Hyperiondev

Task

Unsupervised Learning – Principal Component Analysis

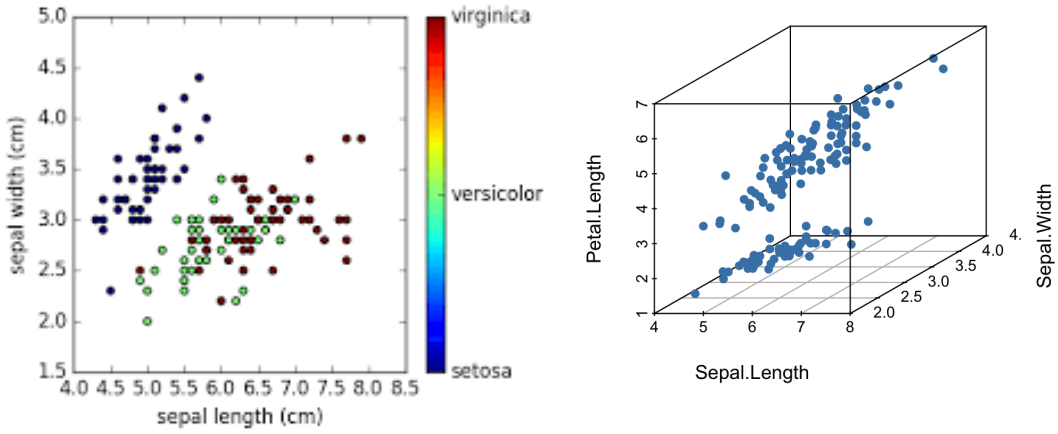
Visit our website

Introduction

In previous examples, we've worked with only a handful of input variables (or features), but in real-world machine learning tasks, the number of variables (or features) is often much larger. As the number of variables grows, data becomes more challenging to work with: relationships between variables become harder to identify, training times increase, and the risk of overfitting rises. To tackle these issues, dimensionality reduction techniques, such as principal component analysis (PCA), t-distributed stochastic neighbour embedding (t-SNE), and linear discriminant analysis (LDA), can help reduce the number of variables while preserving essential information. This task will focus specifically on PCA, a powerful method for simplifying high-dimensional data.

DIMENSIONALITY REDUCTION

Data points can be represented as vectors in a feature space, with each feature corresponding to a dimension. For instance, in the **Iris dataset**, a data point might be represented as [4,3], where 4 is the sepal length and 3 is the sepal width. When using only these two features, the data can be visualised in 2-dimensional space. However, adding another feature, such as petal length, turns the data into a 3-dimensional structure, which can make visualisation and analysis more complex. Dimensionality reduction techniques, like PCA, allow us to simplify this high-dimensional data by reducing it to the most informative dimensions. This makes it possible to visualise even high-dimensional data in 2D or 3D spaces while retaining the essential structure of the dataset.



The image shows two visualisations of the Iris dataset:

- **2D visualisation (left):** This plot uses only two features, **sepal length** and **sepal width**, so each data point is represented as a 2D vector, such as [x, y]. By excluding the petal length feature, the data is projected into a 2D space, simplifying visualisation and allowing easier interpretation of relationships between points based on these two features.

- **3D visualisation (right):** This graph plots the data using three features: **sepal length**, **sepal width**, and **petal length**. Each data point here can be represented as a vector with three components, such as [x, y, z], where x is sepal length, y is sepal width, and z is petal length. Together, these three values uniquely define the position of each data point in 3D space.

Dimensionality reduction with PCA

In cases where we start with a 3D dataset (sepal length, sepal width, and petal length), PCA could help us project this data into a lower-dimensional space, like the 2D plot. PCA identifies combinations of features that capture the most variance in the data, potentially combining sepal and petal features into new dimensions. This retains the core relationships in the data while reducing the number of variables.

For example, in 3D, a point might be represented as [5.1, 3.5, 1.4], where 5.1 cm is the sepal length, 3.5 cm is the sepal width, and 1.4 cm is the petal length. After reducing the data to 2D, the same point could be simplified to [5.1, 3.5], focusing only on the sepal dimensions. In this way, PCA or similar techniques compress the data, making it easier to work with while preserving important patterns and variability.

In datasets where there are only two or three input variables, it's often easy to see how the input variable might impact the dependent variable. Now, suppose that we need to categorise items based on dozens of features, or hundreds, or, as in vision and language problems, thousands. With 10 variables we would need 45 plots to examine the relationship between each pair of variables. Inspecting all of these takes time, and it would be hard to find a pattern that applies to all of the data. This means we need to have methods for dealing with multi-dimensionality in our repertoire.

In many cases, our model building benefits from performing **dimensionality reduction**. Reducing the dimensionality of a dataset:

- simplifies the dataset, facilitating description and visualisation,
- decreases the risk of overfitting in supervised learning, and
- reduces the number of distances clustering algorithms must calculate to determine the dissimilarity between observations (this may improve their efficiency and positively affect their performance).

Feature selection

One way of dealing with multi-dimensional data is to remove features that we know have little impact on the outcome variable of our problem based on domain knowledge. This may involve manually inspecting the input variables and deciding which ones can reasonably be expected to influence the outcome.

However, relying solely on domain knowledge may not always be practical or sufficient. In cases where direct inspection isn't possible or doesn't yield clear insights, we can turn to quantitative methods. These methods help assess the strength of each input variable's relationship with the outcome. One such approach is using univariate statistical tests, which evaluate each feature independently for its relevance to the target variable. For example, scikit-learn provides tools to apply **univariate tests** that score input features based on their individual relevance.

While univariate tests are useful for assessing single feature-to-target relationships, they don't capture interactions or dependencies between features. This is where inter-feature correlation becomes important. Highly correlated features may provide redundant information, and including them could lead to overfitting or model inefficiency. Therefore, examining correlations between features helps identify redundant variables that could be removed, improving the model's performance. Both univariate tests and inter-feature correlations play a crucial role in the feature selection process, helping you refine your dataset by eliminating irrelevant or redundant variables.

INTER-FEATURE CORRELATION

Let's say that you are trying to predict a class in the Iris dataset. There are four features (petal and sepal widths and lengths) and at the target label (which species it belongs to). We know that all four features are valuable in making the prediction. However, we also know that, generally speaking, the longer a petal is, the wider it is (and vice versa). In addition, larger petals generally mean larger sepals, especially if it's just a slightly bigger plant.

What this means is that there is some correlation in the features themselves. To a certain degree, you can predict the size of a petal by looking at the size of a sepal. In other words, you could remove that data and still have a pretty good prediction. This is an example of **redundancy** in the data.

To diagnose redundancy, follow these steps:

1. Create a DataFrame without the target variable (i.e., only features).
2. Use `df.corr()` to get the pairwise correlation coefficients between all features in the dataset.
3. Plot a heatmap of the correlation coefficients.

