot (below -2.5) is filled with countries that do not suffer from extreme poverty, and the quality of health

services is generally good, but varies. It seems, that the more we go to the right side of the plot, the better the health services are. So, on the left-hand side, there are Eastern Europe countries with life expectancy around 77. Life expectancy rises as we go to the right side of the plot. A presence of of Trinidad and Tobago with life expectancy of 73 years next to Sweden where people, on average, live almost 10 years longer is interesting. One possible explanation of this phenomenon could be the fact that some of the Trinidad and Tobago time series are very similar to their Swedish counterparts, just with a different magnitude. For example, the life expectancy curves are almost parallel. It is also important to remember, that the health group includes indicators like fertility rate or population growth which do not always depend on the quality of health services in a particular country.

- Bottom left – Economy indicators. Again, there are a few obvious clusters like Lithuania, Estonia, Latvia, but Poland, for example, is in a cluster with North Macedonia (which is common) and Morocco. The GDP of Poland is almost 5 times bigger than the GDP of Morocco, and the GDP growth plots are different. A possible explanation could be the fact that Poland and Morocco are almost identical when it comes to a few indicators, e.g. imports of goods and services (% of GDP) or value added of industry (% of GDP).
- Bottom right – All indicators. This plot includes many obvious relations. The Eastern Europe countries (and Georgia whose GDP (annual %) is growing similarly to Poland) are in the top left corner. Then, the western countries with developed countries like USA or Canada occupy the left middle side. The post-soviet countries are located in the top right corner. Finally, the bottom right corner belongs to poor countries from Africa.

4 Hierarchical clustering

Hierarchical clustering is an unsupervised clustering approach which builds a hierarchy of clusters. The most common type of hierarchical clustering is agglomerative hierarchical clustering. In this strategy, each observation is initially treated as a separate cluster, then two closest clusters are merged into one. This iterative process repeats until all clusters are merged into one which contains every data point.

In order to decide which clusters should be combined, it is necessary to define distance metric and linkage criterion. The linkage criterion is a measure of the similarity between the two clusters. The results of hierarchical clustering are presented in a tree diagram showing the hierarchy of clusters to which each observation belongs, called a dendrogram. Final clustering is achieved by cutting the dendrogram with a horizontal line at a certain height, usually it is where the line can traverse the maximum distance up and down without intersecting the merging point.

4.1 Analysis of separated indicators

In our hierarchical cluster analysis, we chose euclidean distance as distance metric and Ward's method as linkage criterion. In this method, the choice of two clusters to merge is made on the minimum increment of total within-cluster variance. As it was said previously, each observation is initially treated as a separate cluster, so the sum of squares starts out at zero and grows with merging clusters. Ward's method allows keeping this growth as small as possible.

At the beginning, we clustered countries by each indicator in each group separately. Hierarchical relationships between countries are shown in the dendrograms: selected indicators in Figure 9, agriculture indicators in Figure 10, health indicators in Figure 11 and economy indicators in Figure 12. Then, looking at achieved diagrams, we determined the number of clusters. In the dendrograms, each color represents a different cluster.

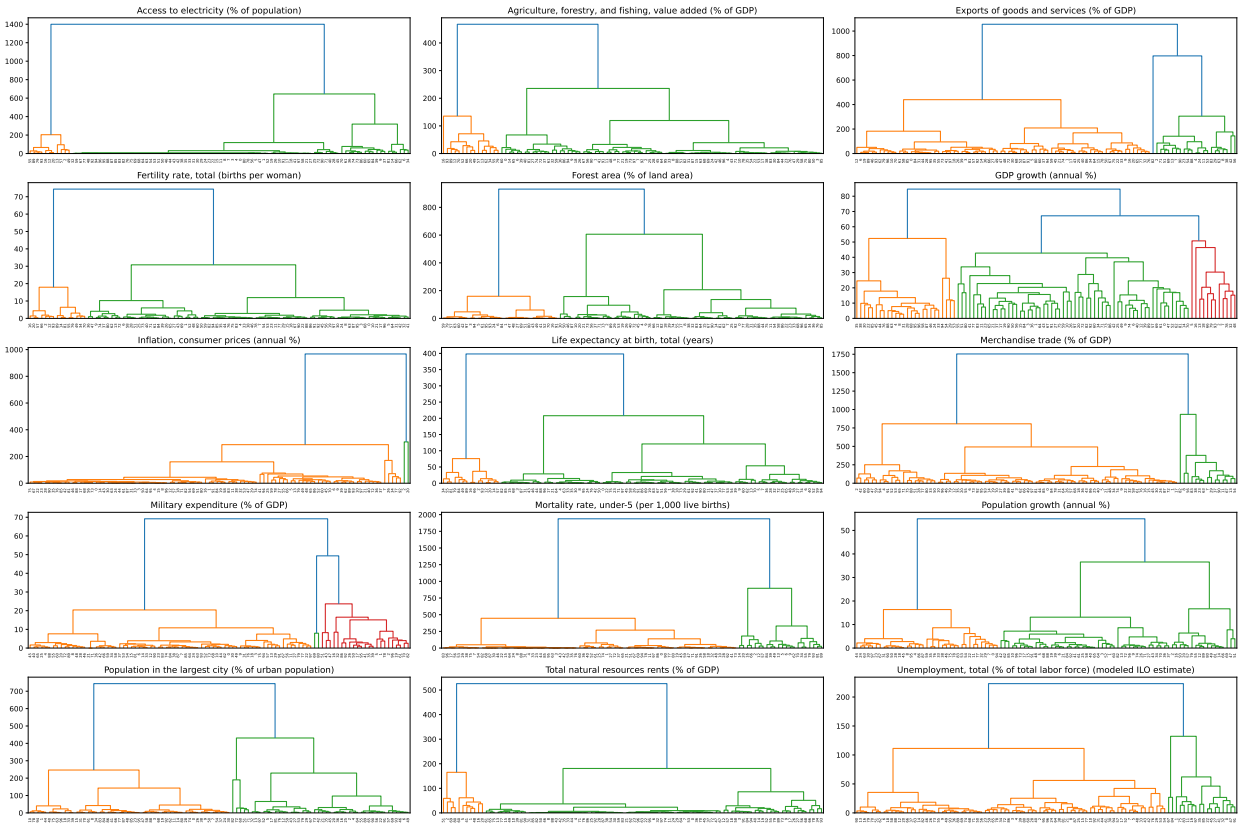


Figure 9: Dendrograms presenting hierarchical relationship between countries by selected indicators

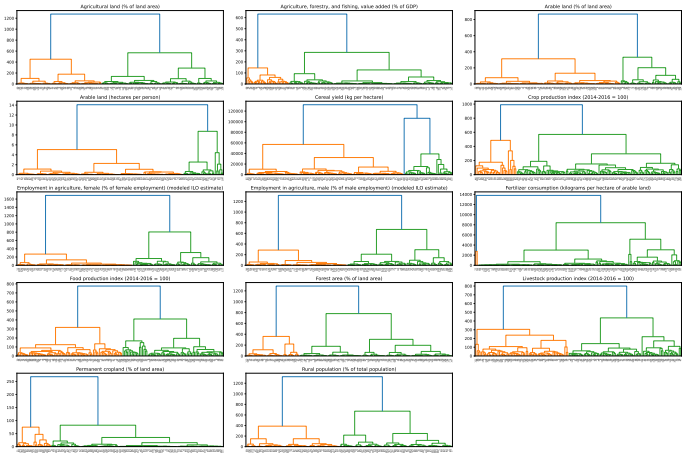


Figure 10: Dendrograms presenting hierarchical relationship between countries by agriculture indicators

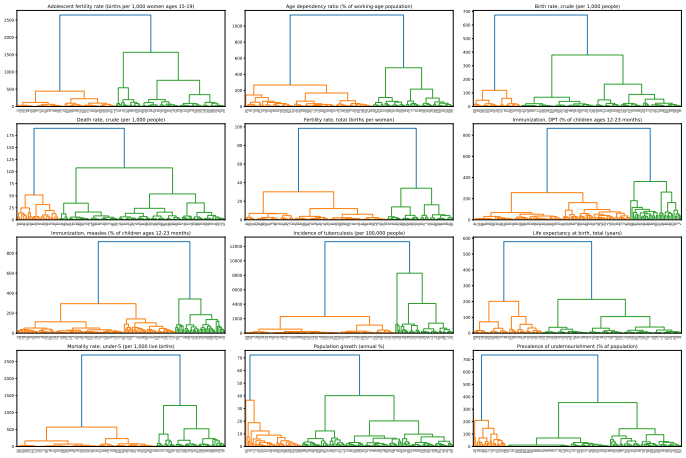


Figure 11: Dendrograms presenting hierarchical relationship between countries by health indicators

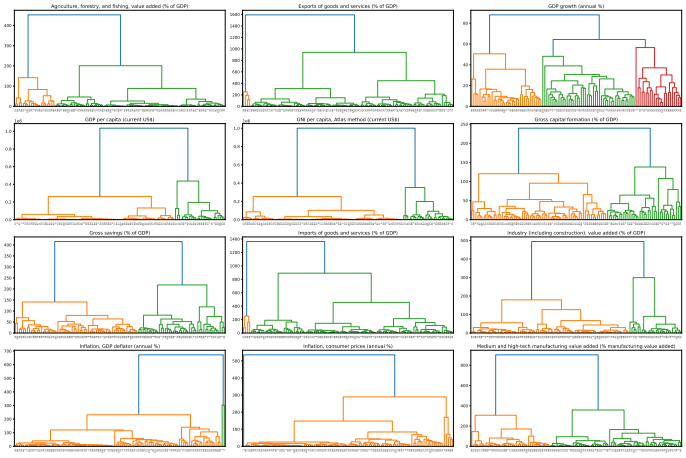


Figure 12: Dendrograms presenting hierarchical relationship between countries by economy indicators

After that, we used the Uniform Manifold Approximation and Projection (UMAP) method in order to reduce dimension. In this way, we were able to visualize the data in two dimensions and see clustering results. Due to the large number of visualizations, we decided not to include all of them in this paper. Also, clustering by each indicator separately was not our main goal, so we decided to focus on groups of indicators analysis. As an example, one of the clustering result is shown in Figure 13. Other visualizations can be found in the project repository[6].

