Academic Project Report

World Development Indicators

Elaborated By : Groupe Break

4ERP-BI2

Maryem Ben Massaoud

Zeineb Eya Rahmani

Rim Chaouch

Abdallah Moueffek Said

Souhail Cherni

Gratitude

We would like to take this opportunity to express our heartfelt gratitude to the coaches and professors who provided invaluable mentorship and guidance throughout the duration of this academic project. Your expertise, knowledge, and dedication have been critical to our success, and we are immensely grateful for the support you have provided us.

Special thanks go to Mrs Ines Slimen and Mrs Mariem Glaa, coaches for Project Bi, for their guidance and unwavering support throughout the BI phase of the project. Their feedback and insights helped us analyze data and identify business insights that could improve decision-making.

We are also grateful to Mrs Inès Mhaya, coach and professor for Data Marketing, who provided us with the tools and resources we needed to formulate effective marketing strategies based on data-driven insights. Her guidance and expertise were essential in ensuring that we were able to develop strategies that would best meet the project's objectives.

We would also like to express our appreciation to Mrs Ines Chouchane, coach and professor for Innovation and Entrepreneurship, who provided us with invaluable guidance and mentorship during the Innovation phase of the project. Her expertise and insights helped us generate creative solutions to address problems and develop original ideas that were both innovative and feasible.

Mrs Wiem Trabelsi, coach for Data Mining, for her guidance and mentorship during the Data Mining phase of the project. Her expertise and insights helped us employ advanced techniques to extract useful information from large datasets, contributing to the success of this phase.

Finally, we extend our gratitude to Mrs Anissa Othmani, Professor of English, for her guidance and support in helping us refine our communication skills. Her feedback and insights were invaluable in helping us present our project effectively to stakeholders.

We would also like to acknowledge and express our thanks to everyone who assisted us in this project. Your contributions and support were crucial to our success, and we are grateful for your efforts. Thank you all for your invaluable support and mentorship throughout this project.

Contents

General introduction

Business Intelligence (BI) encompasses technologies, applications, and practices that aid in the collection, integration, analysis, and presentation of business data. This technology-driven process enables decision-makers to analyze data and obtain actionable insights, facilitating informed business decisions by executives, managers, and corporate stakeholders. In today's cloud-centric and Big data-driven world, BI has become crucial for organizations to harness their enterprise information, allowing them to track, understand, and control critical business data. BI helps companies gain clarity on their current states and make informed decisions, making it an indispensable tool for efficient data exploitation and business success.

BI solutions serve as decision-support information systems that streamline decision-making processes by transforming company data into useful reports and analyses that improve performance. With BI, decision-makers can focus solely on making decisions instead of expending resources on seeking necessary information for decision-making.

This report will provide a comprehensive overview of the BI process, starting with the business and data understanding phase that covers data source identification and description, business objectives, and data warehouse modeling. The subsequent chapter will delve into the data preparation phase, including internal data handling.

Project’s context

During the academic year 2022-2023, as ESPRIT students, we were assigned to work in teams and develop a web application with a dashboard that would assist decision-making across different fields of study. This project was part of the fourth year PI-BI program and was carried out with guidance from our mentors.

The project comprised distinct stages, each with unique directives. Prior to commencing, our mentors provided a presentation to clarify the project's scope.
The project consisted of various phases, each with its own set of instructions. Prior to beginning each stage, we received coaching and guidance from our mentors to ensure we followed a specific plan. Moreover, we underwent validation processes to confirm that we were meeting the project's objectives and adhering to the established guidelines.

In the BI phase, we scrutinized data to uncover business insights that could enhance decision-making. During the Innovation phase, we generated original solutions to address problems. In the Data Marketing stage, we developed marketing strategies founded on data-derived insights. The Data Mining phase necessitated employing advanced techniques to extract valuable information from vast data sets. Finally, in the English course, we refined our communication skills to proficiently present our project to stakeholders.

Our goal was to create a dashboard that would provide a comprehensive analysis of the human development indicator. The dashboard would display various metrics that had a significant impact on human development, including mortality rate, poverty rate, agricultural production, energy consumption, and other socioeconomic indicators.

To achieve the Sustainable Development Goals (SDGs), we studied the correlation between human development and different factors, such as healthcare systems and policies, climate variability, greenhouse gas emissions, energy consumption, freshwater usage, agriculture, and rural environment. By analyzing these factors, we aimed to provide valuable insights into ways to promote sustainable development and enhance human well-being.

Furthermore, we analyzed the influence of socioeconomic factors on the environment and natural resources, such as land use, water quality, air pollution, and ecosystem health. This analysis aimed to identify potential trends and provide insights into ways to mitigate the negative impacts of human activities on the environment and natural resources.

The project also examined the correlation between socioeconomic development and global issues, such as poverty, hunger, and climate change. By studying these factors, we aimed to provide valuable insights into ways to promote sustainable development and address global challenges.

Overall, our application provided a comprehensive analysis of the human development indicator and its relationship with various socioeconomic, environmental, and global factors. The dashboard included data visualization and provided valuable insights to policymakers, researchers, and stakeholders, facilitating evidence-based decision-making and promoting sustainable development. Our project was a valuable learning experience that helped us apply our skills and knowledge in a practical setting and develop essential skills such as teamwork, communication, and problem-solving, which will be useful in our future careers.

Chapter 1 : Project Understanding

Introduction:

To initiate our project, it is imperative that we conduct comprehensive research and exploration. Firstly, we must conduct a thorough analysis and comprehension of our data. Subsequently, we should investigate similar solutions that align with our project goals. Additionally, we need to examine and select appropriate methodologies that can enable us to successfully tackle our project, based on the available information and our desired outcomes.

I .Problematic

Our issue is a fundamental one that touches on the heart of modern civilization: how can we achieve sustainable development goals? The United Nations has identified 17 specific Sustainable Development Goals (SDGs) to guide countries, organizations, and individuals in their efforts to build a better future for all. These SDGs aim to end poverty, protect the planet, and ensure that all people enjoy peace and prosperity by 2030. Achieving these goals requires a collaborative and holistic approach that addresses a wide range of issues in various sectors.

One of the key areas that require attention is the issue of health. Despite significant advancements in healthcare, people still suffer from preventable diseases, and a large number of them die prematurely. In many developing countries, the problem is particularly acute, with millions of people lacking access to basic healthcare services. To achieve sustainable development goals, we must tackle this issue by reducing the mortality rate due to communicable and non-communicable diseases and ensuring good health for populations. Promoting well-being at all ages must be a priority.

Another crucial sector that requires our attention is agriculture. Agriculture is the backbone of many economies, but it faces numerous challenges, including environmental degradation, loss of usable agricultural land, and food insecurity. These challenges are particularly acute in developing countries, where small-scale farmers struggle to make a living, and food is often scarce. To promote sustainable agriculture, we must address all these issues and find ways to promote economically viable, healthy, and socially equitable agricultural practices. We need to ensure that agriculture not only meets our present needs but also the needs of future generations.

Climate change is another pressing issue that demands our attention. The impact of climate change is already being felt worldwide, and if left unchecked, it could have severe consequences for the planet and its inhabitants. The link between climate change and sustainable development is undeniable, as the former hinders the latter. Therefore, we must take urgent measures to combat climate change and its impact on our environment, our societies, and our economies. This requires a collective effort from governments, businesses, and individuals worldwide.

Finally, poverty is a major obstacle to sustainable development. Poverty is a complex issue that manifests in various forms, from lack of access to basic needs like food and shelter to social exclusion and marginalization. Achieving sustainable development goals requires us to take measures to fight and end poverty in all its forms and everywhere in the world. This includes promoting inclusive economic growth, creating jobs, improving access to education and healthcare, and reducing inequality.

Achieving sustainable development goals is a complex and multifaceted challenge that requires us to address a wide range of issues in various sectors. To succeed in this effort, we must work together, collaborate across borders, and focus on promoting holistic solutions that prioritize the well-being of people and the planet.

II. Study of existante

1. Sun power

SunPower is a company committed to sustainability and environmental responsibility. They provide dashboards to assist businesses in their decision-making process.

The SUNPOWER dashboard offers insights into the energy sector, including specific features such as displaying energy rates for each hour of the day, the total energy consumption rate, and the number of CO2 emissions avoided in tons.

2. Agrivi

Agrivi is an agricultural technology company that provides an agricultural management platform. The platform consists of a suite of features that provide the tools and information that some sustainable development companies need to make informed decisions. Among these features are

1)Productivity Presentation: It displays productivity data in the form of a table with a date filter. This allows users to analyze and track productivity trends over specific time periods.

2)Maps for Factor Visualization: The platform also includes maps that visualize the usage of various factors that can impact production. Users can filter the map data by date, enabling them to analyze the spatial distribution and patterns of these factors over time.

3)Agrivi's platform empowers sustainable development companies with valuable insights and visualizations, helping them make informed decisions about their agricultural operations.

3. Health Catalyst

Health Catalyst provides a data analytics platform that includes dashboards and reports to facilitate decision-making for companies working on sustainable development.

Among the functionalities of this dashboard, there is the presentation of hospital-acquired disease rates with a date filter. It also displays the length of stay for patients with the ability to filter them by date. Additionally, there is a presentation of the overall patient satisfaction rate filtered by date in the form of maps.

Health Catalyst's platform empowers companies in the sustainable development sector with valuable insights and visualizations, enabling them to make informed decisions regarding healthcare data and patient outcomes.

III. Functional needs

1. Dashboard Display:

The application displays KPIs such as HDI rate, mortality rate, poverty index, agricultural production, renewable energy consumption, health expenditure, health workers, R&D expenditure, pesticide use, agricultural land, annual precipitation.

2. Prediction Form:

The application also includes a prediction form that enables users to predict future HDI values based on historical data.

3. Data Management:

To ensure data reliability, the application has a database that stores the KPI data, and the data is easily accessible. The application is also capable of calculating KPIs based on the available data.

4. Login System:

A secure login system is implemented to restrict access to the dashboard and the prediction form, enabling users to create an account, log in, and log out.

IV. Non-Functional needs

Non-functional requirements play a significant role in influencing the user's overall experience and performance, making it imperative not to overlook them. To fulfill these requirements, the following criteria need to be fulfilled:

➢**Performance:**

The application should provide fast data retrieval, processing, and rendering capabilities to ensure real-time or near-real-time analysis and visualization of data.

➢**Scalability:**

The application should be able to handle growing data volumes, user loads, and concurrent users without significant degradation in performance. It should scale horizontally or vertically to accommodate increasing demands

➢**Security:**

The application should enforce strict access controls, authentication, and authorization mechanisms to protect sensitive business data and ensure data privacy. It should also implement encryption and secure communication protocols.

➢**Compatibility:**

The application should be compatible with different web browsers, devices (desktop, mobile), and operating systems, ensuring a consistent experience across platforms.

Conclusion

In conclusion, non-functional requirements are essential to consider when developing any application. These requirements, such as performance, scalability, security, and compatibility, play a significant role in shaping the user's experience and performance. Neglecting these requirements can lead to unsatisfactory results, even if the application's functional requirements are met. Hence, it is essential to ensure that the application fulfills these non-functional requirements to deliver an effective and efficient solution. By doing so, the application can perform optimally, handle large amounts of data, maintain security, and ensure a consistent user experience across various platforms. Therefore, considering these non-functional requirements is a crucial factor in creating successful applications.

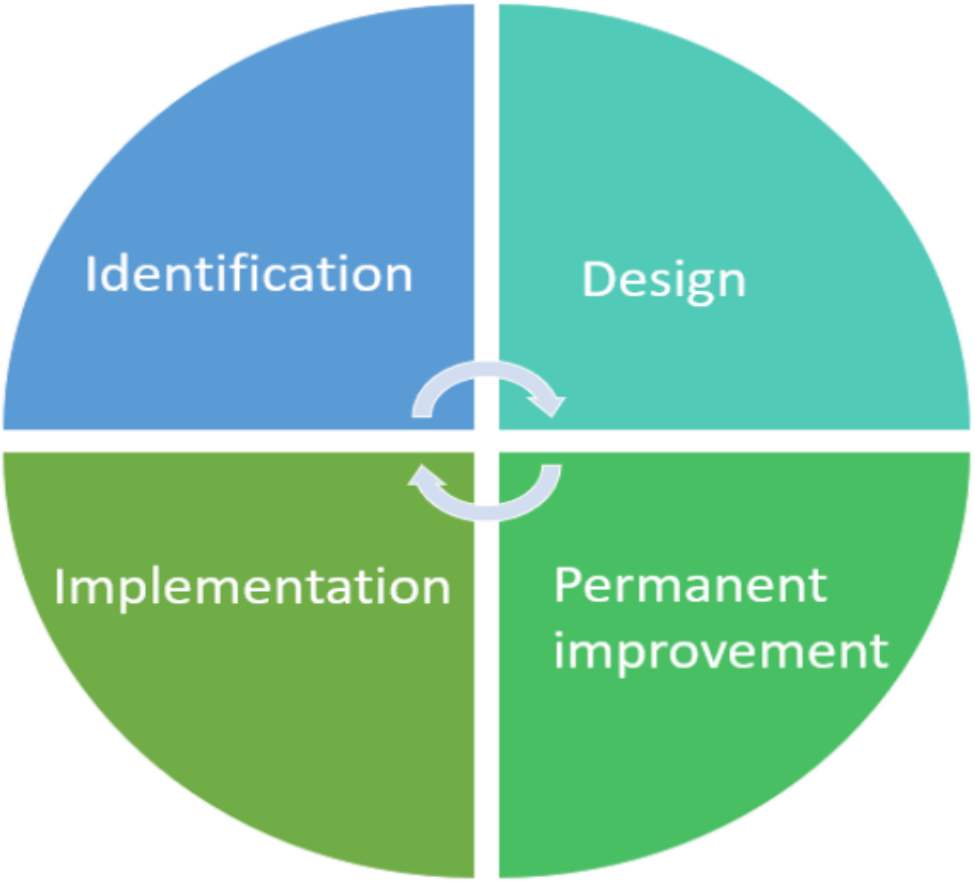
V. Work methodology

1. GIMSI

1.2 Principle :

The GIMSI methodology (General Access to Decision Information using a Methodology-based System that facilitates the expression of a company's individuality) is a comprehensive approach to implementing project steering performance, also known as Business Intelligence, with a focus on human decision-making situations. The methodology

is based on four phases, each with a specific goal and process.



1. Identification

The first phase of GIMSI is Identification, which involves identifying the problem or opportunity that the project aims to address. This phase also involves defining the project's scope, objectives, and stakeholders, as well as determining the available resources and constraints.

2.Design

The second phase of GIMSI is Design, which involves developing a detailed plan for the project based on the information gathered in the Identification phase. This includes defining the project's architecture, selecting the appropriate technologies, and designing the project's data model.

3.Implementation

The third phase of GIMSI is Implementation, which involves putting the plan into action by building and testing the project. This includes developing the necessary software components, integrating the data sources, and testing the project's functionality and performance.

4.Permanent improvement

The fourth phase of GIMSI is Permanent improvement, which involves ongoing monitoring and optimization of the project. This includes identifying and addressing any issues or inefficiencies, as well as implementing new features or functionalities to improve the project's performance and effectiveness. This phase also involves evaluating the project's success and determining whether any changes are necessary to ensure its continued success.

1.2 Advantages :

-The GIMSI approach relies more on cooperation and communication in both directions between leaders .

-Another noticeable difference is the fact that the GIMSI approach aims at the satisfaction of all stakeholders, that is to say that of customers, shareholders, partners, staff and the public

-The GIMSI approach offers regular external audits because the dashboard system offered by the GIMSI method is first and foremost a decision system .

1.3 Weaknesses :

-A strategic management system is not necessary or required by GIMSI but can be implemented if needed

-A standard performance measurement system needs to be implemented.

2. CRISP

Introduction

The CRISP-DM methodology, which stands for Cross-Industry Standard Process for Data Mining, is an open standard model that outlines a set of common approaches used by data mining experts. This methodology is widely used and comprises six distinct phases:

2.1 Principle:

The principle behind CRISP-DM is to provide a structured approach for data mining projects that ensures the end results are accurate and reliable. The six phases of the methodology are designed to guide the data mining process from start to finish and help ensure that the final results meet the needs of the business.

1. Business Understanding

The first phase of CRISP-DM is Business Understanding, which involves identifying the business objectives of the project and determining what questions need to be answered in order to achieve those objectives. This phase also involves determining what resources are available for the project and what constraints may exist.

2. Data Understanding

The second phase of CRISP-DM is Data Understanding, which involves gathering and reviewing the data that will be used for the project. This includes assessing the quality of the data, identifying any missing or incomplete data, and determining what data is relevant to the project.

3. Data Preparation

The third phase of CRISP-DM is Data Preparation, which involves cleaning and transforming the data so that it can be used for analysis. This includes removing duplicates, filling in missing data, and converting data into a format that can be analyzed.

4. Modeling

The fourth phase of CRISP-DM is Modeling, which involves developing and testing various models to determine which one best fits the data and meets the objectives of the project. This includes selecting the appropriate algorithms and techniques to use, as well as refining the models to improve their accuracy.

5. Test and Evaluation

The fifth phase of CRISP-DM is Test and Evaluation, which involves testing the models against real-world data to determine their effectiveness and accuracy. This phase also involves assessing the results of the analysis and determining whether they meet the business objectives of the project.

6. Deployment

The final phase of CRISP-DM is Deployment, which involves putting the results of the analysis into action. This includes creating reports and visualizations that communicate the findings to stakeholders, as well as implementing any changes or recommendations that were identified during the project.

2.2 Advantages :

- CRISP-DM can be implemented without much training, organizational role changes, or controversy.
- CRISP-DM provides strong guidance for even the most advanced of today's data science activities .

2.3 Weaknesses:

- On the other hand, CRISP-DM's documentation requirements might unnecessarily slow the team from actually delivering increments.
- Others argue that CRISP-DM, as a process that predates big data, "might not be suitable for Big Data projects due its four V's"

VI. Used tools

1. Software used

➢ Talend

Talend Open Studio is a software tool developed by Talend that operates as an open-source ETL (Extract Transform Load) software. Its primary function is to contribute to the decision-making process by facilitating the data integration procedure. The software is equipped with the capability to create an all-inclusive ETL and Data Integration pipeline.

➢ Power Bi

Microsoft's Power BI is a business analytics service that is designed to provide interactive visualizations and business intelligence functions. The service offers a simple user interface that enables end-users to create their own reports and dashboards. Additionally, Power BI includes data warehouse capabilities such as data preparation, data discovery, and interactive dashboards.

➢ Excel

Microsoft Excel, commonly known as Excel, is a software application created by Microsoft that provides users with the ability to create, modify, and analyze data utilizing spreadsheets. It is a robust tool for managing and analyzing data, and its usage spans across numerous industries and sectors.

➢ **PostgreSQL**

PostgreSQL, also known as Postgres, is a relational database management system (RDBMS) that operates on an open-source platform. It is renowned for its robustness, scalability, and vast array of features. The software provides support for both structured and unstructured data, and it operates with ACID compliance, ensuring that data integrity and reliability are maintained.

2. Language used

➢ **Python**

Python is an interpreted, high-level programming language that has widespread usage in multiple domains, including web development, scientific computing, data analysis, artificial intelligence, and automation.

➢ **Html**

HyperText Markup Language (HTML) is the primary markup language that web developers use to construct and organize web pages. HTML offers a comprehensive set of tags that determine the layout and substance of a webpage.

➢ **Streamlit**

Streamlit is a Python library that simplifies the process of creating web applications for machine learning and data science projects. With Streamlit, developers can quickly and easily create interactive dashboards, data visualizations, and other web-based tools that allow users to interact with their data in real-time. Streamlit provides an intuitive and

flexible interface for building custom web applications, without requiring extensive knowledge of web development technologies such as HTML, CSS, and JavaScript. It is open-source and has gained popularity in the data science community for its ease of use and speed of development.

Streamlit

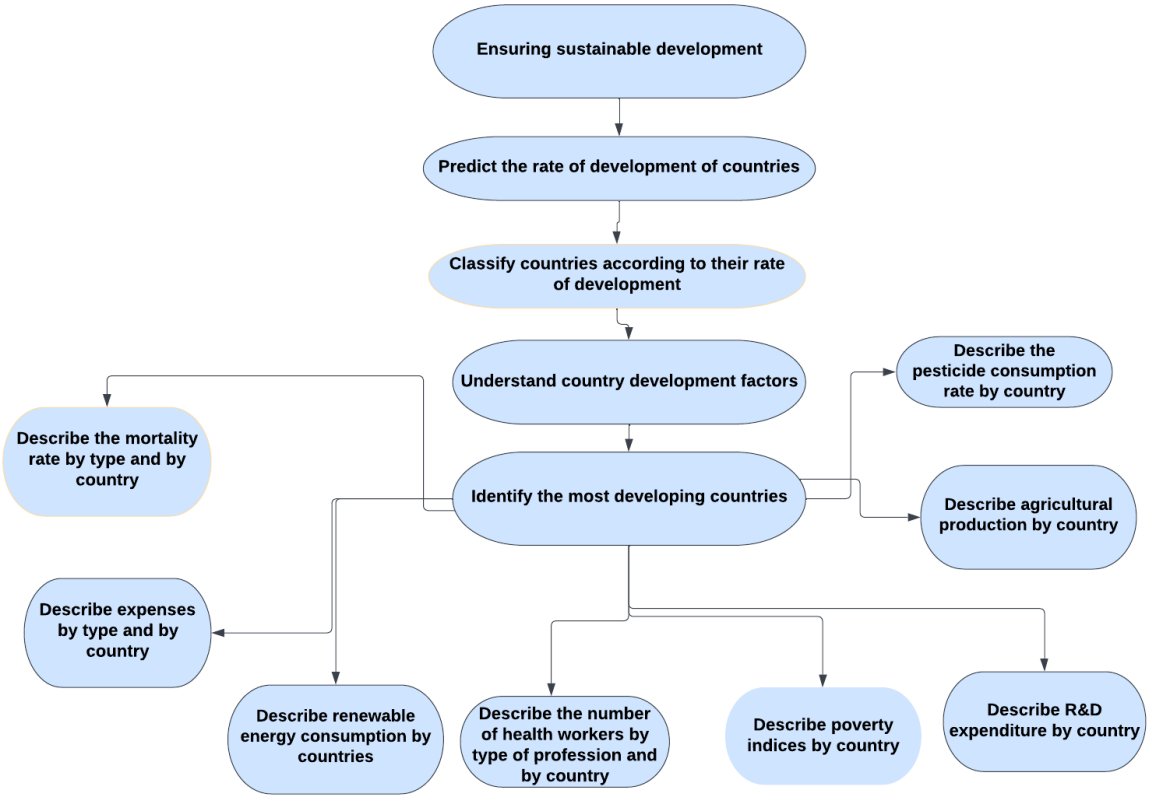
Chapter 2 :Data Warehouse Modeling

Introduction:

The second chapter of our project delves into the technical aspect of our data warehouse. However, before we proceed, it is crucial to familiarize ourselves with data modeling techniques and identify the most suitable approach for our needs. We must create a model for our data warehouse as a part of the conception phase of our GIMSI methodology, where we aim to determine what needs to be accomplished.

I. Objectives

Currently, we are in the first step, which involves establishing the project's goals, as demonstrated in the equivalence tree below:



II. Modeling techniques

In the second chapter of our project, we explore the technical aspects of our data warehouse. However, prior to this, we need to familiarize ourselves with data modeling techniques and identify the most suitable approach for our specific requirements. Ultimately, we must develop a comprehensive model for our data warehouse.

1.Star Schema

Star schema was chosen in our data model ,which is the simplest model and the one most commonly used in the design of Data Warehouses. It consists of a fact table surrounded by many dimension tables that do not have a link between them.

1.1 Advantages

- Ease of navigation : This greatly simplifies queries (the fact table is linked to each dimension table by a single relationship, a single join)
- Efficient : Decreases the execution time.

1.2 Disadvantages

- Redundancy in dimensions: Each dimension is stored in a separate dimension table which results in denormalization.
- Complex diet: All dimensions do not relate to measurements.

2. Data Warehouse

Once we have finalized our data modeling technique, we will proceed to store the structured, non-volatile, subject-oriented data to enable effective analysis.fact table

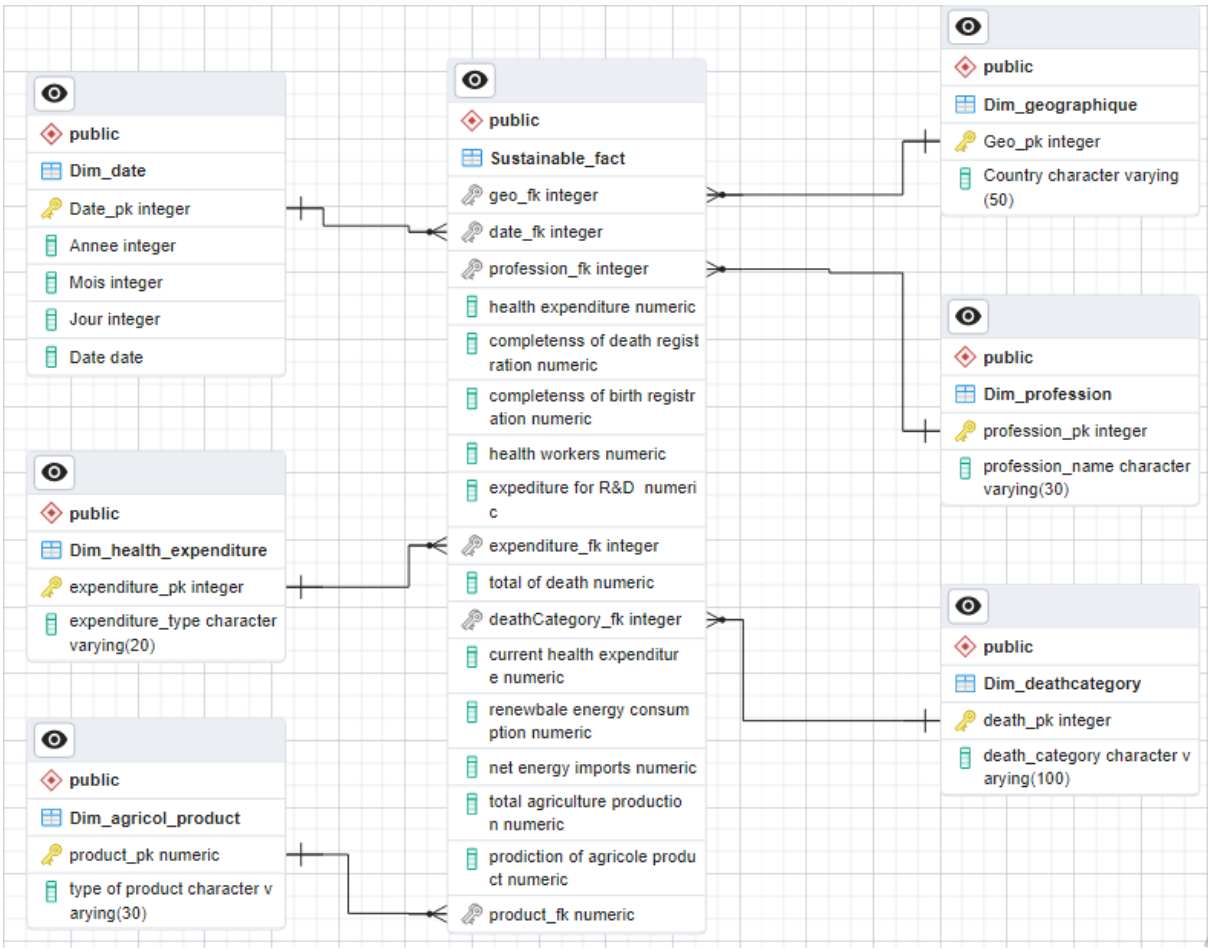
2.1 Fact table identification

The fact table presented is a comprehensive collection of environmental and social indicators that can be used to analyze a particular geographic area's development over time. It serves as a valuable tool for identifying trends and patterns using different measures.
To ensure a multifaceted analysis, the fact table includes several dimensions that allow the data to be analyzed from different angles. These dimensions include date, geography, cause of death, socioeconomic status, energy source, water source, and air quality. This approach allows the user to examine the data over time, by geographic area, or from various other angles.