A heatmap is essentially a 2D grid. When looking at cell (0, 1) in the grid, the value there shows the correlation between features 0 and 1. To analyse the correlation coefficient heatmap, look at each value, which lie between -1 and 1. If it is:

- **close to 1:** it denotes a high positive correlation (i.e., the higher one value gets, the higher the other value gets), and this needs to be removed from the data.
- **close to 0**: there is little to no correlation between features, which is the ideal case.
- **close to -1**: there is a high negative correlation between features (i.e., the lower one value gets, the higher the other gets), which, as with a high positive correlation, is undesirable and needs to be removed.

How do we remove these correlations? One way may be to remove features that are correlated mostly with other features. However, this isn't very practical, as these features still hold some valuable information in them. There is a way to extract this valuable information from the feature and leave the less valuable information behind. This is done by creating **principal components** out of the features using PCA.

PRINCIPAL COMPONENT ANALYSIS

PCA is a dimensionality reduction technique that finds, as the name suggests, the principal components of a set of input variables. Principal components are directions in the feature space of a dataset that reflect where the data are most spread out. Considering a toy example with vectors [[4, 3, 1], [4, 2, 5], [4.1, 5, 6]], the outcome of PCA would reflect that the toy instances have nearly the same value for X_1 (namely: 4, 4, and 4.1). This means feature X_1 does not help us put the data into different categories. However, in the dimensions X_2 and X_3, the data points are more spread out.

How it works

PCA works by computing the **eigenvectors** of the dataset. For any given dataset there are as many eigenvectors as there are features, and each eigenvector has a corresponding eigenvalue. The eigenvectors are the directions in which the data is most spread out, and the eigenvalue describes how spread out the data is (how much variance there is) in that direction. PCA then selects the eigenvectors with the highest eigenvalues and projects the data points into a space that has only those eigenvectors as its dimensions.

Note that principal components (eigenvectors with the least variance) of a dataset are not simply a list of the most informative features. Interpreting the results is therefore not simply a matter of interpreting why certain features got removed. Instead, it is often useful to also think about *correlations* between variables. For example, PCA might successfully reduce the dimensions of a dataset that contains the height, weight, and IQ of a group of adults from 3 to 2 dimensions, since height and weight are more strongly correlated to each other than to IQ.

Extra resource

Please read **this blog post** for more insight into how PCA works, including the role of eigenvectors and eigenvalues.

Let's go through an example where we will explore the correlation between variables, perform PCA on a dataset, demonstrate the importance of scaling data in this context, and use the principal components in a cluster analysis.

EXAMPLE ANALYSIS

In this example, we explore the differences between various countries using unsupervised learning methods such as PCA and clustering techniques. The related code is available in the **unsupervised_example.ipynb** Jupyter Notebook in your folder. The dataset we will be using is from the **World Bank Data Catalog** and contains data on 44 country groups. There are 20 variables for each country group in total, with 19 describing each country group through population statistics, electricity, and technology adoption, as well as economic indicators such as inflation and trade data.

Exploring the data

The country groups can be split into four categories: income, lending, other indicators, and geographic regions. Understanding these categories will become important when we analyse the output of the clustering algorithms later. The income groups are based on the gross national income (GNI) per capita in US dollars. In this dataset, there are also groups where the income groups have been combined, for example low and middle income as shown in Table A.

Table A

	Abbreviation	Income groups
1	LIC	Low income
2	LMC	Lower-middle income
3	MID	Middle income (low and upper-middle income combined)
4	UMC	Upper-middle income
5	LMY	Low and middle income combined
6	HIC	High income

Economies are classified into three groups by the World Bank: the International Development Association (IDA), the International Bank for Reconstruction and Development (IBRD), and “blend”. IDA countries are poor and cannot afford IBRD loans. Blend countries are somewhat better off, qualifying for both IDA and IBRD loans. Table B shows countries in the different lending categories.

Table B

	Abbreviation	Lending categories
1	IBD	IBRD only
2	IBT	IDA & IBRD total
3	IDA	IDA total
4	IDB	IDA blend
5	IDX	IDA only

Other indicators include **demographic dividend** categories, members of the Organisation for Economic Co-operation and Development (OECD), **heavily indebted poor countries**, and **least developed countries** as defined by the United Nations. These additional categories are summarised in Table C.

Table C

	Abbreviation	Other indicators
1	EAR	Early demographic dividend
2	FCS	Fragile and conflict-affected situations
3	HPC	Heavily Indebted Poor Countries (HIPC)
4	OED	OECD members
5	PST	Post-demographic dividend
6	PRE	Pre-demographic dividend
7	LTE	Late-demographic dividend
8	LDC	Least-developed countries: UN classification

Lastly, countries are grouped according to geographic regions in Table D. Some geographic groups are further defined by excluding high-income countries or filtering the countries in a region based on their assigned lending category.

