Universidad Politecnica de Madrid
Technical School of Computer Systems Engineering

Analysis of the Most Influential Factors on Life Expectancy using Machine Learning

Final Degree Project
Degree in Software Engineering

Academic year 2020-2021

Author:
Miguel Roca García

Supervisor:
Raúl Lara Cabrera

Translated from Spanish to English by ChatGPT

I would like to express my sincere thanks to all the people who have made it possible for me to reach this point.

First of all, to my supervisor, Raúl, for introducing me to the world of Machine Learning and his guidance and advice throughout this work.

Secondly, to all my classmates and friends made during my degree, who made university a wonderful experience.

Finally, I want to deeply thank my family for their support and dedication to my education. Especially my parents for supporting me and allowing me to study in Madrid, and my aunt Lelel for encouraging me to always aim for the highest grades.

Thank you from the bottom of my heart.

Abstract

This Final Degree Project uses machine learning models to analyze the influence of various factors on life expectancy, such as economic, development, food diet, health, and political indicators.

The factors studied have been obtained through an ETL process from different international organizations of prestige and organized by country, year, and gender. A preprocessing of the dataset characterized by the imputation of missing values was required.

Several machine learning models have been implemented to predict the life expectancy of each case with the provided indicators. The best results were obtained by an artificial neural network, the multilayer perceptron.

The study of the behavior of the neural network has been approached with three different strategies in order to determine the effect of the indicators on the calculation of life expectancy.

The first approach was the feature selection technique wrapper methods, with which the minimal set of indicators required to get an acceptable error has been determined.

Secondly, a fitness function for the genetic algorithm has been implemented in order to maximize the life expectancy of a specific case with a certain margin change. This algorithm provides the most influential factors for each situation.

Finally, the model has been carried out with SHAP, obtaining the factors that most affect at a general and individual level and in comparisons between cases. The contribution of each factor to the final result in an individual case has been provided as well.

It is important to mention the difference between correlation and causality for the results obtained. This study identifies the effect of the indicators on the calculation of life expectancy, not being able to demonstrate if their values are the cause in real life, although it can serve as a basis for consideration by health or political authorities.

Contents

Table of Contents

Índice de figuras

Capítulo 1

Introduction

Health is a topic that raises special interest and concern in society today. This trend is driven by the pandemic that is sweeping the world and has forced extreme measures to avoid a very high number of deaths.

The health status of a country or population is a characteristic that is very difficult to measure, as there is no clear factor that can give us an idea of this concept. However, there are some metrics that help to understand or get an idea of the health situation of a population at a specific point in time. One of them is life expectancy.

Life expectancy at birth is the average number of years that a newborn will live if the mortality patterns of the population to which it belongs remain constant in the future from its time of birth[12]. This means that life expectancy is calculated for a specific period of time, a specific population, and generally a specific sex, and it will only refer to life expectancy for a given age, in our case, at birth.

It is an extremely useful measure for knowing the health level of a country or population at a specific time, usually a year. Its calculation is done through what are known as mortality tables or *life tables*, which show the probability of death within age ranges[13].

Life expectancy, therefore, is a metric that comes from mortality, a measure that depends on a wide variety of factors, which can influence more or less depending on the case or the importance of each factor. Among the factors to consider, we can include development, economic, food, political, geographical, health, social, and other indicators.

The variety and abundance of these values to consider is so high that a traditional analytical approach to the data to see the influence of each factor on the final result for each case becomes unfeasible. This problem is solved thanks to advances in artificial intelligence and machine learning.

With machine learning models, it is possible to obtain an output from a large number of inputs, thus allowing the model to take into account all the factors that should be considered to calculate life expectancy, and then analyze the model to understand how much and how each of the studied factors or indicators affects the final calculation.

1.1. Objectives

The objective of this work is divided into three blocks that follow a sequential order:

1. Constitute a dataset consisting of life expectancy and factors that may influence it, organized by country, year, and gender, extracted from databases of internationally relevant organizations.

2. Create a machine learning model capable of predicting or calculating life expectancy from the constructed dataset.

3. Study and interpret the functioning of the constructed model using different strategies to understand how and to what extent the established factors influence the final life expectancy calculation.

1.2. Methodology and Structure

The methodology and tools used to achieve the objectives set out in the previous section will establish the structure in which the work is organized.

The programming language used throughout the work has been *Python 3.8* [14], and the chosen framework for building and running the code is *Jupyter Notebook*[15].

The project development begins with the extraction and obtaining of the dataset, for which CSV files and the *pandas*[16] library of *Python* were used to create the unified dataset with all the data to be considered. Using this same programming language, an analysis and preprocessing of the dataset were carried out.

Next, using the *machine learning* libraries *scikit-learn*[17] and *TensorFlow*[18], models were built and trained, from the simplest to the most complex.

Subsequently, several techniques for analyzing the more complex model, the neural network, were established. The first technique described was the feature selection strategy *wrapper methods*. The second model analysis approach proposed was using a genetic algorithm, for which Salga[19], a program developed by the Polytechnic University of Madrid for using the genetic algorithm with the help of a visual interface, was used. The third and last technique applied was using the *SHAP*[20] library of *Python*, based on *Shapley values* for model explanation.

For the use of the neural network and the calculation of life expectancy as well as a possible comparison between cases, a predictor was built in a *Python* notebook.

After the development is completed, the results section presents everything obtained from the previous steps, starting with a study of the correlation of the extracted data with the target variable, life expectancy.

We will report the error obtained by each model in the following subsection and carry out a study of the neural network's error to detect its behavior.

Finally, applying the model interpretation strategies described in the development, the results obtained according to each approach will be described.

Capítulo 2

State of the Art

2.1. Other Studies on Life Expectancy

Life expectancy is a widely studied topic in our society, and the large number of studies that, in one way or another, address this broad field of research is evidence of this.

A comprehensive study published in *Our World In Data* addresses the evolution and changes in life expectancy over the years, broken down by country. It offers extensive visualizations to see this progress and comparisons with key factors such as public health spending and Gross Domestic Product[12].

The article by Casper Worm Hansen goes beyond visualizations and examines the relationship between life expectancy and a factor like average years of education, concluding that for each year increase in life expectancy, years of education rise by 3.5 %[21].

A notable study is the one published by the Iranian Journal of Public Health[22]. This study examines socio-health factors on life expectancy in countries with middle and low-income levels. The factors studied in this article include female fertility, per capita income, years of education, HIV ratio, and physician density per capita. It concludes that to increase life expectancy in low-income countries, HIV prevalence, adolescent birth rates, and illiteracy should be eliminated.

However, studies in the field of life expectancy often examine a limited set of factors due to the constraints of traditional analysis. Through machine learning, we can analyze the influence of an indeterminate number of factors.

2.2. Machine Learning

Machine learning is a discipline within artificial intelligence that is flourishing thanks to new computing technologies. Machine learning is based on creating a model that learns from a starting dataset, reducing error in relation to the goal over time. The learning process is called training. The objective of this process is to find patterns within the dataset that allow decision-making to predict future input data. Since machine learning is data-driven, its quality is crucial for obtaining reliable and accurate results.

Depending on the problem at hand, there are various approaches within machine learning.

Supervised learning is when a dataset with labels is trained, meaning the expected result of each case in the dataset provided to the model is known, allowing us to measure the error of the model's prediction. The advantage of this approach is that it requires fewer training data and provides metrics of precision on the results. However, labeled data is rare and expensive to obtain. This approach is used for classification problems (when the label is a category) and regression problems (when the label is a numerical value).

Unsupervised learning starts with unlabeled data. The objective of this approach is to identify patterns and relationships within the data.

2.2.1. Interpretation of Machine Learning Models

One of the biggest challenges in machine learning is the ability to explain the developed model. There are numerous cases where a machine learning algorithm is used to make a business decision. This decision needs a justification, which models are not always capable of providing. A practical case would be a bank loan application rejection. If a machine learning model is used to reject the application, a logical explanation must be provided to the client.

Depending on the applied model, the capacity for explanation will vary. However, the more complex the model, the less capacity it will have. When a model cannot explain its decision-making process, it is said to be a **black-box** model.

The analysis of black-box models is underdeveloped, and the two most notable techniques in this field are those proposed by LIME[23] and SHAP[24]. Both techniques were developed in the last decade and are capable of explaining the model by showing the contribution of each input to the final result.

However, LIME has some disadvantages compared to its alternative[25]. This model interpretation technique can only explain one case at a time, while SHAP provides this functionality as well as various graphs to interpret the global effect of the data on the entire set. Furthermore, LIME lacks robustness, meaning a small change in the input values of the studied point can lead to drastic changes in the explanation, giving very different results for similar situations. Finally, it is dependent on a set of hyperparameters required for its use. For these reasons, SHAP has been chosen over LIME as the model interpretation technique.

Another very interesting approach for interpreting black-box models is presented in the article by Federico Piccinini[26], where he outlines a strategy for interpretation using a genetic algorithm.

Capítulo 3

Project Development

The workflow followed in this project began with the extraction, transformation, analysis, and preprocessing of data, followed by the application of machine learning models, and finished with the analysis and interpretation of the results.

3.1. Dataset

The dataset we will start with consists of a set of indicators organized by year and country. These indicators provide information about healthcare, poverty, food, economy, technology, access to resources, etc., which characterize a country in a specific year. Additionally, we will have life expectancy associated with each case. The dataset spans from 1990 to 2019.

3.1.1. Data Source

The data has been extracted from various sources, all of which are prestigious global organizations, such as: *World Bank*[27], WHO[28], UNICEF[29], *Our World In Data*[30], and FAOSTAT[31].

The **World Bank** is a multinational financial organization made up of 189 countries. Its goal is to reduce poverty through low-interest loans and economic support to nations with fewer resources. The *World Bank* offers a freely accessible database with hundreds of indicators on global development. This information can be extracted using its data visualization and analysis tool known as *DataBank*, which allows users to generate and store charts, tables, and maps based on this data.

The **World Health Organization** (WHO) is an agency of the United Nations whose function is to lead and organize alliances in complex health situations, as well as to determine research lines to acquire and disseminate new knowledge in the health field. It offers a publicly accessible data repository organized by categories, indicators, and countries.

UNICEF (United Nations Children's Fund) is also an organization under the United Nations. It provides humanitarian aid to eliminate poverty, discrimination, violence, and disease. It operates in developing countries. UNICEF offers a website to access different indicators organized by theme and country.

Our World in Data is an online organization that presents and publishes data and statistics from research and empirical analysis on the changes occurring in the world, aiming to find the reasons behind these behaviors.

FAOSTAT is the acronym for the United Nations Food and Agriculture Organization. It is responsible for collecting, analyzing, and publishing statistics related to food and agriculture for decision-making purposes. It offers free access to its data and statistics.

Finally, it is necessary to mention **Kaggle**[32], a free platform that, among other things, offers datasets created by other users of the platform. Part of the

dataset has been extracted from this platform, which, in turn, was extracted from the previously described sources.

3.1.2. Dataset Definition

Features are the inputs that a machine learning model will use to obtain the output. Some of the indicators we started with have been obtained through statistical calculations and estimates, so they are associated with a 95 % confidence interval. This interval will be reflected in the data, including its lower and upper limits.

The features that initially make up the dataset are specified below:

- **Country**: Name of the country.
- **Year**: Year.
- **Gender**: Gender. It can be *Female*, *Male*, or *Both sexes*.
- **Life Expectancy**: Life expectancy. This is the target variable, i.e., the variable that we will try to predict using the machine learning model.
- **Infant Mortality Rate**: Infant mortality rate. Number of children under one year old who die per 1,000 live births in a year.

  Low CI Value Infant Mortality Rate: Lower limit of the confidence interval.

  High CI Value Infant Mortality Rate: Upper limit of the confidence interval.
- **Under 5 Mortality Rate**: Under-5 mortality rate. Probability of a newborn dying before reaching 5 years of age, expressed per 1,000 live births.

  Low CI Value Under 5 Mortality Rate: Lower limit of the confidence interval.