2.2 Dimension identification

The analysis of key indicators gathered from technical and administrative managers will involve various analytical perspectives, developed based on their specific needs and the availability of data. Following the identification of these needs, we proceeded to create dimensions related to insurance services, including:

- Date dimension
- Geographical dimension
- DeathCategory dimension
- Profession dimension
- Health_expenditure dimension
- Agricol_product dimension



3.KPI identification

The Data Warehouse (DW) is designed to contain valuable information for extracting measurements, known as Key Performance Indicators (KPIs). To identify this information, we conducted a thorough analysis of the data provided.

- Human Development Index
- Mortality rate
- Poverty index
- Agriculture production rate
- Renewable energy consumption rate

Conclusion:

In this chapter, we explored the technical aspects of our data warehouse and the importance of data modeling techniques in developing a comprehensive model for our specific requirements. We chose the star schema model due to its simplicity and efficiency in query execution, despite its disadvantages. We also identified various dimensions related to insurance services, including date, geographical, death category, profession, health expenditure, and agricultural product dimensions. Finally, we identified key performance indicators (KPIs) such as human development index, mortality rate, poverty index, agriculture production rate, and renewable energy consumption rate, which will be used to measure the success of our data warehouse. The information stored in our data warehouse will enable effective analysis and informed decision-making, contributing to the overall success of our project.

Chapter 3:Data integration

Introduction:

This stage involves integrating our data, addressing missing or unusable values, and implementing our dimensions using Talend's data tools.

I. Talend

Talend Open Studio is a software tool developed by Talend that operates as an open-source ETL (Extract Transform Load) software. Its primary function is to contribute to the decision-making process by facilitating the data integration procedure. The software is equipped with the capability to create an all-inclusive ETL and Data Integration pipeline.

1. Components

One of Talend’s great strengths is that it can connect to almost any existing data source,

business application and file type. And, thanks to more than 250 components. Among its components we use:

1.1 TDBInput

This component allows to read a database and extract fields from it using queries and put it in a list which will be transmitted to the next component via a stream connection

1.2 TDBOutput

This component allows writes, updates, makes changes or suppresses entries in a database.

1.3 TMap

TMap is an advanced component, which can be integrated as a plugin into Talend Studio.It allows to transform and route data from single or multiple sources to single or multiple destinations.

II. Internal Data Implementation

The implementation of our data-model is a physical realization on a real machine of the components of the abstract machine that together constitute that model alongside the necessary data.

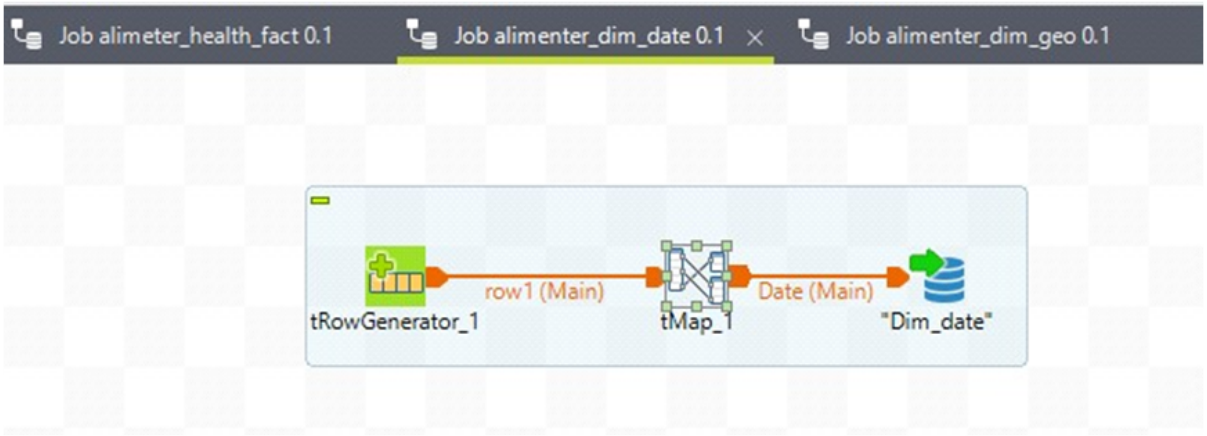
1. Date dimension

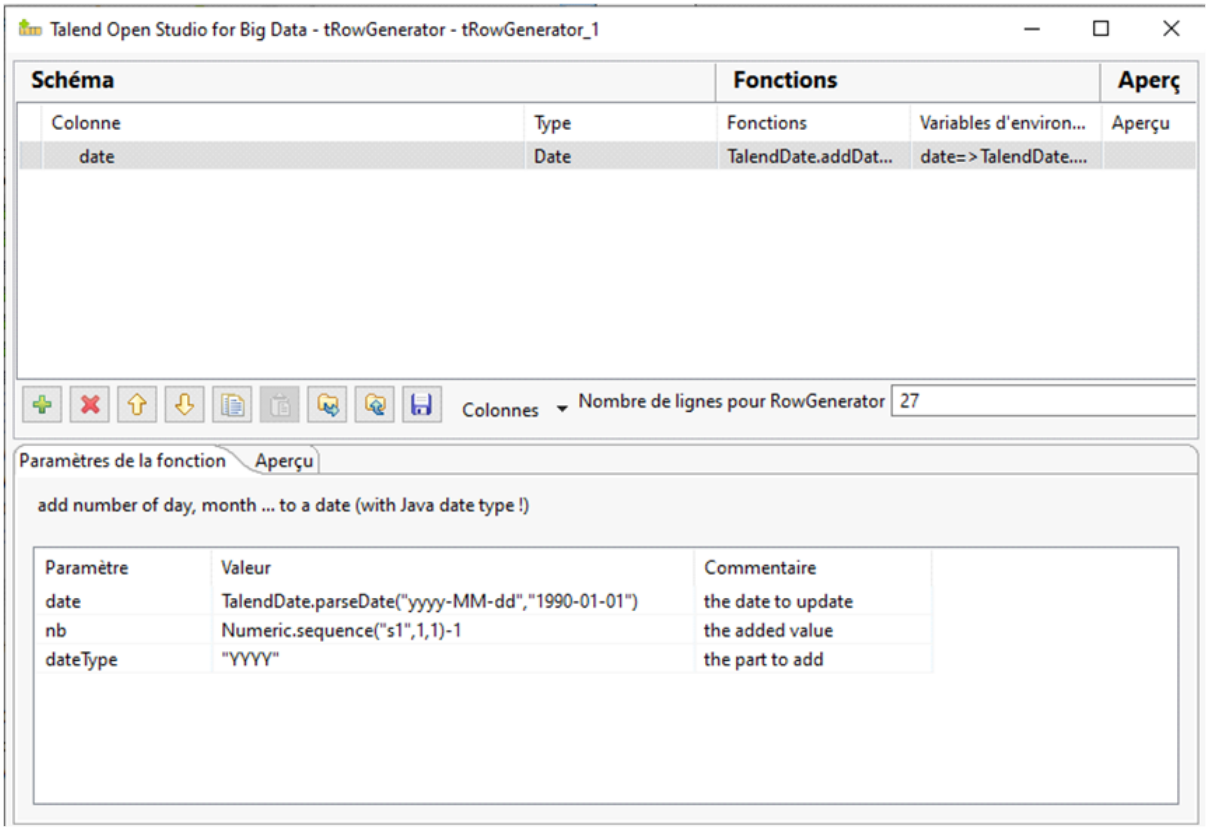
For the feeding of the Date dimension, we used 3 components:

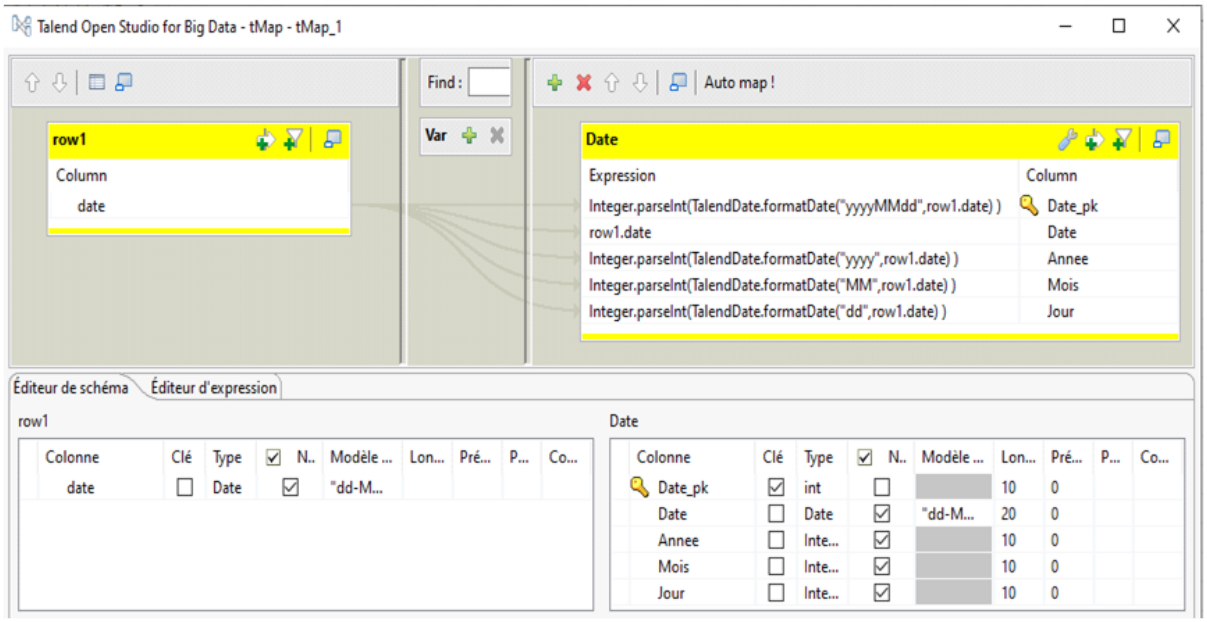
- A tRowGenerator to generate a list of dates (27 rows) starting from "1990-01-01" and incrementing by year.

- A tMap to perform the mapping by connecting the "DATE" column (which we created using tRowGenerator) from the input to the output. For the year, month, and day columns, we used the function "Integer.parseInt(TalendDate.formatDate("", row1.date))". We also used a sequence function for the primary key "Date_pk" to be "auto increment"

- A tDBOutput, which is our database table where we will load our data.



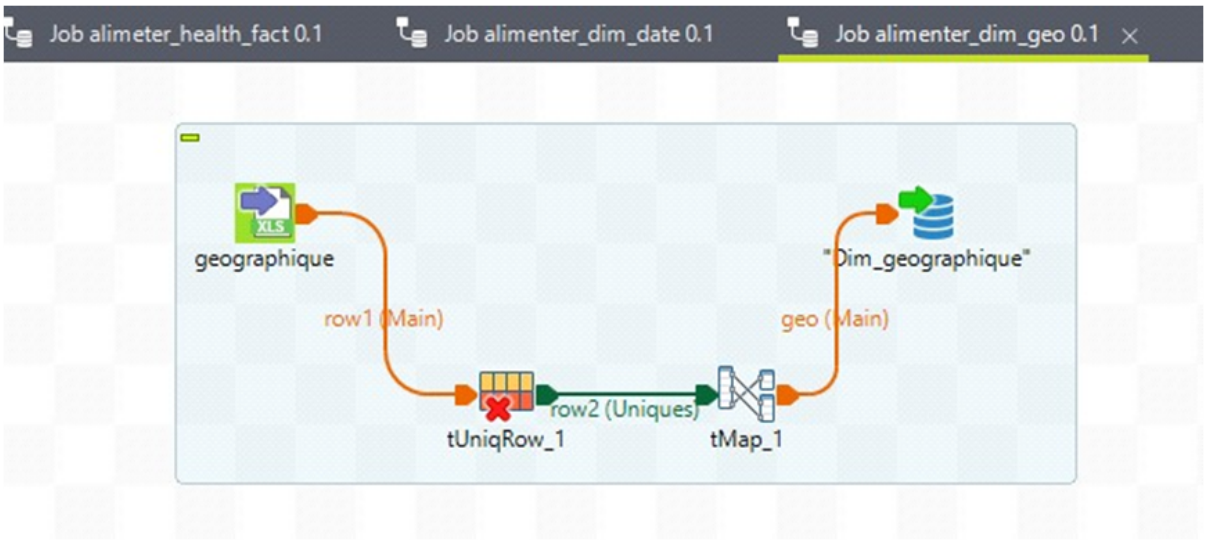


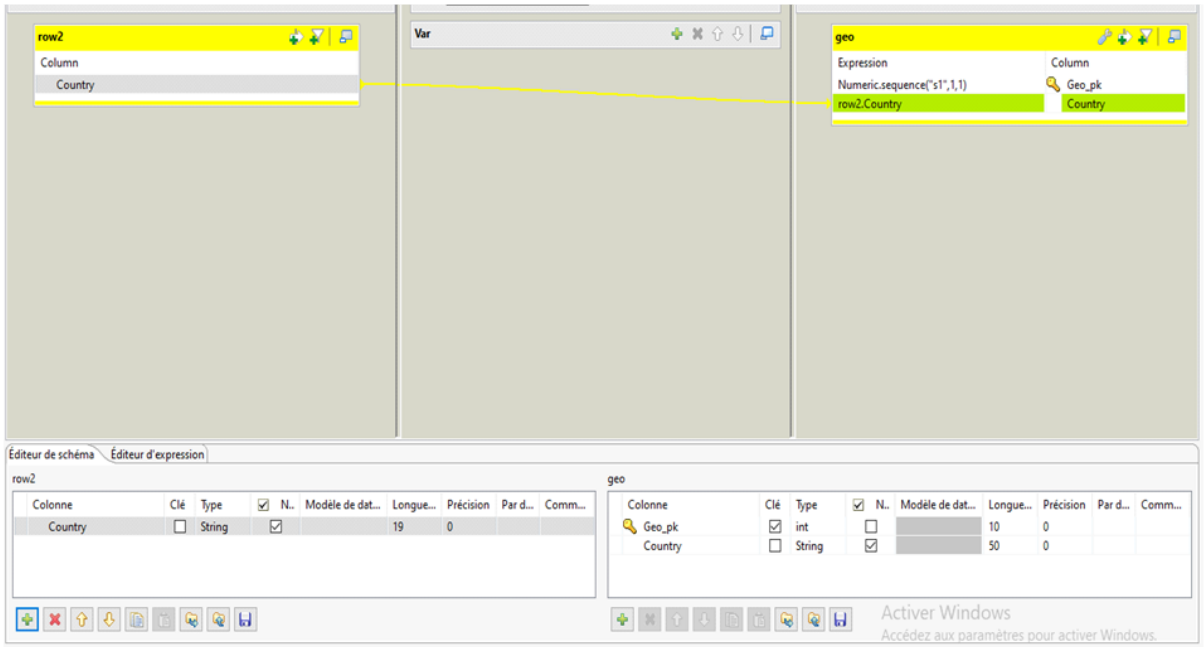


2. Geographical dimension

For the geographical dimension data feeding, we used 4 components:

- A tFileInputExcel to input an Excel file that contains the list of countries.
- A tUniqRow where we checked the "country" column to eliminate redundancy.
- A tMap to perform the mapping by connecting the "country" column from the input to the output. We used a sequence function for the primary key "Geo_pk" to be "auto increment".
- A tDBOutput, which is our database table where we will load our data.

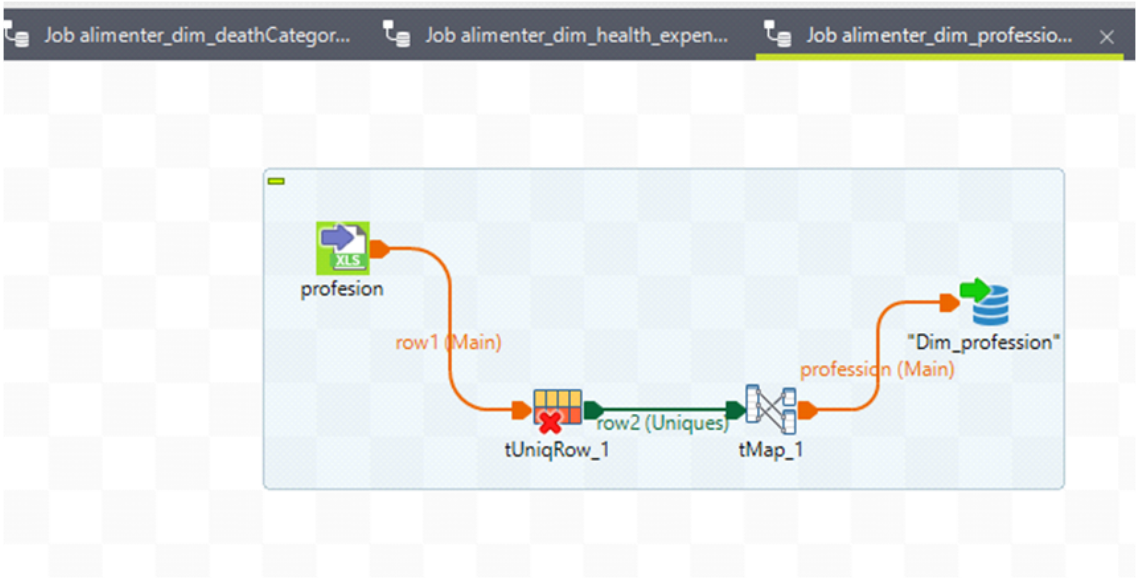


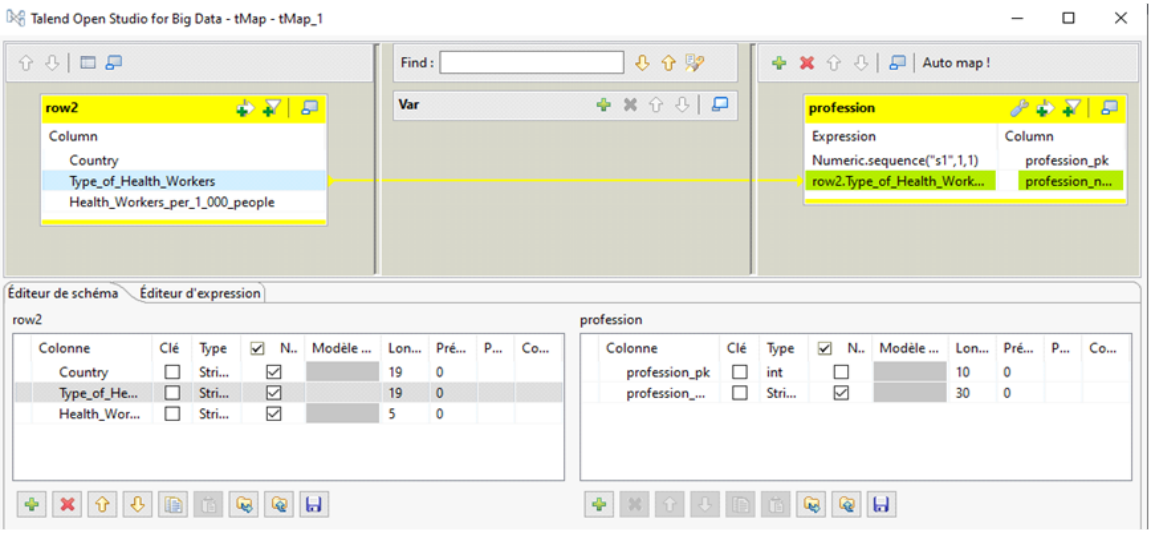


3. Profession dimension

For feeding this dimension, we followed the same steps using 4 components:

- A tFileInputExcel to extract data from the Excel file as input.
- A tUniqRow to eliminate redundancy.
- A tMap to perform the mapping by connecting the input columns to the output columns. We used a sequence function to increment the primary key.
- A tDBOutput, which is our database table where we will load our data.





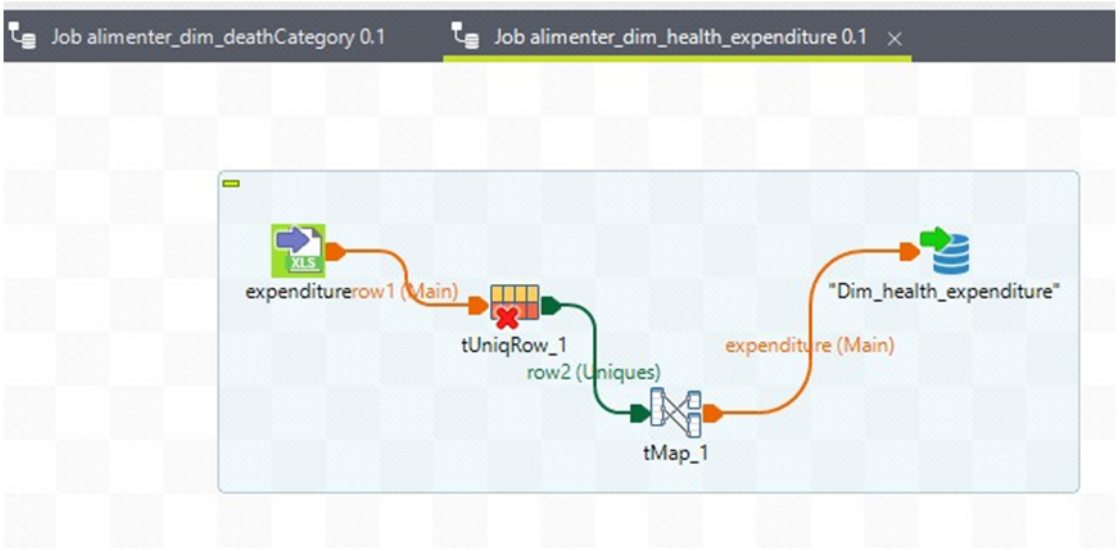
4. Health-expenditure dimension

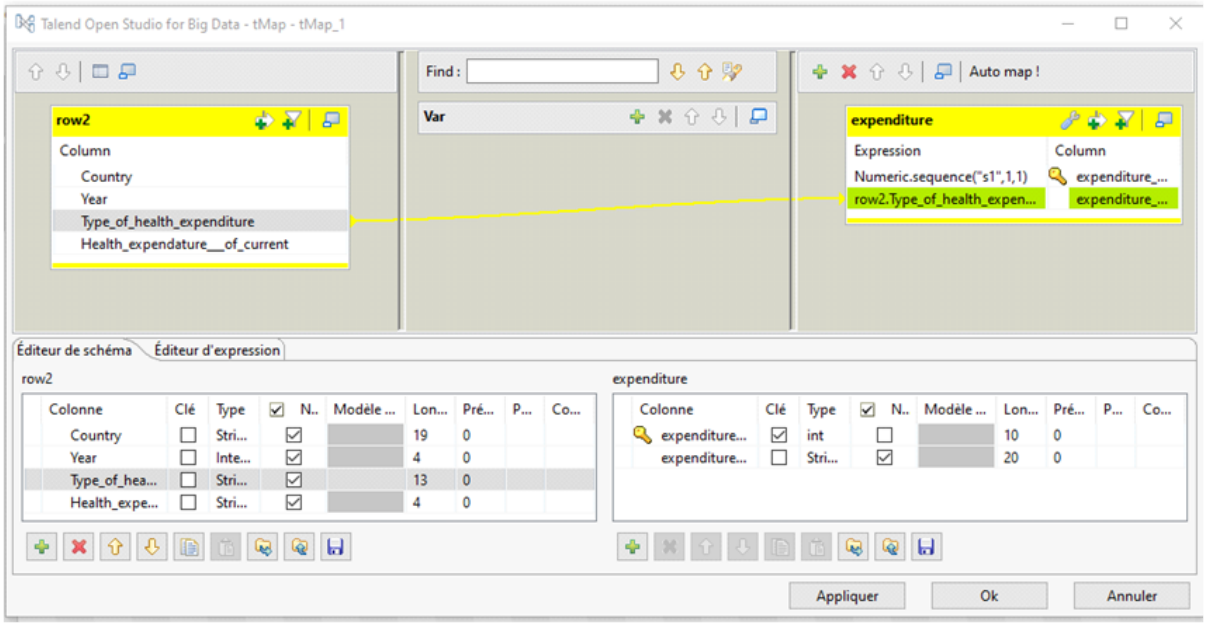
For feeding this dimension, we followed the same steps using 4 components:

- A tFileInputExcel to extract data from the Excel file as input.
- A tUniqRow to eliminate redundancy.
- A tMap to perform the mapping by connecting the input columns to the output columns.

We used a sequence function to increment the primary key.

- A tDBOutput, which is our database table where we will load our data.





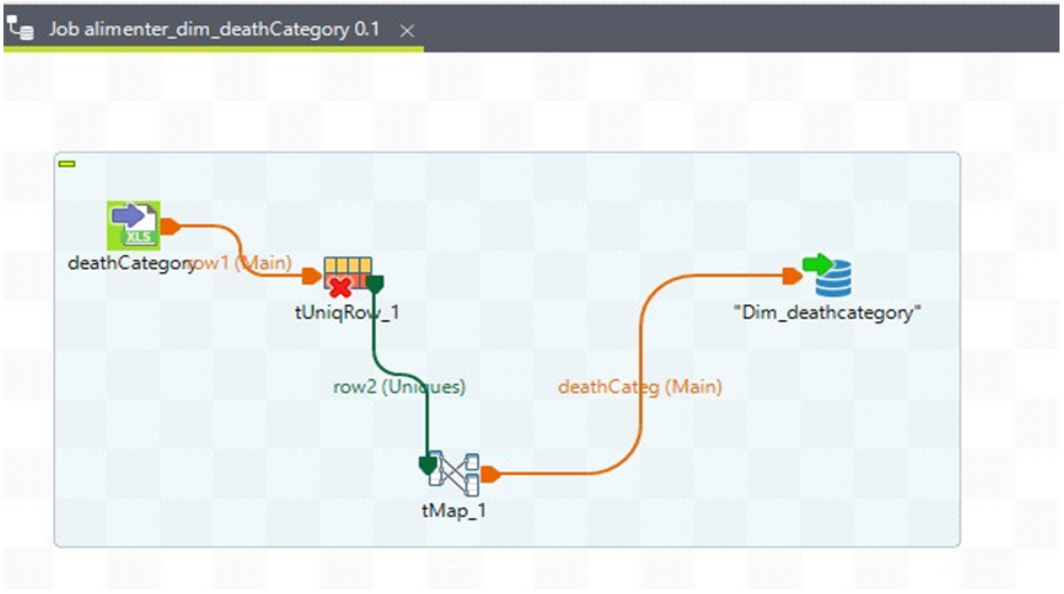
5. DeathCategory dimension

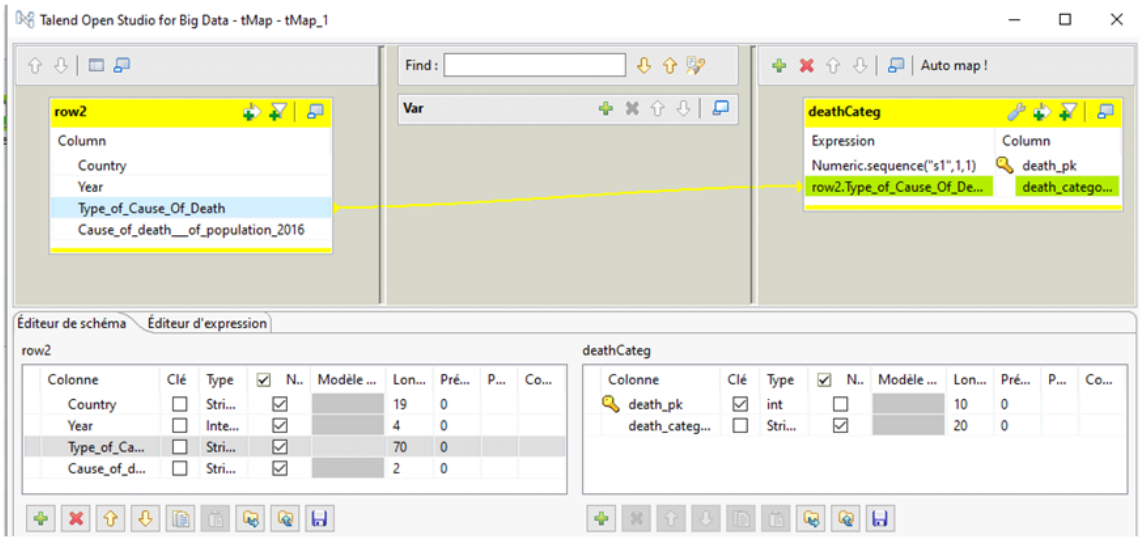
For feeding this dimension, we followed the same steps using 4 components:

- A tFileInputExcel to extract data from the Excel file as input.
- A tUniqRow to eliminate redundancy.