Table D

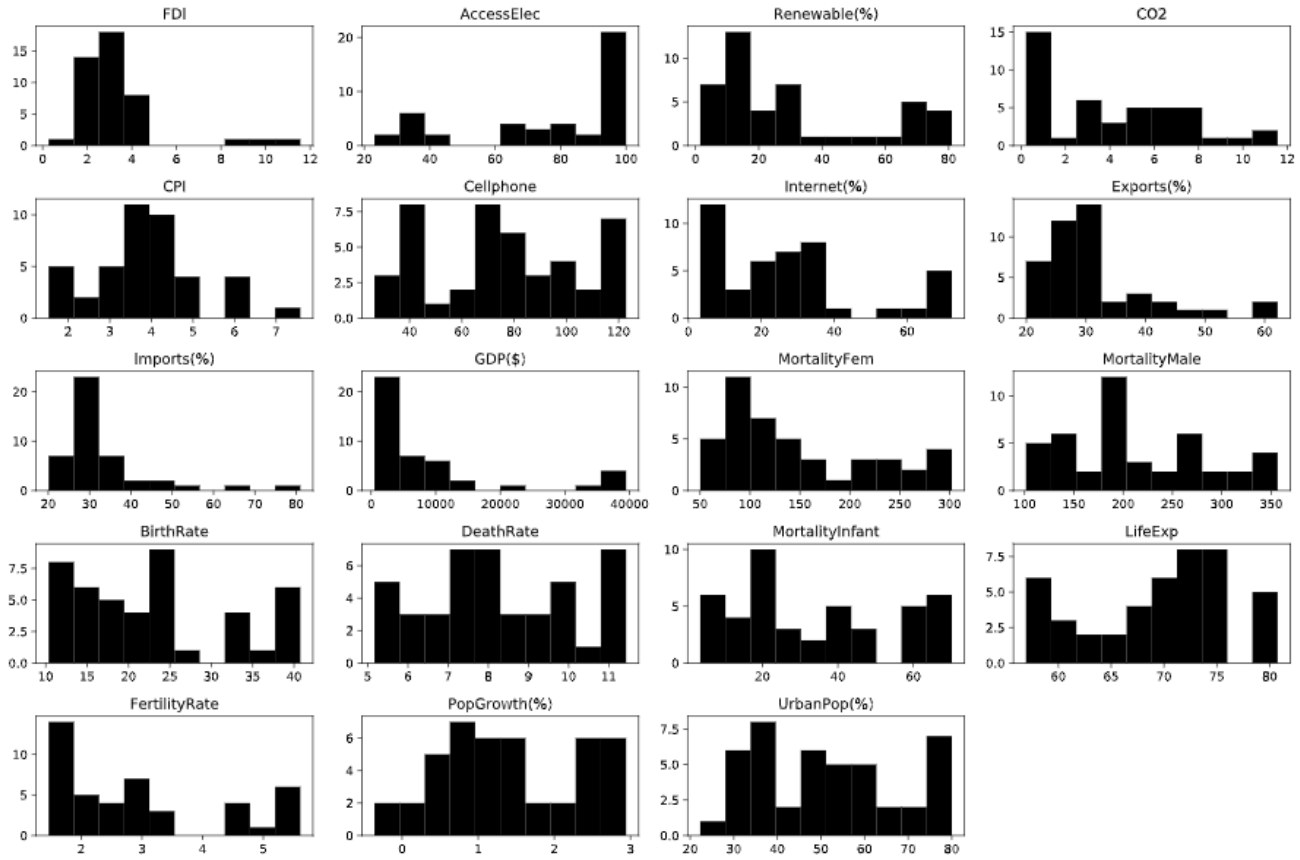
	Abbreviation	Geographic region
1	CEB	Central Europe and the Baltics
2	CSS	Caribbean small states
3	EAP	East Asia & Pacific (excluding high income)
4	EAS	East Asia & Pacific
5	ECA	Europe & Central Asia (excluding high income)
6	ECS	Europe & Central Asia
7	EMU	Euro area
8	EUU	European Union
9	FSM	Micronesia, Fed. Sts.
10	LAC	Latin America & Caribbean (excluding high income)
11	LCN	Latin America & Caribbean
12	MEA	Middle East & North Africa
13	MNA	Middle East & North Africa (excluding high income)
14	OSS	Other small states
15	PSS	Pacific island small states
16	SSA	Sub-Saharan Africa (excluding high income)
17	SSF	Sub-Saharan Africa
18	SST	Small states
19	TEA	East Asia & Pacific (IDA & IBRD countries)
20	TEC	Europe & Central Asia (IDA & IBRD countries)
21	TLA	Latin America & the Caribbean (IDA & IBRD countries)
22	TMN	Middle East & North Africa (IDA & IBRD countries)
23	TSA	South Asia (IDA & IBRD)
24	TSS	Sub-Saharan Africa (IDA & IBRD countries)
25	WLD	World

Now that you have some insight into the country groups, we will have a look at the variables in the dataset. To improve the understanding of the data, the variables are renamed to have more intuitive names such as "**Birthrate**" instead of the original heading of "SP.DYN.CBRT.IN". The full names for each of the features can be found in the **metadata** provided by Carnegie Mellon University. The mean, standard deviation, range, and distribution of each variable, as well as the number of missing values per variable, are summarised in the table below.

	Missing	Mean	StdDev	Min	Max
FDI	0	3.39	2.11	0.27	11.56
AccessElec	0	76.63	26.00	23.09	100.00
Renewable(%)	0	30.77	25.55	1.50	81.01
CO2	0	4.14	3.21	0.22	11.56
CPI	2	3.89	1.29	1.53	7.58
Cellphone	0	76.06	28.70	26.56	122.83
Internet(%)	0	27.39	20.54	3.25	72.29
Exports(%)	0	31.90	9.44	19.93	62.17
Imports(%)	0	32.98	11.35	20.10	80.99
GDP($)	0	**8864.04**	11163.53	522.03	39449.30
MortalityFem	0	**149.35**	76.51	50.16	302.16
MortalityMale	0	**213.08**	72.75	101.46	356.62
BirthRate	0	22.80	9.50	10.36	40.75
DeathRate	0	8.30	1.81	5.17	11.43
MortalityInfant	0	33.49	21.33	3.48	70.22
LifeExp	0	69.19	6.94	56.94	80.75
FertilityRate	0	2.96	1.29	1.47	5.61
PopGrowth(%)	0	1.41	0.92	-0.36	2.94
UrbanPop(%)	0	51.31	16.82	22.30	80.02

At first glance, the GDP per capita variable stands out as having a mean and standard deviation which are significantly higher than the other variables. This makes sense as most of the other variables are percentages or ratios per 1000 people, while GDP per capita is in US dollars. This indicates that **scaling** the data will be useful to keep the GDP per capita from impacting the analysis disproportionately.

We can also get insight into the spread of the data by plotting histograms for each variable:



Missing values

It's common when working with datasets to have missing values. Below is a sample of our data showing some missing data:

Country	Renewable(%)	CO2	CPI	Cellphone
CEB	14.54	6.82	1.84	122.19
CSS	9.10	9.28	3.25	113.63
EAP	16.47	5.11	3.79	69.91
EAR	26.48	2.12	4.58	68.07
EAS	13.73	5.70	3.25	73.66
ECA	6.29	7.52	6.28	122.83
ECS	10.83	7.54	2.39	120.37
EMU	12.83	7.43	1.53	117.10
EUU	12.96	7.35	1.67	118.57
FCS	51.77	0.83	3.62	38.27
FSM	1.50	1.10	nan	26.56
HIC	9.53	11.56	1.96	109.04

There are two missing values in total within the dataset, both in the CPI column. The simplest way to handle missing values is to remove the affected row from the dataset. This can be done with the `dropna` function from pandas. However, sometimes when there are a large number of missing values it would be detrimental to our analysis to remove too many samples.

There are a variety of techniques for substituting missing values with statistical prediction. This process involves estimating a value based on other samples and is generally referred to as "missing data imputation". A simplistic version of this would be to use the `fillna` function to replace the missing values with the mean of the other samples.

A slightly better approach might be to use k-nearest neighbours (k-NN) imputation, which is based on a variation of **Gower's distance.** Consider the first missing variable, the consumer price index (CPI) for the "FSM" country group:

```
Country Groups     Micronesia, Fed. Sts.
FDI                                 0.27
AccessElec                         64.53
Renewable(%)                        1.50
CO2                                 1.10
CPI                                  NaN
Cellphone                          26.56
Internet(%)                        20.00
Exports(%)                         23.51
Imports(%)                         80.99
GDP($)                           2861.77
MortalityFem                      156.79
MortalityMale                     185.22
BirthRate                          23.75
DeathRate                           6.27
MortalityInfant                    32.40
LifeExp                            68.58
FertilityRate                       3.46
PopGrowth(%)                       -0.33
UrbanPop(%)                        22.30
Name: FSM, dtype: object
```

k-NN will investigate other observations with similar values to all the other variables. After the identification, it will get the mean of the consumer prices of these observations and impute the value as the value of consumer price for Bermuda.