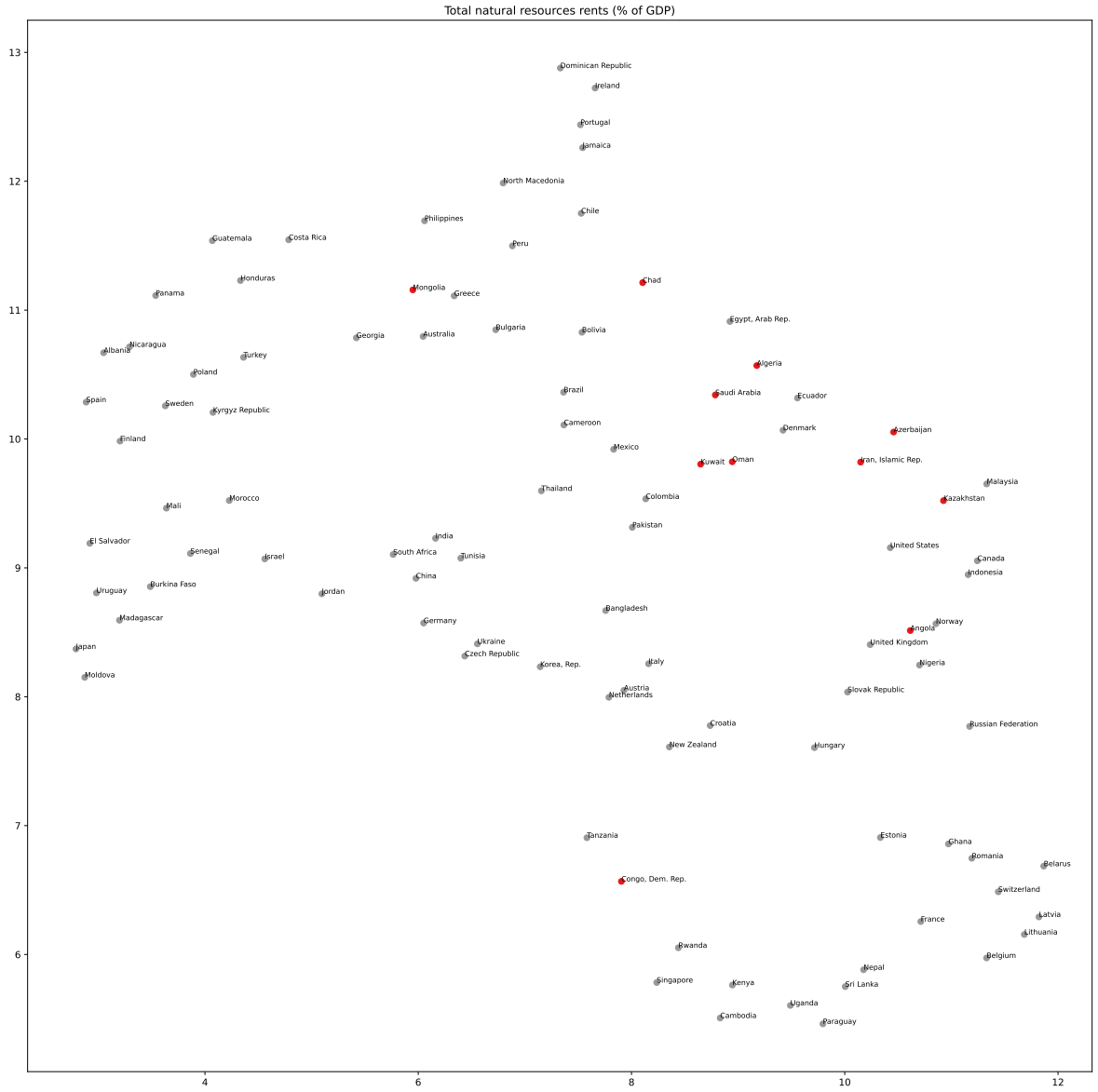


Figure 13: Countries clustered by total natural resources rents (% of GDP), each color represents a separate cluster

Subsequently, we repeated steps described above, but with feature extraction using one of the previously trained autoencoders (`Autoencoder v4`). Hierarchical relationships between objects from each indicator group, after feature extraction, are shown in dendrograms: selected indicators in Figure 14, agriculture indicators in Figure 15, health indicators in Figure 16 and economy indicators in Figure 17.

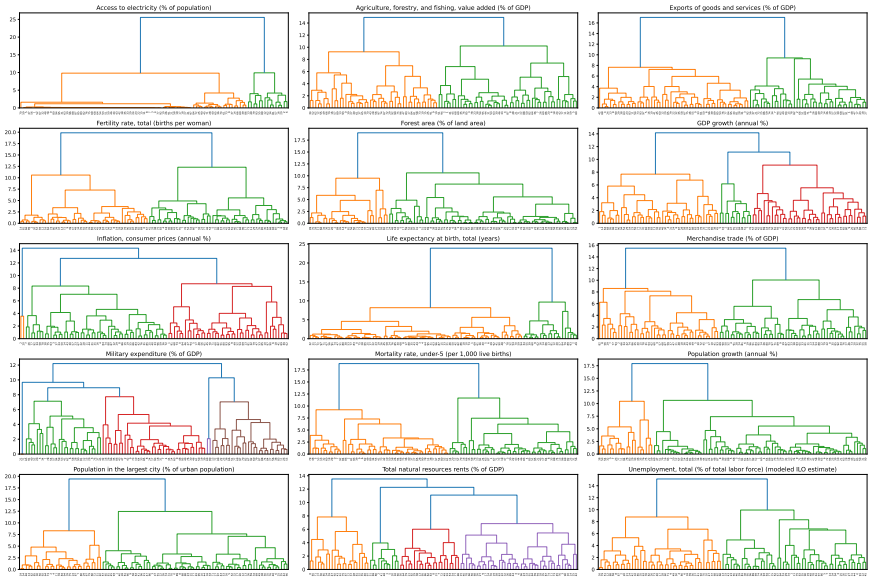


Figure 14: Dendrograms presenting hierarchical relationship between countries by selected indicators, with feature extraction

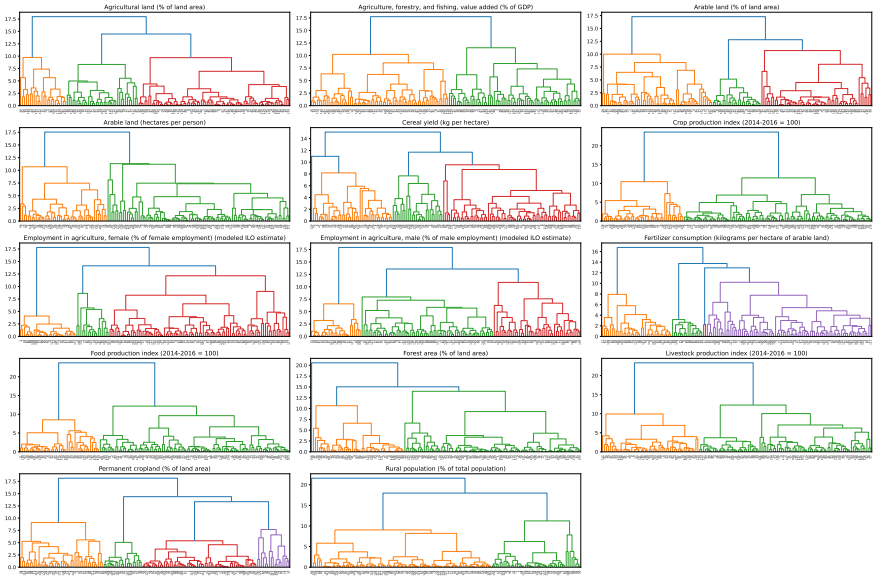


Figure 15: Dendrograms presenting hierarchical relationship between countries by agriculture indicators, with feature extraction

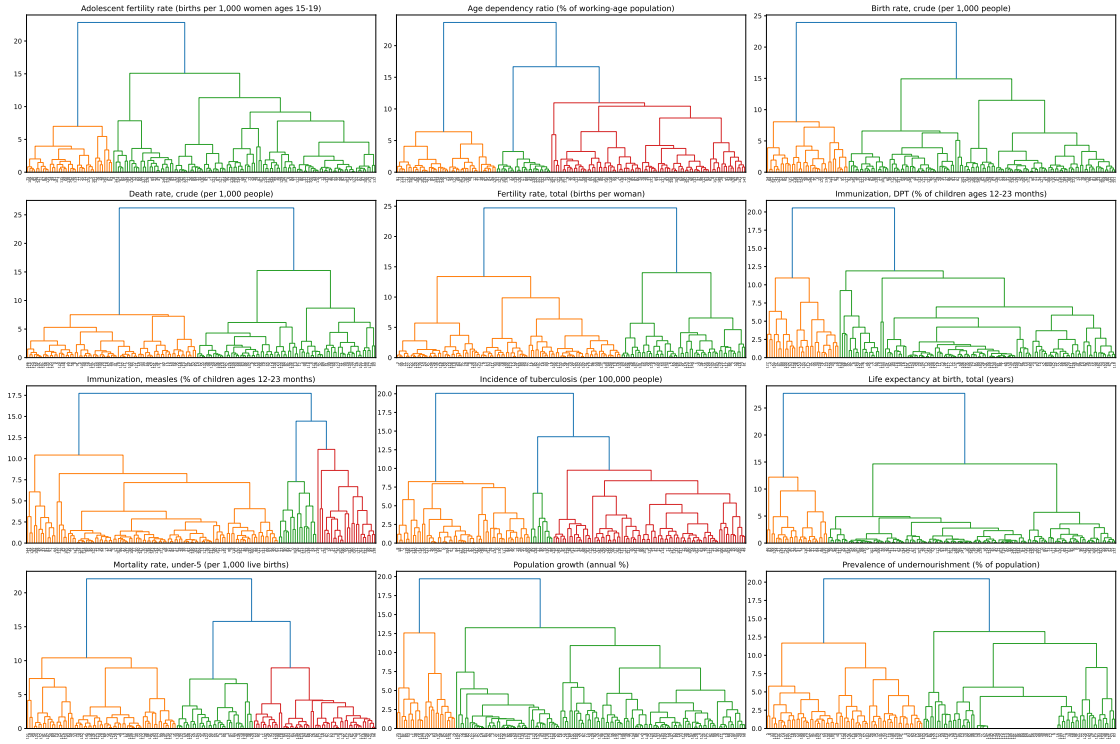


Figure 16: Dendrograms presenting hierarchical relationship between countries by health indicators, with feature extraction

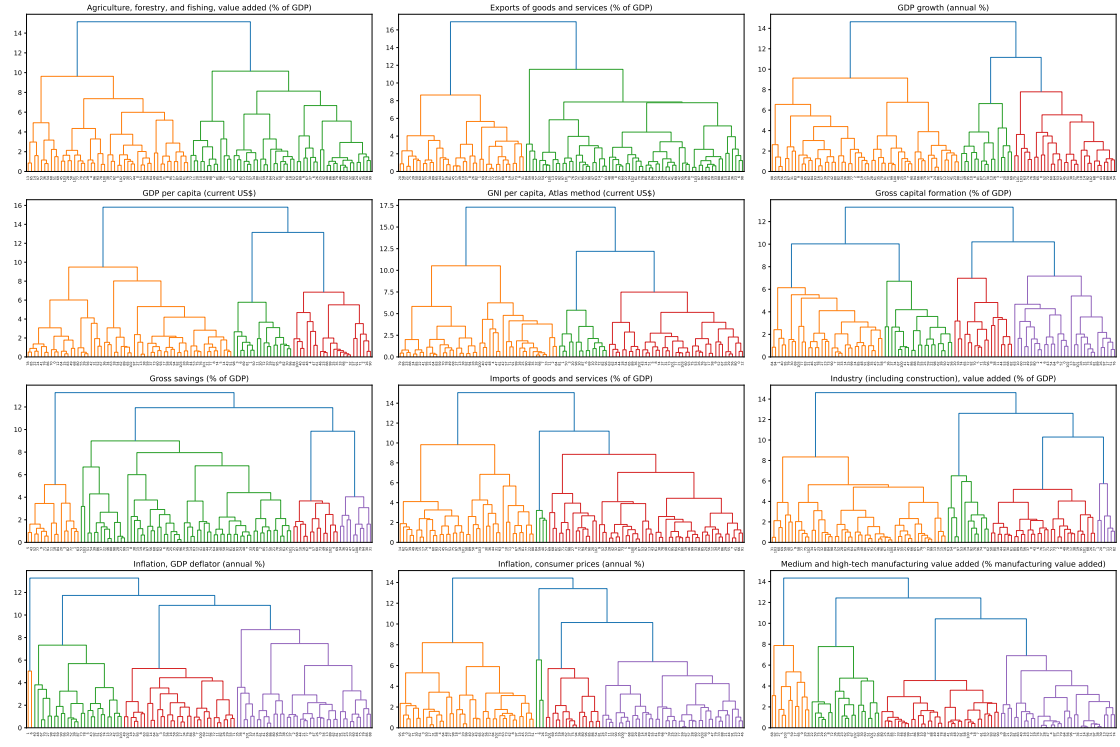


Figure 17: Dendrograms presenting hierarchical relationship between countries by economy indicators, with feature extraction

As diagrams show, results differ from those without feature extraction presented before. The structure of clusters has changed, as well as the number of clusters for some indicators. Clustering by total natural resources rents (% of GDP) is shown in Figure 18. In case of clustering without feature extraction, only two clusters were found. By looking at data we can tell that one cluster, marked in red, includes countries with higher total natural resources rents and the second cluster, marked in grey, includes countries with lower total natural resources rents, so interpretation is pretty simple. Feature extraction revealed much more complex information about data. In this case, four significant clusters were found. We can assume that the cluster marked in brown includes countries in which the value of this indicator was very low (below 2.5% of GDP), compared to other countries, in the analysed period of time. In the cluster marked in grey, we can find countries in which the value of this indicator decreased in the period from 2000 to 2018. The third cluster, marked in red, includes countries in which total natural resources rents were unstable over that time. In the last cluster, marked in purple, there are countries for which the value of the analysed indicator was very high (over 20% of GDP), compared to other countries in the analysed period of time.