• A tMap to perform the mapping by connecting the input columns to the output columns. We used a sequence function to increment the primary key.

•A tDBOutput, which is our database table where we will load our data.

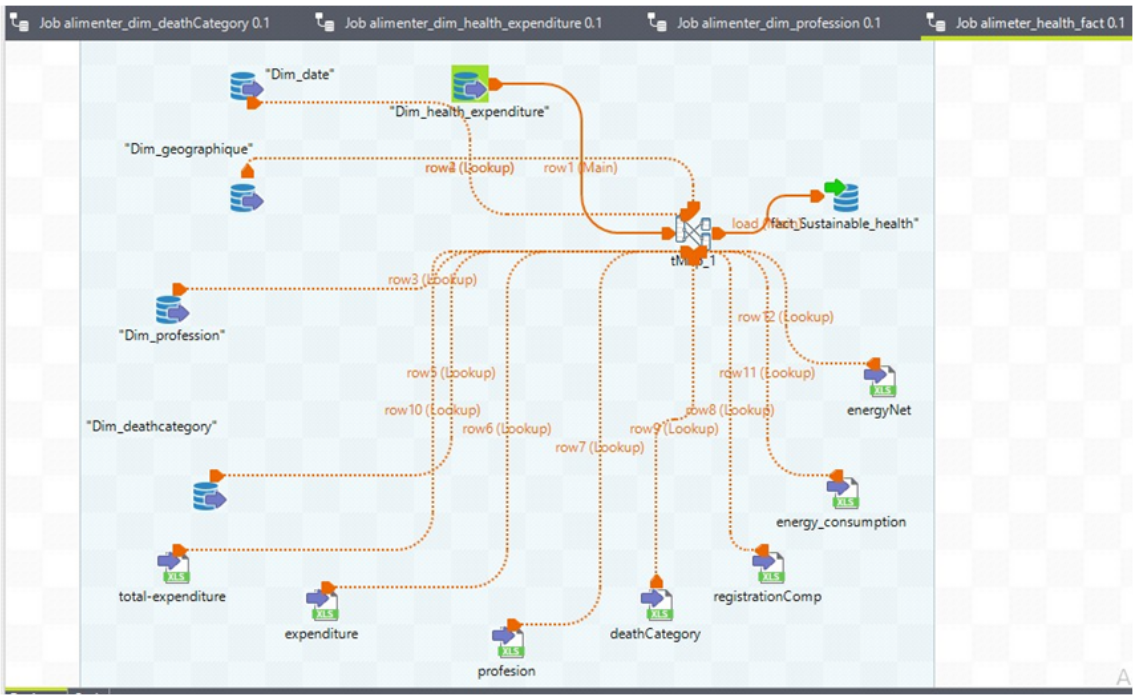


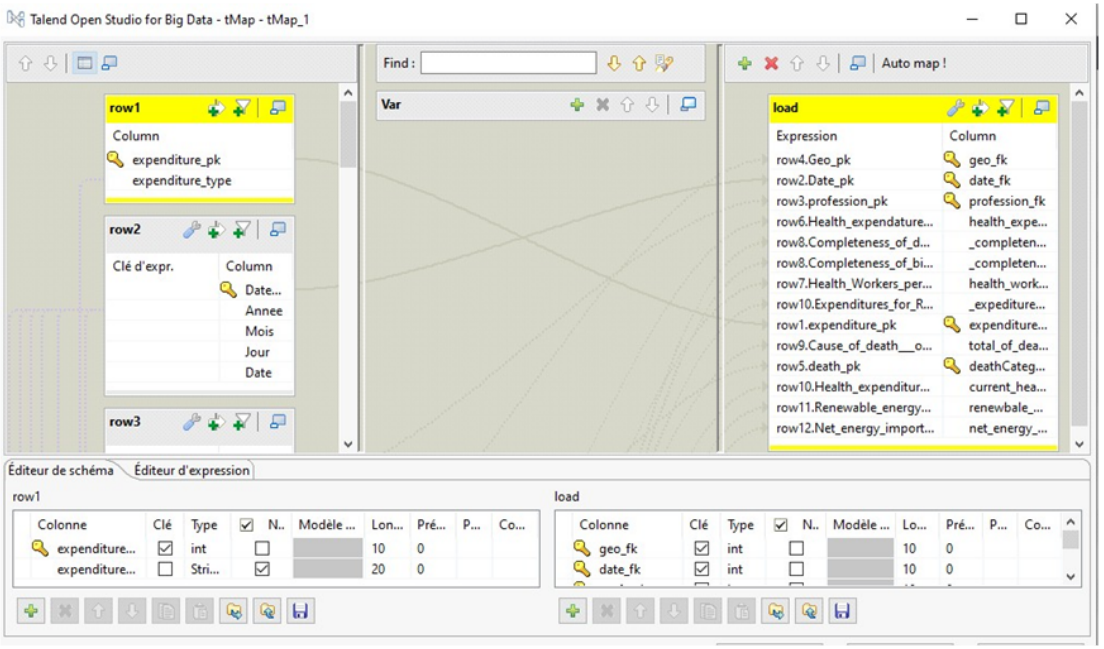


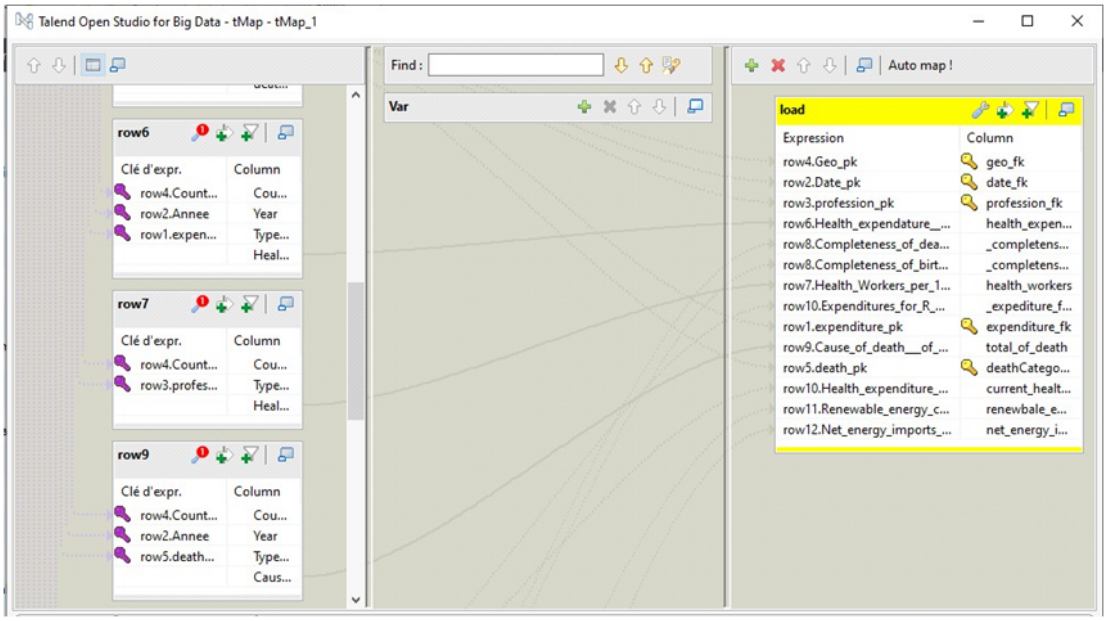
6. Sustainable fact

To load the fact table, we need the 4 dimensions for foreign key feeding and tFileInputExcel components for other measures. We also used the tMap component to

connect the input columns with the output columns and perform joins between the input columns. Subsequently, we connected it with the tDBOutput component to load our data.







III. External Data

1. Human Development Indicator

The Human Development Indicator (HDI) was extracted from the website https://countryeconomy.com/hdi?year=2016 using the python modules of request and BeautifulSoup to extract the required data, which are the countries and their corresponding HDI values. The extracted data was then used to explore the correlation between human development and factors such as mortality rate, agricultural production, energy consumption, and poverty rate. The final output of this analysis was a visualization of these relationships for each country.

2.Mortality Rate

The Mortality Rate data was sourced from https://www.kaggle.com/datasets/navinmundhra/world-mortality?resource=download. The python modules of request and BeautifulSoup were used to extract the required data, which are the countries and the mortality rates for both sexes. A formula was then applied to extract the percentage of the world's total mortality rate for each country. The extracted data was used to investigate the correlation between mortality rate and human development, and the final output was a visualization of this relationship for each country.

3. Pesticide use

The data on Pesticide use was extracted from https://www.worldometers.info/food-agriculture/pesticides-by-country/ using the python modules of request and BeautifulSoup. The required data included the countries and their corresponding pesticide use values, measured in tons and kilograms per hectare. The

extracted data was used to investigate the correlation between pesticide use and human development and the final output was a visualization of pesticide use for each country.

4. Poverty Index Value

The Poverty Index Value data was sourced from https://hdr.undp.org/content/2022-global-multidimensional-poverty-index-mpi?fbclid=IwAR3KA5zga6rTwYm-Dd6NH8LDdcymq0d2Zf5ChPdvSRxAmDVldFSKaMJMhZw#/indicies/MPI

The extracted data was used to explore the correlation between poverty and the development of each country. The final output of this analysis is not specified.

Conclusion:

In this chapter, we learned about data integration and how Talend Open Studio, an open-source ETL software, can be used to facilitate the data integration process. We looked at some of Talend's components, such as TDBInput, TDBOutput, TMap, and TJoin, which help in connecting to different data sources, performing transformations, and ensuring data quality. We also saw how to implement our data model by physically realizing the components of the abstract machine using Talend's tools.

In particular, we saw how to implement various dimensions such as date, geographical, profession, health-expenditure, and death category using Talend's tools. Additionally, we looked at how to load the fact table using these dimensions along with external data such as the Human Development Indicator.

Overall, Talend's data tools can simplify the process of data integration, making it easier to connect to different data sources, perform transformations, and ensure data quality. By following the steps outlined in this chapter, we can effectively implement our data model and load our data into the database tables.

Chapter 4: Data Mining

Introduction:

Data mining is a process used to turn raw data into useful information. By using software to look for patterns in large batches of data, and following the CRISP method, we tried to learn more about road accidents to develop more effective methods to decrease their frequency and numbers

I. Business Understanding

The Business Understanding phase focuses on understanding the objectives and requirements of the project. After analyzing our database, we found that

Our Data Mining Objectives are :

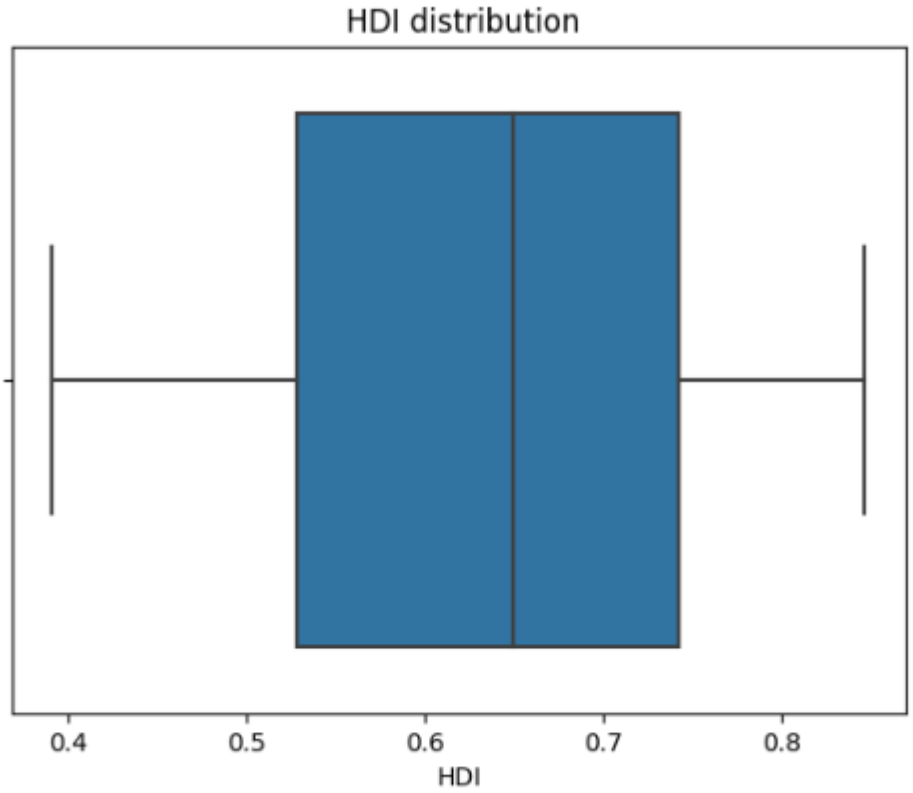
- Describe the health expenditure by type and by country
- Describe the mortality Rate by type and by country
- Segment countries by their research and development expenditure (R&D)
- Segment countries by their pesticide use
- Segment countries by their agricultural production
- Segment countries by their poverty rate
- Segment countries by their renewable energy consumption
- Understand the development factors of countries
- Predict Human development index rate
- Predict Countries by their HDI Rate
- Analyze people's opinions towards the sustainable development subject

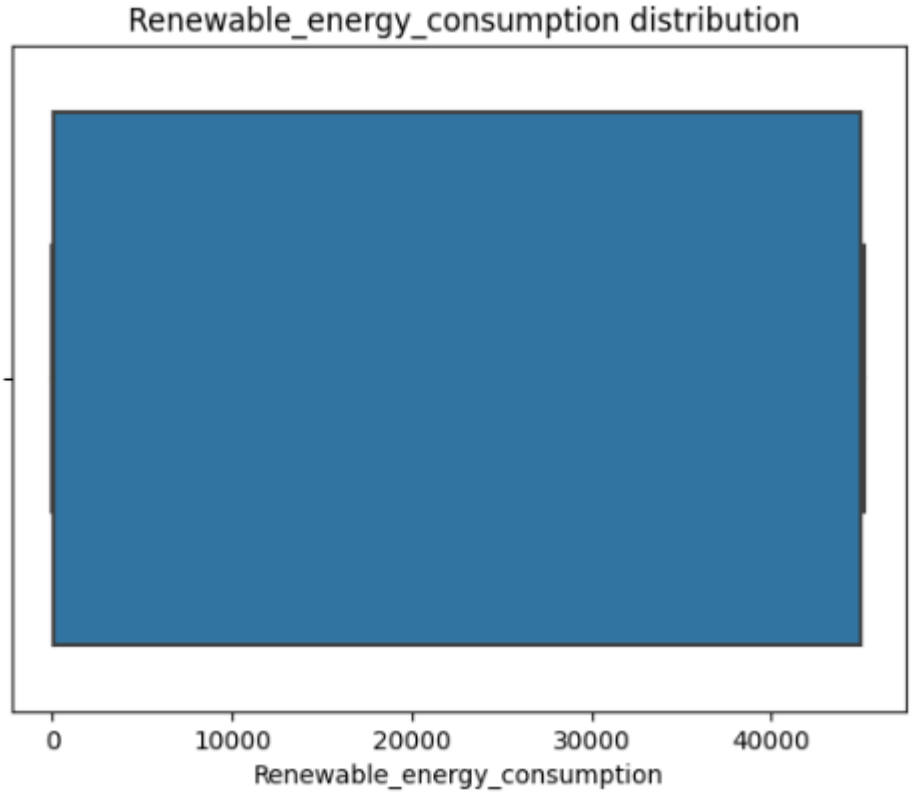
II. Data understanding

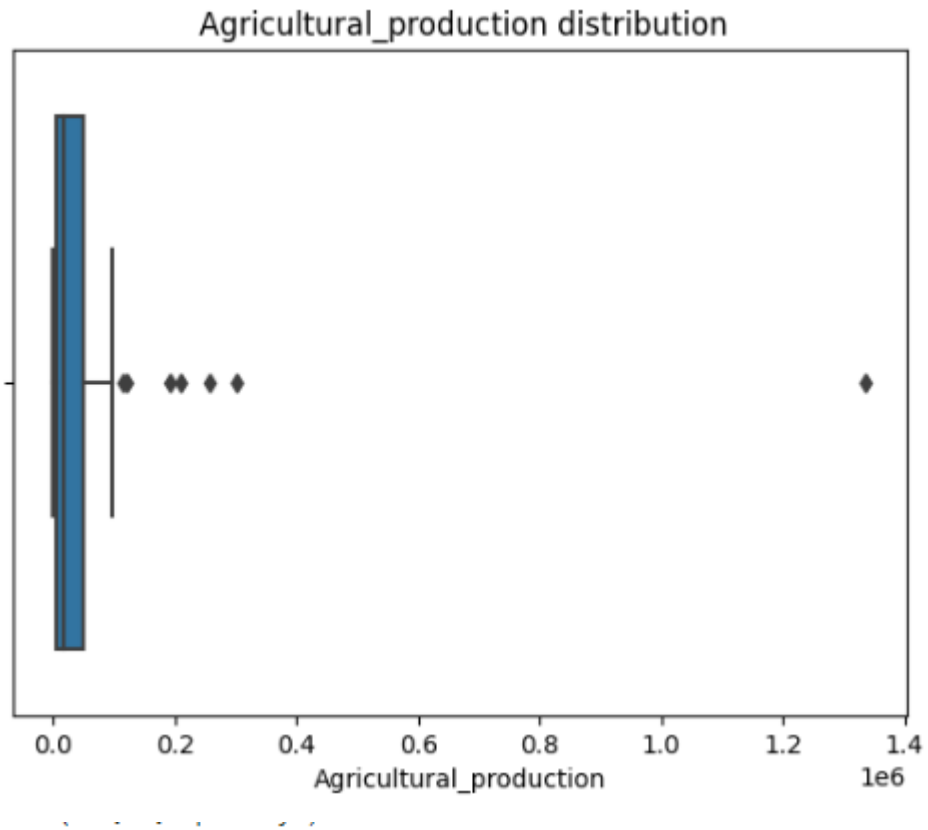
Next is the Data Understanding phase. Adding to the foundation of Business Understanding, it drives the focus to identify, collect, and analyze the data sets that can help you accomplish the project goals.

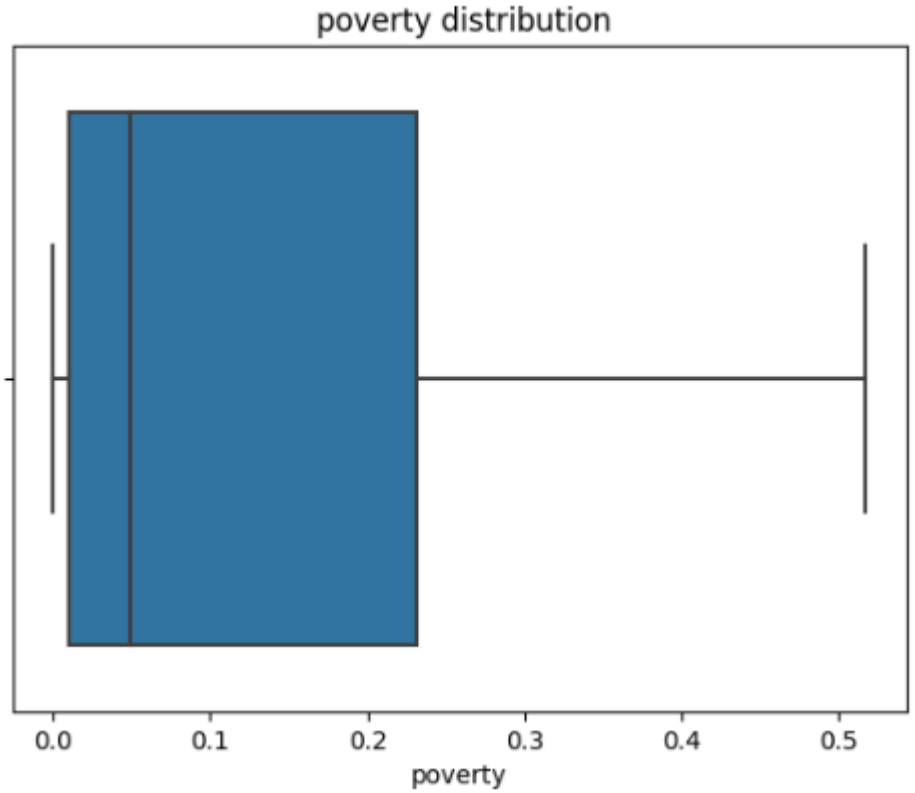
1. External data

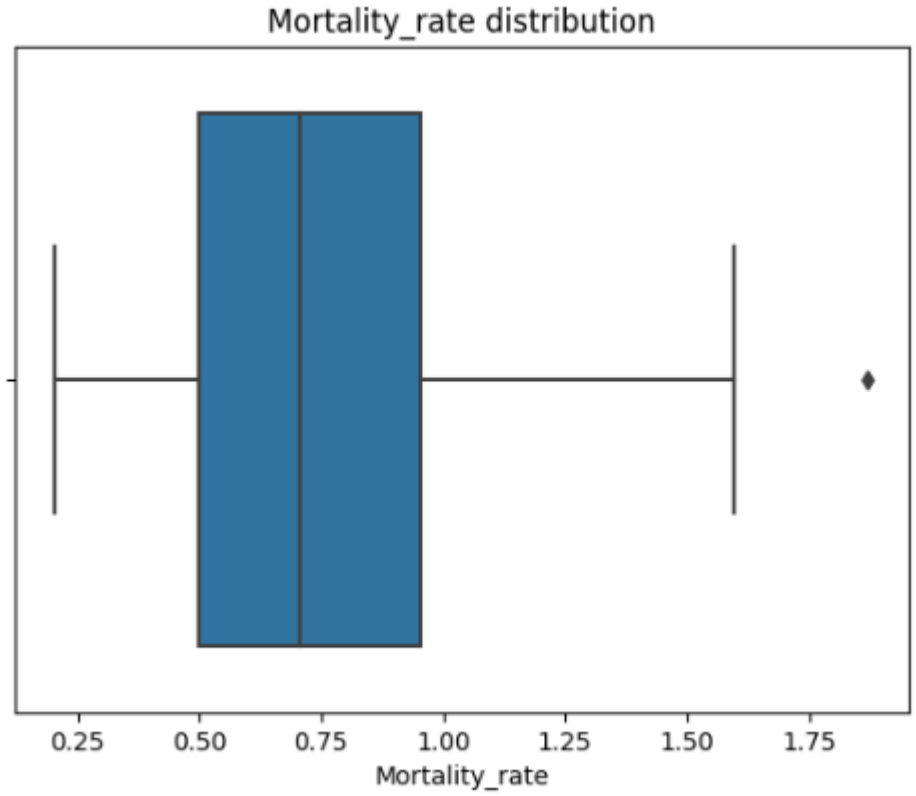
To detect outliers, box plots were used






Agricultural_production distribution
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Agricultural_production
1e6

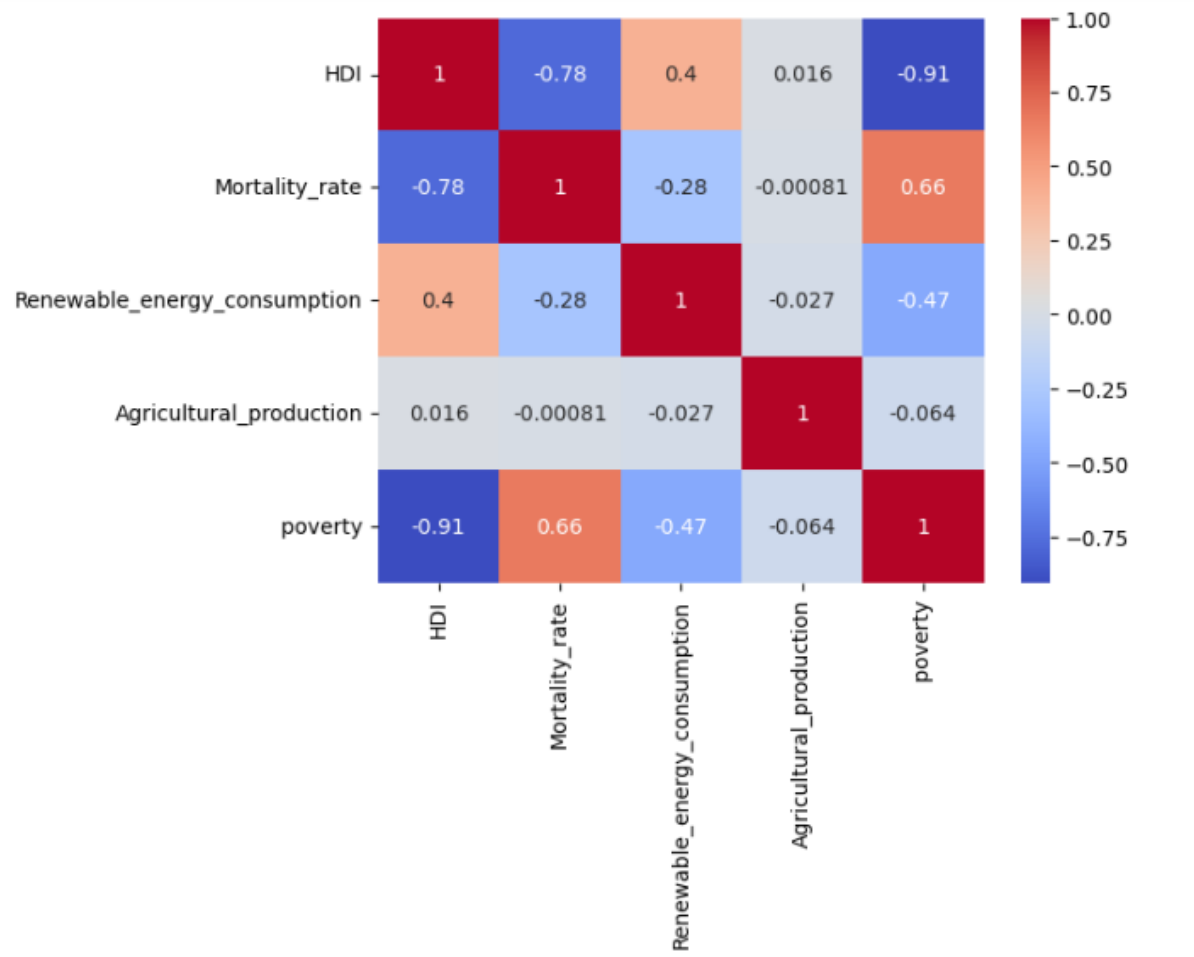

poverty distribution
0.0
0.1
0.2
0.3
0.4
0.5
poverty



2. Correlation test

In this part we studied the correlation between the different variables, that's why we used a correlation matrix

Here is the result of our correlation matrix



From these results it was noticed that there is a strong correlation between HDI and poverty, Mortality rate, renewable energy consumption while the correlation between HDI and agriculture production is low

III. Data preparation

1. Data imputation

We utilized the K-nearest neighbors (kNN) algorithm to impute missing values in the Vehicle dataset. kNN is a supervised learning algorithm that can solve classification and regression problems. We specifically used it for numeric values and employed the Euclidean distance measure to determine the number of neighboring data points, represented by the hyperparameter k, to contribute to each prediction. We also used the DataFrame.fillna() method to fill missing values.