Here is the same sample as before, now with the imputed values. It shows us that all the missing variables have been imputed and the data is now ready for PCA.

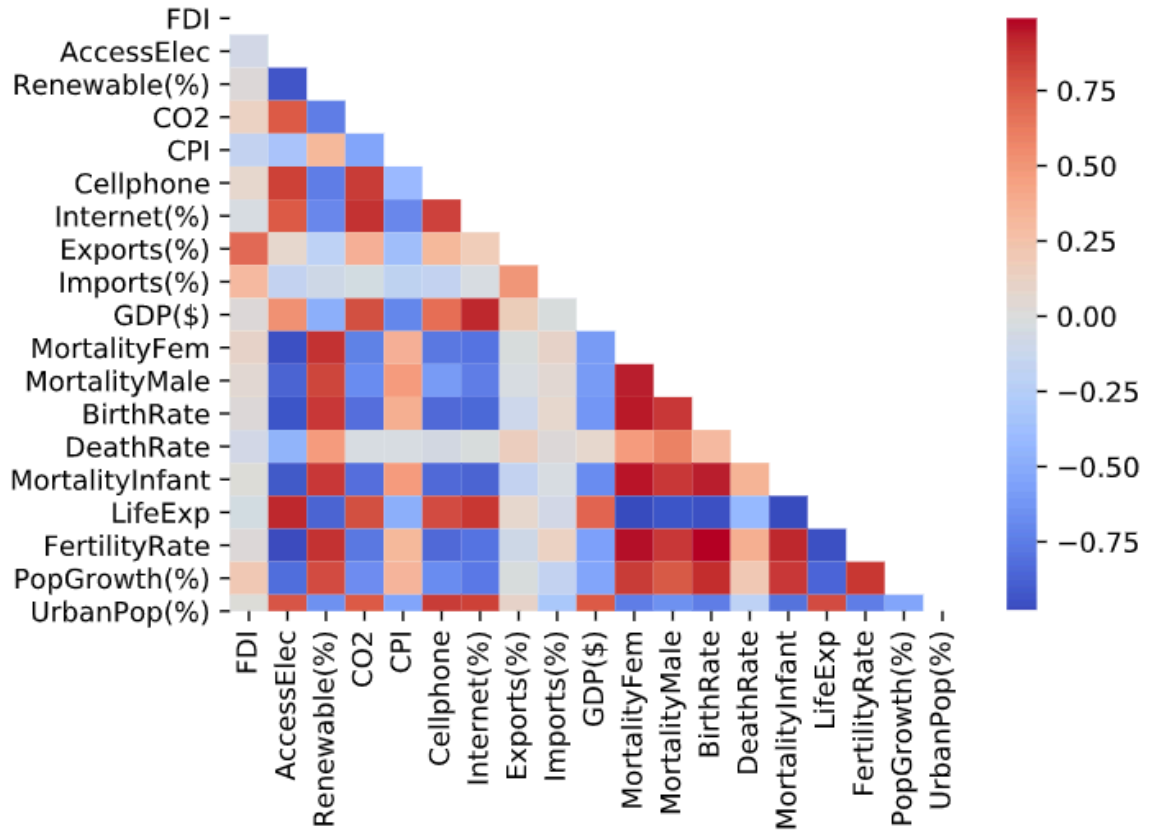
Country	Original	Imputed
CEB	1.84	1.84
CSS	3.25	3.25
EAP	3.79	3.79
EAR	4.58	4.58
EAS	3.25	3.25
ECA	6.28	6.28
ECS	2.39	2.39
EMU	1.53	1.53
EUU	1.67	1.67
FCS	3.62	3.62
FSM	nan	3.89

By using imputation, all 44 country groups can be used in the rest of the analysis.

Correlation analysis

Next, we will explore the correlation between features. In a correlation heatmap, darker colours indicate stronger correlations, while lighter colours represent weaker correlations. Warm colours (like red or orange) typically denote positive correlations, while cool colours (like blue) represent negative correlations.

From the plot below, you can see that most of the variables are highly positively or negatively correlated with each other. For example, access to electricity (% of the population) has a strong negative correlation with renewable energy, female mortality, and the fertility rate. Similarly, the birth rate is strongly positively correlated with infant mortality and fertility rate.



From the correlation plot, it's evident that foreign direct investment (FDI) has a relatively strong positive correlation to imports and exports. Access to electricity is positively correlated to cellphone subscriptions, Internet usage, life expectancy, and percentage of people who live in urban areas, as well as CO_2 emissions!

These correlations are intuitive as people who have electricity can use electronics such as phones, urban areas are more likely to have electricity than rural areas, and generally, countries with electricity access are more likely to have better healthcare,

thereby increasing life expectancy. The predictors that have a strong negative correlation to electricity are the various mortality rates, the fertility rate, and the percentage of renewable energy consumption.

The negative correlation between access to electricity and renewable energy is interesting as it seems to suggest that countries that have high access to electricity are less likely to use renewable energy. This may point to the fact that countries with high access to electricity historically haven't needed to invest as heavily in renewable energy infrastructure as they can already provide for their countries' electricity needs with their existing fossil fuel production or procurement techniques.

There are other intuitive correlations, such as population growth to fertility rates and birth rates. Overall, many variables have strong negative and positive correlations with each other, which makes the data a good candidate for PCA. PCA will be able to reduce variables that encode similar types of differences between countries in a way that requires fewer dimensions.

PCA: Unstandardised data

Recall that PCA is a method for finding the underlying variables (i.e., principal components) that best differentiate the observations by determining the directions along which your data points are most spread out. Since the determination of the principal components is based on finding the direction that maximises the variance, variables with variance that are much higher than the other variables tend to dominate the analysis purely due to their scale.

Importance of components

The table below shows the standard deviation associated with the first six principal components. It also shows the amount of variance that the principal component comprises in comparison to the total variance.

	PC1	**PC2**	**PC3**	**PC4**	**PC5**	**PC6 ...**
Standard deviation	1.12e+04	9.02e+01	2.74e+01	1.52e+01	1.12e+01	8.52e+00
Proportion of variance explained	1.00e+00	6.53e-05	6.01e-06	1.85e-06	1.00e-06	5.82e-07
Cumulative proportion	1.25e+08	1.25e+08	1.25e+08	1.25e+08	1.25e+08	1.25e+08

Let's perform PCA on the "countries" dataset without scaling the data to see what happens. To assess the outcome of our PCA we can use visualisations to explore patterns in the data and the importance of each feature to a principal component. We will use a PCA scatter plot and loading plot combined into a biplot for this purpose.