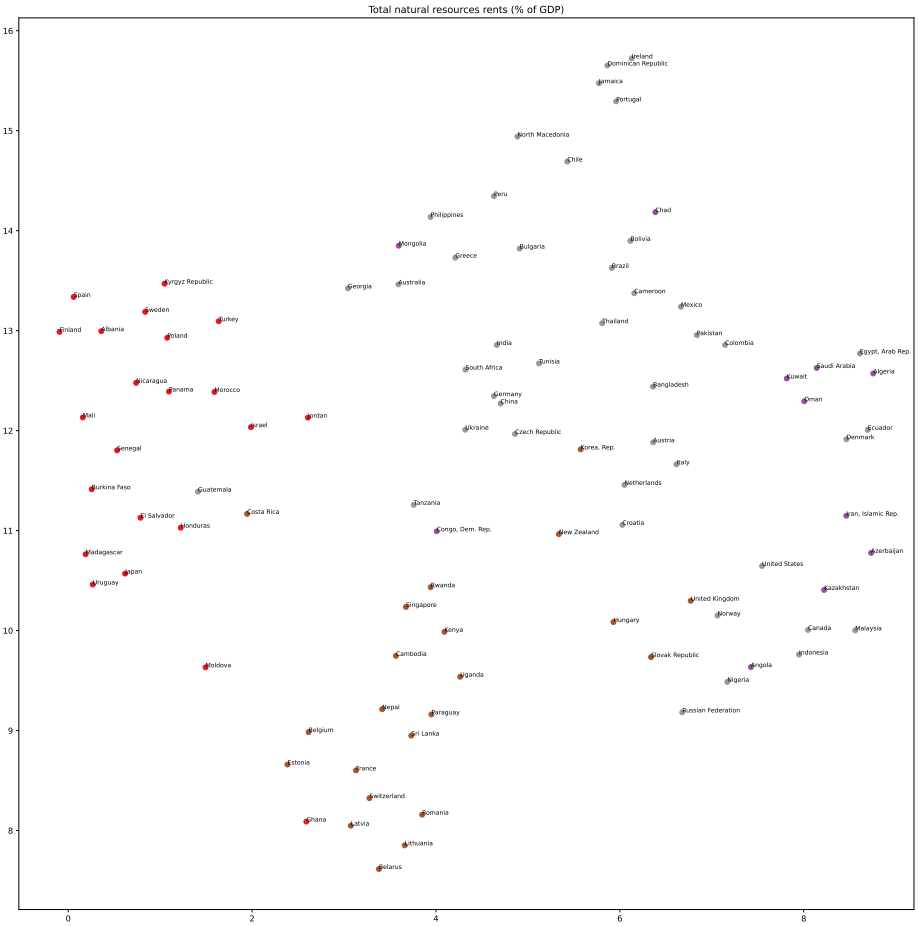


Figure 18: Countries clustered by total natural resources rents (% of GDP) with feature extraction, each color represents a separate cluster

4.2 Analysis of groups of indicators

Hierarchical cluster analysis was also carried out for combination of the indicators in four groups: selected, agriculture, health and economy, first without feature extraction, next with feature extraction. The analysis for the combination of all indicators was also performed. For feature extraction, `Autoencoder v4` was used. Dendrogram for selected indicators is shown in Figure 19 and clustered data is shown in Figure 20. The results can also be found in the project repository [7]. For these indicators, data was divided into two clusters. The first one, marked in red, includes Africa and South Asia countries and the second, grey cluster includes the remaining countries.

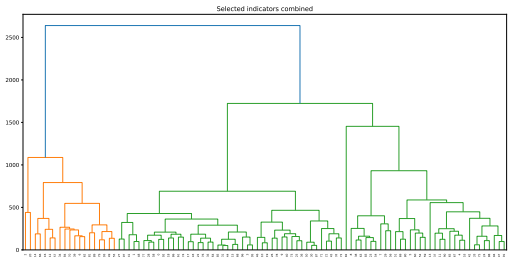


Figure 19: Dendrogram presenting hierarchical relationship between countries by combined selected indicators

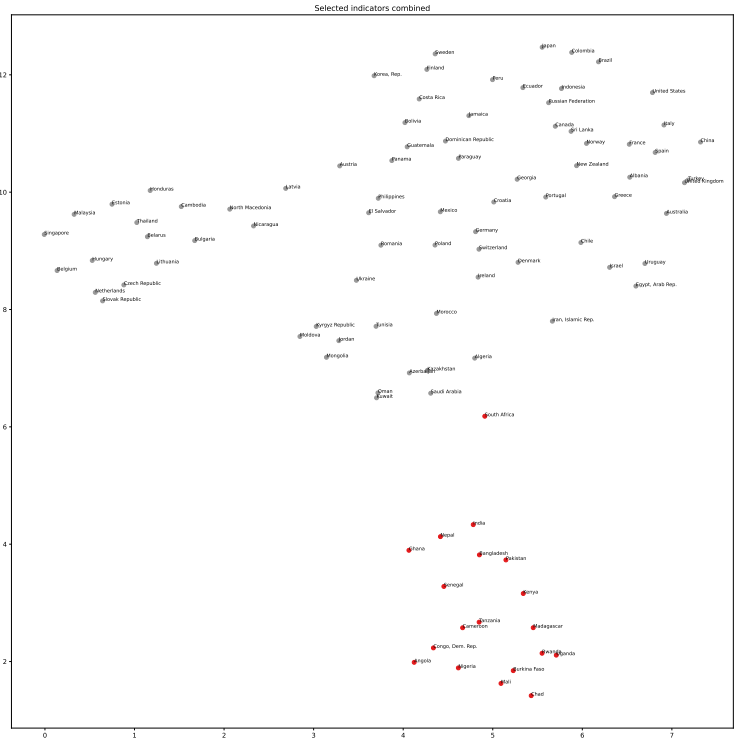


Figure 20: Countries clustered by selected indicators combined, each color represents a separate cluster

In case of clustering by combined selected indicators with feature extraction, three clusters were obtained, as we can see in Figure 21. The final result of clustering is presented in Figure 22. The first cluster, marked in red, includes European countries and more developed countries from other parts of the world such as: United States, Australia, Japan. The second, grey cluster includes Asia, South America and more developed African countries. The last cluster, marked in orange, includes poor countries, most of which are African countries.

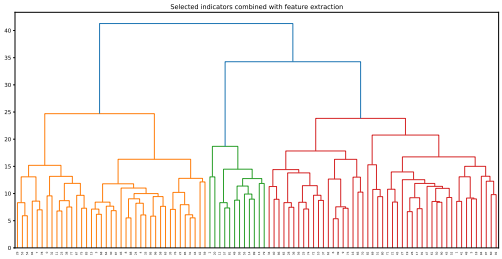


Figure 21: Dendrogram presenting hierarchical relationship between countries by combined selected indicators with feature extraction

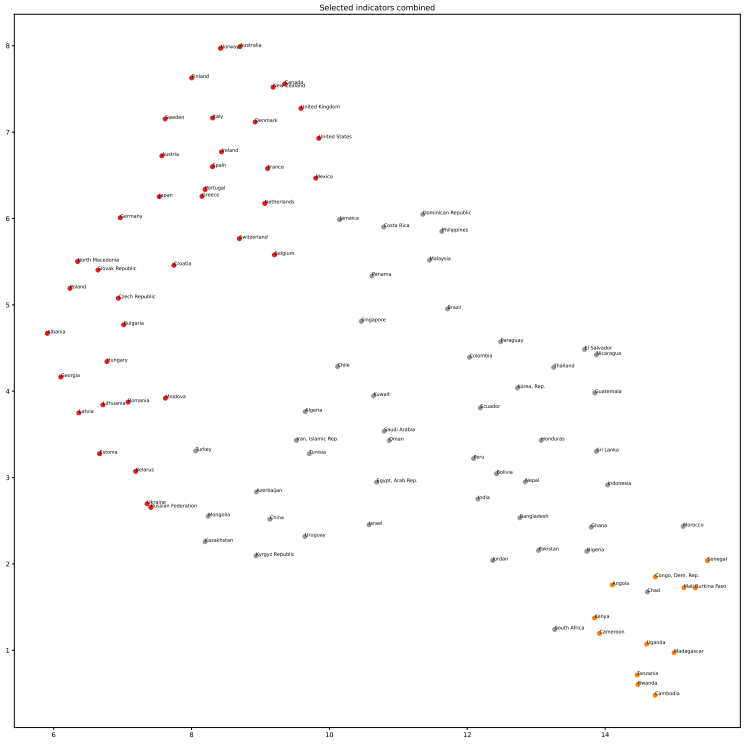


Figure 22: Countries clustered by selected indicators combined with feature extraction, each color represents a separate cluster

Clustering countries by combined agriculture indicators allowed for the designation of three clusters, as it is presented in Figure 23. As it can be seen, one of them includes only one country. which is the United Arab Emirates. This country is very different from the rest when it comes to agriculture, because despite the low importance of this sector, it was characterized by very high values of the cereal yield index in the analyzed period. The clustering result is presented in Figure 24. Cluster marked in orange includes countries for which agriculture is mostly not an important sector. Red cluster is the opposite. The single-element cluster, marked in grey, includes the United Arab Emirates.