2. Preparation of datasets

To meet our objectives we have prepared datasets

2.1 Describe the health expenditure by type and by country

	A	B	C	D
1	Country	Health_expenditure_Public	Health_expenditure_Out_of_pocket	External_health_expenditure
2	Afghanist	5,10	77,40	17,5
3	Albania	41,40	58,00	0,7
4	Algeria	67,70	30,90	0
5	American	52,92	32,66	9,116766467
6	Andorra	49,10	41,70	9,116766467
7	Angola	44,10	35,20	3,6
8	Antigua a	60,60	32,20	0
9	Argentina	74,40	15,80	0,6
10	Armenia	16,50	80,60	1,7
11	Aruba	52,92	32,66	9,116766467
12	Australia	68,30	18,90	0
13	Austria	72,50	18,90	9,116766467
14	Azerbaija	20,00	78,90	0,4
15	Bahamas,	49,90	27,70	0,4
16	Bahrain	61,40	28,00	0
17	Banglade	18,00	71,90	7,6
18	Barbados	45,90	45,20	1,7
19	Belarus	61,40	35,80	0,2
20	Belgium	84,10	15,90	0
21	Belize	66,30	22,90	3,8

2.2 Describe the health workers by type and by country

A	B	C	D
Country	Health workers Physicians	Health workers Nurses and midwives	Specialist surgical workforce
Afghanistan	0.3	0.3	0
Albania	1.2	3.6	11.6
Algeria	1.8	2.2	12.1
American Samoa	NA	NA	NA
Andorra	3.3	4	83.1
Angola	0.2	1.3	NA
Antigua and Barl	2.8	3.1	14
Argentina	4	2.6	50.1
Armenia	2.9	5.6	86.7
Aruba	NA	NA	NA
Australia	3.6	12.7	45.1
Austria	5.1	8.2	109.9
Azerbaijan	3.4	7	67.6
Bahamas, The	1.9	3.1	NA
Bahrain	0.9	2.5	14.4
Bangladesh	0.5	0.3	2.9

2.3 Describe the mortality Rate by type and by country

cause of death 2016

	A	B
1	Country	Type of Cause Of Death
2	Afghanistan	Communicable diseases and maternal, prenatal, and nutrition conditions
3	Afghanistan	Injuries
4	Afghanistan	Non-communicable diseases
5	Albania	Communicable diseases and maternal, prenatal, and nutrition conditions
6	Albania	Injuries
7	Albania	Non-communicable diseases
8	Algeria	Communicable diseases and maternal, prenatal, and nutrition conditions
9	Algeria	Injuries
10	Algeria	Non-communicable diseases
11	American Samoa	Communicable diseases and maternal, prenatal, and nutrition conditions
12	American Samoa	Injuries
13	American Samoa	Non-communicable diseases
14	Andorra	Communicable diseases and maternal, prenatal, and nutrition conditions
15	Andorra	Injuries
16	Andorra	Non-communicable diseases
17	Angola	Communicable diseases and maternal, prenatal, and nutrition conditions
18	Angola	Injuries
19	Angola	Non-communicable diseases
20	Antigua and Barbud	Communicable diseases and maternal, prenatal, and nutrition conditions

2.4 Segment countries by their research and development expenditure

	A	B
1	Country	Expenditures_for_RD
2	Afghanistan	0,8568
3	Albania	0,8568
4	Algeria	0,5
5	American Samoa	0,8568
6	Andorra	0,8568
7	Angola	0
8	Antigua and Barbuda	0,8568
9	Argentina	0,6
10	Armenia	0,3
11	Aruba	0,8568
12	Australia	1,9
13	Austria	3
14	Azerbaijan	0,2
15	Bahamas, The	0,8568
16	Bahrain	0,1
17	Bangladesh	0,8568
18	Barbados	0,8568
19	Belarus	0,5
20	Belgium	2,5
21	Belize	0,8568
22	Benin	0,8568
23	Bermuda	0,2
24	Bhutan	0,8568

Expenditures_for_RD-dataset

2.5 Segment countries by their pesticide use

	A	B
1	Country	Pesticide Use (tons)
2	China	1763000
3	United States	407779
4	Brazil	377176
5	Argentina	196009
6	Canada	90839
7	Ukraine	78201
8	France	70589
9	Malaysia	67288
10	Australia	63416
11	Spain	60896
12	Italy	56641
13	Turkey	54098
14	India	52750
15	Japan	52249
16	Germany	48193
17	Mexico	47128
18	Colombia	37698
19	Thailand	35287
20	Ecuador	34253
21	South Africa	26857

2.6 Segment countries by their agricultural production

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitude	...	Y2004	Y2005	Y2006	Y2007	Y2008	Y2009
0	AFG	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.71	...	3249.0	3486.0	3704.0	4164.0	4252.0	4538.0
1	AFG	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.71	...	419.0	445.0	546.0	455.0	490.0	415.0
2	AFG	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.71	...	58.0	236.0	262.0	263.0	230.0	379.0
3	AFG	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.71	...	185.0	43.0	44.0	48.0	62.0	55.0
4	AFG	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.71	...	120.0	208.0	233.0	249.0	247.0	195.0

5 rows × 63 columns

2.7 Segment countries by their poverty rate

poverty2015

	A	B
1	Country	poverty index value in 2015
2	Afghanistan	0,272
3	Albania	0,003
4	Algeria	0,005
5	Angola	0,282
6	Argentina	0,001
7	Armenia	0,001
8	Bangladesh	0,104
9	Belize	0,017
10	Benin	0,368
11	Bolivia (Plurinational State of)	0,038
12	Botswana	0,073
13	Burundi	0,409
14	Cameroon	0,232
15	Central African Republic	0,461
16	Chad	0,517
17	Colombia	0,020
18	Congo (Democratic Republic of the)	0,331
19	Costa Rica	0,002
20	Côte d'Ivoire	0,236
21	Cuba	0,003
22	Dominican Republic	0,009
23	Ecuador	0,008

2.8 Segment countries by their renewable energy consumption

	Country	Mortality_rate	Health_expenditure_Current	Expenditures_for_RD	Health_Workers	Renewable_energy_consumption	cluster
0	Afghanistan	0.948840	10.200000	0.8568	0.600000	18.4	1
1	Albania	0.371790	6.700000	0.8568	4.811600	38.6	2
2	Algeria	0.367918	6.600000	0.5000	4.012100	0.1	2
3	American Samoa	0.441501	6.715054	0.8568	5.900371	0.9	2
4	Andorra	0.000000	10.400000	0.8568	7.383100	19.7	2
...	...	...	...	...	...	...	...
209	Virgin Islands (U.S.)	0.000000	6.715054	0.8568	5.900371	3.9	2
210	West Bank and Gaza	0.000000	6.715054	0.5000	5.900371	10.5	2
211	Yemen, Rep.	0.000000	5.600000	0.8568	1.000800	2.3	2
212	Zambia	1.107626	4.500000	0.8568	1.001500	88.0	1
213	Zimbabwe	1.293521	9.400000	0.8568	1.301600	81.8	1

2.9 Understand the development factors of countries

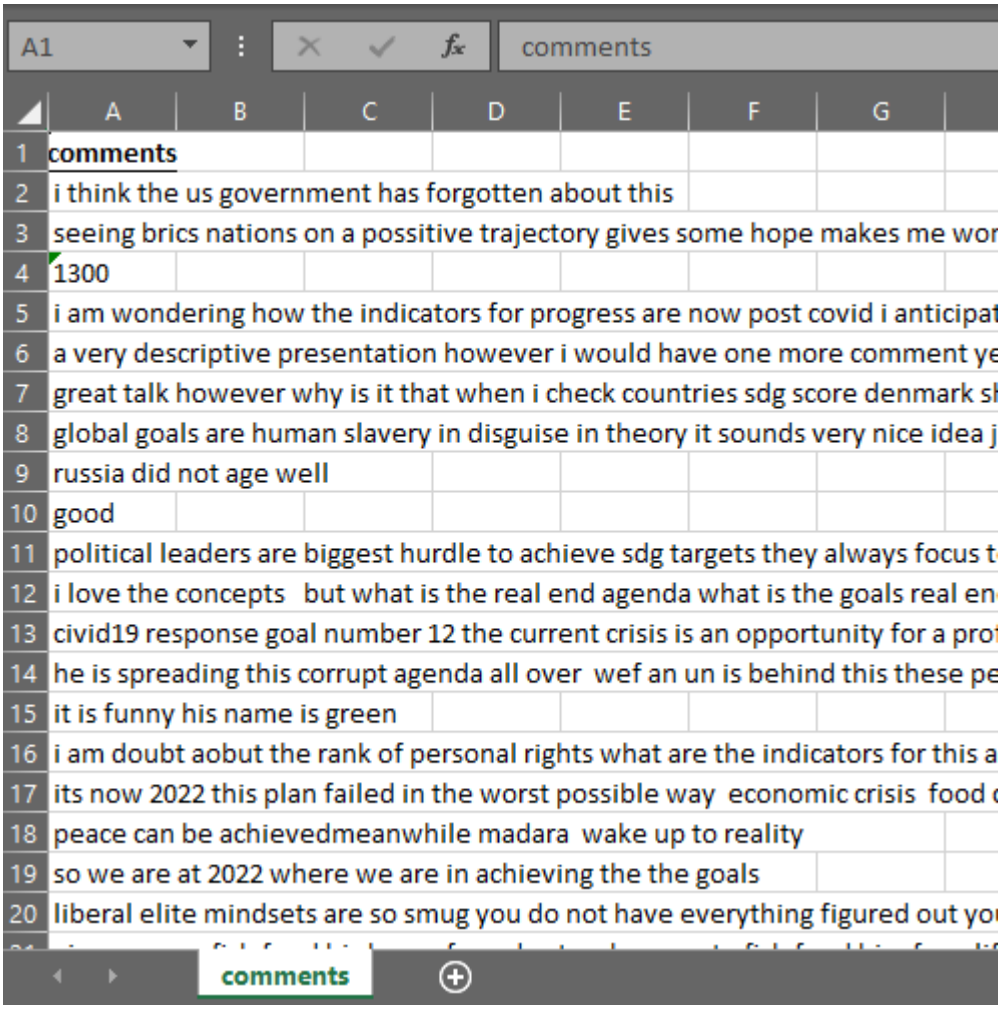
	Country	HDI	RD_Expenditures	Renewable energy consommation	Agricultural Production	Pesticide use kg per hectare	poverty
0	Afghanistan	0.481	Low	Low	Low	NaN	Moderate
1	Albania	0.798	Low	Moderate	Low	Very Low	Very Low
2	Algeria	0.743	Low	Low	Low	Very Low	Very Low
3	American Samoa	NaN	Low	Low	NaN	NaN	NaN
4	Andorra	0.871	Low	Low	NaN	NaN	NaN
...	...	...	...	...	...	...	...
209	Virgin Islands (U.S.)	NaN	Low	Low	NaN	NaN	NaN
210	West Bank and Gaza	NaN	Low	Low	NaN	NaN	NaN
211	Yemen, Rep.	NaN	Low	Low	Low	NaN	NaN
212	Zambia	0.564	Low	High	Low	Very Low	Moderate
213	Zimbabwe	0.588	Low	High	Low	Very Low	Low

214 rows × 7 columns

2.10 Predict HDI Rate

	A	B	C	D	E	F
1	Country	HDI	Mortality_rate	Renewable_energy_consumption	Agricultural_production	poverty
2	Afghanistan	0.481	0.9488400914	45 034,00	23 007,00	0.27172124
3	Albania	0.798	0.3717904032	38.6	8 271,00	0.00274788
4	Algeria	0.743	0.3679175865	0.1	72 161,00	0.0054091
5	American Samoa		0.4415011038	0.9		
6	Andorra	0.871	0.4672897196	45 126,00		
7	Angola	0.596	0.9217303745	49.6	48 639,00	0.28243506
8	Antigua and Bar	0.794	0.464738004	0,00	119,00	
9	Argentina	0.847	0.4298826537	10,00	80 843,00	0.0014693
10	Armenia	0.765	0.4492467372	45 153,00	7 175,00	0.000690069007857106
11	Aruba		0.4672897196	45 113,00		
12	Australia	0.935	0.2362418187	44 966,00	62 446,00	
13	Austria	0.915	0.2401146354	34.4	24 990,00	
14	Azerbaijan	0.75	0.4569923706	44 987,00	20 162,00	
15	Bahamas, The	0.823	0.4672897196	44 958,00	629,00	
16	Bahrain	0.865	0.2207505519	0,00		
17	Bangladesh	0.612	0.5034661709	34.7	120 854,00	0.1040603
18	Barbados	0.794	0.38728167	45 140,00	469,00	
19	Belarus	0.813	0.6235234886	45 144,00	37 852,00	
20	Belgium	0.927	0.2788428024	44 966,00	33 336,00	
21	Belize	0.712	0.6932341892	35,00	552,00	0.01710883

2.11 Analyze people's opinions towards the sustainable development subject



IV. Data Modelling

1. Describe the health expenditure by type and by country

➢The PCA algorithm:

PCA, or Principal Component Analysis, is a dimensionality reduction technique used in statistical analysis and machine learning. It transforms a set of original variables (or features) into a set of new variables called principal components, which are linear combinations of the original variables.

1.1 Application of the algorithm:

```
[ ]  from sklearn.decomposition import PCA
     from sklearn.preprocessing import scale

     # suppression des colonnes non numériques
     system_health_num = system_health.drop(columns = ["Country"])
     pca = PCA()
     print(system_health_num )
     pca.fit(system_health_num)#variables uniquement quantitatives
```

```
var = pandas.DataFrame({'id': system_health_num.columns, 'explained_variance_ratio_': pca.explained_variance_ratio_})
var
```

	id	explained_variance_ratio_
0	Health_expenditure_Public	0.747936
1	Health_expenditure_Out_of_pocket	0.236771
2	External_health_expenditure	0.015293

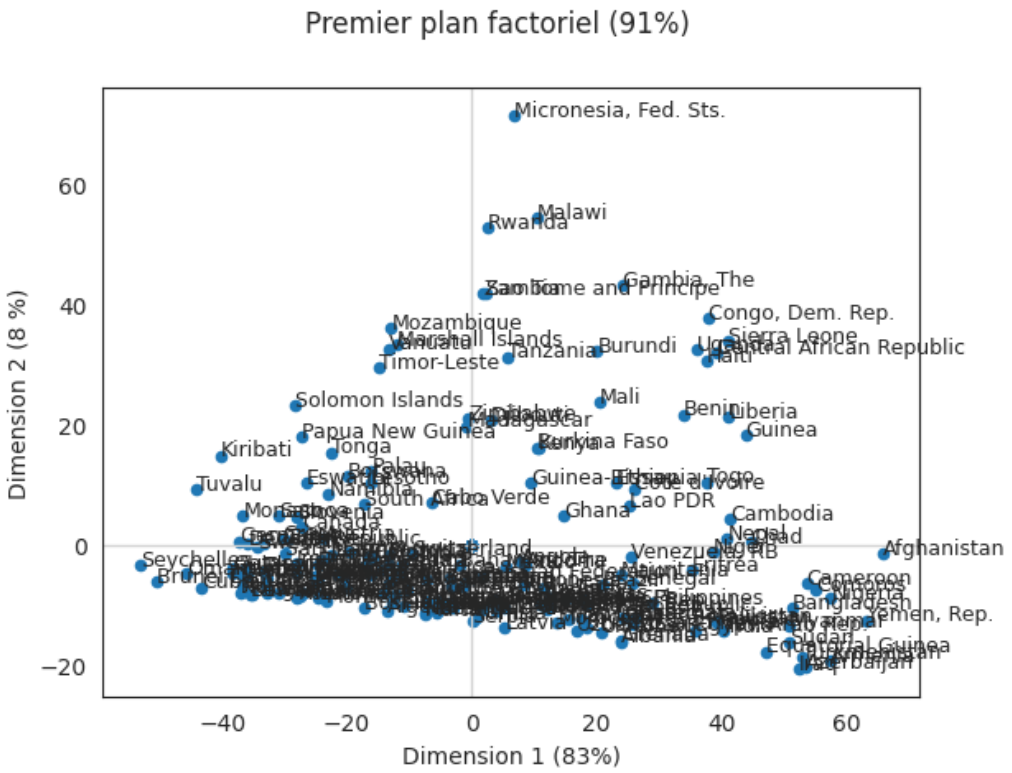
The "health expenditure Public" presents a high value for the proportion of variance explained which indicates that the corresponding principal component captures a large amount of information from our data.

1. 2 Presentation of the individuals on the factorial plane:

```
# utilisation de subplots nécessaire car annotation du graphique
fig, ax = plt.subplots()
health_pca_df.plot.scatter("Dim1", "Dim2", ax = ax) # l'option ax permet de placer les points et le texte sur le même graphique
ax.axvline(x = 0, color = 'lightgray', linewidth = 1)
ax.axhline(y = 0, color = 'lightgray', linewidth = 1)
# boucle sur chaque pays
for k in health_pca_df.iterrows():
    # annotation uniquement si valeur absolue sur une de 2 dimensions importantes (valeurs choisies empiriquement)
    if (abs(k[1]['Dim1']) > 3.5) | (abs(k[1]['Dim2']) > 1.5):
        ax.annotate(k[1]["Country"], (k[1]['Dim1'], k[1]['Dim2']), fontsize = 9)



plt.xlabel("Dimension 1 (83%)")
plt.ylabel("Dimension 2 (8 %)")
plt.suptitle("Premier plan factoriel (91%)")
plt.show()
```

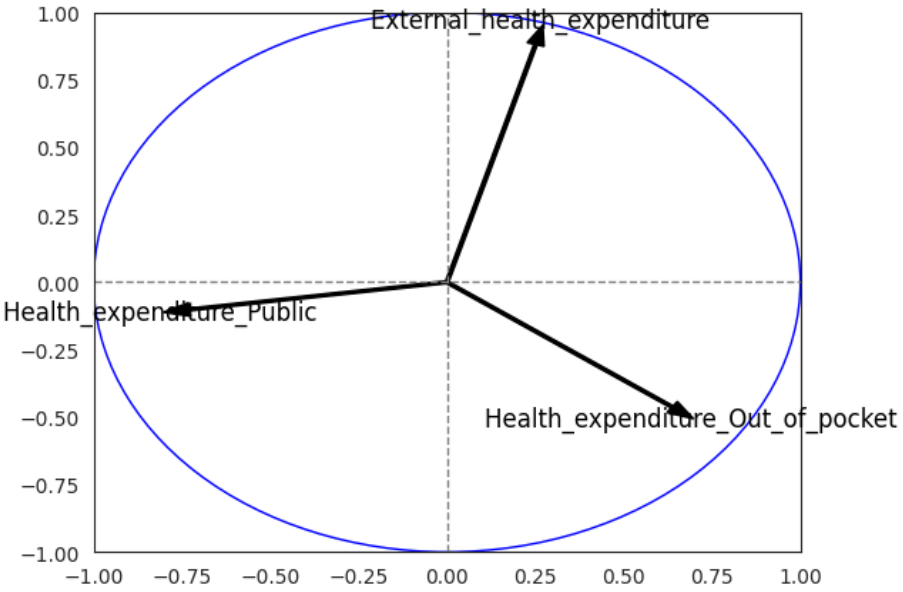


1.3 Presentation of the Variables on the factorial plan:

```
[ ] #présentation des variables
    n = system_health_num.shape[0] # nb individus
    p = system_health_num.shape[1] # nb variables
    eigval = (n-1) / n * pca.explained_variance_ # valeurs propres
    sqrt_eigval = numpy.sqrt(eigval) # racine carrée des valeurs propres
    corvar = numpy.zeros((p,p)) # matrice vide pour avoir les coordonnées
    for k in range(p):
        corvar[:,k] = pca.components_[k,:] * sqrt_eigval[k]
    # on modifie pour avoir un dataframe
    coordvar = pandas.DataFrame({'id': system_health_num.columns, 'COR_1': corvar[:,0], 'COR_2': corvar[:,1]})
    coordvar
```

	id	COR_1	COR_2
0	Health_expenditure_Public	-19.799649	-4.515682
1	Health_expenditure_Out_of_pocket	16.453236	-7.476298
2	External_health_expenditure	2.879740	11.667848

According to this correlation matrix we can notice that the "external" type expenses are positively correlated on axis 2 while the "public" type expenses are negatively colored on axis 1.For the "out of pocket" type expenses are positively correlated with axis 1.



Correlation circle:

According to this circle we can notice that the "external" type of expenditure admits a good quality of representation on axis 2 since it admits a strong positive value on axis 2 and it is very close to the circle.

For the expense type "out of pocket" we noticed that is positively correlated on axis 2.

And for the 3rd type of expenditure (public) it was noticed that is negatively correlated on the axis 1

➢**Interpretation:**

We noticed that countries like Micronesia, Malawi, Rwanda, Mozambique... are countries whose health expenditures are basically of external type
while countries like Afghanistan and Cameroon are countries whose health expenditures are out of pocket type. And for the 3rd type which is public we find it for countries like Seychelles.

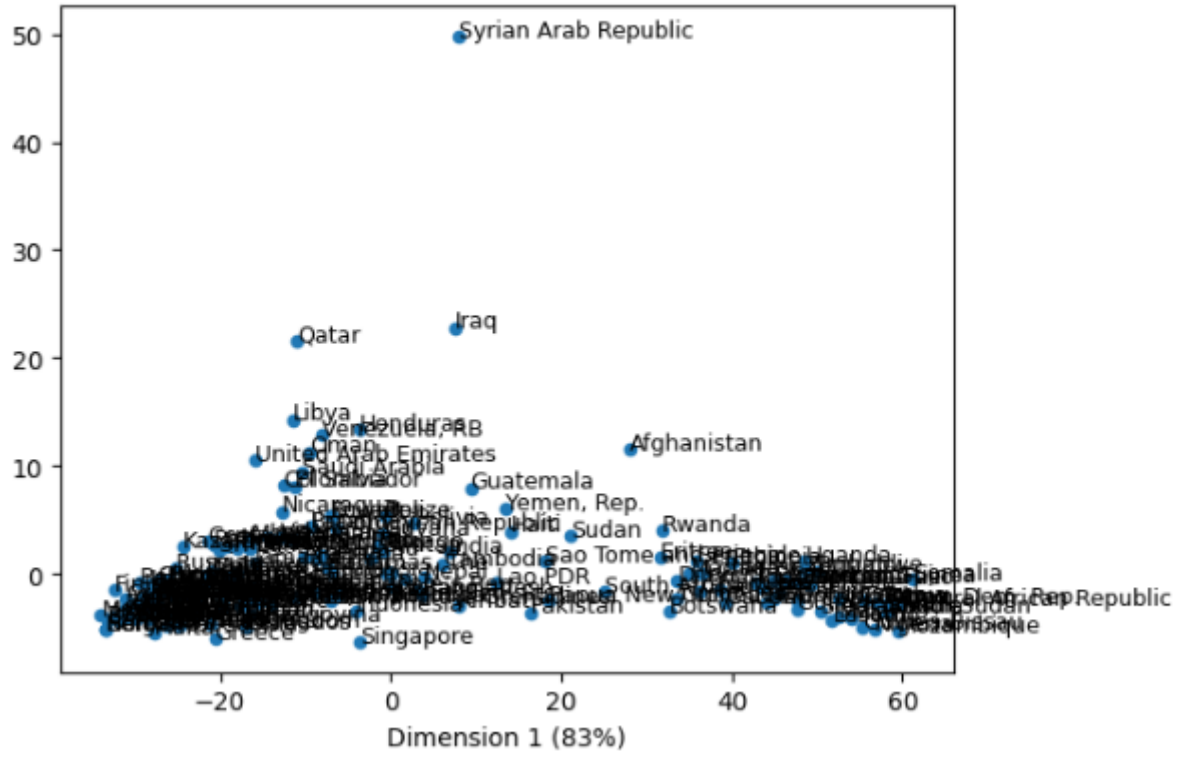
2. Describe the mortality Rate by type and by country

For this objective we will apply the PCA algorithm: It is an algorithm used to analyze the correlations between the types of mortality.
Input data: we used our Dataset 'Health' the columns: country , Communicable diseases , Non-communicable diseases , Injuries %.
These last three present the type of death

	Country	Communicable diseases	Non-communicable diseases	\
0	Afghanistan	36.00000	44.000000	
1	Albania	3.00000	93.000000	
2	Algeria	15.00000	76.000000	
3	American Samoa	22.31694	68.781421	
4	Andorra	22.31694	68.781421	
..	...	...	...	
209	Virgin Islands (U.S.)	22.31694	68.781421	
210	West Bank and Gaza	22.31694	68.781421	
211	Yemen, Rep.	29.00000	57.000000	
212	Zambia	61.00000	29.000000	
213	Zimbabwe	55.00000	33.000000	

	Injuries %
0	20.00000
1	4.00000
2	10.00000
3	9.04918
4	9.04918
..	...
209	9.04918
210	9.04918
211	15.00000
212	10.00000
213	12.00000

[214 rows x 4 columns]



```
n = WGI_num.shape[0] # nb individus
p = WGI_num.shape[1] # nb variables
eigval = (n-1) / n * pca.explained_variance_ # valeurs propres
sqrt_eigval = numpy.sqrt(eigval) # racine carrée des valeurs propres
corvar = numpy.zeros((p,p)) # matrice vide pour avoir les coordonnées
for k in range(p):
    corvar[:,k] = pca.components_[k,:] * sqrt_eigval[k]
# on modifie pour avoir un dataframe
coordvar = pandas.DataFrame({'id': WGI_num.columns, 'COR_1': corvar[:,0], 'COR_2': corvar[:,1]})
coordvar
```

	id	COR_1	COR_2
0	Communicable diseases	18.551893	-2.445240
1	Non-communicable diseases	-20.124363	-1.929043
2	Injuries %	1.494992	4.376652

We can deduce that the rate of type of death (communicable diseases) is positively correlated on the first axis but it is negative on the second axis
and for the second type of death (non-communicable diseases) are negatively correlated on the two axes and for the third type it is positive correlated on the two axes.

➢**Interpretation:**

Syrian arab public is the only country which has the highest death rate by the type 'injuries', while the countries (iraq,qatar,libya) have the highest death rate by the type 'communicable diseases' and finally the highest death rate by the type 'non-communicable diseases' exists in the countries 'Singapore' and 'greece'.

3. Segment countries by their research and development expenditure (R&D)

3.1 CAH algorithm

CAH which is a technique of data analysis that allows to group individuals (in the case of our objective the individuals are considered as being the countries) this grouping will be made in homogeneous groups according to their characteristics (in the case of our objective, it is about the expenses in R&D), whose goal is to group the countries according to their expenses in R&D.