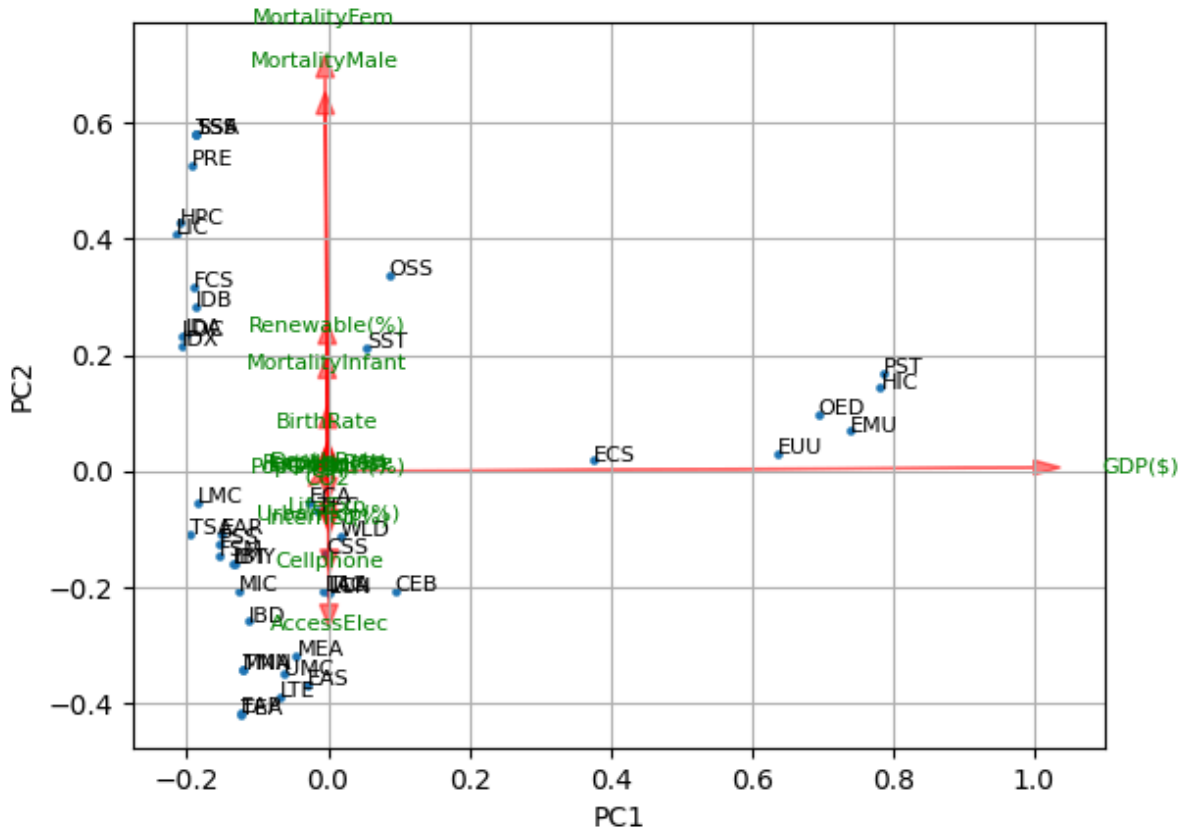
A PCA scatter plot shows the data points transformed from the original data and projected onto the principal component axes. This visualisation helps to understand how the data points are distributed in the reduced-dimensional space. The axes represent the principal components, which capture the most important variance in the data. By examining the scatter plot, you can identify clusters, patterns, and outliers in the data.

A PCA loading plot shows the contribution of each feature to a principal component, influential features, and relationships between features. The direction of an arrow on a PCA loading plot indicates the correlation between the variable and the principal component. A small angle between the principal component and a feature indicates a positive correlation, while a large angle indicates a negative correlation. Similarly, the angle between arrows represents the correlation between the corresponding features. The importance of each feature is indicated by the length of the arrows, which corresponds to the magnitude of the values in the eigenvectors.

A PCA scatter plot and loading plot combined into a single plot is called a biplot. Let's consider the biplot for the PCA we performed without scaling the data.

Countries biplot

If we consider the biplot for the first two principal components, as expected, the first principal component is dominated by GDP, which is on a much larger scale than the other variables (as seen during data exploration). This makes it difficult to see how country groups vary with respect to the other variables, or to read the biplot as most features are overlapping.



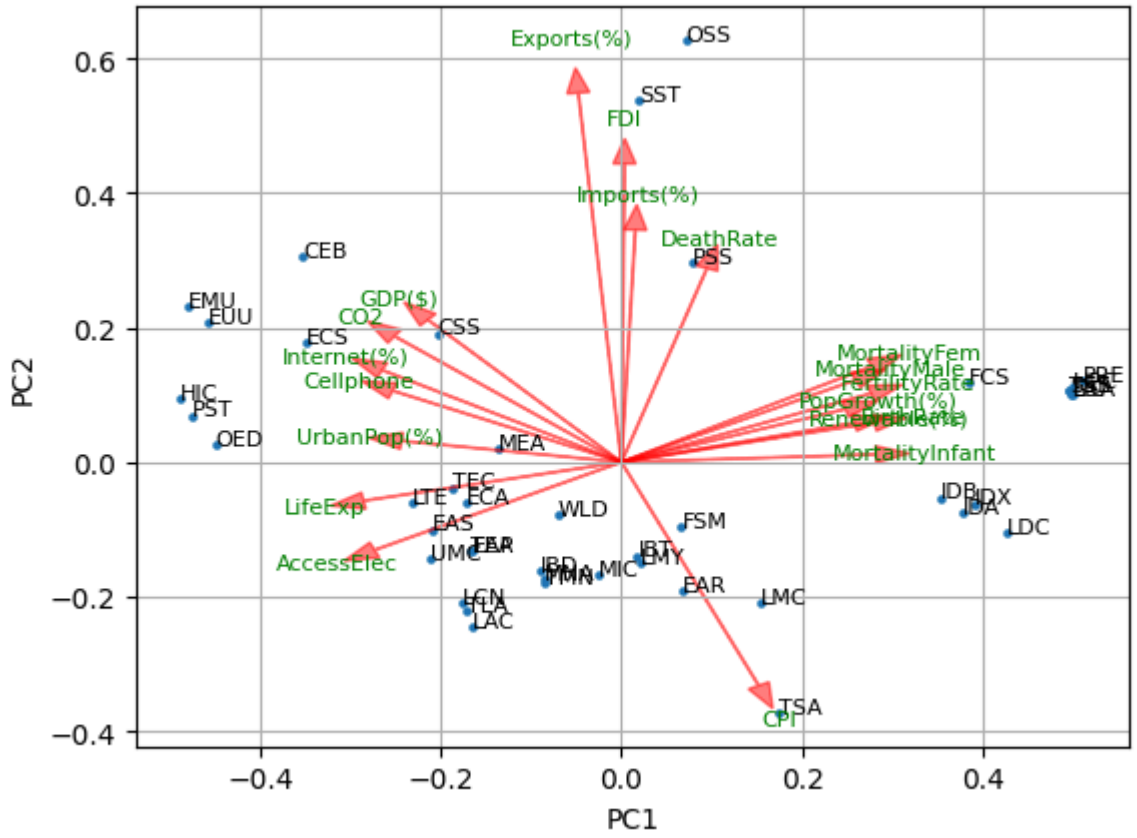
It can, however, be seen that the country groups HIC and PST ("high income" and "post-demographic dividend") have the highest GDP per capita, and the wealth dispersion among country groups is very heavily skewed to the right with there being a large wealth gap between the poor and wealthy country groups. The second principal component deals with the other variables, with the correlated variables going in the same direction. Due to the scale issue, there seems to be little variation based on the second principal component. This is confirmed by looking at the proportion of variance explained, which is 100% and 0.0000653% for the first and second principal components respectively.