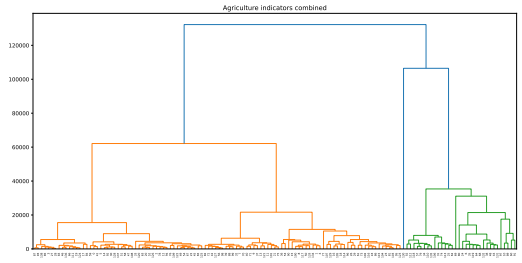


Figure 23: Dendrogram presenting hierarchical relationship between countries by combined agriculture indicators

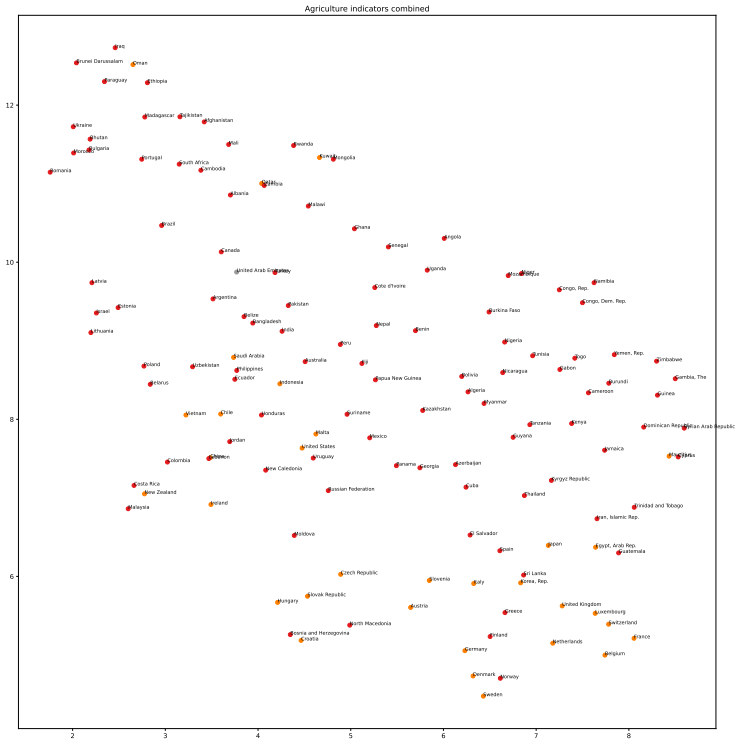


Figure 24: Countries clustered by agriculture indicators combined, each color represents a separate cluster

With feature extraction, we obtained only two clusters. The structure of clusters changed, some countries no longer belonged to the same cluster. Combined agriculture indicators with feature extraction dendrogram is presented in Figure 25 and clustering result in Figure 26.

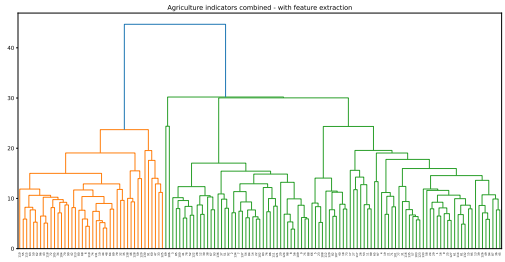


Figure 25: Dendrogram presenting hierarchical relationship between countries by combined agriculture indicators with feature extraction

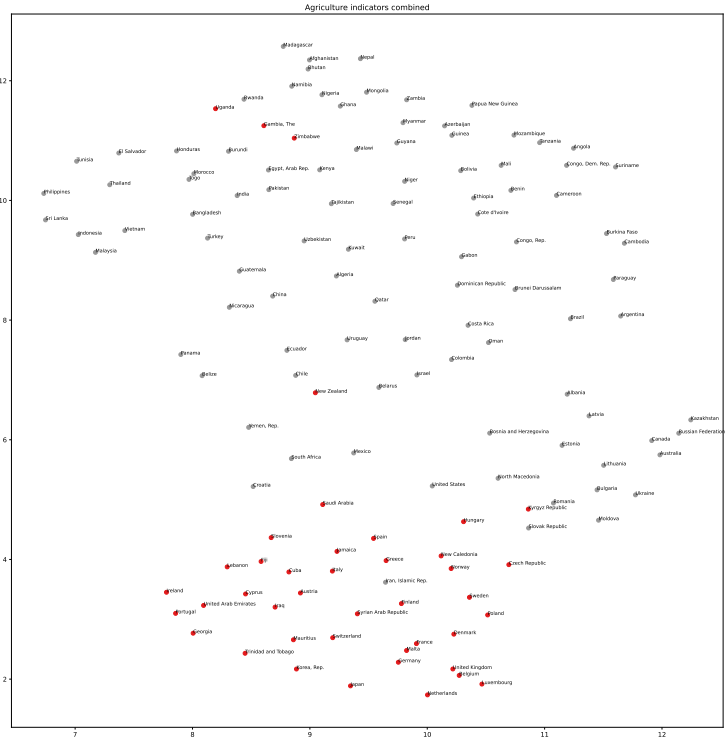


Figure 26: Countries clustered by agriculture indicators with feature extraction combined, each color represents a separate cluster

Health indicators were the next analysed group of indicators. Without feature extraction, three clusters were obtained, as it can be seen in dendrogram (Figure 27). The result of clustering is presented in Figure 28. The cluster having the most elements, marked in red, includes countries with better healthcare. The second, grey cluster contains countries with worse healthcare, higher birth and death rate. The last cluster, marked in orange, is similar to the second one, mainly different from it in more cases of tuberculosis.

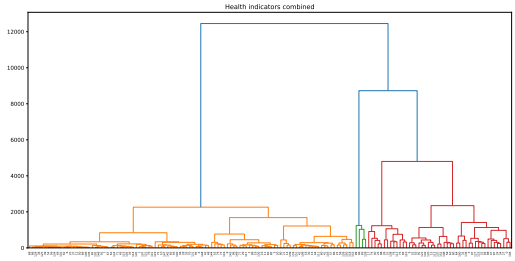


Figure 27: Dendrogram presenting hierarchical relationship between countries by combined health indicators

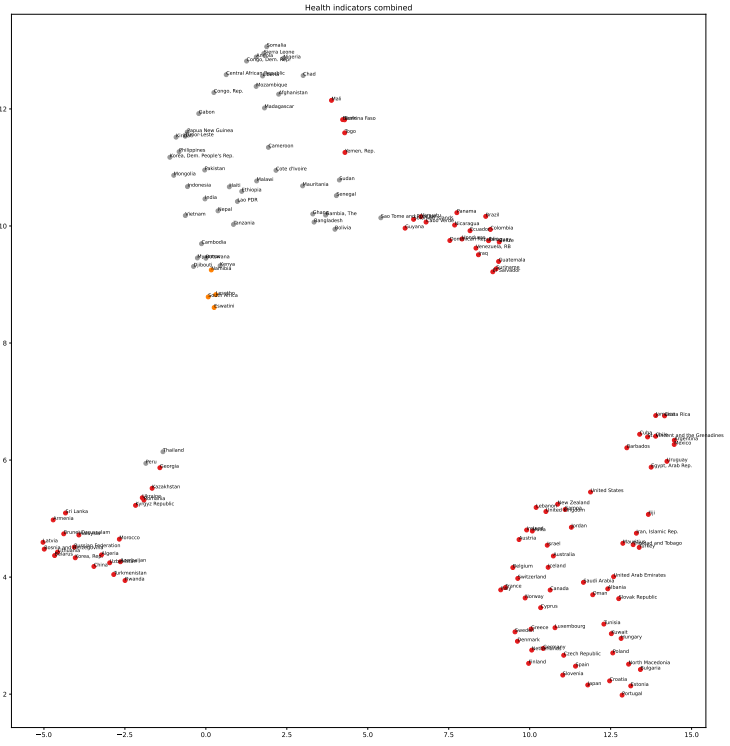


Figure 28: Countries clustered by health indicators combined, each color represents a separate cluster

Analysis of these indicators with feature extraction led to more simple interpretation. In this case, only two significant clusters were found, as it is shown in Figure 29. The result of this clustering is presented in Figure 30. We can tell that the first, more numerous cluster, marked in grey, includes countries with better access to health services, higher immunization rate, higher life expectancy and lower birth and death rates. The second cluster, on the other hand, is the opposite of the first one.

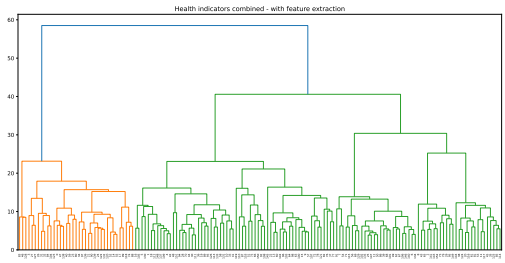


Figure 29: Dendrogram presenting hierarchical relationship between countries by combined health indicators with feature extraction

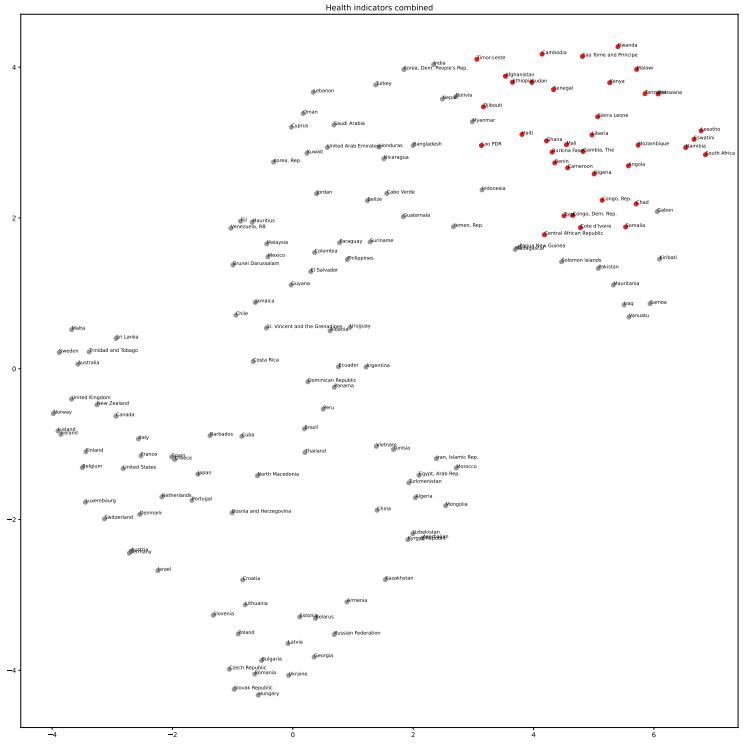


Figure 30: Countries clustered by health indicators with feature extraction combined, each color represents a separate cluster

The last group of indicators that we examined in our analysis were economic indicators. Two clusters were obtained without feature extraction, as Figure 31 presents. Clustered countries are presented in Figure 32. We can tell that a smaller cluster, marked in grey, includes the richest countries in the world like China, the United States or Western European countries. The second cluster includes the remaining countries.