3.1.1-Application of CAH

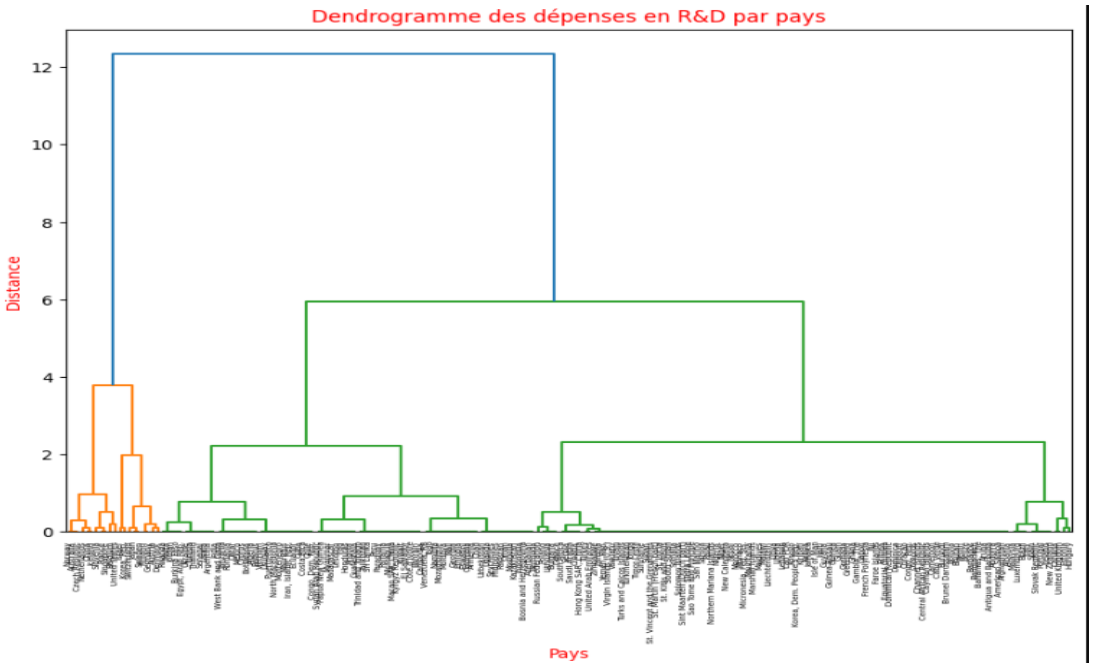
To build our similarity matrix

```
similarity_matrix = linkage(data.iloc[:, 1:].values, method='ward', metric='euclidean')
similarity_matrix
```

3.1.2 Construction and display of the dendrogram:

```
plt.figure(figsize=(10, 7))
plt.title("Dendrogramme des dépenses en R&D par pays")
plt.xlabel('Country')
plt.ylabel('Distance')
dendrogram(similarity_matrix, labels=data['Country'].tolist(), leaf_rotation=90)
plt.show()

plt.savefig('images/dendrogramme-depensess-RD.png', dpi=300, bbox_inches='tight')
```

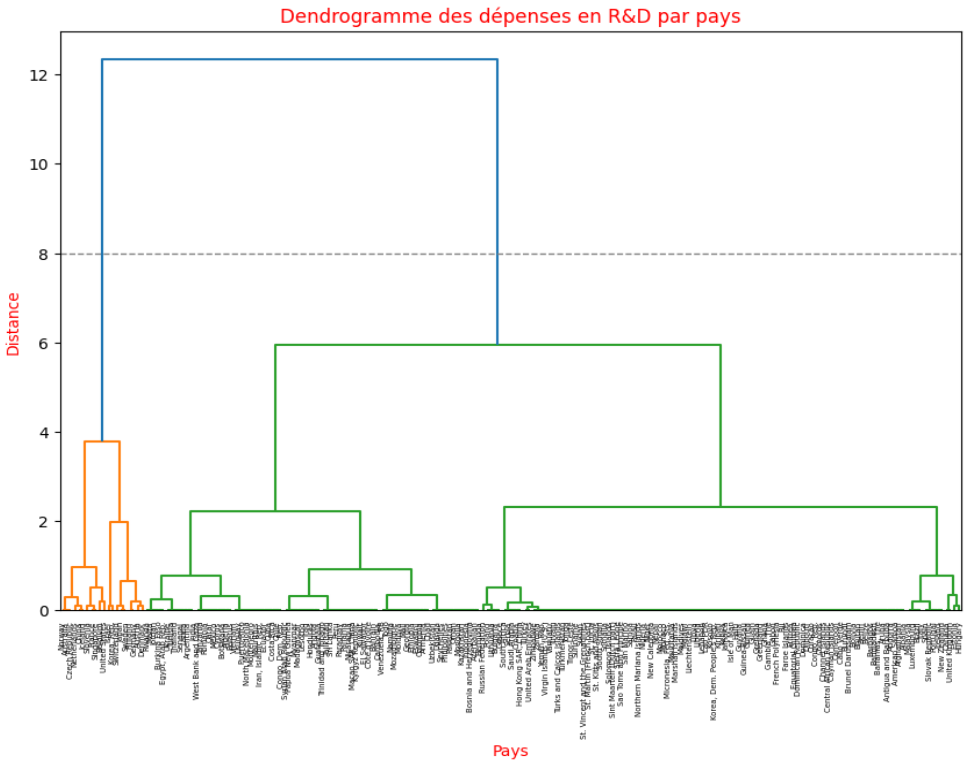


⇒ We can notice from this dendrogram visualization that represents the hierarchy of country groupings according to their similarity in terms of R&D spending.Countries close to each other in the dendrogram have the most similar R&D spending levels.

3.1.3 Choosing the height threshold for clustering:

```
# choisir le seuil de hauteur pour déterminer le nombre de clusters
threshold = 8
#dendrogram(similarity_matrix, labels=data['Country'].tolist(), leaf_rotation=90, color_threshold=threshold
plt.figure(figsize=(10, 7))
plt.title("Dendrogramme des dépenses en R&D par pays", c= "red")
plt.xlabel('Pays', c='red')
plt.ylabel('Distance',c='red')
dendrogram(similarity_matrix, labels=data['Country'].tolist(), leaf_rotation=90)
plt.axhline(y=threshold, c='gray', linestyle='--', linewidth=1)
plt.show()

#plt.savefig('images/dendrogramme-depensess-RD.png', dpi=300, bbox_inches='tight')
```



The dendrogram obtained after the application of CAH shows that there is the possibility to segment the countries according to their R&D expenditures into several clusters, depending on their similarity.

However, to achieve my basic objective, I chose to divide into 2 clusters, the countries that were the furthest apart in height are those that are the least similar (Height threshold to determine the number of clusters = 8) .

⇒ This segmentation into 2 clusters allowed us to group the countries more homogeneously according to their level of R&D expenditure.

3.1.4 Analysis of the clusters

After having segmented the countries into 2 clusters we wanted to determine the cluster with the most R&D expenditure and the one with the least R&D expenditure

```
from scipy.cluster.hierarchy import fcluster

# Seuil de hauteur pour déterminer le nombre de clusters
threshold = 8

# récupérer les clusters avec la fonction fcluster
clusters = fcluster(similarity_matrix, threshold, criterion='distance')

# créer une liste de listes qui contient les noms de pays de chaque cluster
clustered_countries = [[] for _ in range(max(clusters))]

for i, country in enumerate(data['Country']):
    clustered_countries[clusters[i]-1].append(country)

# afficher les noms de pays dans chaque cluster
for i, countries in enumerate(clustered_countries):
    print(f"Cluster {i+1} contient les pays suivants:")
    print(countries)
```

```
Cluster 1 contient les pays suivants:
['Australia', 'Austria', 'Belgium', 'China', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany', 'Iceland', 'Israel', 'Japan'
Cluster 2 contient les pays suivants:
['Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Andorra', 'Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba', '
```

⇒ Each cluster contains the countries for example

1st cluster : France Germany, Australia...

2nd cluster : Algeria,Angola,Argentina...

3.1.5 Calculation of the average

Following we calculated the average of R&D expenses for each cluster to determine which cluster contains the most expenses and which contains the least

```
# Séparer les pays en deux clusters
from scipy.cluster.hierarchy import fcluster
clusters = fcluster(similarity_matrix, threshold, criterion='distance')

# Calculer la moyenne des dépenses en R&D pour chaque cluster
mean_rnd_by_cluster = data.groupby('Cluster')['Expenditures_for_RD'].mean()
print(mean_rnd_by_cluster)
```

```
Cluster
1    2.715000
2    0.665233
Name: Expenditures_for_RD, dtype: float64
```

Average cluster 1 has a value of 2.715

Average cluster 2 has a value of 0.665

=> So cluster 1 with the countries we already mentioned above is the one containing the countries with the most R&D expenditure and the 2nd cluster with the lowest average is the one with the countries with the least R&D expenditure

➢**Interpretation:**

After applying the AHC to the R&D expenditure data of the different countries, we were able to achieve our initial objective: To present the countries according to their R&D expenditure, we have chosen to divide the countries into 2 clusters, we did this with the height threshold, which allowed us to separate the countries with the highest levels of expenditure in the 1st cluster which includes countries like (Australia', 'Austria', 'Belgium', 'China', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany', 'Iceland'. ...) and in the 2nd cluster, are the countries with the least R&D expenditure (Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Andorra', 'Angola'...).

Thus, this algorithm and this method allowed usto identify the levels of R&D spending between the different countries.

2. K-MEANS algorithm

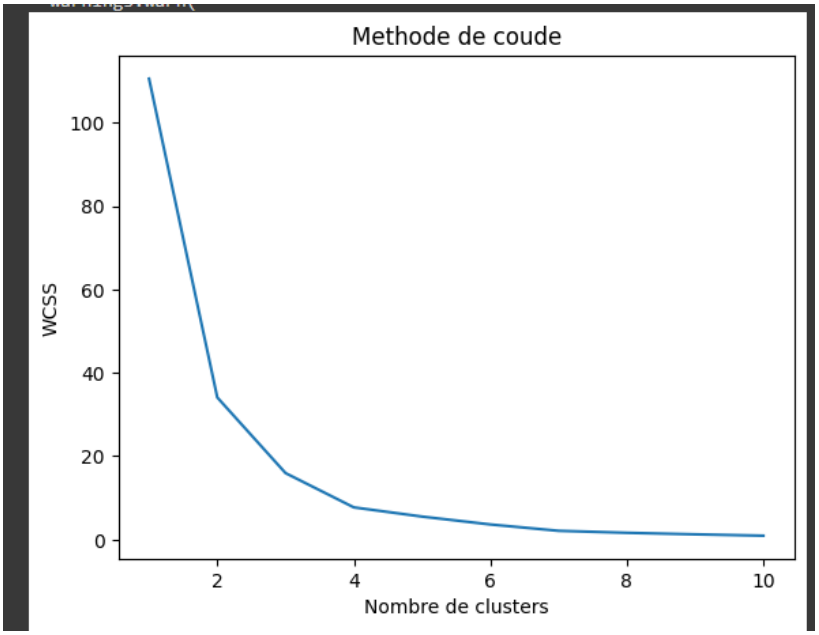
k-means which is a clustering method that allows to group similar observations (R&D expenses in the case of our objective) in a given number of clusters

2.1-The elbow method

Determination of the optimal number of clusters with the elbow method

```
#determination du nbre optimal de clusters avec methode de coude
wcss = []
for i in range (1,11):
  kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
  kmeans.fit(X)
  wcss.append(kmeans.inertia_)

#présentation de graphe
plt.plot(range(1, 11), wcss)
plt.title('Methode de coude')
plt.xlabel('Nombre de clusters')
plt.ylabel('WCSS')
plt.show()
```



=> In order to avoid choosing a random number of clusters that may not represent well the data structure I opted to use the elbow method in the first place there we can notice that the optimal number is 4 but according to my objective already fixed at the beginning I want to have 2 clusters the 1st one for the countries with the most R&D expenses and the 2nd one for those with the least expenses

2.2 Application of the k-means algorithm:

The algorithm will try to divide my data on 4 distinct clusters.

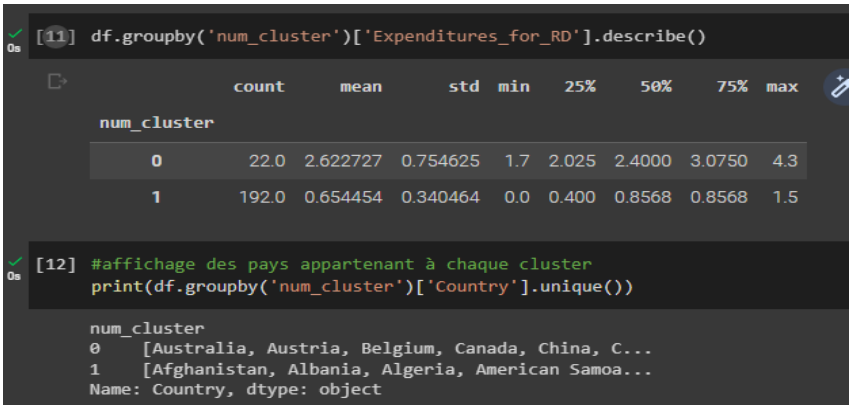
```
#application algo k-means
kmeans = KMeans(n_clusters=2, init='k-means++', random_state=42)
kmeans.fit(X)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:87
  warnings.warn(
```

```
KMeans(n_clusters=2, random_state=42)
```

2.3 Display of the clusters

display of the countries belonging to each cluster



```
[11] df.groupby('num_cluster')['Expenditures_for_RD'].describe()
```

num_cluster	count	mean	std	min	25%	50%	75%	max
0	22.0	2.622727	0.754625	1.7	2.025	2.4000	3.0750	4.3
1	192.0	0.654454	0.340464	0.0	0.400	0.8568	0.8568	1.5

```
[12] #affichage des pays appartenant à chaque cluster
print(df.groupby('num_cluster')['Country'].unique())

num_cluster
0    [Australia, Austria, Belgium, Canada, China, C...
1    [Afghanistan, Albania, Algeria, American Samoa...
Name: Country, dtype: object
```

=> There I could better understand the differences between my clusters in terms of their R&D expenses and can easily see that (either with mean or max) :

Cluster 0 -> has the most spending including countries (Belgium,Canada,Australia...)
Cluster 1 -> Less spending than cluster 0 including countries (Afghanistan,Albania,Algeria..)

➢**Interpretation:**

This analysis of clusters based on R&D spending using the k-means algorithm allowed me to have a clear idea , this last one allowed me to identify 2 different clusters, which have different levels of R&D spending going from countries that have the most spending that is cluster 0 (Belgium,Canada,Australia...) and those that have the least spending cluster 1 (Afghanistan,Albania,Algeria..)

4. Segment countries by their pesticides use

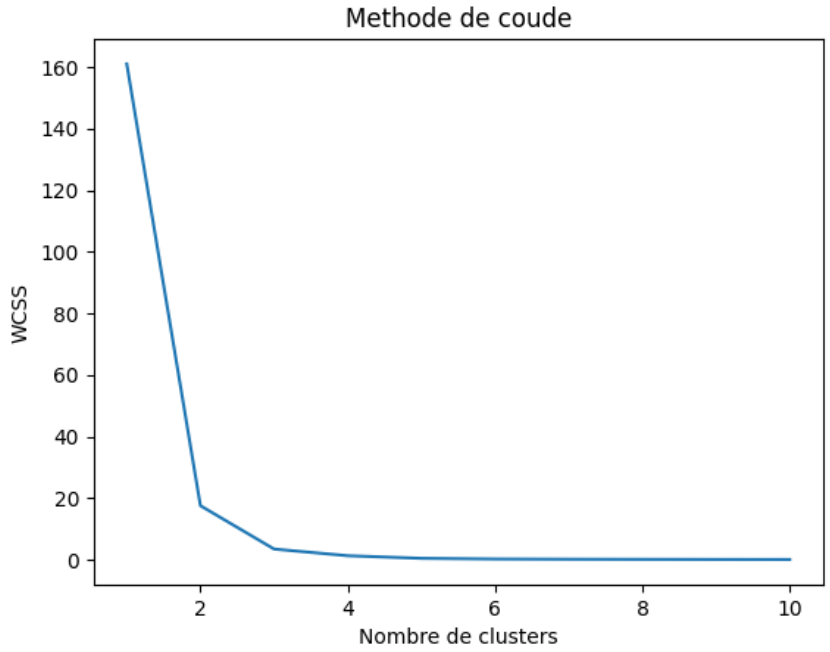
For this segmentation objective we worked with 2 algorithms which are k-means and CAH

5.1 k-means algorithm:

The K-means algorithm is an unsupervised machine learning algorithm used to group data points into groups or clusters based on their similarity. The objective of the algorithm is to minimize the variance or sum of squares of the distances between the data points and the centroid of their respective cluster

```
[ ]  #determination du nbre optimal de clusters avec methode de coude
     wcss = []
     for i in range (1,11):
       kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
       kmeans.fit(X_scaled)
       wcss.append(kmeans.inertia_)

     plt.plot(range(1, 11), wcss)
     plt.title('Methode de coude')
     plt.xlabel('Nombre de clusters')
     plt.ylabel('WCSS')
     plt.show()
```

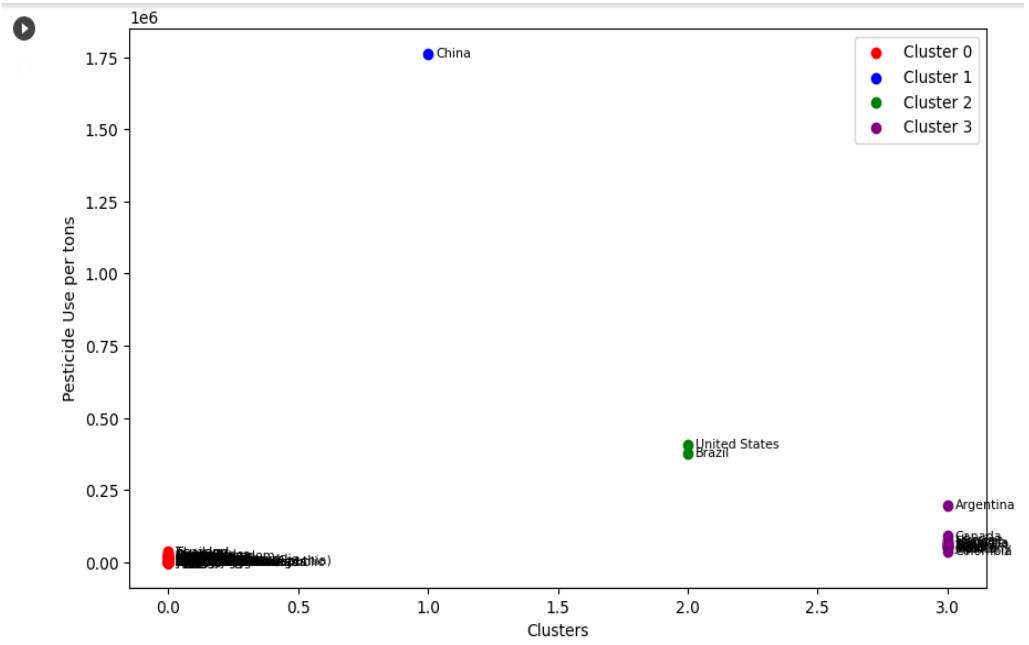


According to this result we have chosen to work with 4 clusters.

```
[ ]  #application algo k-means
     kmeans = KMeans(n_clusters=4, init='k-means++', random_state=42)
     kmeans.fit(X_scaled)
```

When applying this algorithm we obtain the following results:

```
[ ] # création du graphe
    colors = ['red', 'blue', 'green', 'purple']
    fig, ax = plt.subplots(figsize=(10, 6))
    for i in range(4):
        cluster_data = pecticides[pecticides['cluster_labels'] == i]
        ax.scatter(cluster_data['cluster_labels'], cluster_data['Pesticide Use (tons)'], color=colors[i], label=f"Cluster {i}")
        for country, x, y in zip(cluster_data['Country'], cluster_data['cluster_labels'], cluster_data['Pesticide Use (tons)']):
            ax.annotate(country, xy=(x, y), xytext=(5, 0), textcoords='offset points', ha='left', va='center', fontsize=8)
    ax.set_xlabel('Clusters')
    ax.set_ylabel('Pesticide Use per tons')
    ax.legend()
    plt.show()
```




```
[ ] print(pecticides.groupby('cluster_labels')['Country'].apply(lambda x: x.head(10)))
```

```
cluster_labels
0               17          Thailand
                18           Ecuador
                19      South Africa
                20            Russia
                21            Poland
                22         Guatemala
                23       South Korea
                24    United Kingdom
                25          Paraguay
                26           Vietnam
1               0              China
2               1      United States
                2             Brazil
3               3          Argentina
                4             Canada
                5            Ukraine
                6             France
                7           Malaysia
                8          Australia
                9              Spain
                10             Italy
                11            Turkey
                12             India
```

➢**Interpretation:**

According to these results we noticed that the "cluster1" contains only one country which is China which has a very high rate of use of pesticides.

For the United States and Brazil which belong to the "2nd cluster" are countries whose rate of use of their pesticides is average.

In the "3rd cluster" we find countries with low rates of pesticide use. These countries are Argentina,Canada,Ukraine,France,Malaysia,Australia,Spain,Italy,Turkey,India.

While the "cluster 0" contains the countries that have a very low rate of pesticide use.

Thus, this algorithm and method allowed me to identify the different levels of pesticide use by the different countries.

4.2 CAH algorithm:

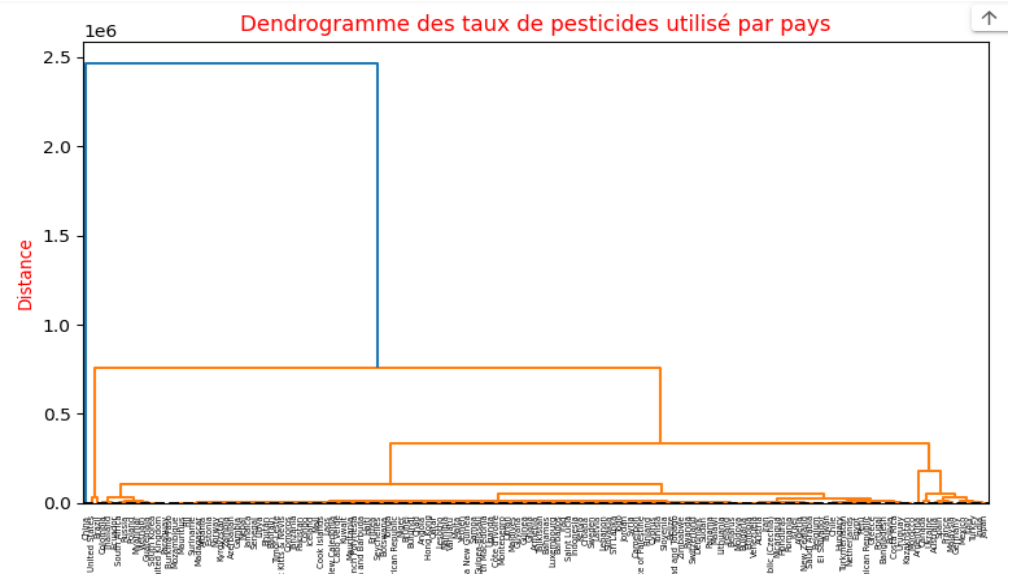
CAH (Classification Ascendant Hierarchical) is a data analysis technique that allows to group individuals (in the case of our objective individuals are considered as countries) this grouping will be done in homogeneous groups according to their characteristics (in the case of our objective, it is the rate of pesticide use), whose goal is to group countries according to their rate of pesticide use.

4.2.1 Construction of similarity matrix with the ward method:

```
[12] similarity_matrix = linkage(pesticides.iloc[:, 1:].values, method='ward', metric='euclidean')
     similarity_matrix
```

4.2.2-Dendrogram

```
plt.figure(figsize=(10, 7))
plt.title("Dendrogramme des taux de pesticides utilisé par pays", c='red')
plt.xlabel('Pays', c='red')
plt.ylabel('Distance' , c='red')
dendrogram(similarity_matrix, labels=pesticides['Country'].tolist(), leaf_rotation=90)
plt.axhline(y=300, c='black', linestyle='--')
plt.show()
```



From this dendrogram we can see a grouping of countries according to their similarity in terms of pesticide levels.

Countries close to each other in the dendrogram have similar levels of pesticide use.

4.2.3 Display of countries with the highest pesticide use rates:

```
# Obtenir les 5 pays  dont le taux d'utilisation de pecticides est  plus élevées
top_5_pays_pecticdes_elevees = pays[pesticides_rd.nlargest(5).index]
print("Les 5 pays  dont le taux d'utilisation de pecticides est  plus élevées :")
print(top_5_pays_pecticdes_elevees)
```

```
Les 5 pays  dont le taux d'utilisation de pecticides est  plus élevées :
0            China
1    United States
2           Brazil
3        Argentina
4           Canada
Name: Country, dtype: object
```

4.2.4 Display of countries with the lowest pesticide use rates

```
# Obtenir les 5 pays  dont le taux d'utilisation de pecticides est  plus faibles
top_5_pays_pecticdes_faibles = pays[pesticides_rd.nsmallest(5).index]
print("Les 5 pays  dont le taux d'utilisation de pecticides est   plus faibles :")
print(top_5_pays_pecticdes_faibles)
```

```
Les 5 pays  dont le taux d'utilisation de pecticides est   plus faibles :
160              Comoros
159                Congo
158             Pakistan
157             Tanzania
156    Saint Kitts & Nevis
Name: Country, dtype: object
```

➢**Interpretation:**

After applying CAH to our data from different countries we were able to achieve our initial objective, we identified the highest countries in terms of their pesticide use rate as china ,united states,Brazil,Argentina,Canada and the lowest countries in terms of their pesticide use rate as Comoros,Congo,Pakistan,Tanzania,Saint Kitts & Nevis development.

5. Segment countries by their agricultural production

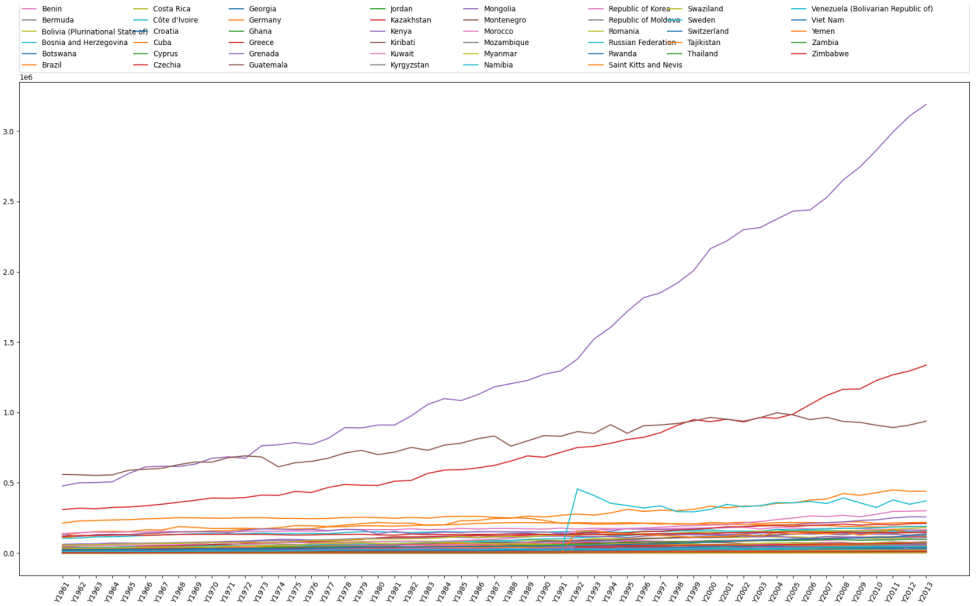
5.1 K_Means algorithm

K-Means clustering was chosen as the method to segment countries according to their agricultural production rate.