In order to learn more about the data through PCA, we scale the data prior to performing PCA. This measure makes it possible to gain more insight into the

relationship between features and principal components, and possible clusters in the data.

PCA: Standardised data

After standardising the data the plot is no longer dominated by features with large numbers.



The first principal component seems to separate the data into 2 directions, which shows the strength of the negative correlations mentioned above. The variables with the largest positive loading values are the various mortality rates, the fertility rate, and renewable energy. The variables with significant negative loading values are technology and electricity access, urbanisation level, GDP per capita, and life expectancy. Therefore, the first principal component seems to summarise a general standard of living.

Countries with a lower standard of living are grouped to the right-hand side in the first principle component, such as the groups FCS (fragile and conflict-affected situations), SSA (Sub-Saharan Africa, excluding high income), etc. These countries tend to have high mortality rates and high population growth rates.

In the centre, we have countries such as LAC (Latin America & Caribbean, excluding high income) and MIC (Micronesia, Fed. Sts.). These are middle-income countries

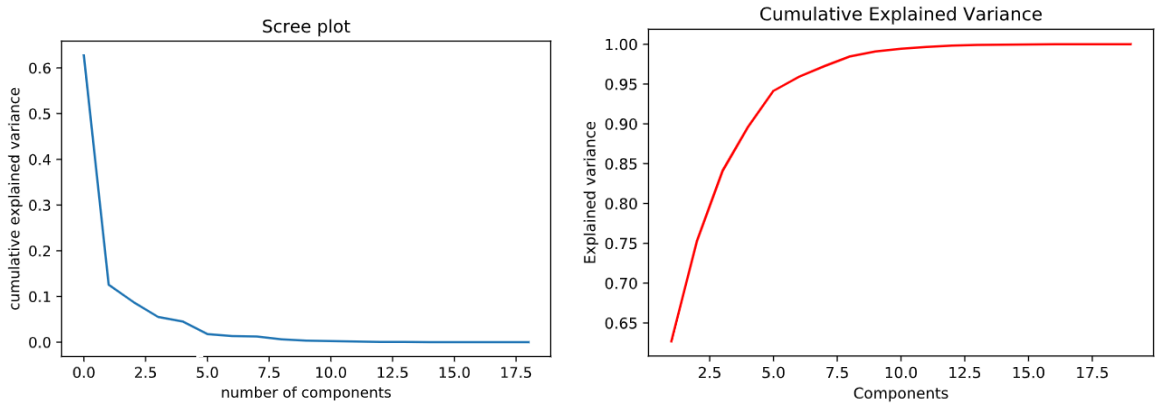
that are still developing but have a higher standard of living than those just discussed.

The countries to the left-hand side in the first principle component are those that have a good standard of living, such as the groups HIC (high income), CEB (Central Europe and the Baltics), EMU (Euro area), EEU (European Union), etc. We see that these countries are correlated with GDP, cell phone and Internet usage, life expectancy, etc.

The second principal component is dominated by exports, imports, and foreign direct investment (FDI), which we saw earlier were positively correlated. This can be summarised as a principal component indicating trade and investment levels. It makes sense that countries that have high investments would be investing in manufacturing products that can be exported. Raw materials for the production may need to be imported leading to the correlation between the variables. The country groups that are extremely above average in these variables are PSS (Pacific island small states), OSS (other small states), and SST (small states). These three likely represent a cluster in the cluster analysis that will be performed below.

Select the number of principal components

In PCA, the first few principal components are the variables that explain most of the variation in the data. As such, when using PCA for dimensionality reduction, we must choose an appropriate number of principal components that explain a significant portion of the variation in our data. This decision will be aided by the scree and cumulative explained variance plots below.



The scree plot displays each principal component on the x-axis and the corresponding explained variance on the y-axis. By observing the “elbow” point in this plot, we can determine the optimal number of components, as it shows where additional components add diminishing returns in terms of explained variance. The

cumulative explained variance plot, on the other hand, helps us understand the cumulative percentage of variance explained as more components are added. This cumulative view allows us to select a number of components that capture a significant portion of the variance, such as 90%, without retaining excessive components.

In this case, the first five principal components together explain around 90% of the variance. We can therefore use them to perform cluster analysis. This is what we refer to as dimensionality reduction: We began with 19 variables and now we have 5 variables explaining most of the variability.

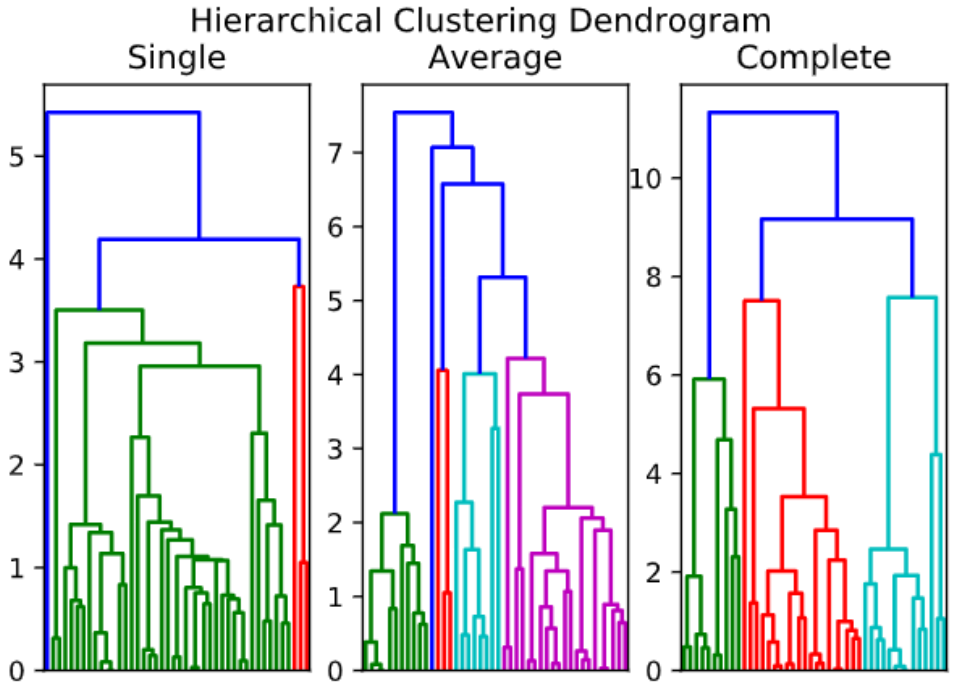
Cluster analysis

We will perform both hierarchical and k-means clustering with the data and compare the results.