  High CI Value Under 5 Mortality Rate: Upper limit of the confidence interval.
- **% Death Cardiovascular**: Probability of dying from cardiovascular diseases, cancer, diabetes, or chronic respiratory diseases between the ages of 30 and 70.

  Low CI Value % Death Cardiovascular: Lower limit of the confidence interval.

  High CI Value % Death Cardiovascular: Upper limit of the confidence interval.
- **Suicides Rate**: Suicide rate. Number of deliberate deaths carried out by the individual with full knowledge or expectation of the outcome, per 100,000 inhabitants.
- **Diet Composition Alcoholic Beverages**: Alcohol per capita. Total amount of alcohol consumed by adults (over 15 years) in a year, in liters of pure alcohol per person.

- **Air Pollution Death Rate**: Death rate from air pollution. Probability of dying from domestic and environmental air pollution. This is divided into:

  Stroke: Stroke.

  Ischaemic Heart Disease: Heart disease.

  Lower Respiratory Infections: Lower respiratory infections.

  Chronic Obstructive Pulmonary Disease: Chronic obstructive pulmonary disease.

  Trachea Bronchus Lung Cancers: Tracheal or lung cancer.

  Total: Total sum.

  It is also divided into:

  Aged Standarized: Age-standardized.

  Aged not Standarized: Not age-standardized.

  All of these features also have lower and upper limits for the confidence intervals.

- **Unsafe Wash Mortality Rate**: Mortality rate due to deaths attributed to exposure to unsafe hygiene, water, and sanitation services per 100,000 inhabitants.
- **Poisoning Mortality Rate**: Mortality rate due to deaths attributed to unintentional poisoning per 100,000 people.
- **Tobacco Prevalence**: Percentage of the population over 15 years old who consume tobacco.
- **% Population Aged 0-14**: Percentage of the population aged 0 to 14.
- **% Population Aged 15-64**: Percentage of the population aged 15 to 64.
- **% Population Aged 65+**: Percentage of the population aged 65 and older.
- **% Population Aged 65-69**: Percentage of the population aged 65 to 69.
- **% Population Aged 70-74**: Percentage of the population aged 70 to 74.
- **% Population Aged 75-79**: Percentage of the population aged 75 to 79.
- **% Population Aged 80+**: Percentage of the population aged 80 and older.
- **Maternal Mortality Ratio**: Maternal mortality ratio. Number of maternal deaths due to childbirth per 100,000 live births.

  Low CI Value Maternal Mortality Ratio: Lower limit of the confidence interval.

High CI Value Maternal Mortality Ratio: Upper limit of the confidence interval.

- **% of Births Attended By Skilled Personal**: Percentage of births attended by skilled healthcare professionals.
- **Neonatal Mortality Rate**: Neonatal mortality rate. Number of deaths during the first 28 days of life per 1,000 live births in a year.

  Low CI Value Neonatal Mortality Rate: Lower limit of the confidence interval.

  High CI Value Neonatal Mortality Rate: Upper limit of the confidence interval.
- **Incidence of Malaria**: Incidence of malaria. Number of malaria cases per 1,000 inhabitants at risk per year.
- **Incidence of Tuberculosis**: Incidence of tuberculosis. Estimated number of tuberculosis cases per 100,000 inhabitants over the course of a year.

  Low CI Value Incidence of Tuberculosis: Lower limit of the confidence interval.

  High CI Value Incidence of Tuberculosis: Upper limit of the confidence interval.
- **Hepatitis B Surface Antigen**: Prevalence of hepatitis B surface antigen.

  Low CI Value Hepatitis B Surface Antigen: Lower limit of the confidence interval.

  High CI Value Hepatitis B Surface Antigen: Upper limit of the confidence interval.
- **Intervention Against NTDs**: Number of people who have required treatment and care for any neglected tropical disease (NTD).
- **Road Traffic Deaths**: Number of deaths due to traffic accidents per 100,000 people.
- **Reproductive Age Women**: Percentage of married or partnered women of reproductive age using contraceptives.
- **Adolescent Birth Rate**: Adolescent birth rate. Number of births by women aged 15 to 19 per 1,000 women in that age range.
- **Universal Health Care Coverage**: Index of basic health services covered on a scale from 0 to 100.
- **Population 10 Percentage SDG Total**: Proportion of the population whose healthcare expenses exceed 10 % of their income.

  Population 10 Percentage SDG Urban: Proportion in urban areas.

  Population 10 Percentage SDG Rural: Proportion in rural areas.

- **Population 25 Percentage SDG Total**: Proportion of the population whose healthcare expenses exceed 25 % of their income.

  Population 25 Percentage SDG Urban: Proportion in urban areas.

  Population 25 Percentage SDG Rural: Proportion in rural areas.

- **Doctors**: Number of doctors per 10,000 inhabitants.
- **Nurses and Midwives**: Number of nurses and midwives per 10,000 inhabitants.
- **Dentists**: Number of dentists per 10,000 inhabitants.
- **Pharmacists**: Number of pharmacists per 10,000 inhabitants.
- **Basic Drinking Water Services**: Percentage of the population with access to basic drinking water services.
- **Basic Sanitation Services Total**: Percentage of the population with access to basic sanitation and hygiene services.

  Basic Sanitation Services Urban: Percentage in urban areas.

  Basic Sanitation Services Rural: Percentage in rural areas.

- **Safely Sanitation Total**: Percentage of the population with access to safe sanitation and hygiene services.

  Safely Sanitation Urban: Percentage in urban areas.

  Safely Sanitation Rural: Percentage in rural areas.

- **Basic Hand Washing Total**: Percentage of the population with access to services for hand washing with soap and water at home.

  Basic Hand Washing Urban: Percentage in urban areas.

  Basic Hand Washing Rural: Percentage in rural areas.

- **Clean Fuel and Technology**: Percentage of the population using clean fuels and technologies as the main source of energy for cooking at home.
- **Birth Rate**: Birth rate. Number of births per 1,000 inhabitants.
- **Battle Related Deaths**: Number of deaths related to battles in an armed conflict.
- **% Injury Deaths**: Percentage of deaths caused by injuries.
- **Death Rate**: Death rate. Number of deaths per 1,000 inhabitants.
- **GDP per Capita**: Gross Domestic Product per capita in dollars.
- **% Population $1.90 a day**: Percentage of the population living on less than $1.90 a day.
- **% Population $3.20 a day**: Percentage of the population living on less than $3.20 a day.

- **% Population $5.50 a day**: Percentage of the population living on less than $5.50 a day.
- **Income per Capita**: Income per capita in dollars.
- **Total Population**: Total number of inhabitants.
- **GNI per Capita**: Gross National Income per capita in dollars, adjusted with the Atlas method.
- **Conflict and Terrorism Deaths**: Number of deaths due to armed conflicts and terrorism.

3.1.3. ETL Processes for Dataset Acquisition

ETL processes (*Extract - Transform - Load*) are procedures for collecting data from an indefinite number of sources, organizing and cleaning it, and unifying it into a single, consolidated repository. It consists of three phases as its name suggests: Extraction, Transformation, and Loading.

The first phase, **extraction**, involves obtaining data from source systems. An analysis of the structure of the extracted data is then carried out to check that it meets the established requirements. Finally, the data is converted into a format that allows its transformation.

The **Transformation** phase involves making certain changes to the content or structure of the data to ensure they follow the established business rules and guidelines. These transformations typically include:

- **Cleaning**: Removing erroneous and duplicate data, translating codes, or splitting discrete data.
- **Filtering**: Removing unnecessary data according to the established requirements.
- **Classification**: Dividing data by type, such as raw data, audio, video, structured, unstructured, etc.
- **Restructuring**: Necessary transformations to ensure the data follows a unified or specified structure. Operations such as sorting rows and columns, renaming, splitting columns and rows, etc.
- **Unification**: Merging data into a single flow or set.

Finally, the **Loading** phase moves the transformed data to its destination. This loading can be done either fully or incrementally, depending on the destination's characteristics and the amount of data.

The scope of ETL processes is broad, although their most common uses include data migration, data replication for backup on other platforms, unification of multiple sources, and storage in a *Data Warehouse* to ingest, transform, and classify data for studies using statistical analysis, artificial intelligence, and *machine learning* in order to derive business intelligence.

The ETL process carried out to obtain the dataset applied to the *machine learning* models followed these steps:

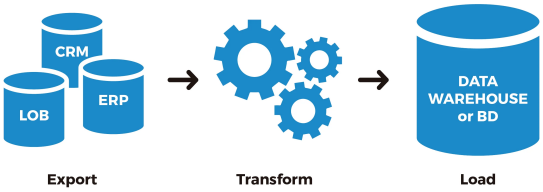

Figura 3.1: Phases of an ETL Process[1]

Extraction Phase

The ETL process began by extracting data from the sources. These **were stored in CSV files**, one for each extracted indicator.

A CSV (*Comma Separated Values*) file is a text file in which each field is separated by a comma, semicolon, or a designated character, forming a table with rows and columns. Typically, the column names, called headers, are specified at the beginning.

These extracted files were **sorted and renamed** for easier use. The new name for each file followed this rule: *Index_Indicator Name*.csv. For files with similar indicators, a subindex was added.

Next, the obtained data were analyzed, discarding those files containing irrelevant information for the project's objective or missing too much data. Finally, the files were loaded into a Python script and **transformed into DataFrames** to apply the necessary transformations.

A DataFrame is a two-dimensional data structure where each column can hold a different type of data. It has a table shape and allows naming of columns and rows.

Transformation Phase

Several **transformations on the structure** of the DataFrames were carried out to ensure they followed a unified format as shown below:

Country	Year	Gender	Indicator

Table 3.1: Data Structure

Some of the transformations applied for this restructuring included:

- Renaming the headers of the CSVs.
- Unifying data from an indicator from various sources to increase the number of known values.
- Merging several CSVs with gender-separated data into a single file.
- Creating a *Gender* column and assigning values based on the data.
- Splitting a CSV into several files based on the indicator.

- Dividing one column into several, with different values depending on the confidence interval in the estimation of an indicator.

After restructuring, all **country names were unified** to ensure they had the same value. For example, for indicators where the United States was listed as *United States of America*, it was transformed to *United States*, which is the format used in the target variable.

Finally, once all the indicators were available in different DataFrames but with the same structure, a **merge** was performed based on the values of *Country*, *Year*, and *Gender*, resulting in a single DataFrame with these three column values plus each of the indicators.

The transformations applied in this phase relate only to the data structure. Data cleaning and filtering will be performed in the Analysis and Preprocessing phase.

Loading Phase

The unified dataset resulting from the transformation phase was loaded into a CSV and stored for later accessibility.

3.1.4. Exploratory Analysis and Cleaning

An exploratory analysis of the unified dataset was conducted to observe the distributions of the various features, search for and eliminate erroneous values, analyze outliers, observe the relationship with the target variable, and gain a general understanding of the dataset's behavior and validity.

Erroneous Data

Through an individual study of each feature, erroneous values were identified and removed from the dataset by replacing them with an empty value.

Particularly noteworthy is the cleaning of data in the *Country* column, where a large number of values in the dataset were not countries but rather continents, regions, islands, etc. Therefore, a list of countries recognized by the United Nations (UN) [33], consisting of 193 member states and 2 observers, was extracted. Records with *Country* values not found in this list were removed.

Outliers

Outliers are observations that are numerically distant from the rest of the data. These values can significantly affect statistical calculations like the mean and can alter the behavior of *machine learning* algorithms, so it is important to detect and study them.

The procedure followed to detect outliers for each feature in the dataset was based on the application of the **Interquartile Range Rule**[34].

The interquartile range (IQR) is defined as the difference between the first and third quartiles of a distribution. Quartiles are values that divide the observations into four equal parts. For example, the first quartile, Q1, corresponds to the value such that one-fourth of the observations are less than or equal to this value, and the rest are greater than or equal.

The interquartile range rule designates as outliers those observations that are either below the first quartile minus 1.5 times the IQR or above the third quartile plus 1.5 times the IQR. That is, an observation will be an outlier if it falls within the range:

$$(-\infty, Q1 - 1{,}5 \cdot IQR) \cup (Q3 + 1{,}5 \cdot IQR, \infty) \tag{3.1}$$

After studying the outliers, the decision was made to keep them in the dataset, as it is small and these values may positively influence the learning of the algorithm by highlighting extreme cases[35].