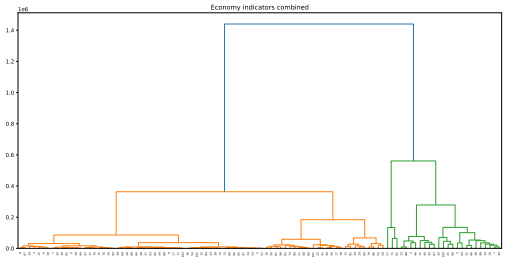


Figure 31: Dendrogram presenting hierarchical relationship between countries by combined economic indicators

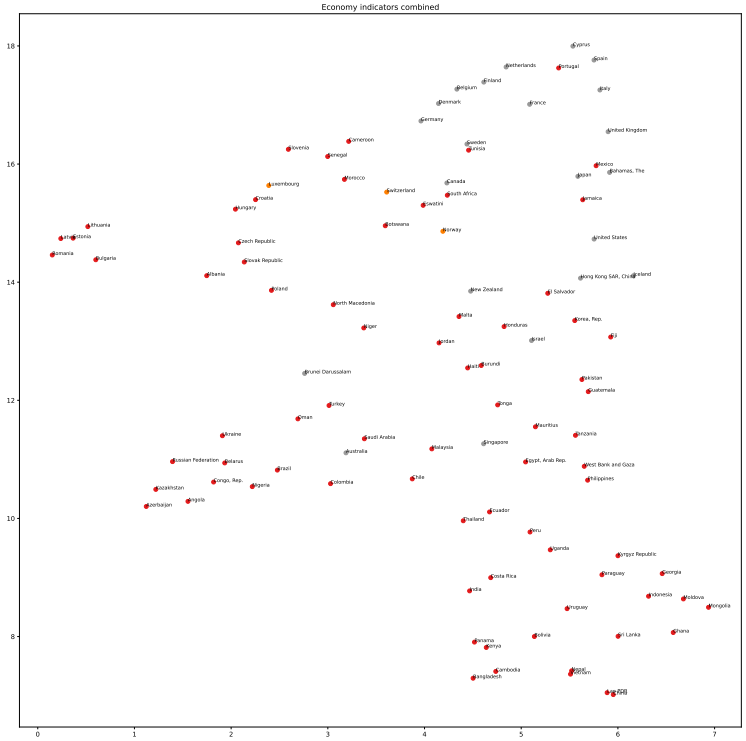


Figure 32: Countries clustered by economic indicators combined, each color represents a separate cluster

As with the previous groups, we then repeated the clustering using feature extraction. In this case, three clusters were received, as it is presented in Figure 33. The clusters and the countries belonging to them are presented in Figure 34. In the first, red cluster, there are mainly European countries, but also developed countries from other continents. The smallest cluster, marked in orange, includes Russian Federation, Ukraine, West Asia and some Africa countries. The last, gray cluster includes the remaining Asian, African and South American countries.

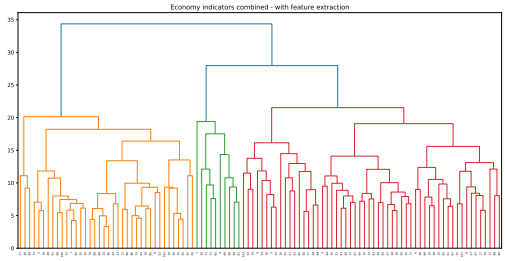


Figure 33: Dendrogram presenting hierarchical relationship between countries by combined economic indicators with feature extraction

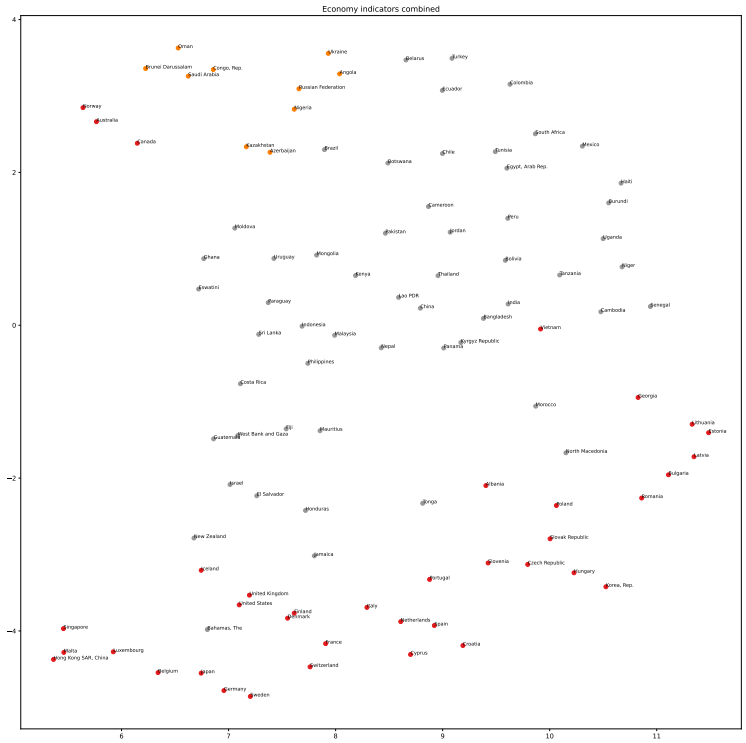


Figure 34: Countries clustered by economic indicators with feature extraction combined, each color represents a separate cluster

Finally, we clustered countries based on all the indicators together. In the case of clustering without feature extraction, only two clusters were obtained, as presented in Figure 35, one of them is clearly larger than the other. The results of this clustering are shown in Figure 36. We can conclude that the smaller cluster, marked in gray, contains the richest countries in the world, more developed than the other. The second, red cluster, on the other hand, includes the remaining countries.

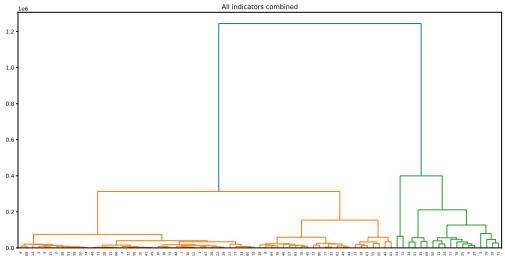


Figure 35: Dendrogram presenting hierarchical relationship between countries by combined all indicators

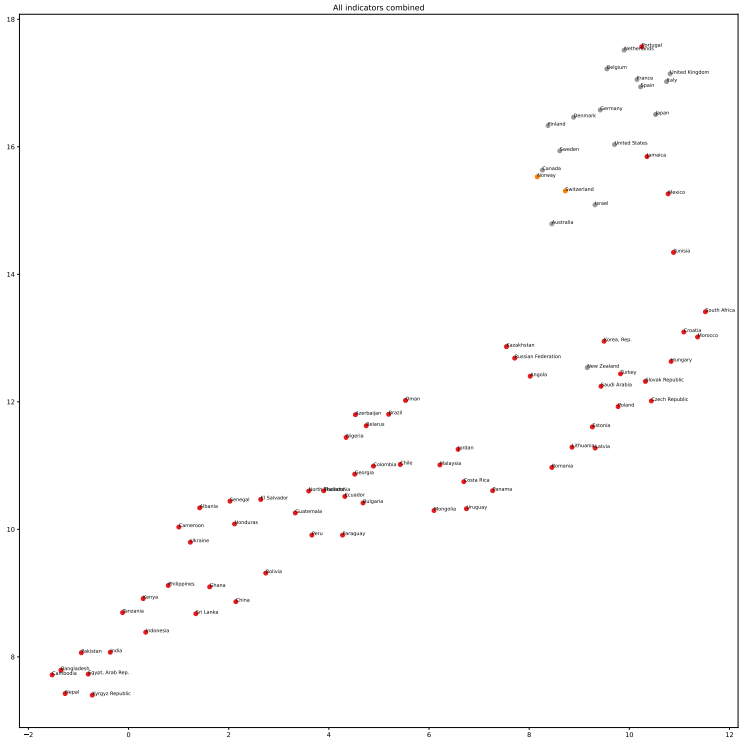


Figure 36: Countries clustered by all indicators combined, each color represents a separate cluster

Clustering with the use of feature extraction in this case gave much more interesting results. We got four, more equal clusters than before, as it can be seen in Figure 37. The final result of clustering based on all indicators with feature extraction is presented in Figure 36. The first, red cluster includes developed countries such as the United States, Australia, New Zealand, Japan and the countries of Western and Northern Europe. The second cluster includes post-communist countries, i.e. East-Central European and Asian countries. The next cluster, marked in brown, includes poor and developing countries, The last, gray cluster includes the remaining Asian, South American and Central American countries.

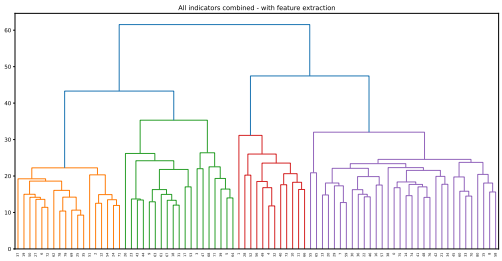


Figure 37: Dendrogram presenting hierarchical relationship between countries by combined all indicators with feature extraction

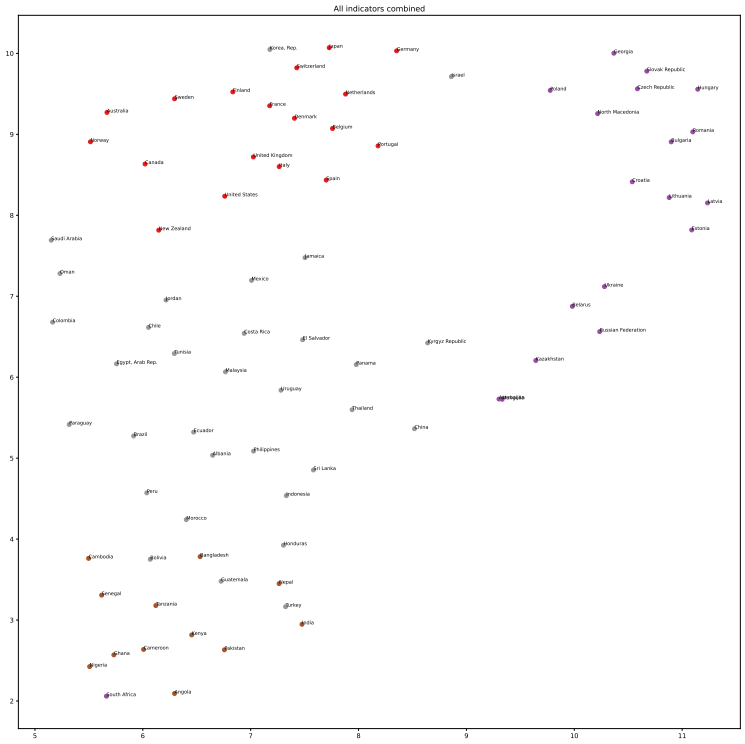


Figure 38: Countries clustered by all indicators with feature extraction combined, each color represents a separate cluster

5 Similarity metrics

The crucial part of time series data analysis is defining a set of tools that enables us to compare and measure a distance between data of this kind. This task has been a subject of academic research in recent years, resulting in defining mathematical metrics and their corresponding software implementations. We have shown before, how we can use autoencoders to create feature vectors to extract meaning from time series data. In this section we will focus on predefined metrics that we can use for this kind of analysis. In particular, we will use *TSdist*[8] library for *R* language. It implements a set of different metrics aimed at time series data distance measurements to choose from.

5.1 Metrics of interest

- **DTWDistance**
  Computes the Dynamic Time Warping distance between a pair of numeric time series. This distance finds the minimal path in a distance matrix that defines a mapping between the two series that are being compared. This distance matrix is built by using the Euclidean distance and the path is sought by using dynamic programming.

- **STSDistance**
  Computes the Short Time Series Distance between a pair of numeric time series. The short time series distance between two series is designed specially for series with an equal but uneven sampling rate. However, it can also be used for time series with a constant sampling rate. It is calculated as follows:

$$STS = \int \sum(((y_{k+1} - y_k)/(tx_{k+1} - tx_k) - (x_{k+1} - x_k)/(ty_{k+1} - ty_k))^2)$$

- **DissimDistance**
  Computes the Dissim distance between a pair of numeric series. The Dissim distance is obtained by calculating the integral of the Euclidean distance between the two series. The series are assumed to be linear between sampling points.