To understand the data, I used a graph for the annual production of different countries, with the quantity on the ordinate and the years on the abscissa.

```
area_list = list(df['Area'].unique())
year_list = list(df.iloc[:,10:].columns)

plt.figure(figsize=(24,12))
for ar in area_list:
    yearly_produce = []
    for yr in year_list:
        yearly_produce.append(df[yr][df['Area'] == ar].sum())
    plt.plot(yearly_produce, label=ar)
plt.xticks(np.arange(53), tuple(year_list), rotation=60)
plt.legend(bbox_to_anchor=(0., 1.02, 1., .102), loc=3, ncol=8, mode="expand", borderaxespad=0.)
plt.savefig('p.png')
plt.show()
```

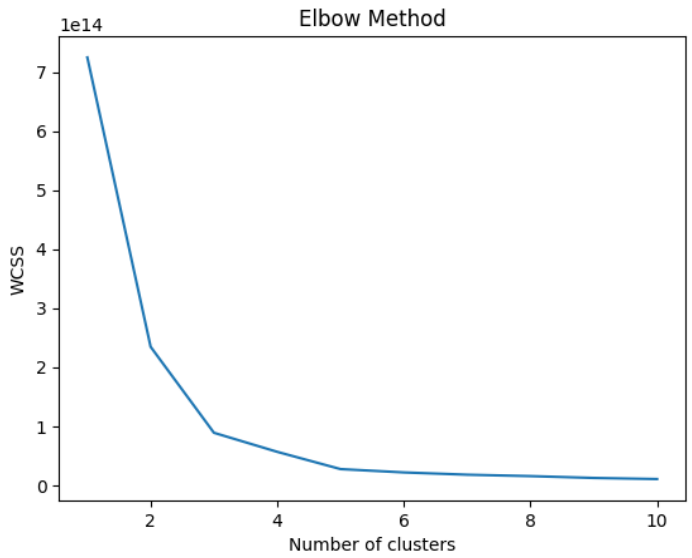


Clearly, China, India and the United States stand out here. These are therefore the countries with the largest production of food and feed.

5.1.1 Application of the Elbow method

The number of clusters is equal to the value on the x-axis of the point that corresponds to the corner at which the graph starts to stabilize (the graph often looks like an elbow).

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i,init='k-means++',max_iter=300,n_init=10,random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



As the graph starts to stabilize at x=4, we will have to form 4 clusters.

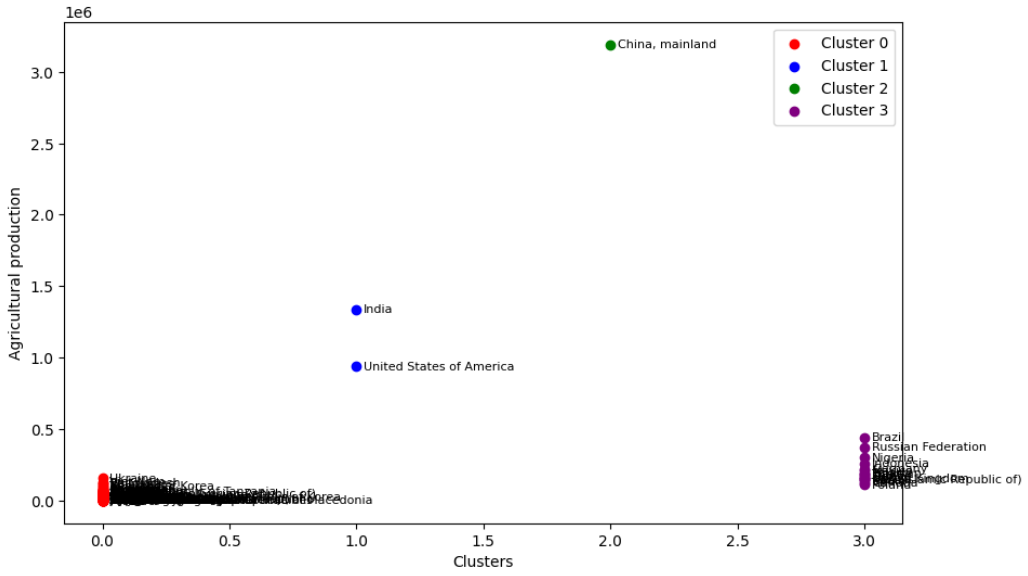
5.1.2 Application of the k-means algorithm:

```
kmeans = KMeans(n_clusters=4,init='k-means++',max_iter=300,n_init=10,random_state=0)
y_kmeans = kmeans.fit_predict(X)

X = X.as_matrix(columns=None)
```

5.1.3 Visualization of the clusters:

```
colors = ['red', 'blue', 'green', 'purple']
fig, ax = plt.subplots(figsize=(10, 6))
for i in range(4):
    cluster_data = new_df2[new_df2['cluster'] == i]
    ax.scatter(cluster_data['cluster'], cluster_data['Y2013'], color=colors[i], label=f"Cluster {i}")
    for country, x, y in zip(cluster_data['Country'], cluster_data['cluster'], cluster_data['Y2013']):
        ax.annotate(country, xy=(x, y), xytext=(5, 0), textcoords='offset points', ha='left', va='center', fontsize=8)
ax.set_xlabel('Clusters')
ax.set_ylabel('Agricultural production')
ax.legend()
plt.show()
```



- The graph we obtained shows that the number of clusters is equal to 4.

- We can deduce from the previous graph which presents the grouping of countries according to their agricultural production, that the countries are divided into 4 clusters:

- Cluster 0 which contains countries like Ukraine, Afghanistan and Antigua and Barbuda produce agricultural products in very small quantities.

- Cluster 1 which contains countries like India and the United States produce agricultural products in moderate quantities.

- Cluster 2 which contains countries like China produces agricultural products in large quantities.

- Cluster 3 which contains countries like Brazil, Russia and Nigeria produce agricultural products in low quantities.

6. Segment countries by their poverty rate

6.1 K-Means algorithm

For this purpose we will apply the k-means algorithm (K-MEANS) which is a partitioning method that allows to group countries according to their poverty rate

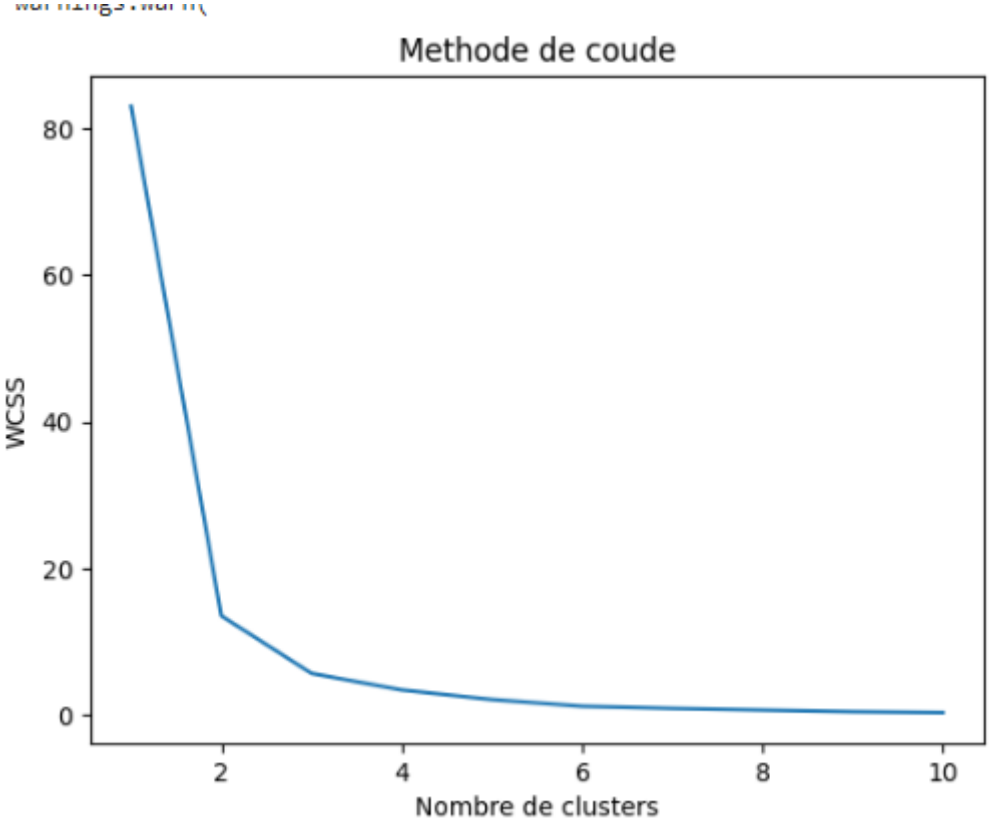
```
[ ]  print(data.columns.get_loc('poverty index value'))
     2

[ ]  #selection de la col de poverty index
     X = data.iloc[:, 1].values.reshape(-1,1)

[ ]  #normalisation de la col sélectionnée
     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X)
```

```
[ ]  #determination du nbre optimal de clusters avec methode de coude
     wcss = []
     for i in range (1,11):
       kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
       kmeans.fit(X_scaled)
       wcss.append(kmeans.inertia_)

     plt.plot(range(1, 11), wcss)
     plt.title('Methode de coude')
     plt.xlabel('Nombre de clusters')
     plt.ylabel('WCSS')
     plt.show()
```

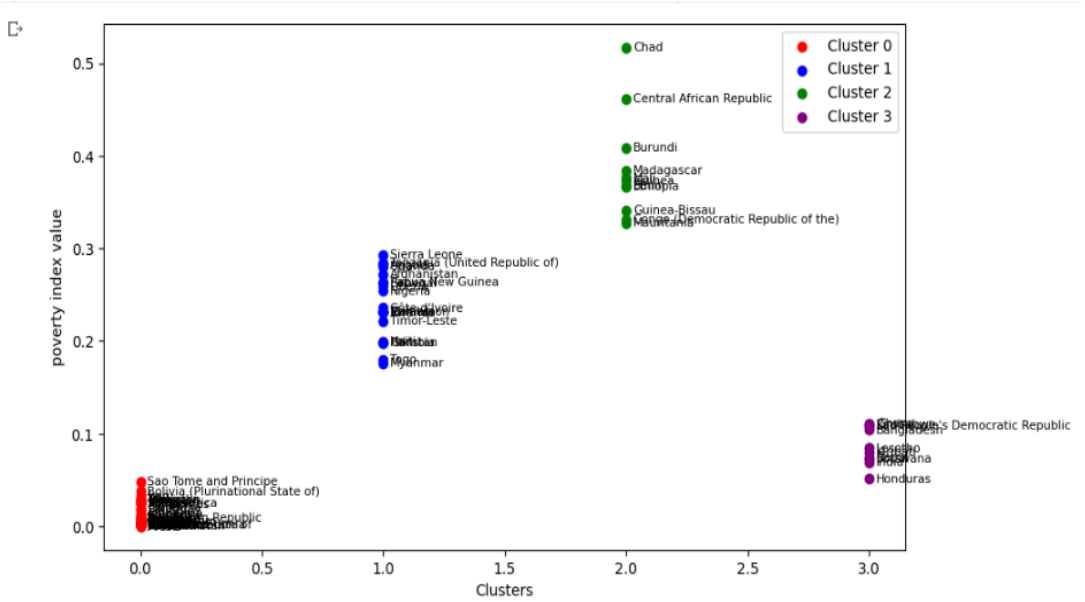


After this result we chose 4 classes (segmentation on 4 clusters)

```
[ ] # Création d'une table pivot
    pivot_table = pd.pivot_table(data, values='Country', index='cluster_labels', aggfunc=lambda x: list(x))

    # Affichage de la table pivot
    print(pivot_table)
```

```
                                                  Country
cluster_labels
0               [Albania, Algeria, Argentina, Armenia, Belize,...
1               [Afghanistan, Angola, Cameroon, Côte d'Ivoire,...
2               [Benin, Burundi, Central African Republic, Cha...
3               [Bangladesh, Botswana, Ghana, Honduras, India,...
```



➢ **Interpretation:**

After this result we can deduce that in 'cluster 2' we have countries with very high poverty rate (Chad, Central African Republic, Burundi)

In 'cluster 1' we have countries with average poverty rate, we can quote as example (sierra leone, côte d'ivoire, afghanistan)

(Ghana, Honduras, Botswana, Zimbabwe) belong to 'cluster 3' are countries that have low poverty index compared to countries that belong to cluster 2 and cluster 1

Finally for the 'cluster 0' which contains the countries that have very low poverty rate like (Algeria , costa rica ,albania,argentina

7 Segment countries by their renewable energy consumption

```
[ ]  kmeans = KMeans(n_clusters=3, random_state=42)
     kmeans.fit(data_k)

     /usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870:
       warnings.warn(
```

KMeans

KMeans(n_clusters=3, random_state=42)

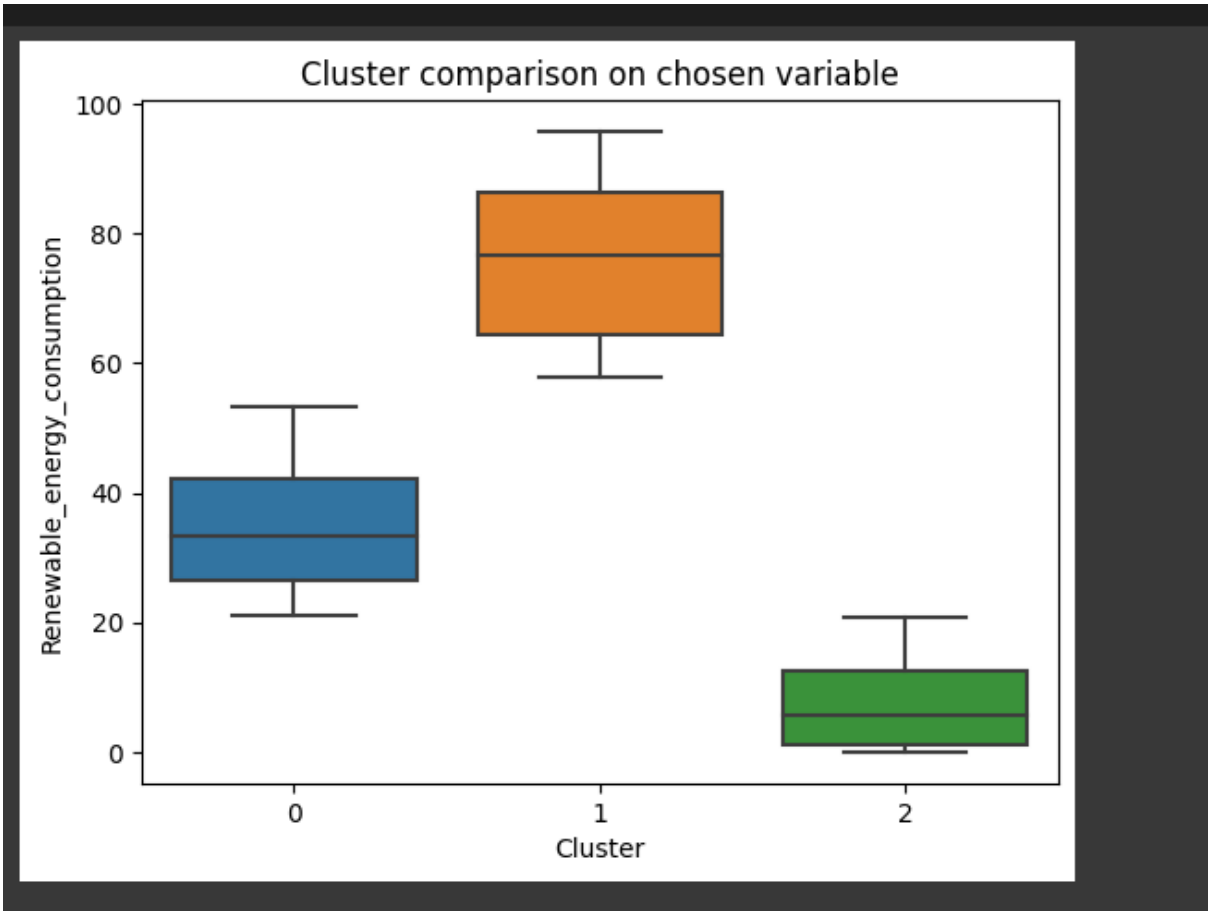
```
[ ] kmeans.labels_
```

```
array([2, 0, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 0, 2,
       1, 2, 0, 0, 0, 2, 2, 1, 1, 0, 1, 1, 0, 2, 1, 1, 0, 0, 2, 0, 0, 1,
       1, 0, 1, 0, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 1, 0, 1, 1, 2, 0,
       0, 2, 2, 1, 0, 0, 2, 0, 2, 2, 2, 2, 1, 1, 1, 0, 1, 0, 2, 2, 1, 0,
       0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 0, 2, 2, 2, 0, 1, 0, 2, 0,
       1, 2, 1, 0, 2, 2, 1, 1, 2, 2, 1, 2, 2, 0, 2, 2, 2, 2, 0, 2, 0, 2,
       1, 1, 0, 1, 2, 2, 0, 0, 1, 1, 0, 2, 1, 2, 0, 2, 0, 0, 1, 0, 0, 2,
       0, 2, 2, 0, 2, 1, 0, 0, 0, 2, 0, 0, 2, 1, 2, 2, 2, 2, 1, 1, 2, 0,
       2, 0, 2, 2, 0, 2, 1, 0, 0, 0, 2, 0, 1, 0, 2, 1, 2, 2, 2, 2, 2, 2,
       2, 1, 2, 2, 2, 2, 1, 2, 0, 2, 0, 2, 2, 2, 1, 1], dtype=int32)
```

```
[ ] data['cluster'] = kmeans.labels_
```

```
[ ] data.head(3)
```

	Country	Mortality_rate	Health_expenditure_Current	Expenditures_for_RD	Health_Workers	Renewable_energy_consumption	cluster
0	0	0.948840	10.2	0.8568	0.6000	18.4	2
1	1	0.371790	6.7	0.8568	4.8116	38.6	0
2	2	0.367918	6.6	0.5000	4.0121	0.1	2



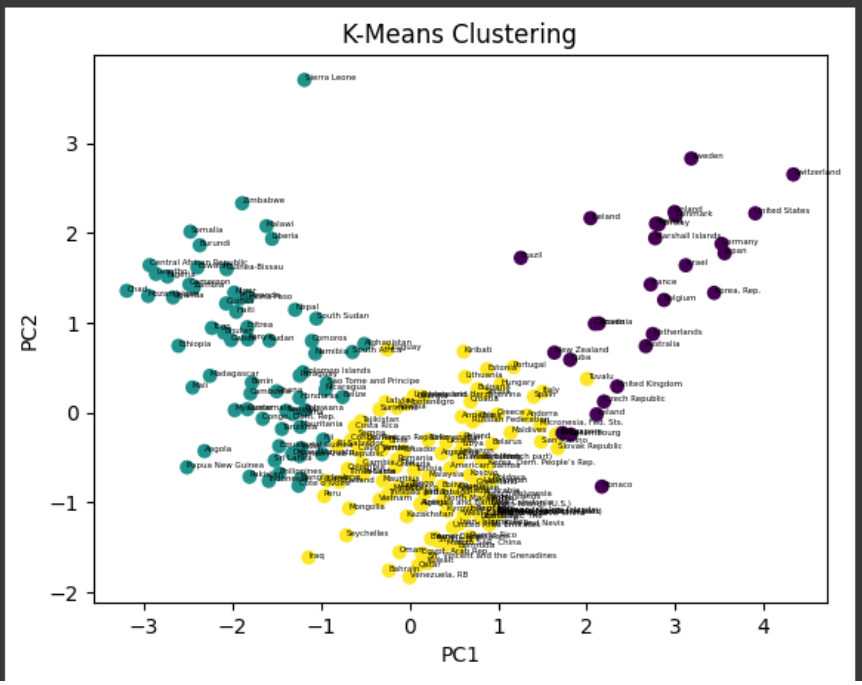
```
[ ]  # Scatter plot of PC1 and PC2 with different colors for each cluster
     plt.scatter(X_pca[:,0], X_pca[:,1], c=data['cluster'], cmap="viridis")

     # Annotate each point with its corresponding country name
     for i, country in enumerate(data['Country']):
         plt.annotate(country, (X_pca[i,0], X_pca[i,1]),fontsize=4)

     # Set the x-axis and y-axis labels
     plt.xlabel("PC1")
     plt.ylabel("PC2")

     # Set the plot title
     plt.title("K-Means Clustering")

     # Show the plot
     plt.show()
```



8. Understand the development factors of countries

The main objective of using this algorithm is to discover hidden patterns or associations in the data that can help us better understand how these variables are interconnected.

```
[8]  frequent_itemsets = apriori(df.drop('HDI', axis=1), min_support=0.5, use_colnames=True)
```

8.1 Application of the association rules

Application of the association rules function on frequent itemsets with a minimum confidence of 0.7 to generate association rules.

```
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Agricultural Production_Low)	(RD_Expenditures_Low)	0.668224	0.906542	0.607477	0.909091	1.002812	0.001703	1.028037
1	(Pesticide use kg per hectare_Very Low)	(RD_Expenditures_Low)	0.621495	0.906542	0.565421	0.909774	1.003566	0.002009	1.035826
2	(HDI_Binary)	(RD_Expenditures_Low)	0.733645	0.906542	0.649533	0.885350	0.976624	-0.015547	0.815161
3	(RD_Expenditures_Low)	(HDI_Binary)	0.906542	0.733645	0.649533	0.716495	0.976624	-0.015547	0.939507
4	(Agricultural Production_Low)	(Pesticide use kg per hectare_Very Low)	0.668224	0.621495	0.514019	0.769231	1.237710	0.098720	1.640187
5	(Pesticide use kg per hectare_Very Low)	(Agricultural Production_Low)	0.621495	0.668224	0.514019	0.827068	1.237710	0.098720	1.918529
6	(HDI_Binary)	(Agricultural Production_Low)	0.733645	0.668224	0.560748	0.764331	1.143824	0.070508	1.407805
7	(Agricultural Production_Low)	(HDI_Binary)	0.668224	0.733645	0.560748	0.839161	1.143824	0.070508	1.656034
8	(Pesticide use kg per hectare_Very Low)	(HDI_Binary)	0.733645	0.621495	0.537383	0.732484	1.178583	0.081426	1.414887
9	(HDI_Binary)	(Pesticide use kg per hectare_Very Low)	0.621495	0.733645	0.537383	0.864662	1.178583	0.081426	1.968069
10	(HDI_Binary, RD_Expenditures_Low)	(Agricultural Production_Low)	0.668224	0.649533	0.500000	0.769784	1.151985	0.065966	1.441151
11	(HDI_Binary, Agricultural Production_Low)	(RD_Expenditures_Low)	0.560748	0.906542	0.500000	0.891667	0.983591	-0.008341	0.862689
12	(RD_Expenditures_Low, Agricultural Production_...	(HDI_Binary)	0.607477	0.733645	0.500000	0.823077	1.121901	0.054328	1.505486
13	(Agricultural Production_Low)	(HDI_Binary, RD_Expenditures_Low)	0.668224	0.649533	0.500000	0.748252	1.151985	0.065966	1.392134

➢**Interpretation:**

We can see that there are several interesting associations between the analyzed variables. For example:

-Low agricultural production and low pesticide use are associated with lower research and development (R&D) spending.

Conversely, a high HDI (human development index) is associated with higher R&D spending

-And high pesticide use is associated with higher agricultural production.

-There are also interesting combination rules, such as the one in row 10 that indicates that R&D spending is likely to be low when agricultural production is low and the HDI is high.

This algorithm allowed us to identify important relationships between the different variables.

9. predict HDI rate

For this objective, we have chosen to work with the multiple linear regression algorithm to predict the development rate based on the mortality rate, agricultural production rate, poverty rate, and renewable energy consumption rate.

The multiple linear regression algorithm is a statistical analysis method that allows modeling the relationship between a dependent variable (in this case, the development rate) and several independent variables (in this case, the mortality rate, agricultural production rate, poverty rate, and renewable energy consumption rate).

10. Predict countries by their HDI rate

By utilizing Knn for prediction and K means for segmentation, a model was trained on a dataset that comprises different factors and their corresponding HDI rates, which then we employed to anticipate the HDI classification of a new country based on its factor values. This approach enables the prediction of country classification based on factors that contribute to the HDI rate.

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.neighbors import KNeighborsClassifier
```

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.neighbors import KNeighborsClassifier
```

```
[ ]  # Charger le fichier Excel en DataFrame Pandas
     df = pd.read_excel("final DataSet.xlsx")
```

```
[ ] # Nombre total de valeurs manquantes pour chaque caractéristique
    print(df.isnull().sum())

    Country                          0
    HDI                             39
    Expenditures_for_RD              0
    Health_expenditure_Current       0
    Health_Workers                   0
    Mortality_rate                   0
    Renewable_energy_consumption     0
    Agricultural production         54
    dtype: int64

[ ] df["HDI"].fillna(df["HDI"].mean(), inplace=True)
    df["Agricultural production"].fillna(df["Agricultural production"].mean(), inplace=True)

[ ] df
```

After importing the necessary libraries and loads our dataset stores in the Excel file named final DataSet, we printed the total number of missing values for each feature in the df dataframe ,then we filled the missing values in the HDI and Agricultural production column with the mean value using the **fillna().**

```
[ ] #Encodage countries
    data = pd.get_dummies(df, columns=['Country'])
```

With this code we performed the one-hot encoding in the **country** column of the data frame **df** in order to convert the categorical data into numerical data.

```
# Normalisation de données HDI
df['HDI'] = (df['HDI'] - df['HDI'].min()) / (df['HDI'].max() - df['HDI'].min())
```

Then we performed the data normalization on the **HDI** column in order to rescale the data.

```
[ ] # Initialiser le modèle KMeans avec deux clusters
    kmeans = KMeans(n_clusters=4)

    # Effectuer l'analyse de segmentation
    kmeans.fit(data)

    # Obtenir les étiquettes de cluster pour chaque pays
    cluster_labels = kmeans.predict(data)

    # Ajouter les étiquettes de cluster au DataFrame Pandas
    #df['cluster'] = cluster_labels
    #ajout de clusters dans le dataframe
    df['num_cluster'] = kmeans.labels_
    df

    /usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the
      warnings.warn(
```

	Country	HDI	Expenditures_for_RD	Health_expenditure_Current	Health_Workers	Mortality_rate	Renewable_energy_consumption	Agricultural production	num_cluster
0	Afghanistan	0.171030	0.8568	10.200000	0.600000	0.948840	18.4	23007.0000	2
1	Albania	0.724258	0.8568	6.700000	4.811600	0.371790	38.6	8271.0000	3
2	Algeria	0.628272	0.5000	6.600000	4.012100	0.367918	0.1	72161.0000	3

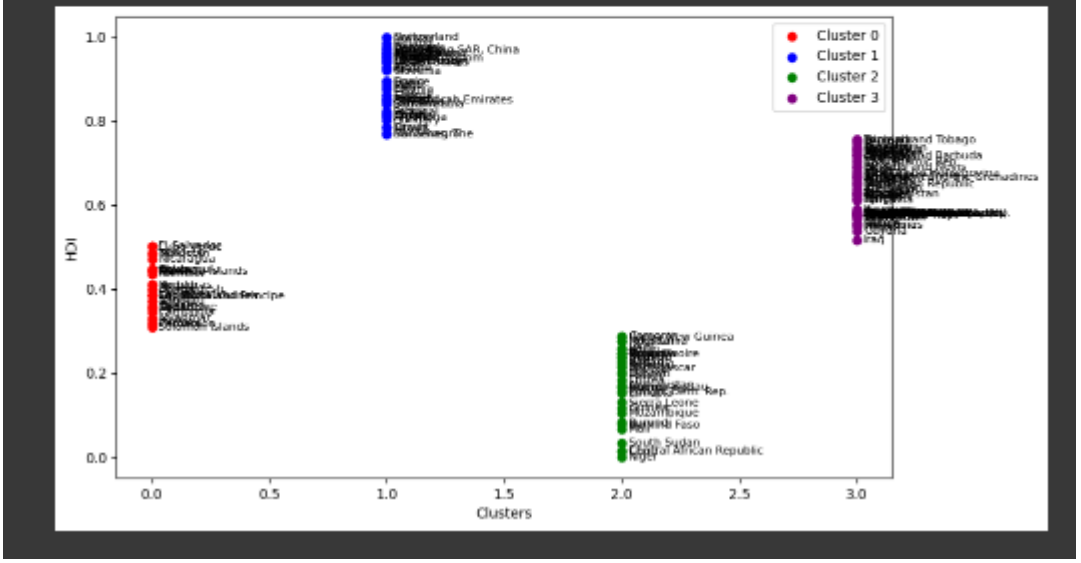
Then we performed clustering analysis using the K-Means algorithm on our dataset,we initialized the K-Means with clusters and fit it on the encoded and normalized data.Then it added the cluster labels to the dataframe as a new column named **num_cluster**.