Handling Absolute Values

For those features or indicators with absolute values, such as the number of people, the values were converted to relative ones by dividing by the corresponding value from the *Total Population* feature, thus obtaining the relative value according to the country's population size.

Distribution

The distribution is a key characteristic to study in the starting dataset, as many *machine learning* models assume a normal distribution, also known as the Gaussian bell curve[36], which is the basis for many statistical methods and algorithms in machine learning[37].

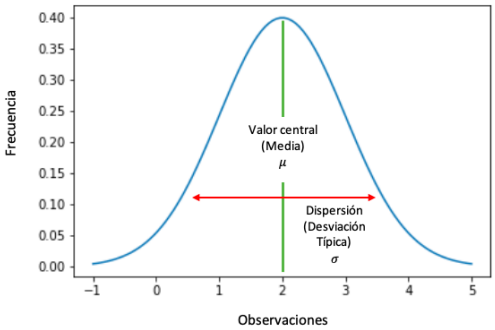

Figura 3.2: Normal Distribution[2]

To deduce the distribution of the input data, the histogram of each feature has been plotted. Although the Gaussian distribution is the most common[38], it is only present in a small group of features in our dataset. The most prevalent data distribution in our dataset is the negative exponential distribution. This is because, for most of the columns in our dataset, the most frequent values are small, and as the value increases, there are fewer instances.

Unknown values

The most notable issue with the dataset is the absence of a high number of values. That is, the value of many features is unknown for a specific year or country. The number of unknown values per instance in the dataset has been

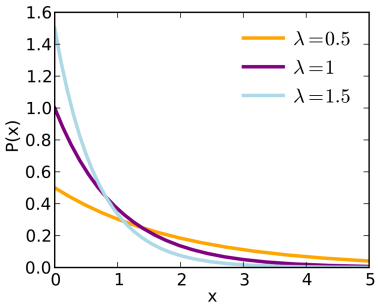

Figura 3.3: Negative Exponential Distribution[3]

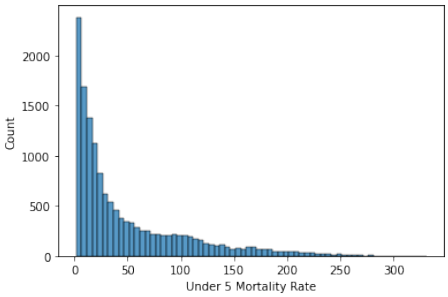

Figura 3.4: Histogram of the infant mortality rate under 5 years old

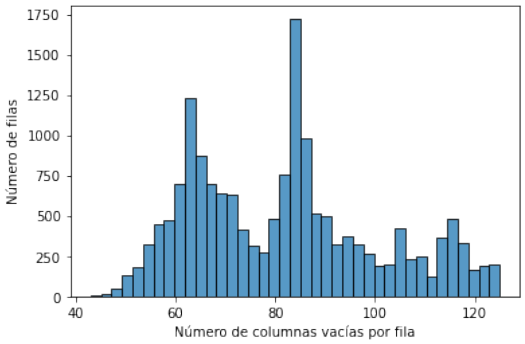

Figura 3.5: Histogram of the number of unknown values per record

studied, resulting in the histogram in Figure 3.5, which shows the frequency of unknown values per row.

We sought to identify the countries with the most unknown values in their rows. We can observe the number of empty values per country in the histogram shown in Figure 3.6.

3.1.5. Handling empty values

Machine learning models require a complete starting dataset for execution. This means they do not tolerate the presence of unknown values. As mentioned earlier, a high number of empty values were found during the exploratory data analysis. For this reason, various techniques have been applied to fill these fields or strategies to remove them.

Interpolation

Interpolation is the calculation or estimation of intermediate values of known points. These intermediate points are obtained by approximating the function that passes through the known values. Interpolation can be linear or polynomial (among others), depending on the type of function used to approximate the points.

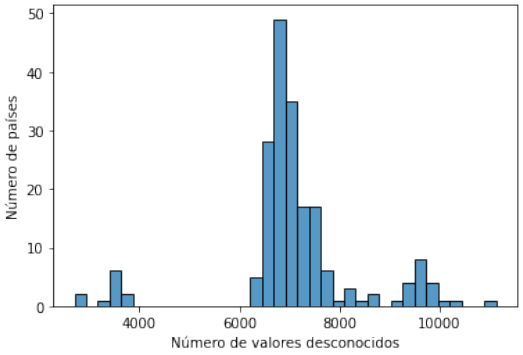

Figura 3.6: Histogram of the number of unknown values per country

Linear interpolation uses the linear function generated by two known exterior points to calculate the intermediate values.

Polynomial interpolation defines a polynomial function of degree N that passes through $N+1$ points to obtain the internal values between those points. We can observe its behavior in Figure 3.7.

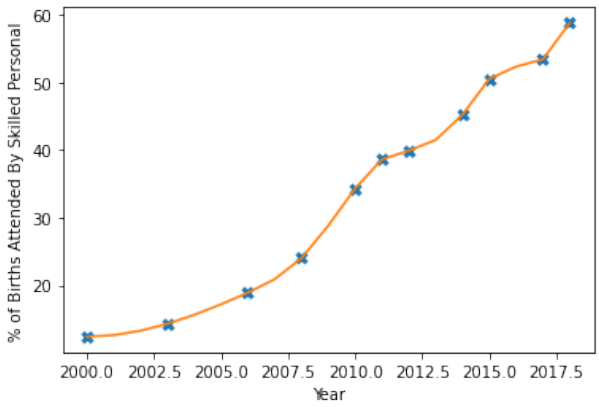

Figura 3.7: Interpolation of the percentage of births attended by healthcare personnel in Afghanistan

This mathematical technique[39] has been applied to obtain the intermediate values of the features, separating them by country and gender. The x-axis of the approximation function represents the year, and the y-axis represents the feature value. A polynomial interpolation of maximum degree 3 was applied to predict the intermediate values. For cases where fewer than 4 known values were available, a polynomial interpolation of degree 2 was applied. Finally, for cases where only two years' values were known, linear interpolation was applied.

After applying this technique, it was noticed that for certain predictors, impossible values had been generated. For example, negative values for the number of dentists per 100,000 inhabitants. This occurred because when there were sharp declines approaching zero, the interpolation function went out of the allowed value range. We can observe this behavior in Figure 3.8.

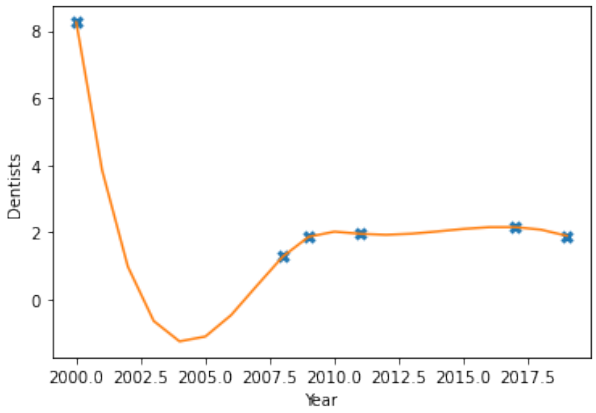

Figura 3.8: Interpolation of the dentist ratio in the Dominican Republic

For this reason, the function was modified to ensure it could only produce values within the established range defined by the minimum and maximum value of each feature.

Feature removal

For a large number of indicators, only values from a single year or a small set of values were known. Due to the high number of unknown values in these columns, they were removed from the dataset. These are:

- Low CI Value Air Pollution Death Rate Lower Respiratory Infections
- High CI Value Air Pollution Death Rate Lower Respiratory Infections
- Air Pollution Death Rate Lower Respiratory Infections Age Standardized
- Low CI Value Air Pollution Death Rate Lower Respiratory Infections Age Standardized
- High CI Value Air Pollution Death Rate Lower Respiratory Infections Age Standardized
- Air Pollution Death Rate Chronic Obstructive Pulmonary Disease
- Low CI Value Air Pollution Death Rate Chronic Obstructive Pulmonary Disease
- High CI Value Air Pollution Death Rate Chronic Obstructive Pulmonary Disease
- Air Pollution Death Rate Chronic Obstructive Pulmonary Disease Age Standardized
- Low CI Value Air Pollution Death Rate Chronic Obstructive Pulmonary Disease Age Standardized
- High CI Value Air Pollution Death Rate Chronic Obstructive Pulmonary Disease Age Standardized

- Air Pollution Death Rate Total
- Low CI Value Air Pollution Death Rate Total
- High CI Value Air Pollution Death Rate Total
- Air Pollution Death Rate Total Age Standardized
- Low CI Value Air Pollution Death Rate Total Age Standardized
- High CI Value Air Pollution Death Rate Total Age Standardized
- Air Pollution Death Rate Trachea Bronchus Lung Cancers
- Low CI Value Air Pollution Death Rate Trachea Bronchus Lung Cancers
- High CI Value Air Pollution Death Rate Trachea Bronchus Lung Cancers
- Air Pollution Death Rate Trachea Bronchus Lung Cancers Age Standardized
- Low CI Value Air Pollution Death Rate Trachea Bronchus Lung Cancers Age Standardized
- High CI Value Air Pollution Death Rate Trachea Bronchus Lung Cancers Age Standardized
- Unsafe Wash Mortality Rate
- Hepatitis B Surface Antigen
- Low CI Value Hepatitis B Surface Antigen
- High CI Value Hepatitis B Surface Antigen
- Reproductive Age Women

Removal of Countries

It has been observed that a large portion of the records with the highest number of unknown values were grouped in a small number of countries. For this reason, we identified which countries had so many indicators missing and then removed them from the dataset. The criterion chosen to select the number of unknown values that would determine the removal of a country was half the total number of values. In other words, if a country has half of its values missing, it will be removed. Following this strategy, the countries removed are listed below:

- Bahrain
- Bhutan
- Bolivia
- Brunei
- Burundi
- Comoros

- Democratic Republic of Congo
- Equatorial Guinea
- Eritrea
- Libya
- Micronesia (country)
- Montenegro
- North Korea
- North Macedonia
- Palestine
- Papua New Guinea
- Qatar
- Saint Vincent and the Grenadines
- Serbia
- Seychelles
- Singapore
- Somalia
- South Sudan
- Sudan
- Syria
- Timor
- Tonga
- Turkmenistan

Median

The median is the point or observation that divides the distribution of the sample into two halves, meaning it leaves the same number of observations to the right and left of that value.

By calculating this statistical value grouped by country, we proceeded to fill in the missing values for features where we knew values for one or more years. The values were assigned to those observations that could not be interpolated because the year was outside the known range.

After these initial techniques for imputing missing values and eliminating records, we observed a decrease in the number of missing values. This evolution can be consulted in graphs 3.9 and 3.10.

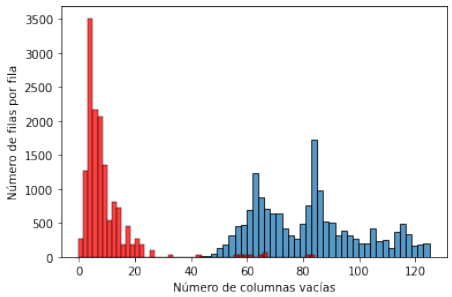

Figura 3.9: In blue: Histogram of the number of missing values per row original. In red: Histogram of the number of missing values per row after applying the initial techniques.

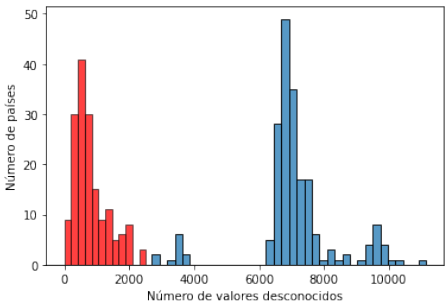

Figura 3.10: In blue: Histogram of the number of missing values per country original. In red: Histogram of the number of missing values per country after applying the initial techniques.