- **CorDistance**
  Computes two different distance measure based on Pearson's correlation between a pair of numeric time series of the same length. Computes the value

$$d_1 = \sqrt{2(1 - \rho)}$$

  is computed, where ρ denotes the Pearson's correlation between series x and y

- **CCorDistance**
  Computes the distance measure based on the cross-correlation between a pair of numeric time series. The cross-correlation based distance between two numeric time series is calculated as follows:

$$D = \sqrt{((1 - CC(x, y, 0)^2)/\sum(1 - CC(x, y, k)^2))}$$

  where $CC(x, y, k)$ is the cross-correlation between x and y at lag k.

- **FourierDistance**
  Computes the distance between a pair of numerical series based on their Discrete Fourier Transforms. The Euclidean distance between the first n Fourier coefficients of series x and y is computed. The series must have the same length. Furthermore, n should not be larger than the length of the series.

5.2 Methodology

We have calculated similarity matrices for examined countries based on the metrics of interest for each indicator included in the study. Please find below an example of such calculation. Using these matrices, we can calculate an overall similarity matrix for all examined indicators, divided into groups of interest, by calculating a sum of component matrices. We can analyze results of this kind by plotting them on a 2D plane using UMAP or similar embedding.

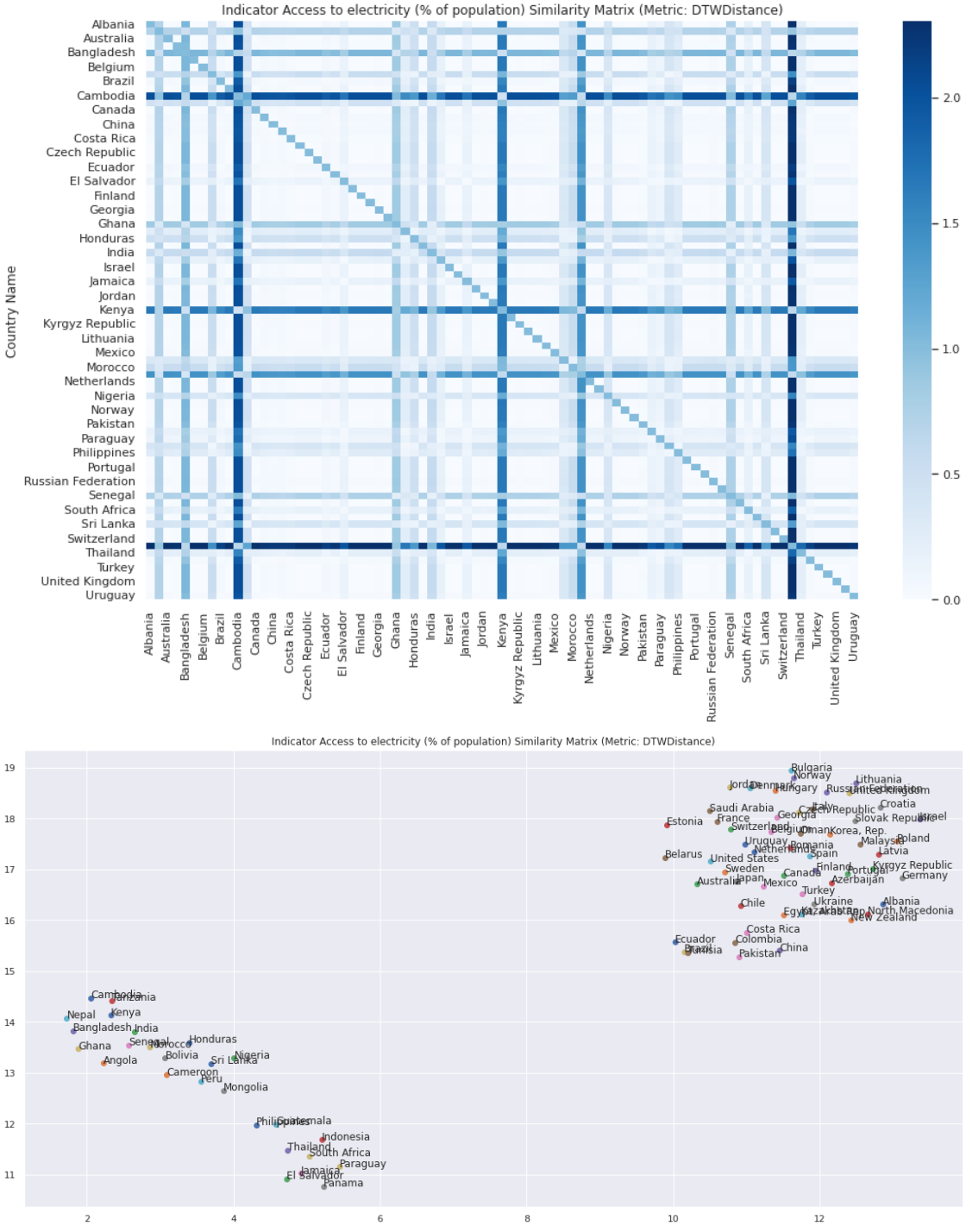


Figure 39: Example: Similarity matrix and corresponding 2D representation (UMAP).

5.3 Analysis results

Using such visualizations, we can analyze examined groups of indicators. Selected results used in following analysis are presented in this paper, all created results and plots can be found in project repository [9].

- Agriculture indicators - This group has to produce a multitude of unexpected results. For instance, Ukraine is placed quite far from similar in terms of climate to Belarus, separated from it by the United States, though very close to not at all similar to itself Panama. Some of these results can possibly be caused by a very big difference in a size of these countries, distorting the results (mind you that some metrics may be more or less sensitive to data magnitudes). Although these quirks, we can also see countries to be placed in positions that are more expected. For instance, we can point to a cluster of small pacific countries.
- Economy indicators - There are some directions that countries of a similar economic positions tend to gravitate towards. For instance, the upper left corner of the plot is occupied by rather poor countries, while the bottom right almost exclusively by more developed ones, like the United States or European Union members. The bottom left seems to attract Asian developing countries (like the Philippines), while the upper right European developing countries (like Belarus). There are some exceptions from these trends, though, and it cannot be said that countries at the most outer positions are outstanding in the regard to appropriate trends.
- Health indicators - Some more obvious clusters than we have expected can indeed be observed on the plot. For instance, most of the rich, western European countries are placed in a cluster near the center of the plot. Interestingly, eastern European countries that have fallen on the east side of the iron curtain and thus has been more influenced regarding the organization of their healthcare system by the Soviet Union, create a separate from mentioned before cluster. This cluster includes former USSR countries, like Georgia or the russian federation. Germany is placed near this cluster, though on it's outskirts.
- All indicators - This combined plot consists of countries divide into some visible cluster. Overall, the clusters are neither fully obvious, nor very odd looking. Some similar countries, like eastern European ones (Estonia, Bulgaria, Romania, etc.) has been placed in a separate clusters. Although, those clusters often includes quirks, like Japan being placed into the eastern European one.

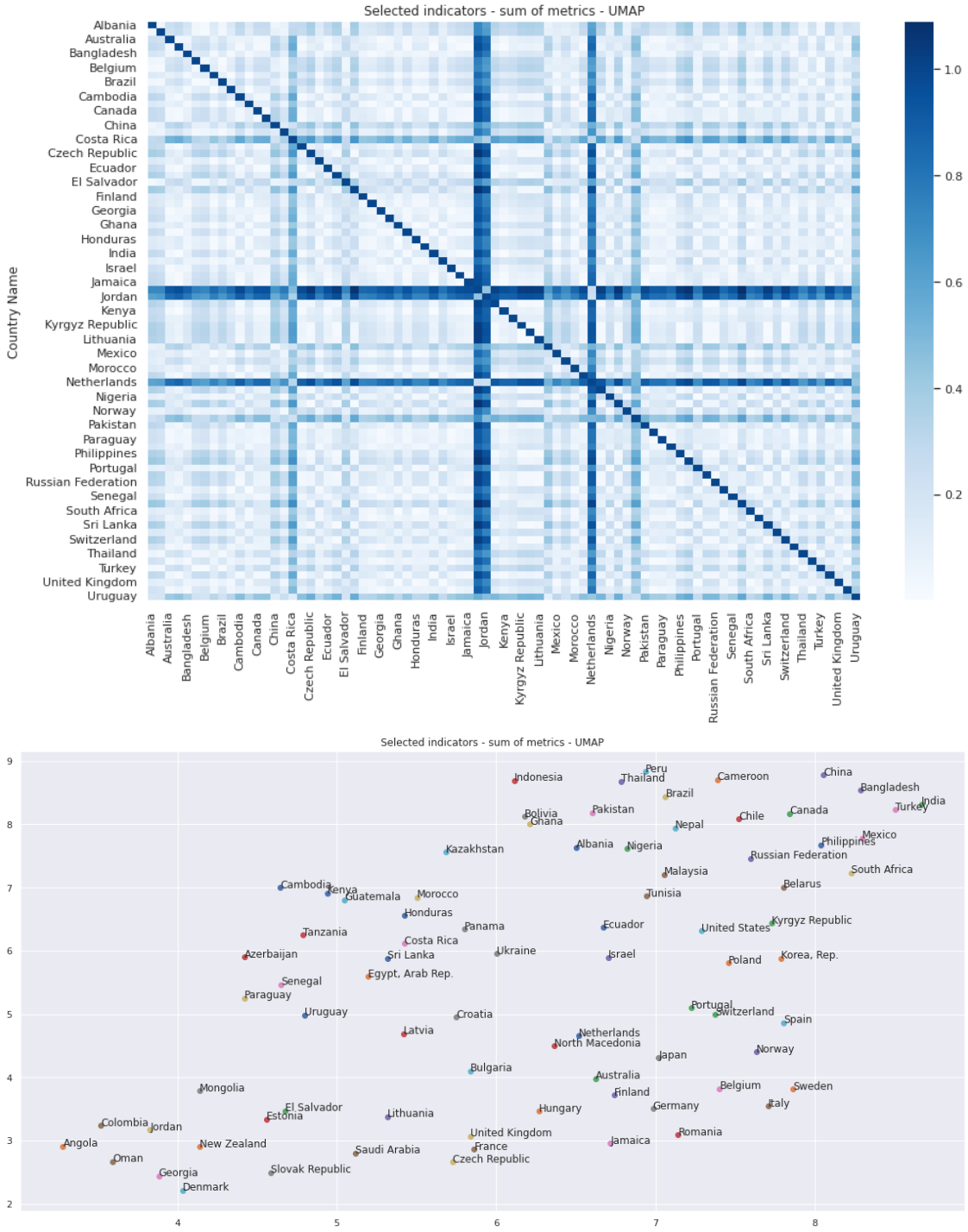


Figure 40: Agriculture indicators: Similarity matrix and corresponding 2D representation (UMAP).