Hierarchical clustering

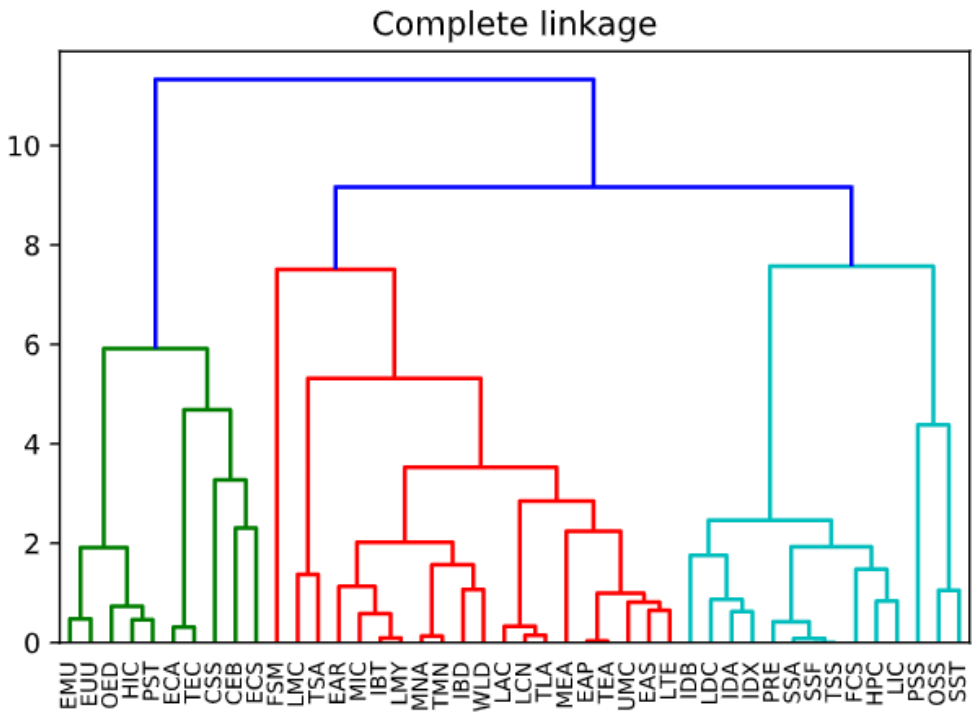
Hierarchical clustering has the advantage that we can see the clusters visually in a dendrogram and don't have to specify the number of clusters before running the algorithm. However, we will have to decide the number of clusters after the algorithm runs.

For the distance metric between observations, Euclidean distance was used, which is the most common way to measure distance. In order to determine the method used to measure the distance between clusters, we plotted the various dendrograms for the single, average, and complete linkage methods.



From the dendrograms above, the complete linkage method creates the most balanced dispersion of clusters and will therefore be the method of choice for the rest of this analysis.

A clearer dendrogram for the complete linkage method is shown below:

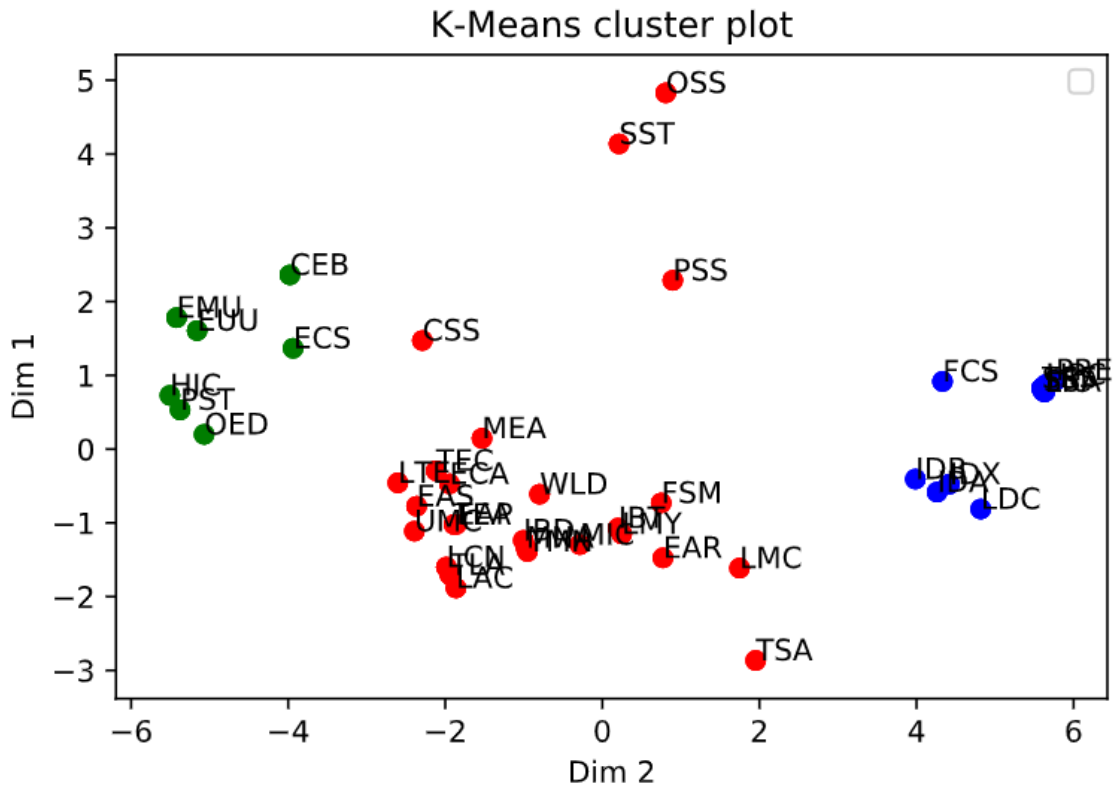


With *k=3*, the clusters are of size 10, 20, and 14 respectively. Within the pale blue cluster, the right-most branch at height 6 contains the "Small states" country groups we noted earlier: PSS, OSS, and SST. These are the same countries mentioned earlier that we expected to form a cluster because of their high values for trade and FDI. The pale blue cluster at large contains country groups that are the least developed, and the red cluster contains developing countries. The pale blue clustered countries are clustered together because of having a lower standard of living based on high mortality rates, lower incomes, and limited access to electricity and technology. The green cluster contains all the wealthy, most developed countries. This cluster has the middle- to upper-income countries who have access to electricity and technology and have high life expectancies. This divide closely mimics what we saw in PCA.