K-Nearest Neighbors Imputer

The K-Nearest Neighbors Imputer algorithm is an imputation technique based on finding the K nearest samples to the observation we want to estimate[40]. Once these points are found, the value is assigned using different possible techniques, such as the mean, mode, or median. Additionally, this algorithm is configurable to either assign the same value to the K samples or to weight their value based on the distance from the target sample.

This imputation technique was applied to assign a value to the remaining observations that had missing values. These observations are indicators for which, for that country, no value is known for any year, so it must be deduced from the others.

Before applying this technique, the target variable was separated from the set so that it would not influence the value assigned by the algorithm.

Since this missing value imputation algorithm works based on distances, it is necessary to normalize the data before applying it, as otherwise, the different scales in our dataset would cause biased values for the missing points. Therefore, the data was scaled to the range $[0, 1]$, and after applying KNN imputation, this transformation was undone, returning to the original values. Additionally, to preserve the information from categorical data when applying this technique, a One Hot Encoding was performed, which was undone after the imputation[41].

After applying this final imputation technique, the dataset was finally free of missing values.

3.1.6. Applied Transformations

Once the dataset is complete, certain transformations are necessary to apply the chosen machine learning models in order to achieve optimal results.

3.1.6.1. Categorical Data

Categorical data refers to data types whose possible values are limited to a finite set of values established as categories. This type of data, if non-numeric, is

problematic for most machine learning algorithms because they use mathematical operations for learning. Therefore, it is necessary to transform these categorical variables into numeric values.

The only categorical text data in the dataset are *Country* and *Gender*. For the feature *Country*, it was decided to remove it so that it would not influence the learning process, as the algorithm could simply learn by country, without considering the other indicators whose influence we want to study.

On the other hand, for the feature corresponding to gender, *Gender*, it was split into two columns, one for the male category (*Male*) and another for the female category (*Female*). These columns were set to 1 if the value of *Gender* corresponded to that value and 0 otherwise. This treatment is known as *One Hot Encoding*. However, in our case, for the category *Both sexes*, both columns were set to 1, making it a variant of this technique.

3.1.6.2. Continuous Data

Continuous data refers to data that can take any real numerical value. These types of data can exist in different scales or ranges. The learning method for a large percentage of machine learning models involves giving more importance to higher values, which could bias the results for features in different scales.

For this reason, it is important to transform continuous data so that they are within the same range and have an equivalent order of magnitude[42]. This transformation will be more or less relevant depending on the applied model. For example, when using models based on decision trees, this transformation is unnecessary[43]. However, for neural networks, it is a crucial preprocessing step, as it will enable the network to learn much faster by accelerating gradient convergence.

Normalization

Normalization, in statistics, refers to the transformation of the scale or range of values of a variable's distribution, primarily to make comparisons at the same order of magnitude with other possible distributions. There are several options for applying normalization, depending on the characteristics and distribution of the data to be transformed. The two most commonly used methods are:

- **Standardization**: Also known as Z-score, standardization transforms the data distribution to a normal distribution with a mean of 0 and a standard deviation of 1. In this way, all data points below the mean will be negative, while those above it will be positive. This transformation is ideal for distributions that resemble a bell curve. The formula for its application is:

$$x' = \frac{x - \mu}{\sigma}$$

  Where x is the value to be standardized, x' is the new value obtained, μ is the mean of the original distribution, and σ is the standard deviation.

- **Min-Max Scaling**: Especially used in preprocessing, scaling involves rescaling the data range to, generally, [0,1] or [-1,1], but this transformation allows choosing any range based on any tuple of values. The choice of this range will depend on the nature of the data. It is often applied to

distributions that do not allow standardization. The formula for the [0,1] range is:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

Where x is the value to be standardized, x' is the new value obtained, $max(x)$ is the maximum value of the original dataset, and $min(x)$ is the minimum.

Distribution Adjustment

Machine learning algorithms tend to perform significantly better when the data follows a normal distribution[44], as they assume such a distribution to perform their operations. As mentioned earlier, a large part of the data we start with, unlike usual, does not follow or resemble a normal distribution. In order to improve the learning of the models, there are different techniques to transform the distribution of the data so that it resembles a Gaussian bell-shaped distribution.

Box-Cox Transformation The Box-Cox transformation [45] encompasses a set of transformations used to correct biases in error distribution, unequal variances, and improve correlation between variables. Generally speaking, what this type of transformation achieves is converting a non-normal distribution into a normal form. This is achieved by the following logic:

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^{\lambda}-1}{\lambda} & if \quad \lambda \neq 0 \\ \\ log(y_i) & if \quad \lambda = 0 \end{cases} \tag{3.2}$$

Where y is the initial dataset, i is the index of the set, and λ is a hyperparameter to configure.

The optimal value of λ will be the one between -5 and 5 that minimizes the standard deviation of the transformed data. However, to apply this transformation, it is necessary to ensure that the data does not contain any negative or zero values, meaning that this transformation is only applicable to exclusively positive data.

Yeo-Johnson Transformation The Yeo-Johnson transformation [46] is an evolution of Box-Cox that allows transforming data with negative values. This transformation is achieved by applying the following formula:

$$y_i'^{(\lambda)} = \begin{cases} \frac{(y_i+1)^{\lambda}-1}{\lambda} & if \quad \lambda \neq 0, y_i \geq 0 \\ log(y_i+1) & if \quad \lambda = 0, y_i \geq 0 \\ -\frac{(-y_i+1)^{2-\lambda}-1}{2-\lambda} & if \quad \lambda \neq 2, y_i < 0 \\ -log(-y_i+1) & if \quad \lambda = 2, y_i < 0 \end{cases} \tag{3.3}$$

Where y is the initial dataset, i is the index of the set, and λ is a hyperparameter to configure.

As we can see, the behavior of this transformation is very similar to the Box-Cox for positive values, except for the increment of one to y. This would be equivalent to the Box-Cox transformation of $y + 1$. In the case where y is negative, the transformation would be a Box-Cox, but in this case of $-y + 1$ with $\lambda = 2 - \lambda$.

In Figures 3.11 and 3.12, we can observe the histograms of the distribution before and after the transformations, applied to a distribution resembling a normal distribution and one that does not. We can appreciate the subtle difference between the Box-Cox and Yeo-Johnson transformations. After these transformations, standardization was also applied.

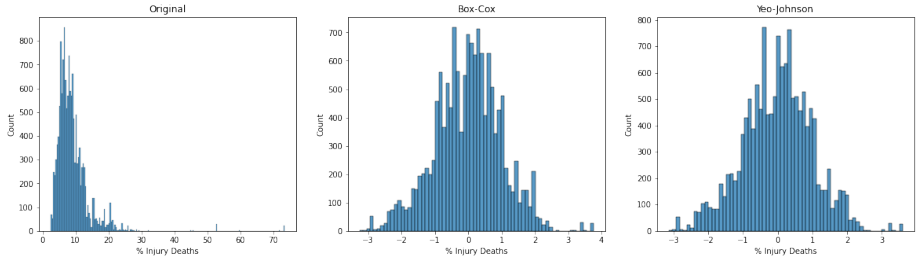

Figura 3.11: Transformations on a distribution resembling a normal distribution. Histogram of the percentage of deaths by injuries.

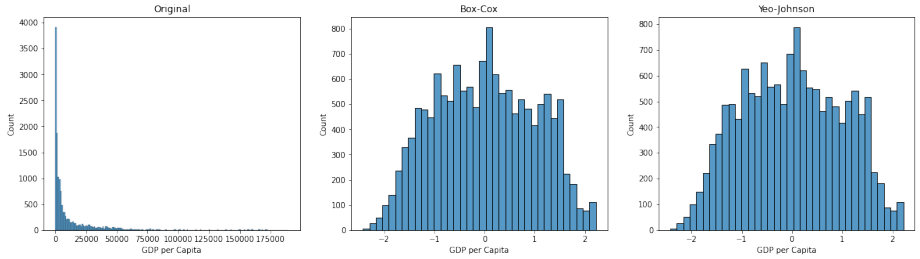

Figura 3.12: Transformations on a distribution not resembling a normal distribution. Histogram of GDP per capita.

In our dataset, standardization has been applied to all distributions that follow or at least resemble a normal distribution. Next, the Yeo-Johnson transformation has been applied to all features except those related to gender. After this last transformation, with all the features in the shape of a Gaussian bell curve, standardization was applied, resulting in all features having the same order of magnitude, with a mean of 0 and a standard deviation of 1. The standardization prior to Yeo-Johnson for variables following a normal distribution is applied because it tends to improve the results[47].

It is worth noting the special treatment applied to the target variable *Life Expectancy*. This feature was not subjected to the Yeo-Johnson transformation; it was only standardized. The reason for this decision lies in the limitations of the transformation function. In order to measure the model's error more representatively, it is necessary to undo the transformation applied to the obtained value. To invert the applied transformation, the standardization done afterward

will be undone, and the inverse Yeo-Johnson function will be used, that is:

$$y_i^{(\lambda)} = \begin{cases} (y' \cdot \lambda + 1)^{\frac{1}{\lambda}} - 1 & if \quad \lambda \neq 0, y_i' \geq 0 \\ 10^{y'} - 1 & if \quad \lambda = 0, y_i' \geq 0 \\ 1 - (-(2-\lambda) \cdot y' + 1)^{\frac{1}{2-\lambda}} & if \quad \lambda \neq 2, y_i' < 0 \\ 1 - 10^{-y'} & if \quad \lambda = 2, y_i' < 0 \end{cases} \tag{3.4}$$

Where y is the initial dataset, i is the index of the set, and λ is a hyperparameter to configure.

However, unlike the original, this function is not continuous for all real numbers. In practice, this resulted in cases where predicted life expectancy values that were lower than a certain threshold (specifically, the minimum value prior to the transformation) would perform a fractional exponent of a negative number to make the inversion, which would result in a complex number and, therefore, invalid.

Once preprocessing is complete, we now have a unified, clean, correct, adjusted, and complete dataset. Therefore, machine learning models can now be applied.

3.2. Feature Selection

After running the first iteration of the workflow on the dataset, it has been detected, based on the preliminary results obtained (4.5.1), that some of the most important variables are too closely related to life expectancy, and their value depends on factors that are not clearly identifiable. Therefore, the decision was made to eliminate those indicators that are not actionable, treatable, or modifiable through monitoring policies in a feasible and clear way. Immutable factors such as gender and year were retained.

The goal of this change is to determine which indicators can be modified in order to achieve a positive change in the prediction of the country's life expectancy. In this way, key points can be identified that a country could address to improve the expected lifespan of its population.

Therefore, the following features were removed:

- Infant Mortality Rate (and its Low CI Value and High CI Value)
- Under 5 Mortality Rate (and its Low CI Value and High CI Value)
- % Population Aged 0-14
- % Population Aged 15-64
- % Population Aged 65+
- % Population Aged 65-69
- % Population Aged 70-74
- % Population Aged 75-79
- % Population Aged 80+

- Neonatal Mortality Rate (and its Low CI Value and High CI Value)
- Maternal Mortality Rate (and its Low CI Value and High CI Value)
- Death Rate
- Total Population

Additionally, to broaden the range of factors characterizing a country at a specific time, the following new indicators were added, extracted from the same sources as the previous ones:

- **Homicide Rate**: Homicide ratio. Number of homicides per 100,000 inhabitants.
- **Government Expenditure Education**: Public expenditure on education as a percentage of total public expenditure.
- **Government Expenditure Military**: Public expenditure on defense as a percentage of total public expenditure.
- **Government Expenditure Health**: Public expenditure on healthcare as a percentage of GDP.
- **Diet Composition**: Consumption of food types in kilocalories per person per day. It is divided into:

  Sugar: Sugar

  Oil and Fats: Oil and fats

  Meat: Meat

  Dairy and Eggs: Dairy and eggs

  Fruit and Vegetables: Fruits and vegetables

  Starchy Roots: Starchy roots

  Pulses: Pulses

  Cereal and Grains: Cereal and grains

  Alcoholic Beverages: Alcoholic beverages

  Other: Other foods
- **Vegetable Consumption**: Vegetable consumption in kilograms per person per year.
- **Fruit Consumption**: Fruit consumption by type in kilograms per person per day. It is divided into:

  Bananas: Bananas

  Dates: Dates

  Other Citrus: Other citrus

  Orange and Mandarines: Oranges and mandarins

  Apple: Apple

  Lemons and Limes: Lemons and limes

  Grapes: Grapes

Grapefruit: Grapefruit

Pineapple: Pineapple

Platains: Plantains

Other: Other fruits

- **Cereal Consumption**: Cereal consumption in kilocalories per person per day. It is divided into:

  Oats: Oats

  Rye: Rye

  Barley: Barley

  Sorghum: Sorghum

  Maize: Maize

  Wheat: Wheat

  Rice: Rice

- **Diet Calories**: Diet by macronutrients in kilocalories per person per day. It is divided into:

  Animal Protein: Animal protein

  Plant Protein: Plant protein

  Fat: Fat

  Carbohydrates: Carbohydrates

3.3. Correlation Analysis

Correlation analysis between variables is one of the most commonly used techniques to interpret the behavior of the dataset with respect to the target variable. It is a necessary first step in the construction of more complex predictive models.