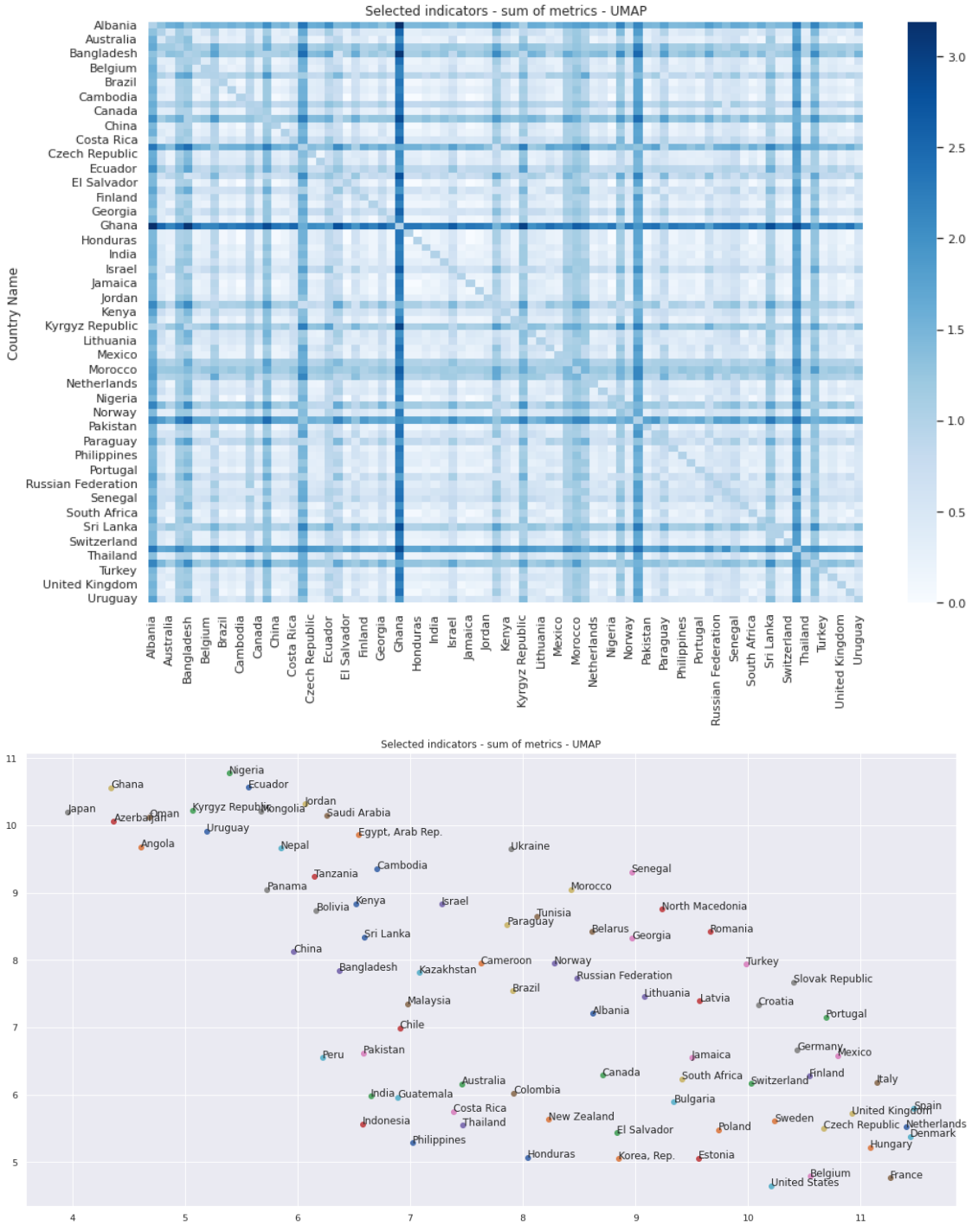


Figure 41: Economy indicators: Similarity matrix and corresponding 2D representation (UMAP).

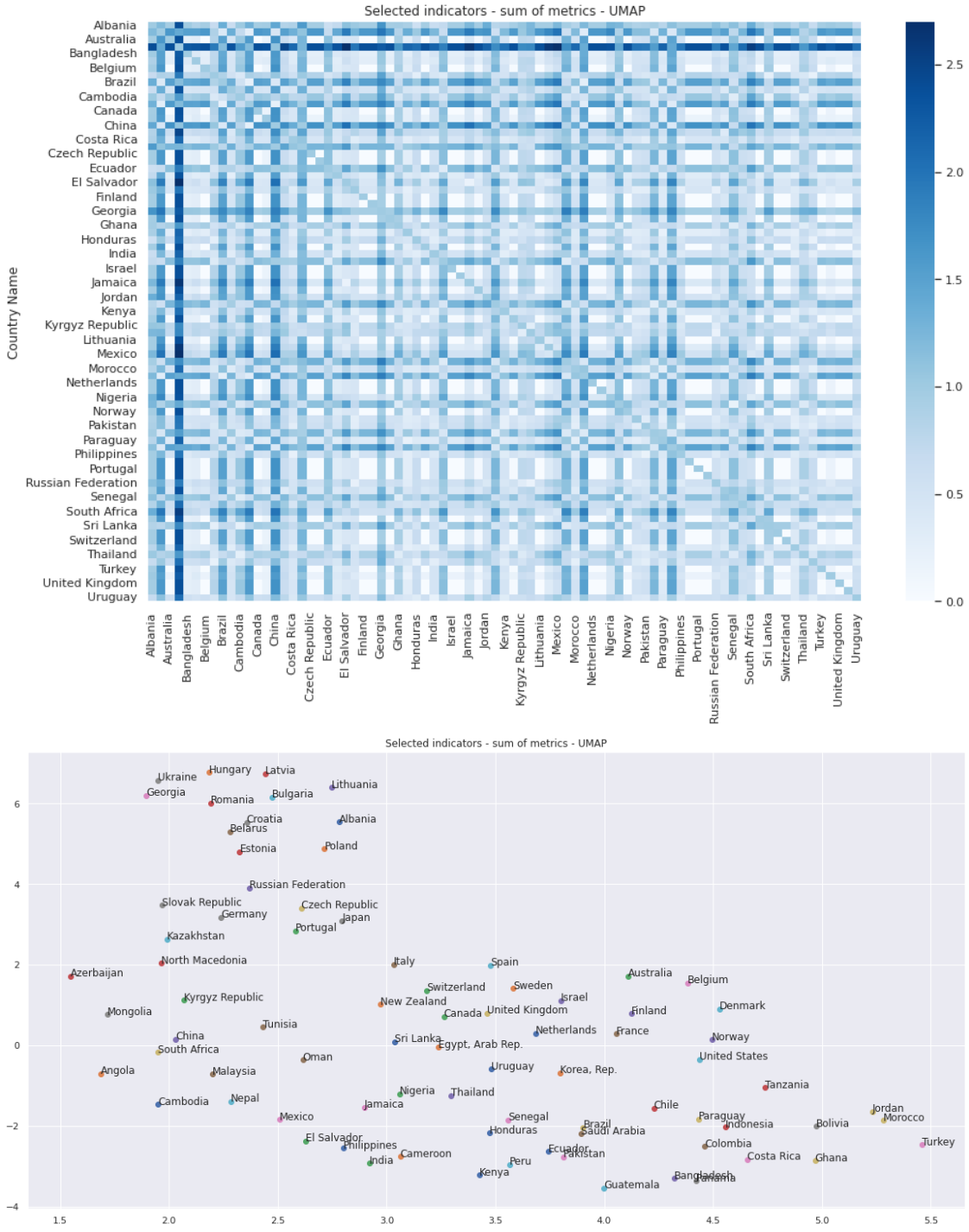


Figure 42: Health indicators: Similarity matrix and corresponding 2D representation (UMAP).

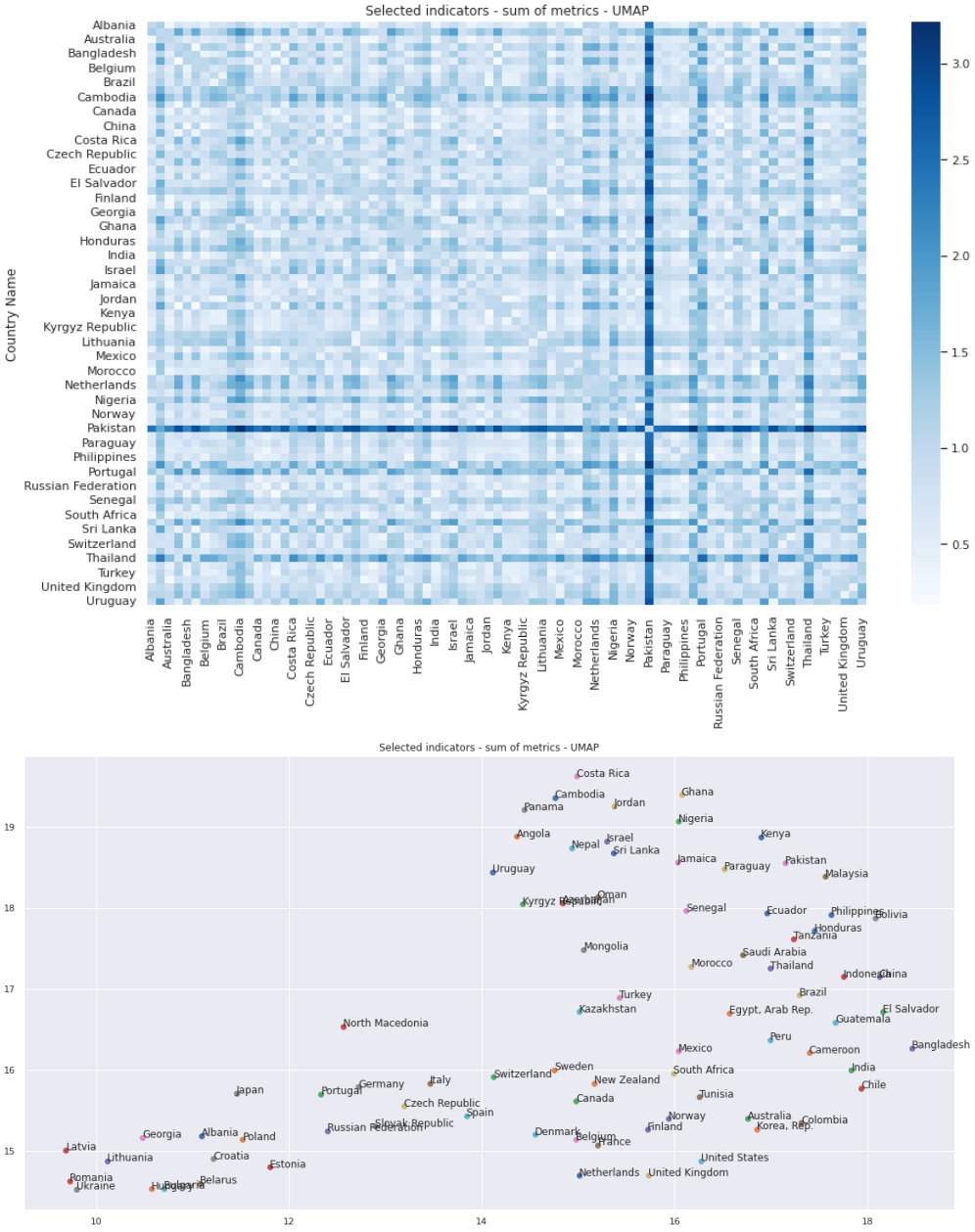


Figure 43: Selected indicators: Similarity matrix and corresponding 2D representation (UMAP).