```
[ ]  df.groupby('num_cluster')['HDI'].describe()
```

num_cluster	count	mean	std	min	25%	50%	75%	max
0	30.0	0.400233	0.057496	0.308901	0.355148	0.391798	0.444590	0.502618
1	47.0	0.894880	0.068851	0.767888	0.840314	0.893543	0.954625	1.000000
2	34.0	0.176573	0.085298	0.000000	0.119983	0.198953	0.244764	0.287958
3	103.0	0.624758	0.061998	0.516579	0.581401	0.581401	0.670157	0.755672

```
[ ]  import matplotlib.pyplot as plt

    # création du graphe
    colors = ['red', 'blue', 'green', 'purple']
    fig, ax = plt.subplots(figsize=(10, 6))
    for i in range(4):
        cluster_data = df[df['num_cluster'] == i]
        ax.scatter(cluster_data['num_cluster'], cluster_data['HDI'], color=colors[i], label=f"Cluster {i}")
        for country, x, y in zip(cluster_data['Country'], cluster_data['num_cluster'], cluster_data['HDI']):
            ax.annotate(country, xy=(x, y), xytext=(5, 0), textcoords='offset points', ha='left', va='center', fontsize=8)
    ax.set_xlabel('Clusters')
    ax.set_ylabel('HDI')
    ax.legend()
    plt.show()
```



11. Analyze people's opinions towards the sustainable development subject

In order to analyze the opinion of people to see their interest in the subject of sustainable development we decided to analyze the comments of a youtube video that addresses the subject of sustainable development, its aspects and its problems.

11.1 Establishing a connection to youtube's API

```
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError
from textblob import TextBlob
import re
import pandas as pd
import matplotlib.pyplot as plt
```

```
[2] DEVELOPER_KEY = "AIzaSyCob7XxrdN_ZBgM9ixmmyo2z7NyC0RC2IE"
    YOUTUBE_API_SERVICE_NAME = "youtube"
    YOUTUBE_API_VERSION = "v3"

    youtube = build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION, developerKey=DEVELOPER_KEY)
```

This code allows us to establish a connection to the YouTube Data API using an API key. and by creating a **youtube** object,we can interact with the API and retrieve information related to the YouTube videos,in our case we wanted to retrieve the comments of the video so we can analyze user behavior and his engagement with sustainable content on the platform.

```
# Connexion à l'API YouTube
youtube = build('youtube', 'v3', developerKey=DEVELOPER_KEY)
```

By initializing a connection to the YouTube Data API using the API key ,we were able to connect to the YouTube API and retrieve information related to the YouTube Video.

11.2 Retrieving the comments

```python
video_id = "N3SQlrmV1cE"

commentss = []
results = youtube.commentThreads().list(
    part="snippet",
    videoId=video_id,
    textFormat="plainText"
).execute()

while results:
    for item in results["items"]:
        comment = item["snippet"]["topLevelComment"]["snippet"]["textDisplay"]
        commentss.append(comment)

    if "nextPageToken" in results:
        results = youtube.commentThreads().list(
            part="snippet",
            videoId=video_id,
            pageToken=results["nextPageToken"],
            textFormat="plainText"
        ).execute()
    else:
        break
```

This part of code allowed us to retrieve the comments from the YouTube video that we have specified by the **video_id** , after retrieving the comments from the video the retrieved comments were appended to a list called **comments.**

11.3 Cleaning the comments

```python
# Nettoyage des commentaires
clean_comments = []
for comment in commentss:
    # Suppression des liens
    comment = re.sub(r'http\S+', '', comment)
    # Suppression des emojis
    comment = comment.encode('ascii', 'ignore').decode('ascii')
    # Suppression des caractères spéciaux
    comment = re.sub(r'[^\w\s]', '', comment)
    # Conversion en minuscules
    comment = comment.lower()
    clean_comments.append(comment)
```

```
[ ] clean_comments

['i think the us government has forgotten about this',
 'seeing brics nations on a possitive trajectory gives some hope makes me wonder where south africa would be in 2030',
 '1300',
 'i am wondering how the indicators for progress are now post covid i anticipate all countries are struggling and very much behind
 'a very descriptive presentation however i would have one more comment yes water and sanitation is a solved problem in developed
requires high financial resources and technical knowledge and therefore can not be easily applied to developing countries',
```

This code allowed us to clean the comments retrieved from the YouTube video using regular expressions and other string operations.The cleaned comments are then stored in a new list called **clean_comments.**

This process of cleaning the comments is important in order to prepare the data for further analysis.

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
import pandas as pd
import nltk
from nltk.probability import FreqDist
from nltk.collocations import BigramAssocMeasures, BigramCollocationFinder
from nltk.corpus import stopwords

# Charger les commentaires à partir du fichier Excel
comments_df = pd.read_excel('comments-clena-final.xlsx')
commentss = comments_df['comments'].apply(str)

# Tokenization des commentaires en mots
tokens = [nltk.word_tokenize(comments) for comments in commentss]

# Créer un finder de collocations avec une mesure de fréquence
finder = BigramCollocationFinder.from_documents(tokens)
measures = BigramAssocMeasures() #raw frequency

# Exclure les mots vides en anglais
stopwords_en = set(stopwords.words('english'))
finder.apply_word_filter(lambda word: word.lower() in stopwords_en)

# Extraire les collocations les plus fréquentes
collocations = finder.nbest(measures.raw_freq, 100)

# Calculer la fréquence de chaque collocation
```

```python
# Extraire les collocations les plus fréquentes
collocations = finder.nbest(measures.raw_freq, 100)

# Calculer la fréquence de chaque collocation
fdist = FreqDist(finder.ngram_fd)

# Imprimer les collocations avec leur fréquence
for collocation in collocations:
    print(collocation, fdist[collocation])
```

```
('ted', 'talk') 2
('united', 'nations') 2
('wan', 'na') 2
('world', 'government') 2
('wouldnt', 'want') 2
('1', 'undermine') 1
('10', 'base') 1
('100', 'lack') 1
('1015', 'billion') 1
('124', 'id') 1
('14000', 'usd') 1
('16', 'countries') 1
('2', 'flood') 1
('2018', 'sad') 1
('2030', 'hope') 1
('21', 'exposed') 1
('21', 'mass') 1
```

This code reads a previously cleaned Excel file containing the comments that we have extracted from the YouTube video, then tokenizes the comments into individual words,removes stop words in English,and finds the most frequent bigram collocations in the comments using raw frequency measures. and finally,prints the 100 most frequent collocations along with their frequency in the comments.

11.4 Display of the most frequent words

- New World -> 2
- raise gdp -> 2
- social progress -> 2
- world government-> 2
- global goals -> 4
- human rights -> 2
- agenda 21 -> 6 (Action plan for sustainable development adopted by 178 governments.)
- sustainble development -> 6
- agenda 2030 -> 3 (Agenda defined by the United Nations members in September 2015, consisting of 17 objectives.)
- One world -> 3
- development goals -> 2

This screenshot displays the most frequent words extracted from the comments of the YouTube video that are related to the topic of sustainable development.
Based on this, we were able to see people's opinions,which truly demonstrate their interest in this subject.

With this algorithm,we were able to analyze people’s interest in this topic by analyzing comments that come from a video discussing sustainable development.

V. Test And Evaluation

This phase is the verification phase of the performance of each method.

1. Segment countries by their research and development expenditure

To interpret and verify the performance of each method I used 2 measures: the silhouette coefficient (measures the quality of the separation between the different clusters) and the inertia (measures the distance between the points and their cluster center)

➢**K-means**

```
Inertie K-means : 34.098522520303035
Coefficient de silhouette K-means : 0.8131357202438535
```

➢**CAH**

```
Inertie CAH : 13.73378187482452
Coefficient de silhouette CAH : 0.6987268968079081
```

According to the results obtained we have the inertia for the CAH is lower than the inertia for the k-means

The silhouette coefficient of CAH is also smaller than the silhouette coefficient of k-means.

2. Segment countries by their pesticide use

➢**K-means**

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# K-means clustering
kmeans = KMeans(n_clusters=2)
kmeans.fit(pecticides.iloc[:, 1:].values)
inertia_kmeans = kmeans.inertia_
silhouette_kmeans = silhouette_score(pecticides.iloc[:, 1:].values, kmeans.labels_)

print("Inertie K-means :", inertia_kmeans)
print("Coefficient de silhouette K-means :", silhouette_kmeans)
```

```
Inertie K-means : 452060815435.6032
Coefficient de silhouette K-means : 0.9689026332840228
```

➢**CAH**

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# CAH clustering
similarity_matrix = linkage(pesticides.iloc[:, 1:].values, method='ward', metric='euclidean')
clusters = fcluster(similarity_matrix, 3, criterion='maxclust')
inertia_cah = similarity_matrix[-1, 2]
silhouette_cah = silhouette_score(pesticides.iloc[:, 1:].values, clusters)

print("Inertie CAH :", inertia_cah)
print("Coefficient de silhouette CAH :", silhouette_cah)
```

```
Inertie CAH : 2464791.718605953
Coefficient de silhouette CAH : 0.9474785714079121
```

From the results the inertia for CAH is smaller than the inertia for k-means
The silhouette coefficient of CAH is also smaller than the silhouette coefficient of k-means.

3. Segment countries by their agricultural production

In order to assess the effectiveness of the clustering methods, I used two metrics: the silhouette coefficient, which evaluates the quality of separation between different clusters, and the inertia, which measures the distance between points and their cluster centers. The K-means and CAH methods were applied, and the results showed that the inertia of the CAH method was lower than that of the K-means method. On the other hand, the silhouette coefficient of the CAH method was smaller than that of the K-means method.

4. Segment countries by their poverty rate

➢K-means

```
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics import silhouette_score

X, _ = make_blobs(n_samples=100, centers=4, random_state=42)

kmeans = KMeans(n_clusters=4, random_state=42)

kmeans.fit(X)

labels = kmeans.labels_

silhouette_avg = silhouette_score(X, labels)

print("Coefficient de silhouette du modèle K-means :", silhouette_avg)
```

```
Coefficient de silhouette du modèle K-means : 0.7937460187892489
```

```
[48] X, _ = make_blobs(n_samples=100, centers=4, random_state=42)

[47] kmeans = KMeans(n_clusters=4, random_state=42)

[49] X, _ = make_blobs(n_samples=100, centers=4, random_state=42)

     kmeans.fit(X)

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to
  warnings.warn(
KMeans
KMeans(n_clusters=4, random_state=42)

+ Code  + Texte

[51] inertia = kmeans.inertia_

     print("Inertie du modèle K-means :", inertia)

Inertie du modèle K-means : 169.64008915092137
```

the silhouette coefficient is 0.79 which is close to 1 this indicates a relatively good performance for our K-means model

The inertia has a value of 169.64 which indicates the cohesion of the clusters. This is a low value which suggests that the points within each cluster are closer to their centroid, which is desirable for good clustering

Overall, with a high silhouette coefficient and relatively low inertia, our K-means model seems to have succeeded in clustering the data in a meaningful and consistent way.

5. Predict HDI Rate

For the performance evaluation of linear regression model the following results were displayed

```
# Évaluer les performances du modèle
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print('Mean Squared Error (MSE):', mse)
print('Coefficient de détermination (R2):', r2)


Mean Squared Error (MSE): 0.002425882764672842
Coefficient de détermination (R2): 0.8308221400822785
```

According to the results this model has a coefficient of determination close to 1 (0.083) and a very low MSE of 0.002 so it is a good model

VI. Deployment

1. Segment countries by their research and development expenditure

The results obtained show that the analysis done with the k-means algorithm with 2 clusters shows a better segmentation quality of my data and a better segmentation between countries with high R&D expenditures and those with low R&D expenditures.

=> K-means seems to perform better than CAH

2. Segment countries by their pesticide use

According to the test results, the silhouette coefficient of k means is higher than that of CAH.So for this objective K-Means is the best performing algorithm

3. Segment countries by their agricultural production

Based on the results obtained, it can be concluded that the k-means algorithm with 4 clusters provides a more effective segmentation of the data and distinguishes more accurately between countries variating in agricultural production from low to very high, indicating that it outperforms the CAH algorithm.

4. Segment countries by their poverty rate

According to the test results we have the coefficient of silhouette of k means is high while the low inertia is therefore this algorithm is efficient and serve us to meet this objective

5. Predict HDI Rate

According to the results obtained, the linear regression model is a powerful model that perfectly meets our prediction objective

Conclusion:

In conclusion, data mining and the CRISP-DM methodology have been used in this project to analyze and understand the relationship between various development factors of countries. The project has used a variety of techniques, including clustering and linear regression, to segment countries based on their research and development expenditure, pesticide use, agricultural production, poverty rate, and renewable energy consumption.

The results of the analysis have shown interesting associations between the variables. For example, low agricultural production and low pesticide use are associated with lower

research and development spending, while a high human development index is associated with higher R&D spending. Additionally, high pesticide use is associated with higher agricultural production.

The project has also shown that the K-means algorithm generally outperforms the CAH algorithm in clustering the data for several objectives, including segmenting countries based on their pesticide use and agricultural production. However, for some objectives, such as segmenting countries based on their research and development expenditure, the CAH algorithm was found to perform better.

Overall, the results of the analysis have provided insights into the factors that contribute to a country's development and can be used to inform policy decisions aimed at promoting sustainable development.

Chapter 5:Data Visualization

Introduction:

Data visualization is a way of presenting information and data through the use of visual elements such as charts, graphs, and maps. Through interactive dashboards, we can effectively display our data to facilitate analysis and decision-making processes. This marks the third phase of the GIMSI methodology, which is the Implementation stage.

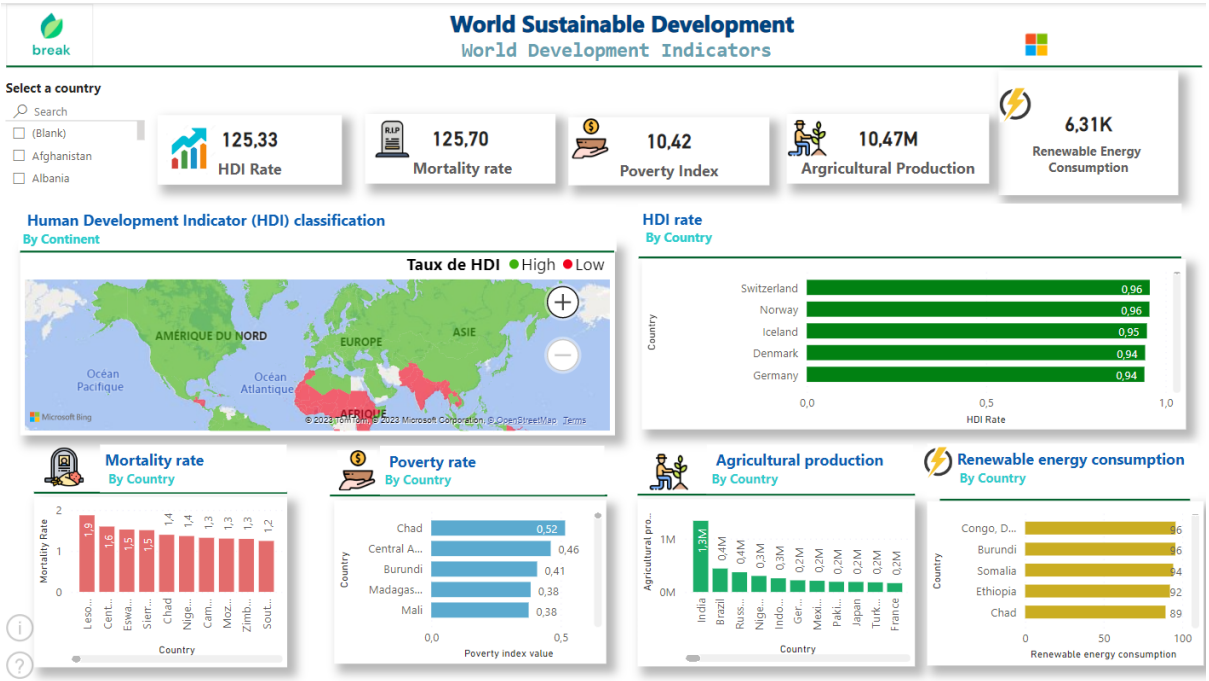
I. Power Bi

Microsoft's Power BI is a business analytics service that is designed to provide interactive visualizations and business intelligence functions. The service offers a simple user interface that enables end-users to create their own reports and dashboards. Additionally, Power BI includes data warehouse capabilities such as data preparation, data discovery, and interactive dashboards.

II. Dashboards

In this section, we will provide an overview of the complete set of dashboards we have developed using MS Power BI Services. In the following section, we will delve into each component in detail.

1. Overview of the general information



1.1 HDI (Human Development Index) rate

interpretation:

This map represents the global HDI rate in the world with a value of 125.33.

Impact

This map will allow the company to have a global idea of the human development rate in the world with the possibility to visualize it for each country.

Action to take:

Stay up to date and follow the evolution of this rate in the world.

1.2 HDI rate by continent

interpretation:

This map represents the global HDI rate by continent. According to this map, we can notice that Africa encompasses both developed and less developed countries while Europe and South America have countries with high rates and are considered developed countries.

Impact

This map will allow the company to easily identify the developed and less developed continents.

Action to take:

Stay up to date and follow the evolution in each continent of the world.

1.3 HDI rate by country

interpretation:

This bar chart represents a classification of countries according to their HDI rate. We have noticed that Switzerland, Norway, and Denmark have the highest HDI rates with 0.962 / 0.961 / 0.64 respectively, while those with the lowest HDI rates are Niger, Chad, and Sudan with 0.38 / 0.39 / 0.40 respectively.

Impact

This chart will allow the company to easily identify the ranking of countries with the value of the rate for each country.

Action to take:

The action to take is to improve or reduce the development factors that have an impact on this rate.

1.4 Mortality rate by country

interpretation:

This bar chart represents a classification of countries according to their mortality rate. Sudan has the highest mortality rate at 1.2%, and the lowest mortality rate is 0.19%.

Impact

This chart will allow the company to identify the mortality rate for each country.

Action to take:

The action to take here is to reduce the mortality rate while improving or reducing the factors that have an impact on it (these factors will be seen later in the second dashboard).

1.5 Poverty rate by country

interpretation:

This bar chart represents a classification of countries according to their poverty rate.

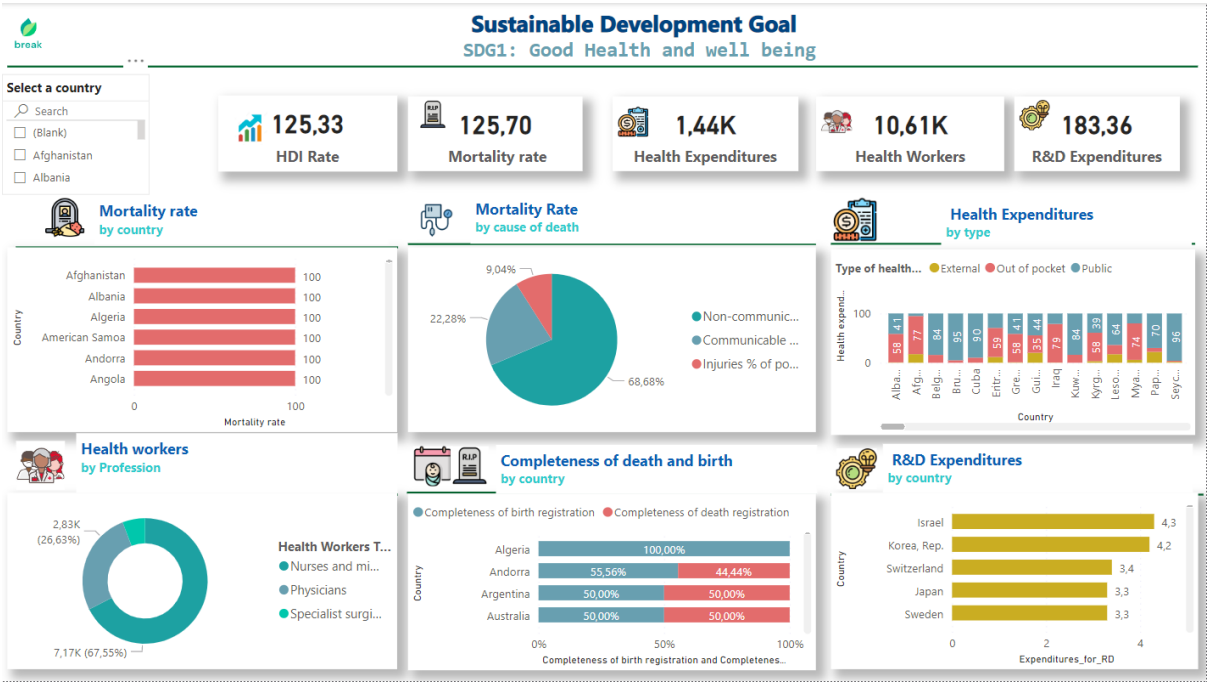
Impact

This chart will allow the company to identify the poverty rate for each country.

Action to take:

The action to take here is to reduce the poverty rate while improving or reducing the factors that have an impact on it (these factors will be seen later in the second dashboard).

2. Overview of SDG1 Good Health and well-being



2.1 Mortality rate by category of cause of death

interpretation:

This pie chart represents the distribution of mortality by cause of death.

We have 68.68% of mortality due to non-communicable diseases, 22.28% due to communicable diseases, and 9.04% due to injuries.

Impact

This chart will allow the company to distinguish the different causes of mortality.

Action to take:

The action to take is to minimize the mortality rate by opting for a more selective strategy according to the main cause of mortality.

2.2 Healthcare spending by type and by country

interpretation:

This bar chart represents healthcare spending by type of spending and by country.
For example, Afghanistan has 77.40% of direct spending, 5.1% for public spending, and 17.5% for external spending.

Impact

This chart will allow the company to identify the different types of healthcare spending for each country.

Action to take:

Adjust the company's strategy according to the type of healthcare spending.

2.3 Health workers by profession

interpretation:

This pie chart represents the distribution of health workers by profession.
We can observe that 67.55% are nurses and midwives, 26.63% are doctors, and 5.83% are specialist surgeons.

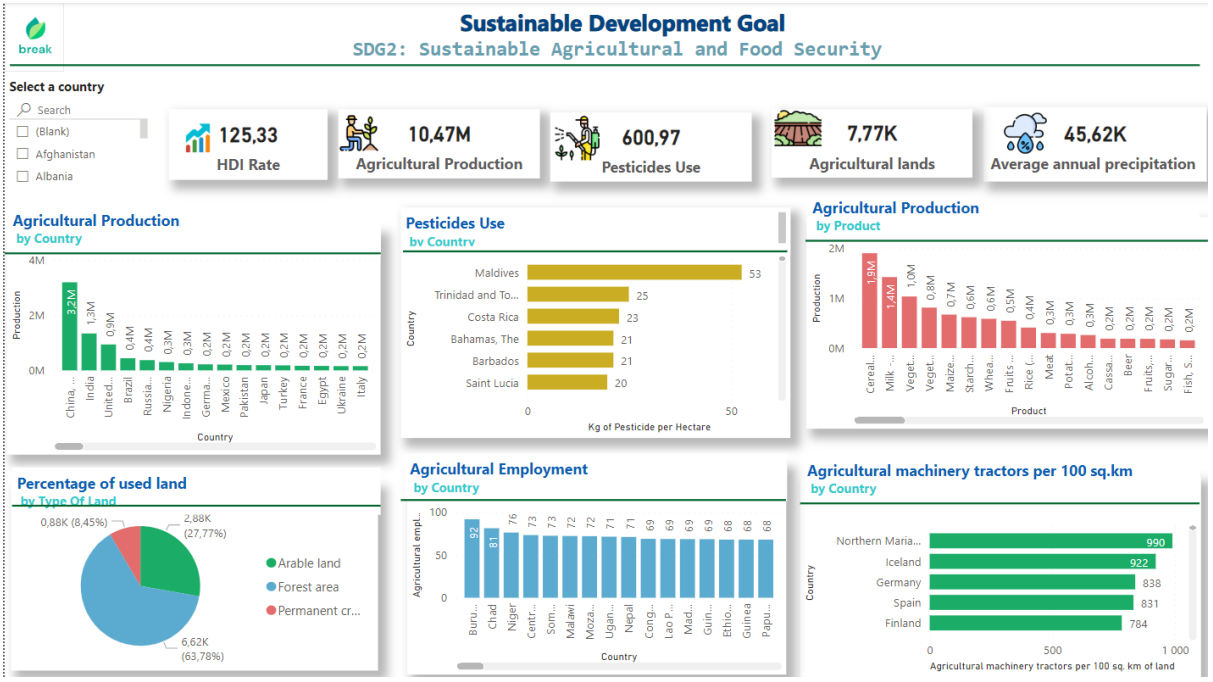
Impact

This chart will allow the company to visualize the distribution of health workers by profession to better understand the needs of each profession and adapt solutions accordingly.

Action to take:

By identifying shortages in certain professions in this field, the company can propose solutions to attract or recruit other profiles.

3. Overview of SDG2 Sustainable Agricultural and Food Security



3.1 Use of pesticides by country

Interpretation:

This bar chart represents different countries with their use of pesticides.

We have Maldives which uses the most pesticides with 53 kilograms of pesticides per hectare
and Zimbabwe which uses the least pesticides with 1 kilogram of pesticides per hectare.

Impact

This chart will allow the company to visualize the quantity of pesticides used by country and enable comparisons between countries. This can be useful to identify which countries have the most sustainable practices in terms of pesticide use, as well as those that should improve their practices to reduce their impact.

Action to take:

The action to take here is to reduce the use of pesticides.

3.2 Type of land used in percent

Interpretation:

This pie chart represents the distribution in % of land use by type.
We have forest land with 63.78%, arable land with 27.77%, and cultivated land with 8.45%.

Impact

This chart will allow the company to visualize the percentage distribution between different types of land use: arable land, forests, and permanent cultivated land.

Action to take:

The action to take here is to find a balance between forest conservation and the use of arable land.

3.3 Agricultural machinery tractors per 100 km² by country

Interpretation:

This bar chart represents a classification of countries according to their use of agricultural machinery tractors per 100 km².

We have Iceland, Germany, and Spain which use the most agricultural machinery with 922 machines per 100 km² / 838 machines per 100 km² / 831 machines per 100 km² respectively, and the countries that use machines at a low rate, such as Ghana with 5 machines per 100 km².