The correlation between two variables evaluates the direct relationship between them. The correlation index is a value within the range $[-1, 1]$. A positive value of this index will indicate a directly proportional relationship, while negative values will indicate an inversely proportional relationship. The closer the value is to zero, the weaker the relationship between the studied variables.

Correlation can be calculated using different techniques. The most common ones are **Pearson's linear correlation** and **Spearman's correlation**. Pearson's linear correlation coefficient measures the linear trend between the studied variables, while Spearman's correlation coefficient measures the monotonic (increasing or decreasing) trend, meaning that both variables move in the same relative direction, even if not constantly.

This correlation analysis was applied to the dataset using Pearson's linear correlation coefficient. As mentioned earlier, it assumes the linearity of the data, so it will only highlight linear relationships, which means it will not be very useful if the problem is nonlinear. However, it is very useful for gaining an initial insight into the possible type of problem the data may present, allowing for a

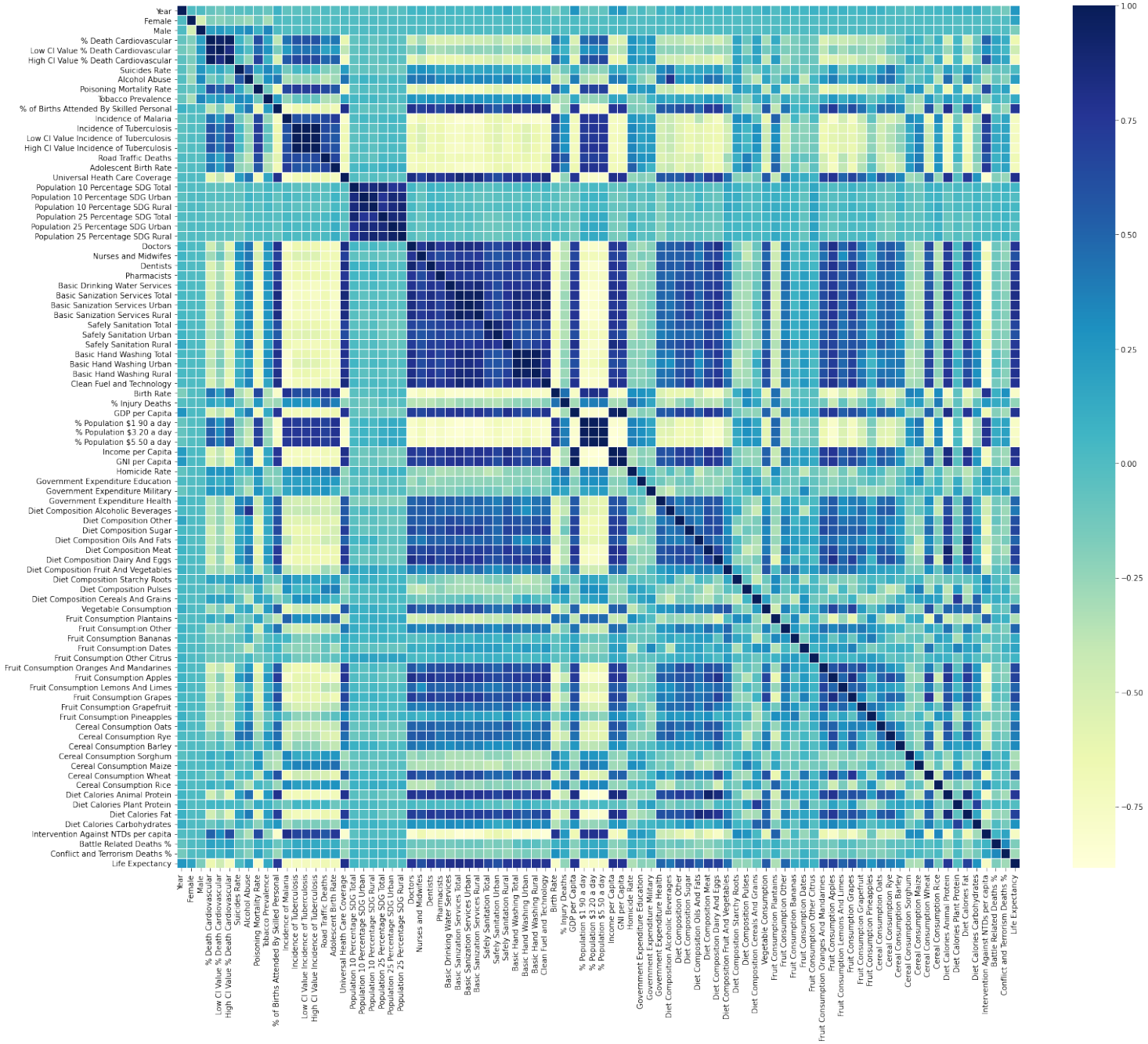

Figura 3.13: Pearson's linear correlation between features

preliminary analysis of which features simpler models will give more importance to. Additionally, we can rule out whether or not the problem is linear.

We can observe the correlation matrix in figure 3.13. By analyzing the matrix, we can conclude that there are groups of indicators with high relationships among them, such as those referring to the healthcare staff ratios per capita. Regarding the target variable, we observe a high correlation with a significant number of features, while with others, the correlation coefficient is close to zero. There are feature filtering techniques to select only those with high correlation to the dependent variable. However, since we are unaware of the linearity of the problem, we decided not to remove these variables, as it could harm the model's performance.

We analyzed the purity of the linear correlation between life expectancy and the features most proportional to it by observing their behavior through the scatter plots developed in the previous data analysis. In the scatter plot of figure 3.14, we can observe that for *Income per Capita*, despite the high correlation coefficient (0.798), the relationship between both variables is clearly nonlinear.

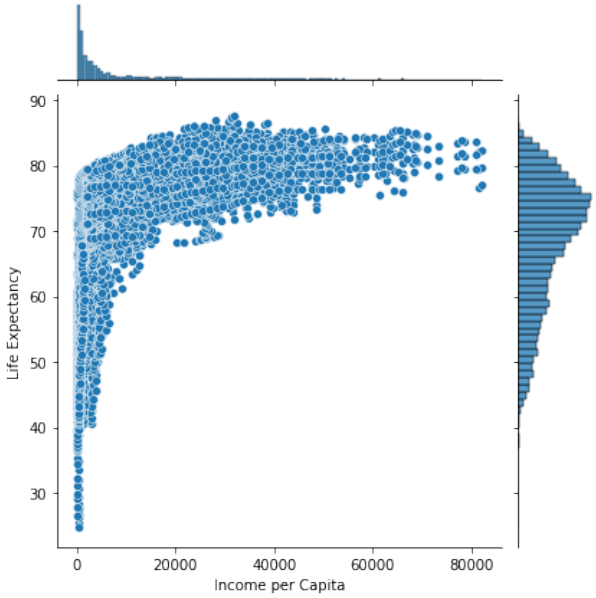

Figura 3.14: Comparison of life expectancy and income per capita

This analysis indicates that most of the variables in the dataset are significant, as they show a notable correlation with the target variable, even though it is not a direct relationship.

3.4. Training and Test Set Split

The dataset has been divided into two sets: the training set and the test set. This division is necessary because we need a dataset to evaluate the quality of the model, and that data should not have been used during training.

The test set must be large enough to generate meaningful results and be

representative of the entire dataset to provide a valid error estimate. Thus, the model's goal will be to generalize and make correct predictions for the test dataset.

The split is typically done in an 80 % training and 20 % test ratio. In our case, this division was made following these proportions. This separation was done randomly, trusting that the random process would yield an equitable and meaningful division of the dataset.

However, for training some machine learning models, it is necessary to validate how well the model is performing. This check cannot be done on the test set, as it must remain completely separate from the training and should not have been used before the evaluation. For this reason, for these models, the training set has been further split, creating a validation set, with a smaller proportion than the training set. This new set allows us to evaluate and validate the model during training while keeping the test set completely independent.

3.5. Applied Machine Learning Models

To address the problem, various regression models were applied, from simpler to more complex ones, to gradually observe the progression of error and the distinctive behavior of each approach. The applied models are described below.

3.5.1. Multiple Linear Regression

Linear regression seeks to find a direct and linear relationship between the input variable (X) and the target variable (y), also known as the dependent variable[48]. The equation defining this model is:

$$y = \beta_0 + \beta_1 x \tag{3.5}$$

The algorithm will attempt to optimize the unknown values of β_0 and β_1 to minimize the difference between the predicted and the target value. In this way, it tries to draw a line that approximates the desired values. We can observe the behavior of this model with an input set through the graph in figure 3.15.

Multiple linear regression is an extension of linear regression, and it is one of the simplest and most commonly used models in machine learning. This model extends the input to an indefinite number of dimensions, following this equation:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{3.6}$$

where n is the number of input dimensions, $x_1 ... x_n$ are the input values of each dimension, and $\beta_0 ... \beta_n$ are the coefficients to be determined by the algorithm. This simple algorithm was applied to observe the linearity between the predictors and the dependent variable, as well as to get a small understanding of the data quality and the complexity of the problem we are dealing with.

3.5.2. K-Nearest Neighbors Regressor

This machine learning method is based on the nearest or most similar samples to predict the value of the target variable[48]. It is an instance-based model, which means it doesn't explicitly learn a model but memorizes the training set to predict the test set. The process follows these steps:

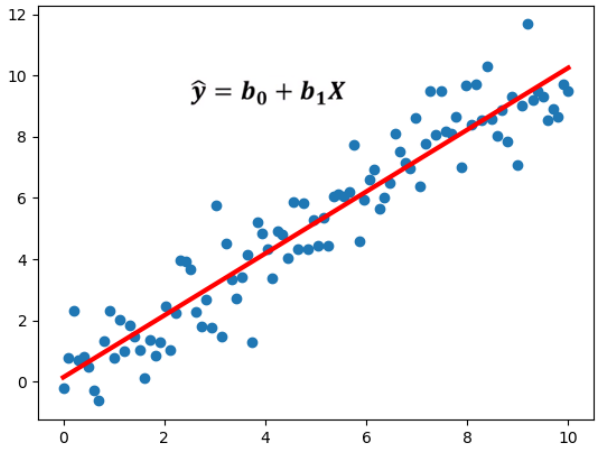

Figura 3.15: Behavior of linear regression[4]

1. Determine the distance between the sample to predict and the training set.
2. Select the k closest points, known as neighbors.
3. Predict the value of the sample by averaging the values of the neighbors, according to the chosen weighting.

The hyperparameters to determine for this model are two:

- k: The number of neighbors to use for prediction.
- *weights*: The weighting type chosen to establish the prediction. It can be *uniform*, where all neighbors have the same weight, or by *distance*, where closer neighbors have more weight.

We can observe the behavior of this model in the graph of figure 3.16 for a one-dimensional input problem.

k-Nearest Neighbor was applied to observe the results with a simple model that, unlike multiple linear regression, does not require linearity in the target variable. However, it is important to note that the learning and prediction of this model are not recommended for datasets with a high number of rows or predictors, which is the case here. High dimensionality makes the execution time, both for training and prediction, significantly higher and requires high computational capacity.