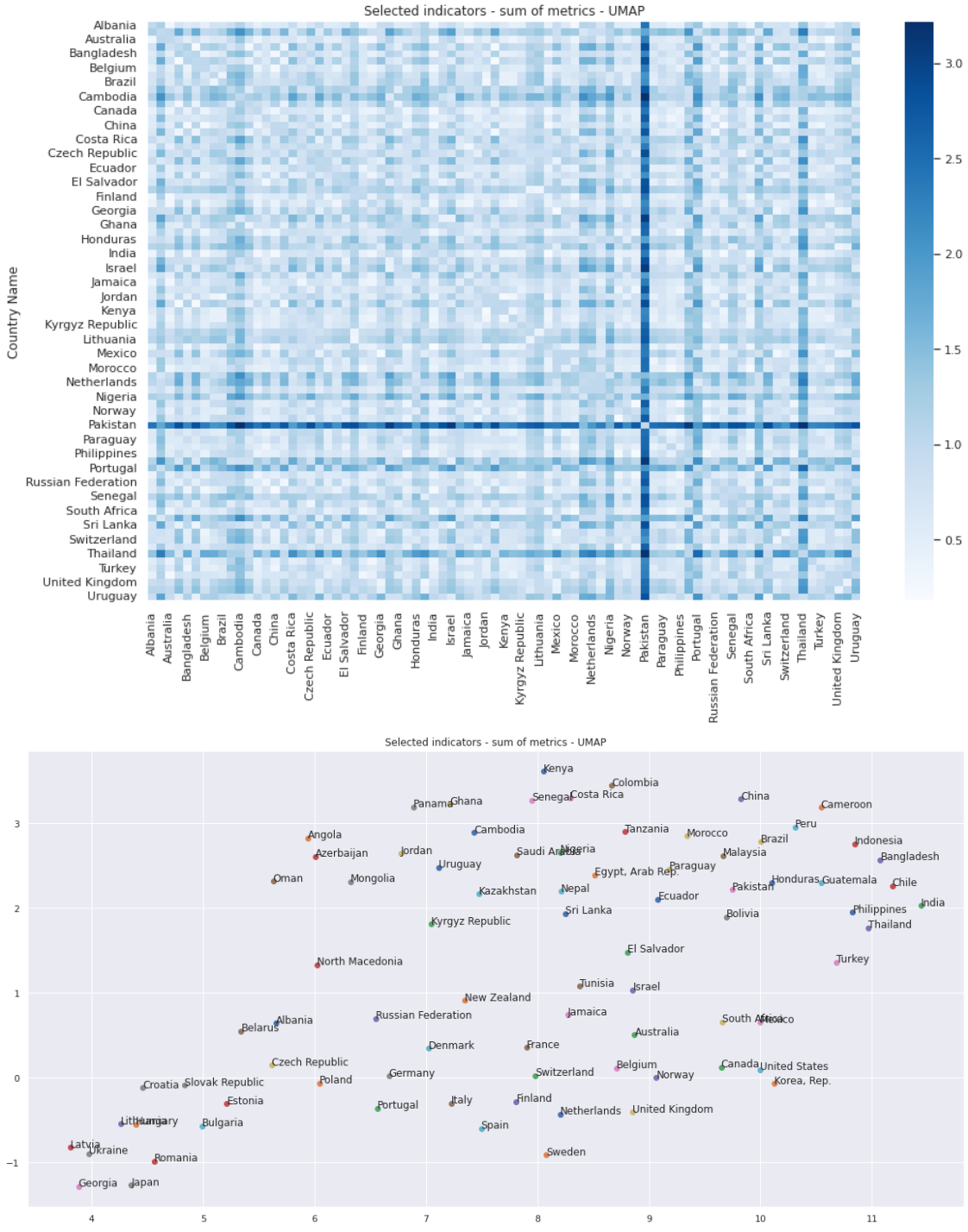


Figure 44: All indicators: Similarity matrix and corresponding 2D representation (UMAP).

6 Summary

The methods we employed enabled us to look at the time series data from many perspectives. We were able to see many obvious international relations, but also managed to discover some weird, or at least nonobvious, ones.

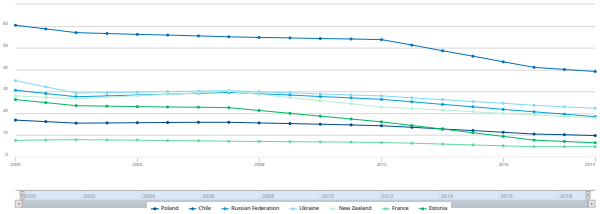
What is more, we could also see the three methods we selected in action and compare their results. One of more interesting comparison can be done between the 2D plots of the default time series and the encoded features. The 2D plots depicting the features extracted by autoencoders show more nonobvious relations – some countries are in clusters they do not belong to at first glance. However, it is important to remember that when the default time series are compared, they are compared point-by-point. Usually, a time series norm is highly dependent on the values; therefore, the norms is implicitly compared at every sample. On the other hand, the autoencoders operate on normalized data, and the norms are additionally added only as one of the features in the latent space. Therefore, the autoencoder plots are based more on the shapes of curves than magnitudes. Usually, we do not compare countries by the shapes of their time series, but just by the values, and thus the autoencoder plots contain the relations which might be initially strange.

Appendix A. Architecture of the autoencoder used for feature extraction

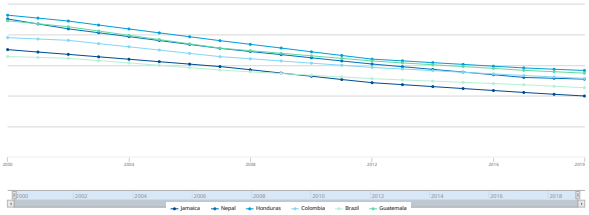
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 19)]	0
reshape (Reshape)	(None, 19, 1)	0
conv1d (Conv1D)	(None, 19, 32)	128
max_pooling1d (MaxPooling1D)	(None, 10, 32)	0
conv1d_1 (Conv1D)	(None, 10, 64)	6208
max_pooling1d_1 (MaxPooling1D)	(None, 5, 64)	0
conv1d_2 (Conv1D)	(None, 5, 128)	24704
max_pooling1d_2 (MaxPooling1D)	(None, 3, 128)	0
flatten (Flatten)	(None, 384)	0
dense (Dense)	(None, 4)	1540
dense_1 (Dense)	(None, 384)	1920
reshape_1 (Reshape)	(None, 3, 128)	0
conv1d_3 (Conv1D)	(None, 3, 128)	49280
up_sampling1d (UpSampling1D)	(None, 6, 128)	0
conv1d_4 (Conv1D)	(None, 6, 64)	24640
up_sampling1d_1 (UpSampling1D)	(None, 12, 64)	0
conv1d_5 (Conv1D)	(None, 12, 32)	6176
up_sampling1d_2 (UpSampling1D)	(None, 24, 32)	0
flatten_1 (Flatten)	(None, 768)	0
dense_2 (Dense)	(None, 19)	14611

Total params: 129,207
Trainable params: 129,207
Non-trainable params: 0

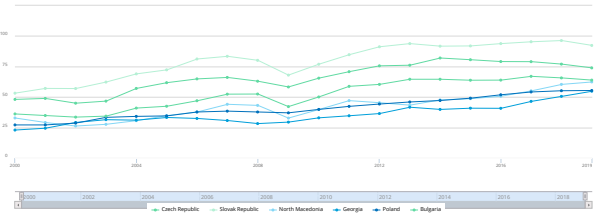
Appendix B. Time series previewed in on the World Bank Open Data website on which we base the analysis in Section 3



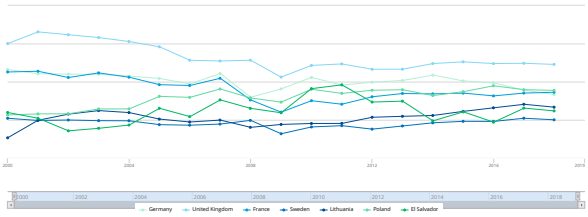
Series : Adolescent fertility rate (births par 1,000 women ages 15-19)
Source: World Development Indicators
Created on: 06/04/2022



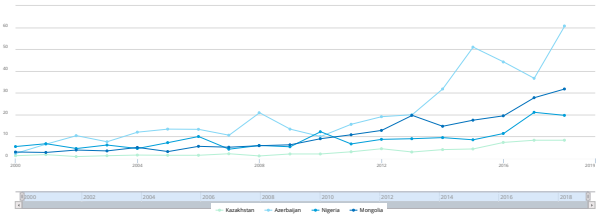
Series : Adolescent fertility rate (births per 1,000 women ages 15-19)
Source: World Development Indicators
Created on: 06/04/2022



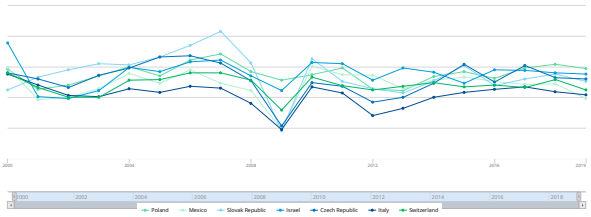
Series : Exports of goods and services (% of GDP)
Source: World Development Indicators
Created on: 06/04/2022



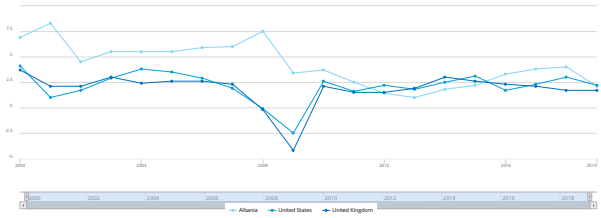
Series : Fertilizer consumption (kilograms per hectare of arable land)
Source: World Development Indicators
Created on: 06/04/2022



Series : Fertilizer consumption (kilograms per hectare of arable land)
Source: World Development Indicators
Created on: 06/04/2022



Series : GDP growth (annual %)
Source: World Development Indicators
Created on: 06/04/2022



Series : GDP growth (annual %)
Source: World Development Indicators
Created on: 06/04/2022

References

[1] *World Bank Open Data*. URL: https://data.worldbank.org/.

[2] *The project repository – autoencoder experiments*. URL: https://github.com/maciektr/worldbank_data_exploration/blob/main/notebooks/feature_extraction/autoencoding_experiments.ipynb.

[3] *The project repository – training the autoencoders*. URL: https://github.com/maciektr/worldbank_data_exploration/blob/main/notebooks/feature_extraction/train_autoencoders.ipynb.

[4] *The project repository – single time series types analysis*. URL: https://github.com/maciektr/worldbank_data_exploration/blob/main/notebooks/feature_extraction/single_types_feature_extraction.ipynb.

[5] *The project repository – time series groups analysis*. URL: https://github.com/maciektr/worldbank_data_exploration/blob/main/notebooks/feature_extraction/groups_feature_extraction.ipynb.

[6] *The project repository – hierarchical clustering by each indicator separately*. URL: https://github.com/maciektr/worldbank_data_exploration/blob/main/notebooks/hierarchical_clustering/hierarchical_clustering_separate_indicators.ipynb.

[7] *The project repository – hierarchical clustering by combined indicators*. URL: https://github.com/maciektr/worldbank_data_exploration/blob/main/notebooks/hierarchical_clustering/hierarchical_clustering_combined_indicators.ipynb.

[8] *Distance Measures for Time Series in R: The TSdist Package*. URL: https://mran.microsoft.com/snapshot/2014-11-17/web/packages/TSdist/vignettes/TSdist.pdf.

[9] *The project repository – Similarity metrics and matrices for indicators*. URL: https://github.com/maciektr/worldbank_data_exploration/blob/main/notebooks/similarity_matrices.ipynb.