K–means

K-means is a very popular clustering partitioning algorithm that is fast and efficient and scales well for large datasets. It's an iterative process, so observations can switch between clusters while the algorithm runs until it converges at a local optimum. This method is not robust when it comes to noisy data and outliers and is not suitable for clusters with non-convex shapes. Another drawback with k-means is the necessity of specifying "k" in advance.

For our analysis, it seems that the shape of clusters is likely to be regular based on the PCA biplot. K will be set to 3 based on the dendrogram. A visualisation of the clusters is shown in the figure below:



The k-means cluster plot groups countries into three main clusters. Each cluster seems to represent different levels of living conditions or development:

- **Developing countries:** One cluster (the blue one) likely represents countries with lower living standards. These countries may have higher poverty, higher mortality rates, and less access to electricity and technology.
- **Developed countries:** Another cluster (the green one) seems to represent developed countries with higher living standards. These countries likely have lower poverty and mortality rates and better access to resources and technology.
- **Semi-developed countries:** The third cluster (the red one in the centre) appears to represent countries with average living conditions that fall somewhere between developing and developed countries. They may have a mix of characteristics from both groups.

Why do we think these clusters represent living conditions?

The clusters are likely based on living conditions because of how the data is grouped:

- Developing countries with lower living standards often share common characteristics (like higher poverty and lower access to resources), which would make them similar in a dataset. This similarity is what k-means clustering uses to group them into the same cluster.
- Developed countries with higher living standards also tend to share characteristics like access to technology and low poverty, grouping them together in another cluster.
- Semi-developed countries show mixed characteristics, fitting them into a middle cluster.

Why did high-trade countries end up in the semi-developed cluster?

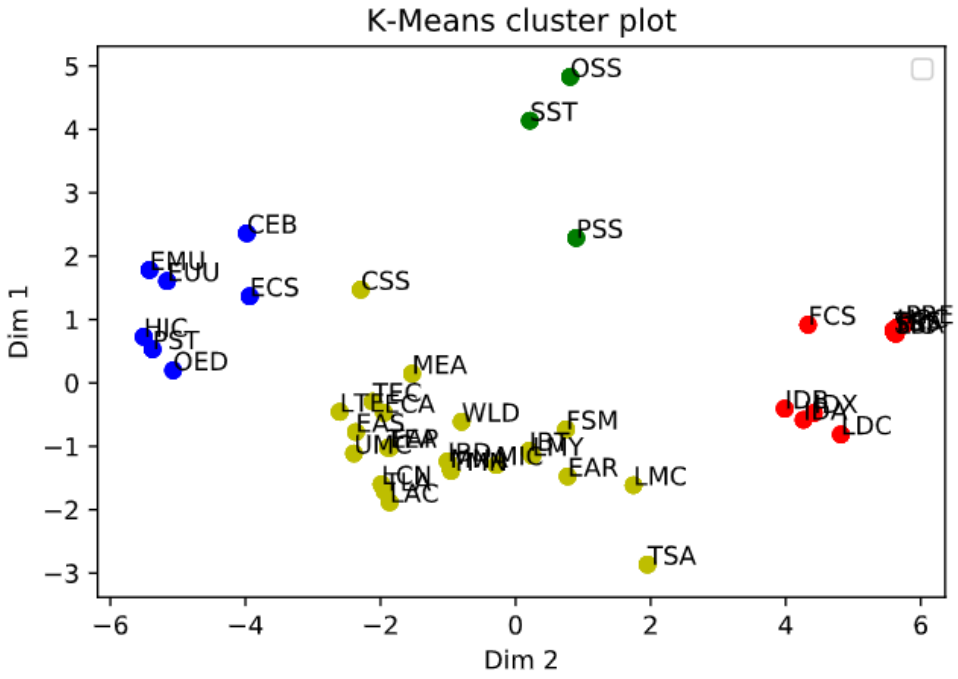
Even though some countries have high trade (high exports, imports, and foreign investments), they fall in the semi-developed cluster. This is because k-means clustering grouped countries more by overall living standards than by trade data. So, while trade alone is important, it isn't enough to place these countries in the developed group since their living conditions are still average overall.

Differences between k-means and hierarchical clustering

Both k-means and hierarchical clustering grouped the countries similarly, but they differ slightly:

- Hierarchical clustering placed more emphasis on trade levels, which separated high-trade countries into different clusters.
- K-means focused more on overall living conditions, resulting in the high-trade countries staying in the semi-developed cluster.

If we were to increase the number of clusters to four, k-means would be able to isolate those high-trade countries into their own separate clusters, as there would be an extra group for distinguishing them based on trade.



Explore the **unsupervised_example.ipynb** notebook for the code behind the PCA example analysis we worked through in this lesson.

Extra resource

An example of an important application of PCA is described in a paper by Labib and Vemuri:

"With the widespread use of computer networks and telecommunication devices, network security has become a primary concern for the developers and users of these networks. As a result, the problem of intrusion detection has grasped the attention of both research and corporate institutions with the aim of developing and deploying effective Intrusion Detection Systems (IDS) that are capable of protecting critical system components against intruders.

Network traffic data collected for intrusion analysis is typically high-dimensional making it difficult to both analyse and visualise. PCA is used to reduce the dimensionality of the feature vectors extracted from the data to enable simpler analysis and visualisation of the traffic."

Read more in the paper about the **application of PCA** for the detection and visualisation of computer network attacks.

Practical task

In this practical task, you will perform an analysis similar to the example provided in this task.

1. Load the Iris dataset into a new notebook, **PCA_task.ipynb**.
2. Please ensure that all steps taken to complete this task are included in the notebook.
3. Explore the features and statistical properties of the dataset.
4. Create a plot of the data using all features to show the positive and negative correlations between them.
5. Scale the data (normalisation or standardisation), then apply PCA to it. Use `n_components = 3`.
6. Plot a biplot with the transformed data.
7. Cluster the data using hierarchical and k-means clustering. Did the clustering techniques identify the three species?
8. After applying PCA, recreate the correlation plot with the transformed data. Compare it to the original, noting any reduction in redundancies.

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Share your thoughts

Please take some time to complete this short feedback **form** to help us ensure we provide you with the best possible learning experience.