GridSearch

Determining the hyperparameters of a model is not easy, and often the best method to optimize hyperparameters is to perform tests with combinations of multiple possible values.

This procedure was followed to set the hyperparameters for the KNN Regressor model. These tests were implemented using the *GridSearch* method, available in the *sklearn* library[17]. Tests were carried out with the following values for the number of neighbors: 2, 3, 4, 5, and 10, and with two possible weighting types:

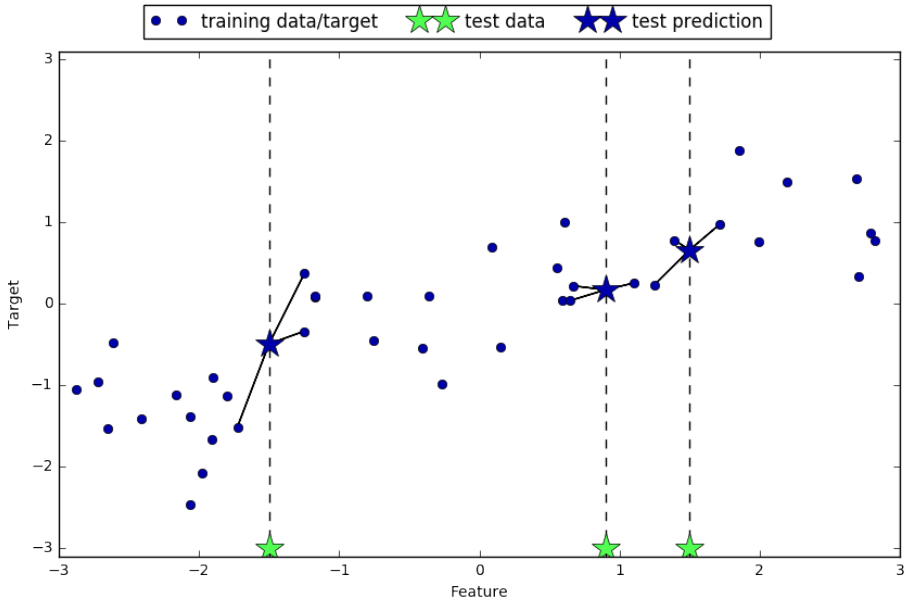

Figura 3.16: Behavior of KNN regression[5]

by distance and uniform, resulting in the lowest error with k=4 and distance weighting.

3.5.3. Random Forest Regressor

A **decision tree** is a machine learning algorithm based on decision-making in a tree-like structure. In this decision tree, each node establishes a condition on a feature of the input set, with a boolean answer, i.e., true or false. This node then splits into two branches, one for each possible answer. Finally, through the condition nodes, we reach a leaf node, which determines the predicted value for the analyzed sample. A decision tree for regression is built using a greedy algorithm that optimizes the following cost function:

$$J(a, l_a) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right} \tag{3.7}$$

where a is an attribute or feature, l_a is the attribute limit, m is the number of samples, and MSE is the mean squared error. The greedy algorithm will determine the best attributes and limits for decision-making. We can see the behavior of a decision tree for regression in figure 3.17.

The **Random Forest** regressor is a set or *ensemble* of decision trees with *bagging*, which means trees are generated from different portions of the training set. These trees are created randomly, speeding up the execution process. The final result for this regressor is the arithmetic mean of the results from each random tree. The hyperparameters to note are:

- Number of trees in the forest
- Maximum depth

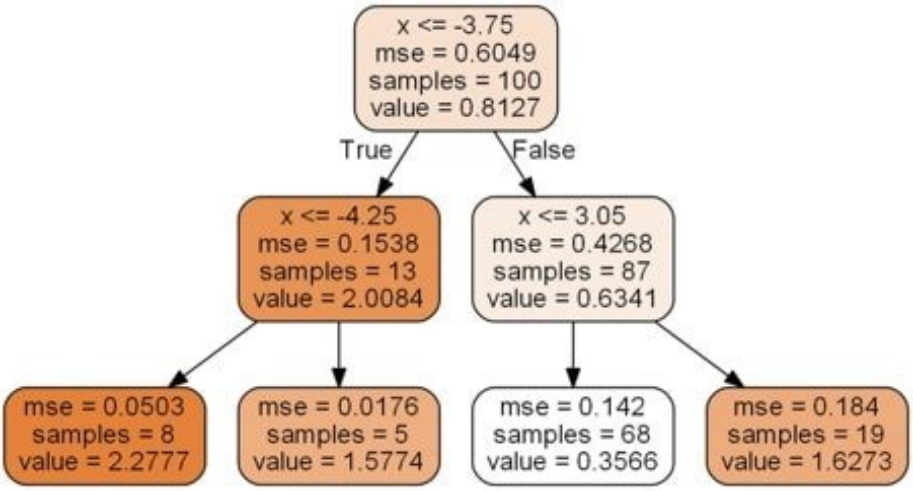

Figura 3.17: Decision tree for regression[6]

- Minimum samples required to split an internal node
- Minimum samples required for a leaf node

Contrary to what might seem intuitive, this machine learning model usually yields very good results because the high number of trees compensates for errors. Additionally, it has a greater ability to generalize compared to regular decision trees and can handle high-dimensionality data. Another advantage of this model is that it can return the importance of each feature it uses, which is very useful for analyzing its behavior.

However, this model has the disadvantage of not being able to predict values outside the input range, which limits its application.

The Random Forest applied to predict life expectancy was implemented with a maximum tree depth of 100 nodes and a forest size of 100 trees (since increasing the number of trees did not result in significantly better outcomes and increased computation time).

3.5.4. Neural Network

Neural networks are based on the neuronal behavior of the human brain[49]. They consist of a set of nodes called artificial neurons that are connected to each other and transmit a signal. Depending on the topology of the network, there are various types of neural networks, such as:

- Simple Perceptron
- Multilayer Perceptron
- Convolutional Neural Network
- Recurrent Neural Network
- Radial Basis Networks

The type of neural network applied in this work has been the multilayer perceptron. The multilayer perceptron is an extension of the simple perceptron.

3.5.4.1. Simple Perceptron

The simple perceptron is a neural network with a single layer, the output layer. All inputs are transmitted to the neurons that make up this layer. A neuron consists of inputs, each with an associated weight that will be modified; a bias, which will be a real number that will also be updated; and an activation function, which for the simple perceptron will always be the step function, defined as:

$$f(x) = \begin{cases} 0 & if \quad x < 0 \\ 1 & if \quad x \geq 0 \end{cases} \tag{3.8}$$

The initialization of both the weights of the inputs and the bias is typically done randomly.

The functioning of a neuron in the output layer consists of two operations: propagation and weight update.

Propagation

Propagation determines the network's output, and it is obtained by calculating the following operation:

$$s = \sum_{i=1}^{n} F_{activation}(e_i \cdot w_i) + b \tag{3.9}$$

Where $F_{activation}$ is the activation function, n is the number of inputs, e is the input value, w is the weight value, and b is the bias. We can observe the behavior of a neuron in a simple perceptron in Figure 3.18.

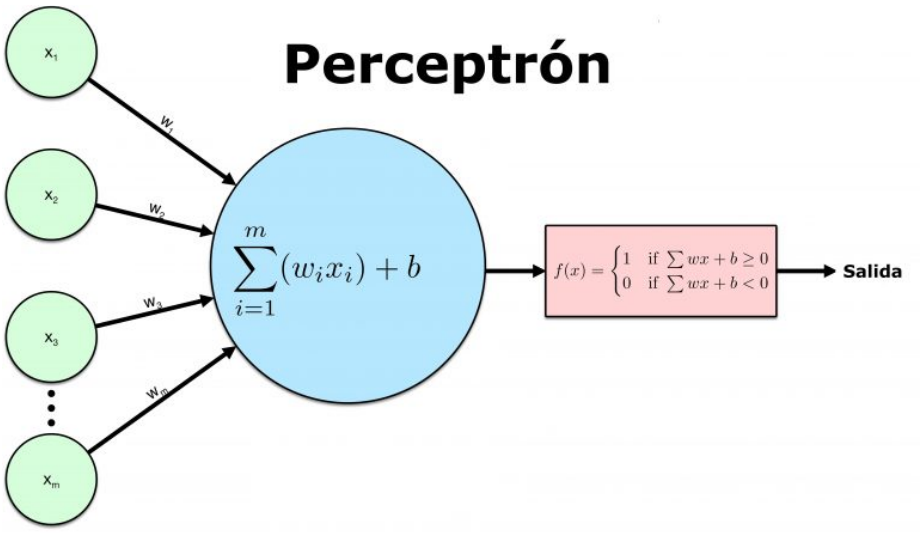

Figura 3.18: Artificial Neuron[7]

Weight Update

The weights will be updated according to the error provided by the error function or *loss*, which is typically the difference between the expected value and the obtained value $(d - s)$. The function that determines the weight update is:

$$\Delta W = \alpha \cdot e \cdot F_{loss} \tag{3.10}$$

Where W are the weights, e is the input, F_{loss} is the error function, and α is the learning rate.

The learning rate is a scalar with a reduced value that controls the variation of the weight increment. The higher the learning rate, the faster the model, but it is more likely to reach a local minimum or for the error to oscillate. The smaller this value, the slower the learning, but better results will be obtained.

For the bias update, the following is done:

$$\Delta b = \alpha \cdot F_{loss} \tag{3.11}$$

The propagation and backpropagation operations are performed for the entire input set. This is known as an *epoch*. The higher the number of *epochs*, the more the network will learn, but if exceeded a certain limit, overfitting may occur.

The simple perceptron is the simplest of the neural networks, although it lays the foundation for the multilayer perceptron. It is characterized because it can only solve linear problems.

3.5.4.2. Multilayer Perceptron

The multilayer perceptron extends the behavior of the simple perceptron by adding intermediate layers between the inputs and the output layer; these are known as hidden layers. The problem posed by hidden layers is the lack of knowledge of the desired output for those layers, so the weight update cannot be calculated.

The solution implemented by the multilayer perceptron to circumvent this problem is gradient backpropagation.

Gradient Backpropagation

The gradient is the directional derivative that results in the direction of maximum growth of the function, allowing us to obtain the vector direction to converge where the error is smaller. This technique uses the derivative of the activation function. However, the step function described earlier does not have a derivative, so it is not applicable to the multilayer perceptron. In replacement, the most commonly used are those listed in Table 3.2.

Normally, for neural networks with a small number of hidden layers, the sigmoidal function is used as the activation function. However, as the number of hidden layers increases, using this function or the linear function presents the problem of gradient vanishing.

Gradient vanishing affects gradient backpropagation, making it so that the more layers are backpropagated, the smaller the gradient value becomes, and thus the propagation of the weights. As a result, the neural network is not able to learn.

Function	Formula	Derivative
Linear	$f(x) = x$	$f'(x) = 1$
Sigmoidal	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x) \cdot (1 - f(x))$
ReLU	$max(0, x)$	$f'(x) = 1$ if $x > 0$

Table 3.2: Derivable Activation Functions

This problem does not occur with the linear function because its derivative is a constant. The downside of this is that, being linear, when applied, all layers would be equivalent to a single matrix operation, which would be equivalent to a simple perceptron, which can only solve linear problems, as mentioned earlier.

Therefore, the necessary function had to be differentiable, non-linear, and differentiable.

A solution was found with the use of the ReLU function, i.e., the Rectified Linear Activation Function. This allowed backpropagation without vanishing and was not linear. The disadvantage of this activation function is that it can never return negative values[50].

Finally, the gradient backpropagation equations are as follows:

$$\Delta W^{(k)} = \alpha \cdot s^{(k-1)} \cdot \delta^{(k)}$$

$$\Delta b^{(k)} = \alpha \cdot \delta^{(k)}$$

For the output layer:

$$\delta^{(n)} = F_{loss} \cdot f'_{activation}(s^{(n)})$$

For the hidden layers:

$$\delta^{(k)} = W^{(k+1)} \cdot \delta^{(k+1)} \cdot f'(s^{(k)})$$

Where W are the weights, k is the layer number, α is the learning rate, b is the bias, n is the total number of layers, and s is the output.