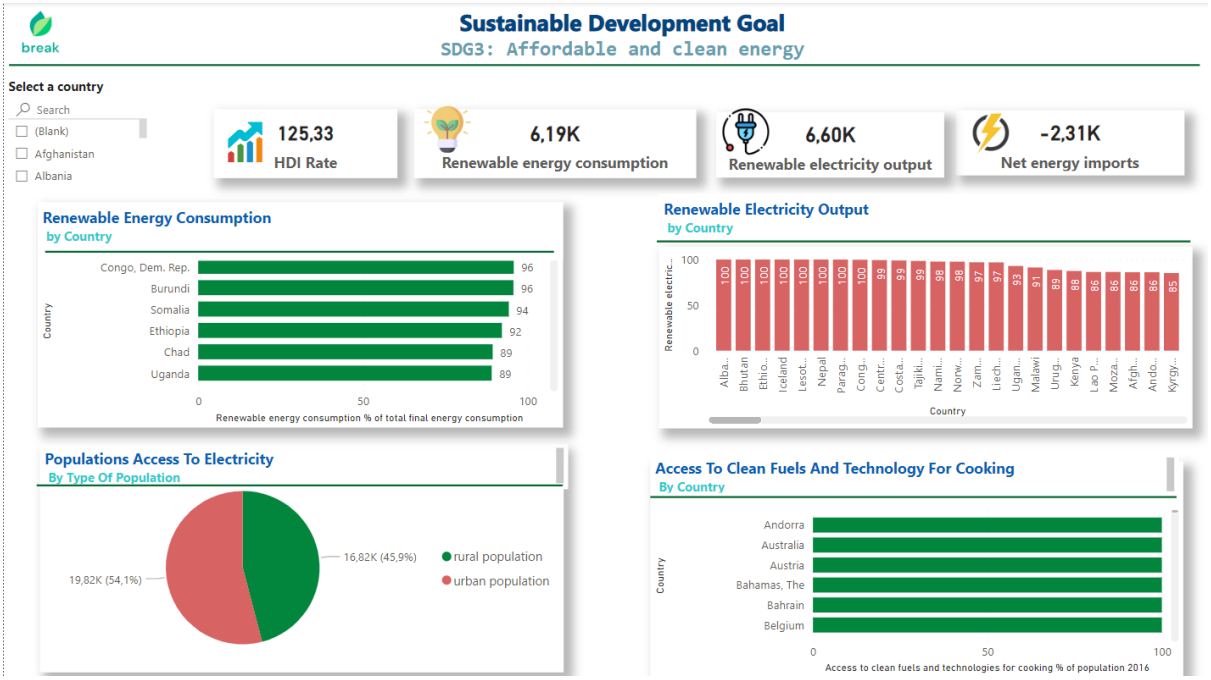
Impact

This chart will allow the company to identify countries that use the most agricultural machinery compared to those that use the least.

Action to take:

The action to take here is to adapt a strategy according to the needs.

4. Overview of SDG3 Affordable and clean energy



4.1 Renewable energy consumption by country

Interpretation:

This bar graph represents different countries with their renewable energy consumption rates.

We have Congo as the country with the highest renewable energy consumption rate of 95.80% and Iran as the country with the lowest consumption rate of 1%.

Impact

This diagram will allow the company to visualize the percentage of renewable energy consumption by country and enable comparisons between countries. This can be useful for identifying countries with sustainable practices in renewable energy consumption as well as those that need to improve their practices to reduce their impact.

Action to take:

The action to take here could be to look into establishing partnerships with these countries to purchase green energy or invest in renewable energy projects.

4.2 Renewable electricity production by country

Interpretation:

This bar graph represents the renewable electricity production by country.
We have Albania as the country with the highest renewable electricity production rate of 100% and Iran as the country with the lowest rate of 5.10%.

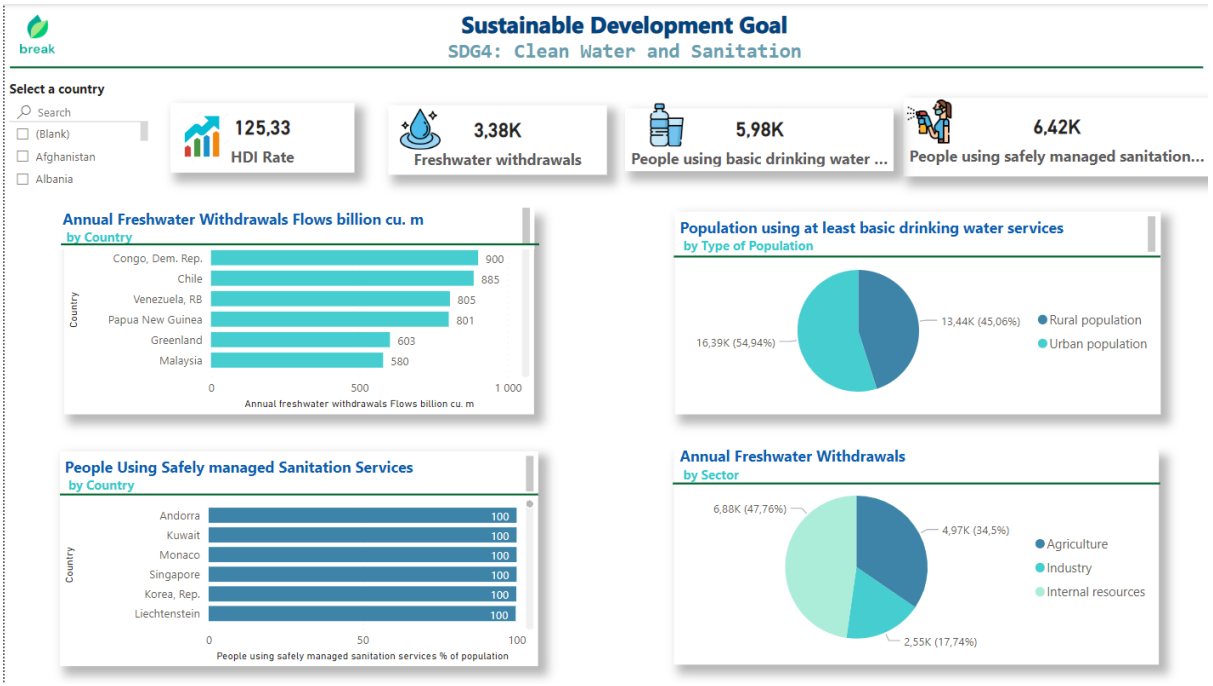
Impact

This diagram will allow the company to visualize the percentage of renewable electricity production by country and identify the most and least productive countries in renewable electricity production.

Action to take:

The action to take here could be for the company to invest in renewable energy technologies to improve the efficiency and profitability of renewable electricity production.

5. Overview of SDG4 Clean Water and Sanitation



5.1 Freshwater quantity by country

Interpretation:

This diagram presents a classification of countries according to their freshwater quantity.
Congo has the highest rate of freshwater with 900 billion cubic meters, followed by Chile.

Impact

This diagram allows the company to know the quantity of freshwater in different countries and compare them.

Action to take:

It is necessary to protect wetlands, rivers, and lakes as they play a necessary role in maintaining the availability of freshwater.

5.2 Annual freshwater withdrawals by sector

Interpretation:

This graph presents the three different sectors for freshwater withdrawals. The highest amount of freshwater withdrawn is for the "internal resources" sector with a percentage of 47%, followed by the agriculture sector with a percentage of 34.5%, and finally, the industry sector with 17.4%.

Impact

This graph shows the company how much water each sector consumes.

Action to take:

Recycling water in industrial processes, reducing losses and leaks in the industry sector.

5.3 People using safely managed sanitation services by country

Interpretation

This diagram presents the population of people who use safely managed sanitation services, and it is noted that Monaco, Japan, and Belgium have the highest rates, while Niger and Tuvalu have the lowest rates.

Impact

This diagram allows the company to know which countries have safely managed sanitation services.

Action to take:

Informing the population about the importance of sanitation and hygiene practices.

5.4 Availability of basic drinking water services

Interpretation

This graph is divided into two: the percentage of availability of drinking water in urban areas is 54%, while the percentage of availability of drinking water in rural areas is 46%.

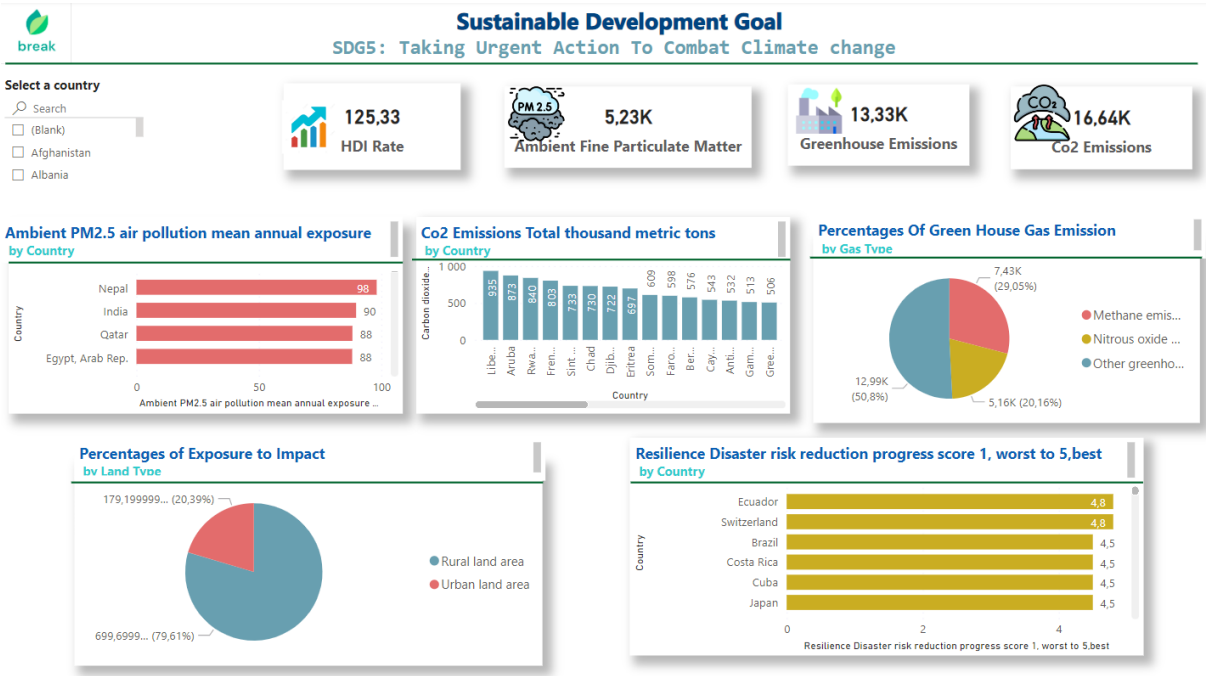
Impact

This graph shows the company the percentage of available drinking water in each area.

Action to take:

Invest in the construction and improvement of drinking water infrastructure in rural areas, including the establishment of improved water sources such as wells equipped with manual pumps.

6. Overview of SDG5 Taking Urgent Action To Combat Climate change



6.1 Ambient PM2.5 air pollution mean annual exposure by Country

Interpretation

The diagram displays the mean annual exposure to Ambient PM2.5 air pollution by country. Based on the data, Nepal has the highest level of pollution with a PM2.5 of 98, while Finland has the lowest with a PM2.5 of 6.

Impact

The graph is useful for the company as it provides information on the level of Ambient particle matters PM2.5 in various countries.

Action to take:

Reduce its carbon footprint and contribute to combating climate change

6.2 Percentages Of GreenHouse Gas Emission by Gas Type

Interpretation

This graphic represents the distribution of GreenHouse Gas between Methane Nitrous and other types of gasses . We can notice that theMethane percentage represents 29,05% while Nitrous percentage represents 20,16% and the other types of gasses are equal to 50,8% of the general GreenHouse Gas.

Impact

This graph allows the company to have an overall view on the percentage of GreenHouse Gas as well on its distribution between the different types of gas : Methane Nitrous and other types of gasses

Action to take

The company could explore and invest in technologies and practices that help reduce emissions of these gasses. Another action could be to increase awareness and education on the impact of GreenHouse Gas emissions and encourage employees and stakeholders to adopt more sustainable practices.

Conclusion:

In conclusion, data visualization is a powerful tool that enables businesses to gain insights into their data through the use of visual elements such as charts, graphs, and maps. The use

of interactive dashboards, such as those developed in Microsoft Power BI, can facilitate analysis and decision-making processes. The dashboards discussed in this chapter covered a range of topics, from human development rates to sustainable agricultural practices, providing a comprehensive view of the data. By interpreting the visualizations and taking appropriate actions, businesses can make informed decisions to improve their operations, reduce poverty rates, and promote good health and well-being.

Chapter 6:Realization of the application

Introduction:

This chapter focuses on the realization of a web application using Streamlit, a Python library for building data science and machine learning web applications. The chapter begins with an overview of the development environment, including HTML, which is the primary markup language for web development, and Streamlit, which simplifies the process of creating web applications for data science and machine learning projects.

The chapter then explains the process of setting up Streamlit by downloading Anaconda and installing the library. It also describes how to import a trained machine learning model into the workspace and implement some basic Python functions to make the model operational. A simple HTML form is then created to collect input from the user to be fed into the model.

I. Development environment

1. HTML

HyperText Markup Language (HTML) is the primary markup language that web developers use to construct and organize web pages. HTML offers a comprehensive set of tags that determine the layout and substance of a webpage.

2. Streamlit

Streamlit is a Python library that simplifies the process of creating web applications for machine learning and data science projects. With Streamlit, developers can quickly and easily create interactive dashboards, data visualizations, and other web-based tools that allow users to interact with their data in real-time. Streamlit provides an intuitive and flexible interface for building custom web applications, without requiring extensive knowledge of web development technologies such as HTML, CSS, and JavaScript. It is open-source and has gained popularity in the data science community for its ease of use and speed of development.

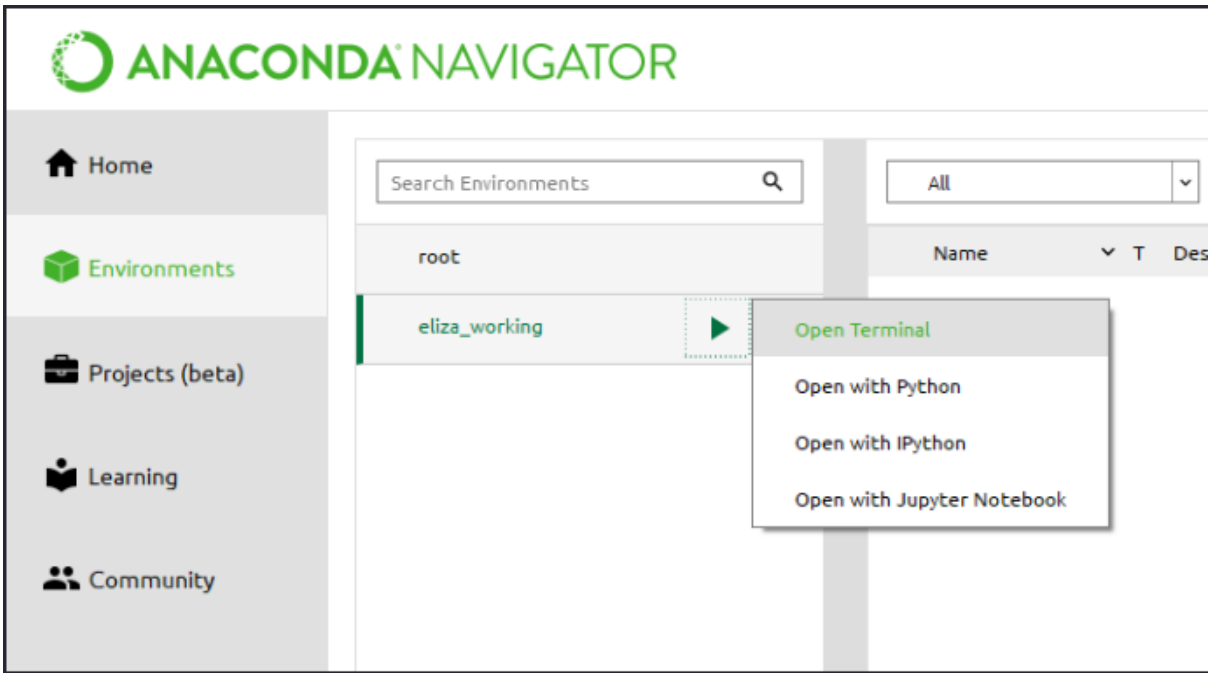
Streamlit

II. Web application

Streamlit is a user-friendly software that can be easily set up by following a few simple steps. To get started with Streamlit, the first thing you need to do is download Anaconda, which is an open-source distribution of the Python and R programming languages designed for data science.

Once you have Anaconda installed, you can either use the root environment or create a new root environment to download Streamlit. Simply enter a single command in the Anaconda prompt, and Streamlit will be installed and ready to use in no time.

-**pip install streamlit**



Once you have completed the installation process, you can verify that Streamlit is functioning properly by testing the default app called "hello". This can be done simply by typing "Streamlit hello" into the terminal.

Once you have confirmed that the server is working correctly, you can begin writing your code. In this case, we will be using Streamlit to deploy a machine learning model. However, before deploying the model, it is important to first test and export it.

```
[ ]  import pickle

[ ]  filename='trained_mod.sav'
     pickle.dump(gbr, open(filename, 'wb'))

[ ]  loaded_model = pickle.load(open('trained_mod.sav','rb'))
```

```
[ ]  input_data = (0.429883,10.0,80843.0,0.001469)

[ ]  input_data_array = np.asarray(input_data)

[ ]  input_data_reshape = input_data_array.reshape(1,-1)

[ ]  prediction = loaded_model.predict(input_data_reshape)
     print(prediction)

     [0.78199759]
```

In this particular scenario, linear regression was deemed the most effective model. The next step involved downloading the 'trained_mod.sav' file, which contains the trained model, and importing it into the workspace. For this purpose, the IDE of choice was VS Code.

```python
import streamlit as st
import pandas as pd
import numpy as np
import pickle as pk

st.set_page_config(page_title="HDI Calculator", page_icon=":guardsman:")

col1, col2, col3, col4 = st.columns(4)

with col1:
    st.write(" ")
with col2:
    st.write(" ")
with col3:
    st.image("assets/logobreak.png")
with col4:
    st.write(" ")


# loading the saved model
loaded_model = pk.load(
    open("C:/Users/MOUEFEK/OneDrive/Bureau/streamlittest/trained_mod.sav", "rb")
)


def MLtest(input_data):
    # changing the input data to numpy array
    input_data_array = np.asarray(input_data)
    # reshape the array as we are predicting for one instance
    input_data_reshape = input_data_array.reshape(1, -1)
    prediction = loaded_model.predict(input_data_reshape)
    return prediction
```

The next step was to add some code to the 'main.py' app. Firstly, the model was loaded using the variable name 'loaded_model'. Then, a few simple Python functions were implemented to make the model operational.

Finally, a basic form was created to collect input from the user to be fed into the model.

```python
st.subheader("Predicting Human Development Index (HDI)")


# methode 1 with form is with
with st.form(key="form3"):
    Mortality_rate = st.text_input("Mortality rate")
    Renewable_energy_consumption = st.text_input("Renewable energy consumption")
    Agricultural_production = st.text_input("Agricultural production")
    poverty = st.text_input("Poverty")
    submit_button3 = st.form_submit_button(label="Try Me!!")

# result can be either inside ou outside
if submit_button3:
    dia = MLtest(
        [Mortality_rate, Renewable_energy_consumption, Agricultural_production, poverty]
    )

    st.write("your current HDI is :")
    st.success(dia)
```

To run your project, simply enter the command "Streamlit run main.py" in the terminal, and your project will be up and running.

Predicting Human Development Index (HDI)

Mortality rate

Renewable energy consumption

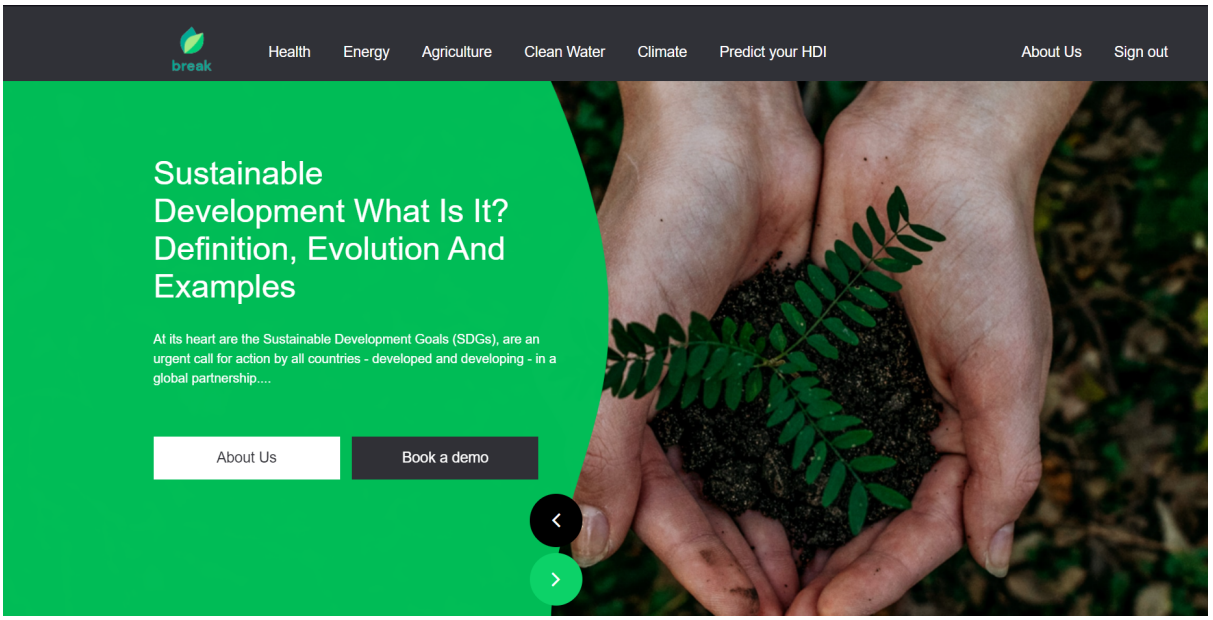
Agricultural production

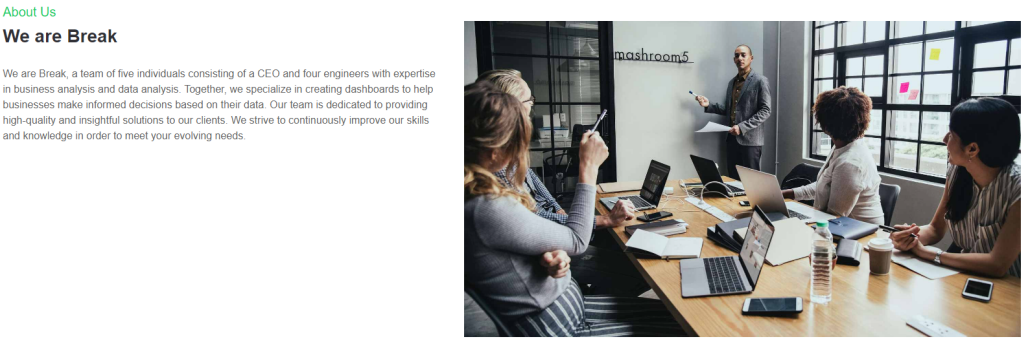
Poverty

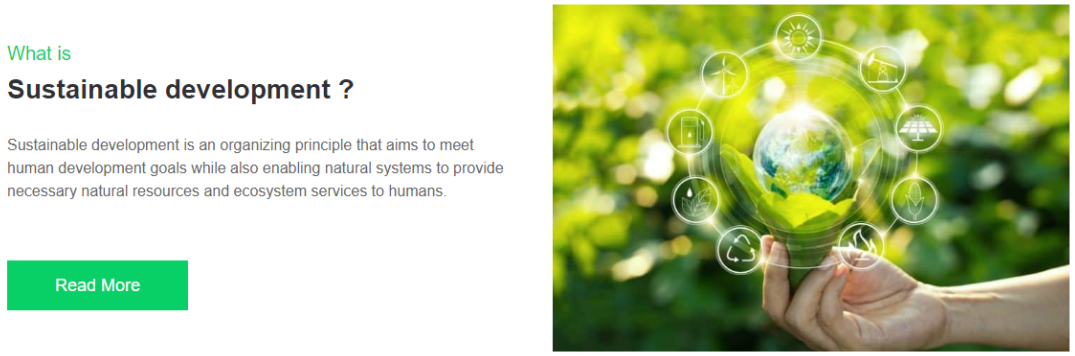
Show Result

For our web application, we decided to use standard HTML, CSS, and JavaScript, as we felt that it would be unnecessary to use a framework with only one API consumer, such as Power BI.

In the end, this resulted in a final product that was visually appealing and fully functional.







Conclusion:

In conclusion, the development of a web application using Streamlit, a Python library for building data science and machine learning web applications, has been outlined in this chapter. With Streamlit, developers can quickly and easily create interactive dashboards, data visualizations, and other web-based tools that allow users to interact with their data in real-time.

By following a few simple steps, such as downloading Anaconda, installing Streamlit, and importing a trained machine learning model, developers can create custom web applications without requiring extensive knowledge of web development technologies like HTML, CSS, and JavaScript.

In this scenario, a linear regression model was used to develop a fully functional web application that allowed users to input data and receive predictions from the model. The decision to use standard HTML, CSS, and JavaScript instead of a framework with only one API consumer was made to optimize the final product's visual appeal and functionality.

Overall, Streamlit has proven to be an intuitive and flexible interface for building custom web applications for data science and machine learning projects. With its ease of use and speed of development, Streamlit has gained popularity in the data science community and is expected to become an increasingly valuable tool in the development of web applications in the future.

Conclusion

At the conclusion of our project, we were able to leverage the data we had collected to create a comprehensive data warehouse, which served as a central repository for all the data related to our project. We used a combination of tools and techniques to integrate the

data from various sources, identify and impute missing information, and implement the data warehouse.

One of the key benefits of having a data warehouse is the ability to create graphical representations of the data that can be easily understood by a wide range of stakeholders, including those who may not be technically savvy. We used a variety of data visualization tools to create compelling and informative visualizations that helped us analyze and make decisions based on the large amount of data we had collected.

To further analyze the data and extract insights, we used advanced data mining techniques such as modeling and clustering algorithms. These techniques helped us identify patterns and correlations within the data that would have been difficult to identify using traditional methods.

All of the work we did on the data warehouse and data mining was implemented on a web platform with a user-friendly interface and visualizations that made it easy for even non-technical customers to understand the data. This platform also allowed us to easily share the data with other stakeholders who needed access to it.

Throughout the project, we learned how to use a variety of software solutions, including Talend and MS Power BI, to integrate, analyze, and visualize the data. We also gained experience in the methodologies used in projects like these, such as agile project management and data warehousing best practices.

One of the most valuable lessons we learned during the project was the importance of collaboration and teamwork. We had a diverse team with a range of skills and backgrounds, and it was through working together that we were able to successfully complete the project on time and on budget. We also learned how to manage our time and resources effectively, and how to cope with the stress of a fast-paced work environment.

Overall, the project was a great learning experience for all of us, and we are proud of the work we were able to accomplish. We believe that the data warehouse and web platform we created will have a lasting impact on the organization, helping them make better informed decisions based on the data they have collected.

Bibliography

[1] SDG:

THE 17 GOALS | Sustainable Development (un.org)
[2] CRISP -DM:
https://www.sv-europe.com/crisp-dm-methodology/
[3] Talend:
https://www.tutorialspoint.com/talend/talendtalendopenstudio.html
[4] Power BI :
https://www.tutorialspoint.com/powerbi/index.html
[5] web scraping :
https://meteostat.net/en/station/60715
[6] Human Development Indicator :
https://countryeconomy.com/hdi?year=2016
[7] Mortality Rate :
https://www.kaggle.com/datasets/navinmundhra/world-mortality?resource=download.
[8] Pesticide Use :
https://www.worldometers.info/food-agriculture/pesticides-by-country/
[9] Poverty Rate :
https://hdr.undp.org/content/2022-global-multidimensional-poverty-index-mpi?fbclid=IwAR3KA5zga6rTwYm-Dd6NH8LDdcymq0d2Zf5ChPdvSRxAmDVldFSKaMJMhZw#/indicies/MPI