In Figure 3.19, we can see the structure of a multilayer perceptron with a dense hidden layer.

Application

The neural network used to predict life expectancy consists of a multilayer perceptron with two hidden layers of 350 neurons and a ReLU activation function. The output layer has a single neuron that will return the life expectancy. This last layer has a linear activation function. This is because the data has been standardized, so life expectancy encompasses a range of both positive and negative values, which is why it is not possible to use the ReLU function, which only returns positive values as mentioned earlier. After testing with more and fewer hidden layers and neurons, it was concluded that this was the best structure because it is complex enough to solve the problem but not too much

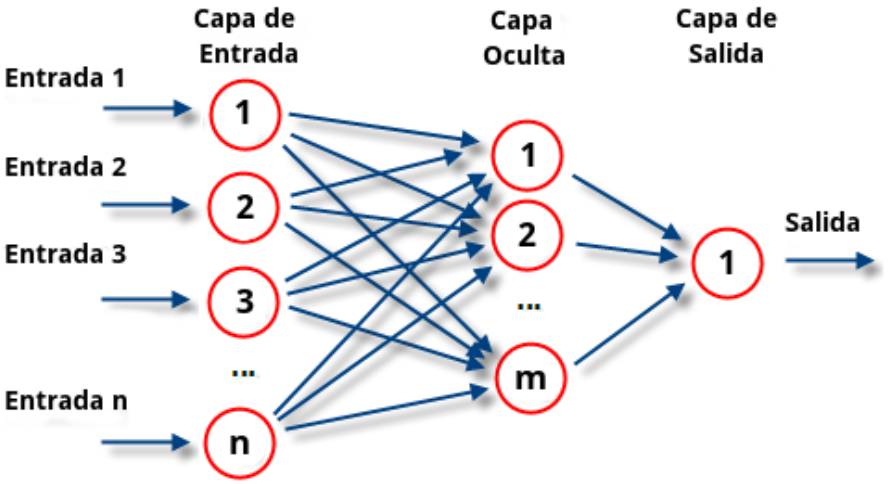

Figura 3.19: Multilayer Perceptron[8]

to fall into *overfitting*, meaning the model does not excessively fit the training data.

The applied *loss* function has been the mean squared error, known as MSE.

The chosen learning rate, aiming to reduce error oscillation and avoid local minima, has been very low, at 0.0001. Additionally, the Adam optimizer[51] has been used. The Adam optimizer (*adaptive moment estimation*) is an extension of gradient descent, meaning it is used in updating the network weights. While classical gradient descent maintains a learning rate for all weight updates, Adam varies this value according to the moments of the gradient. There are other well-known optimization techniques for neural networks, such as RMSprop (in fact, Adam is based on this technique), but worse results were obtained for our problem.

The learning of the network is highly conditioned by the number of *epochs* the network will be trained. The strategy applied to determine this number has been early stopping or *early stop*[52]. To apply this technique, a small proportion of the training set is reserved as a validation set, which the network will not learn from. The neural network will be tested on the validation set at the end of each *epoch*. *Early stop* states that if the validation error increases over a certain number of *epochs* (in our case, 30 has been chosen due to high error oscillation), the learning of the neural network will be stopped before reaching the established number of *epochs*, hence *early stop*. If the validation error never worsens, the network will run as many *epochs* as were originally set, in our case 1000.

3.5.5. k-Fold Cross Validation

Cross-validation arises from the idea that when randomly separating the data into a training set and a test set, there may be bias, with the training set lacking certain characteristics in the data[48].

To avoid this randomness, cross-validation is used. It divides the data into k

different random splits, applies the model to each, and measures the error. The average of these metrics will be the model's error. We can observe the behavior of this division in Figure 3.20.

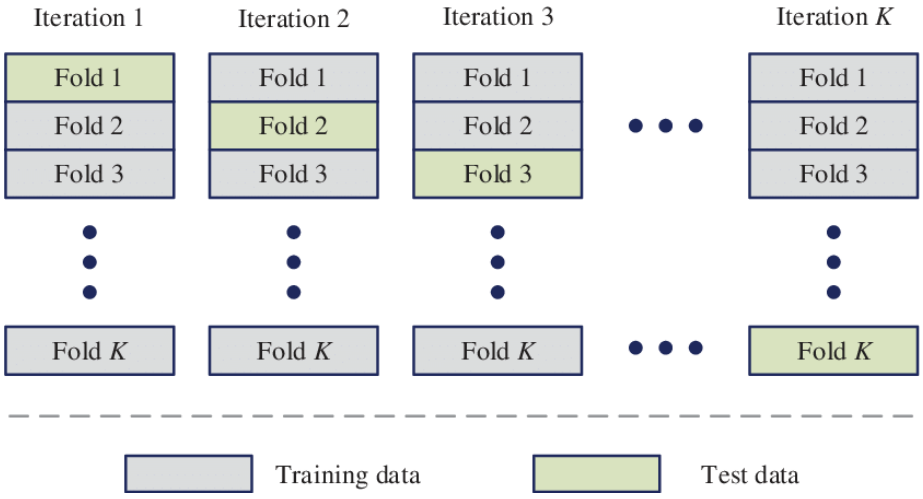

Figura 3.20: K-Fold Cross Validation[9]

Cross-validation has not been possible to apply to the *Random Forest* and multilayer perceptron models due to the computational complexity of both models. As for linear regression, since it was only used as a reference to apply more complex models, there was no need to apply this validation. However, it was applied in the *KNN Regressor* model with k equal to 5. It is important to highlight that this division was made on the training set, leaving the test set completely free of any contact with the model, in order to obtain a more reliable error. This division of the training set is called the validation set.

The detailed process followed can be seen in Figure 3.21.

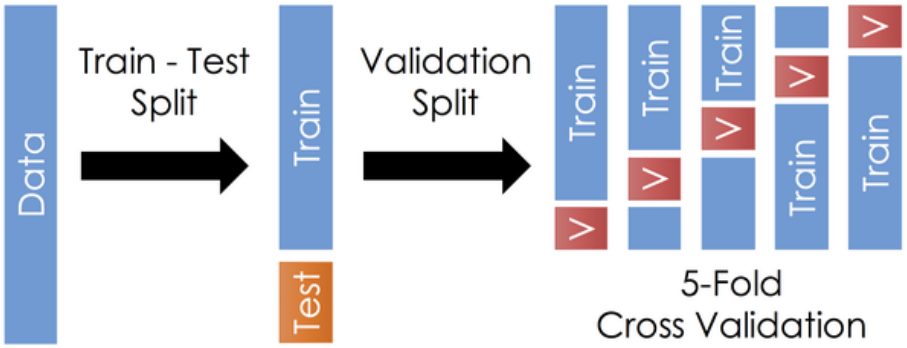

Figura 3.21: Training, Validation, and Test Set Division[10]

3.6. Techniques Applied for Result Analysis

3.6.1. Wrapper Methods

Wrapper methods are a feature selection technique where features are evaluated based on how much they improve the model's error with or without them[49]. They work by training and evaluating the model with a subset of features, selecting the subset with the least error or that reduces it the most.

The selection or order of choosing these subsets of predictors to train and evaluate the model can be done from several possible approaches:

- **Forward Feature Selection**: It adds one feature at a time to the model, starting from zero.
- **Backward Feature Elimination**: Unlike the previous strategy, this one starts with the model with all the features and eliminates one by one from the subset.
- **Exhaustive Feature Selection**: This is equivalent to a brute-force approach, trying every possible combination of features until the optimal subset is found.

Due to the computational complexity and execution time required to train a neural network, as well as the enormous number of features in the dataset, the exhaustive selection method has been discarded. Therefore, the other two approaches have been applied.

The order of creating subsets, either by adding or removing features, may or may not have a certain criterion. In our case, to capture the most relevant subset, the features were ordered based on two criteria:

1. Correlation with the target variable.
2. Based on the importance obtained from the results of the *Random Forest Regressor* model.

3.6.2. Genetic Algorithm

The genetic algorithm[53] is a search and optimization technique based on Charles Darwin's theory of the origin of species, also known as natural selection[54]. It posits three fundamental principles:

1. Each individual tends to transmit its traits to its offspring.
2. Nature produces individuals with different traits.
3. The most adapted individuals tend to produce more offspring.

This algorithm starts with an initial population, whose individuals are possible solutions to the problem, whether valid, invalid, of higher or lower quality. The quality of the solution, known as *fitness*, will be the value the algorithm seeks to maximize. To do so, it will use its three fundamental operations: selection, pairing, and mutation.

Initialization

The initial population will consist of N individuals. Each individual corresponds to a chromosome, a string of genes representing a possible solution to the problem. Depending on the problem, the type of each gene may vary. In our case, each gene in the chromosome will correspond to the value of a feature. Thus, each chromosome will be a real number within a specified range, called the alphabet. The alphabet is the set of possible values for a gene.

Fitness Function

The quality of an individual is determined by the fitness function, which the algorithm will try to maximize. Given a chromosome, this function will calculate the quality of the solution and thus the quality of the individual. The higher this value, the more adapted the individual will be considered to be, and, according to Darwin's laws, the more offspring it will leave.

Selection

Selection is the process by which certain individuals from the population are chosen to pair and create offspring. Several strategies can be used for this selection process, the most common being:

- **Random**: Individuals are chosen randomly with equal probability.
- **Roulette**: More adapted individuals have a higher probability of being selected. Due to the poor quality of the initial individuals, this technique can yield bad results. It can be improved by applying what is known as normalization, which applies the exponential function of the individual's fitness to perform the selection.
- **Tournament Selection**: T random individuals are chosen to compete and determine who will reproduce, with the individual with the highest fitness winning.

Pairing

Pairing consists of dividing and joining two individuals to create offspring. Each pair of individuals will produce two offspring, who will inherit the genetic material of the parents. The pairing strategy will vary depending on the type of problem, whether it is classical, permutation, numeric, etc. For our case, pairing is done following the formulas:

$$\begin{aligned} a' &= \beta \cdot a + (1 - \beta) \cdot b \\ b' &= (1 - \beta) \cdot a + \beta \cdot b \end{aligned} \tag{3.12}$$

Where a and b are the parent chromosomes, a' and b' are the offspring, and β is a random number within the range $(0, 1)$.

Mutation

Mutation is the process by which new genetic material is introduced into the population. Mutating consists of changing the genes of a chromosome with a certain probability. To do this, the chromosome will be traversed gene by gene. The mutation probability will determine whether the gene will mutate or not. In case of mutation, the new value will be a randomly chosen element from the alphabet, with each element having the same probability.

These three operations will be executed across the entire population, creating a new generation.

Application

The strategy applied to determine the importance of the indicators through this algorithm is based on the article by Federico Piccinini[26] on this subject and follows this logic:

Given a real case from a country in a specific year and a particular gender (or both), we will aim to optimize the features that maximize life expectancy by modifying the original feature values as little as possible. Thus, those features that are most modified will be considered more important in calculating the target variable.

Therefore, we need the solution to be as close as possible to the input, but the output should be as far as possible (positively) from the original prediction. To achieve this, the fitness function has been implemented with the following hyperparameters:

- *margin_input*: The margin of modification for the input. It represents the maximum allowed difference between the original input and the chromosome values (total sum). In other words, it determines how much the original input can differ from the proposed solution.
- w_{input} and w_{output}: The weights for the importance of the similarity between the input and output of the chromosome. The total sum will equal 1.

The quality calculation follows this procedure:

Input To determine the quality of the input, the sum of the differences between each feature and its original value is calculated. Features referring to year and gender should not be modified since we are analyzing the influence of the indicators, not these two factors. Therefore, to penalize the difference of these immutable features with respect to the original input, their difference is multiplied by 100.

Since we want to minimize this value, the inverse is applied so it can be maximized, as the genetic algorithm only maximizes quality.

Once the difference is below the threshold set as *margin_input*, the quality will be equal to 1.

$$dif_{input} = \left(\sum_{no\ mutables} |x_{original} - x_{cromosoma}| \right) \cdot 100 + \sum_{mutables} |x_{original} - x_{cromosoma}|$$

$$f_{input} = \begin{cases} \frac{1}{1+dif_{input}} & if \quad dif_{input} \geq margin\_input \\ 1 & if \quad dif_{input} < margin\_input \end{cases} \tag{3.13}$$

Where x is the value of each feature, the *non-mutable* set contains the features *Year*, *Male*, and *Female*, and the *mutable* set contains the rest.

The chromosome will be a list containing the values of each feature that the model takes as input.

Prediction The quality of the output will always be zero until the input similarity reaches the value of *margin_input*, which will determine how much the input is allowed to change at most. Once this point is reached, the quality of the output will be calculated as the difference between the predicted life expectancy from the neural network and the original life expectancy value. To give more importance to this difference, the value is cubed after adding 1 (to avoid penalizing values between 0 and 1).

$$f_{output} = \begin{cases} (le\_predicted - original\_le + 1)^3 & if \quad dif_{input} < margin\_input \\ 0 & if \quad dif_{input} \geq margin\_input \end{cases} \tag{3.14}$$

Where *le_predicted* is the predicted life expectancy of the chromosome and *original_le* is the original value.

Union Finally, the differences are summed, weighted by their respective weights.

$$f = w_{input} \cdot f_{input} + w_{output} \cdot f_{output} \tag{3.15}$$

The values chosen for the input and output weights were 0.1 for w_{input} and 0.9 for w_{output}, to give more importance to the output result once sufficiently similar values have been reached for the inputs. The selection technique chosen was roulette and normalization with a crossover probability of 0.7. The mutation probability was set to 0.75. A high probability was chosen to allow the algorithm to try a greater number of different solutions, expanding the search space. To counteract this decision and avoid losing the best solution, elitism was used, so the best individual from a generation always passes to the next one without being modified. Finally, a population size of 100 individuals was chosen.

The genetic algorithm was executed using Salga, a visual interface implementation of the genetic algorithm developed by the Polytechnic University of Madrid[19].

3.6.3. Shapley Additive Explanations (SHAP)

SHapley Additive exPlanations, or SHAP for short, is a model interpretation technique applicable to any type of *machine learning* model. A black-box model is one that has inputs and produces outputs, but its decision-making process for determining the output is not easily understood.

The algorithm, developed by Lundberg and Lee in 2017[24], is based on Shapley values[55], a concept from game theory. Game theory involves a game and its players. The sum of each player's contributions will determine the final

state of the game. This theory is applied to model interpretation, where the resolution of the game represents the output of the black-box model, and the players are the different features that make up the inputs.

The value contributed by each player to the final result is specified by its Shapley value. In our case, the values that determine the quantitative contribution of each feature to the final result calculated by the model are known as *SHAP values*.

The game they contribute to, for our algorithm, corresponds to a single prediction. So, for each set of inputs, there will be a different SHAP value for each feature.

The calculation of the *SHAP values* is done through the marginal contribution of each feature to the final result. The marginal contribution is the difference between the model's prediction including a feature and excluding it.

To calculate the marginal contribution of each feature, the model will be executed 2^F times, where F is the number of features. In each execution, different combinations of features will be included, starting from zero and ending with all of them. The value of a feature not included in the execution will be the arithmetic mean. In our case, as all features are standardized, this value will always be 0.

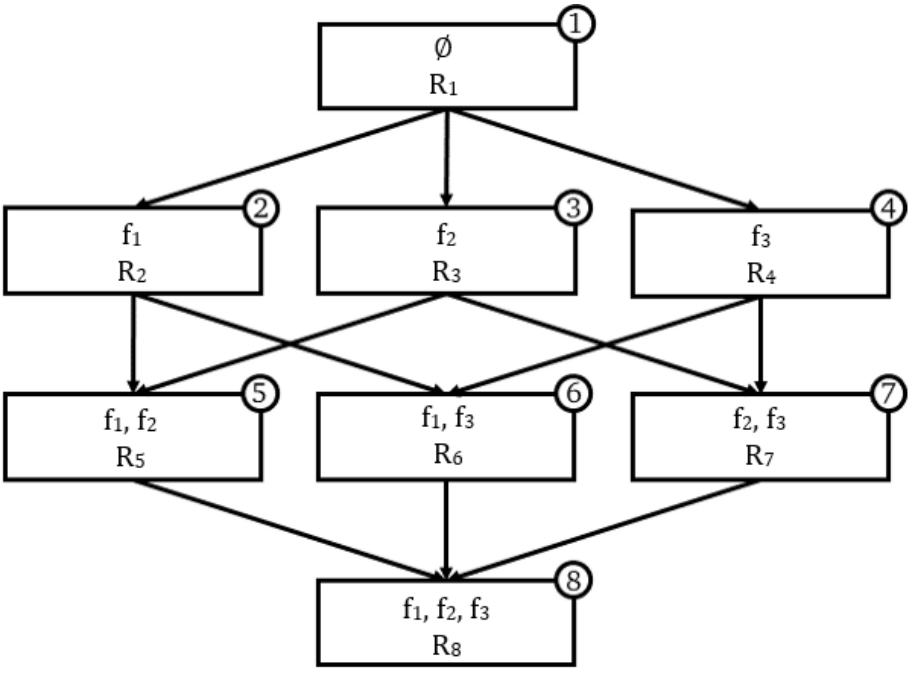

Figura 3.22: Model executions for calculating the marginal contribution of each feature[11]

Finally, we have an execution tree like the one in figure 3.22 for $F = 3$, where f is each feature and R is each result, obtaining $2^F = 2^3 = 8$ results. Each connection between the nodes represents the marginal contribution of a feature from one level to another. For example, at node 5, the marginal contribution

of feature 2 is calculated by the subtraction $CM_{f_2,\{f_1,f_2\}} = R_5 - R_2$, i.e., the difference between the model's prediction with feature 2 and without it.

To calculate the total marginal error and, therefore, the *SHAP value* of a feature, all its marginal contributions are summed, weighted:

$$SHAPvalue_{f_i} = w_1 \cdot CM_{f_i,\{f_i\}} + ... + w_{F-1} \cdot CM_{f_i,\{f_1,...,f_F\}} \tag{3.16}$$

Where f_i is the feature, w are the weights, and F is the total number of features. The weight values must sum to 1 and be equal for each level. Therefore, the weight is given by:

$$w_{level} = level \cdot \binom{F}{level}. \tag{3.17}$$

Finally, the calculation of a SHAP value for a feature for a model input is summarized as:

$$SHAP_{feature}(x) = \sum_{set:feature \in set} \left[|set| \cdot \binom{F}{|set|} \right]^{-1} [Prediction_{set}(x) - Prediction_{set-feature}(x)] \tag{3.18}$$

Where x is the input, *set* is the set of features, and F is the total number of features.

The sum of all the SHAP values for each feature will result in the model's prediction for the given sample.

For example, in figure 3.23, we can analyze the influence of each SHAP value of each feature on the final prediction. The blue bars represent negative SHAP values, which will reduce the life expectancy value, and the red bars represent those that will increase it to contribute to the value calculated by the model, in this case, 0.87 years in its standardized value.

Figura 3.23: Example of the influence of features on the final result using *SHAP values*

It is important to understand that SHAP values are unique for each prediction and depend on the rest of the features. Therefore, it is possible that for two identical input values for a single feature, their SHAP values could differ greatly, one influencing the result positively and the other negatively.

Performing this calculation when the number of features is extremely high is computationally impossible, as the model would need to be executed 2^F times just to explain the result for one sample. For our dataset, this would require 2^{87} executions, resulting in $1{,}5 \cdot 10^{26}$. However, the library used implements a series of approximations that allow for obtaining results. Moreover, it offers a powerful optimization specifically for neural networks.

3.7. Predictor

Finally, a predictor was created in a *Jupyter Notebook*, where the corresponding values for each indicator can be specified to obtain life expectancy through the neural network, as well as the explanation of each input feature's contribution using SHAP.

As a second option, the values for the indicators of a country in a specific year for a gender can be chosen, after which the desired factors can be modified to compare the real value, the original one calculated by the *machine learning* model, and the final result after the changes.

Capítulo 4

Results

The results obtained have been divided according to the different approaches covered in this work, including determining which country development indicators are the most influential for the calculation or outcome of life expectancy, the error obtained in the developed machine learning models, the optimization of the indicators to maximize life expectancy, and the interpretation of the model functions.

4.1. Preprocessed Dataset

After data processing, three datasets were obtained corresponding to three phases of the process flow.

- The first is the unified dataset with all the development indicators organized by country, year, and gender, obtained after the extraction, transformation, and loading process.
- The second resulting dataset was obtained by applying data cleaning to the previous set and eliminating and imputing missing values, resulting in a complete dataset.
- The third and final dataset obtained was the one corresponding to applying distribution transformations, standardization, and handling of categorical data to the complete data.

4.2. Importance According to Correlation

The correlation analysis between the independent variables and the target variable showed a high relationship between them. This initial analysis allowed us to clarify which variables are, a priori, the most relevant for determining life expectancy. We can observe the ten most correlated features in order in Table 4.1.

It is noteworthy that all these indicators, except for the fourth and the last, are related to health and hygiene issues. Therefore, we deduced that these types of indicators would play a fundamental role in the models to be implemented. On the other hand, we observed the high inverse correlation of the target variable with the birth rate, meaning that the more people are born, the lower the life expectancy. This will be another feature that required special attention.

4.3. Obtained Error

The difference between the value obtained by a machine learning model and the expected value is known as the error. However, the calculation of the error

Position	Feature	Correlation Coefficient
1	Basic Sanitation Services Total	0,817128
2	Basic Sanitation Services Urban	0,802336
3	Basic Drinking Water Services	0,800003
4	Income per Capita	0,798284
5	Basic Sanitation Services Rural	0,794671
6	Universal Health Care Coverage	0,794614
7	Poisoning Mortality Rate	−0,790273
8	Basic Hand Washing Rural	0,786186
9	Basic Hand Washing Total	0,782979
10	Birth Rate	−0,782901

Table 4.1: Top 10 features most correlated with life expectancy

can be implemented from several different approaches. For regression, as in our case, the most common error metrics applied to evaluate the developed models are:

Mean Absolute Error (MAE)

The Mean Absolute Error is the average of the errors made in absolute value. It is calculated using the following formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i| \tag{4.1}$$

Root Mean Squared Error (RMSE)

The RMSE is obtained by taking the square root of the squared difference between the expected and obtained value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2} \tag{4.2}$$

A particularity of RMSE is that, by applying the square of the error, it penalizes larger errors more. It is interesting to calculate the error using this technique when the goal is to increase the influence of large errors, in cases where these need to be avoided.

Coefficient of Determination (R^2)

The errors described above are error evaluation techniques whose range will vary depending on the problem at hand. For this reason, it is necessary to have

reference values to understand the performance of the regression models.

$$R^2 = 1 - \frac{\sigma^2_{error}}{\sigma^2_{output}} \tag{4.3}$$

The Coefficient of Determination, or Explained Variance Ratio, commonly known as R^2, calculates the error relative to the variance. The value of this error will range from $[0, 1]$, meaning that the lower the error, the closer the R^2 value will be to 1.

Due to the distribution transformation and standardization process, the error obtained for RMSE and MAE did not provide any meaningful information, since there was no reference value, which is necessary for interpreting these errors as previously mentioned. For this reason, to interpret the error, a de-standardization was applied, resulting in the error for RMSE and MAE in years.

Once the different implemented models were trained, they were evaluated by predicting the test set and calculating the error to assess the generalization ability and performance of the models with values that the model had not faced previously during training.

The error obtained by each model is shown in Table 4.2 for the training set and in Table 4.3 for the test set. The error values in years of life have been added to help interpret the results on the test set.

Model	RMSE	MAE	R^2
Linear Regression	0,2729	0,199	0,9247
KNN	$2{,}015 \cdot 10^{-8}$	$5{,}54164 \cdot 10^{-9}$	0,9999
Random Forest	0,0281	0,0172	0,9991
Neural Network	0,0276	0,0177	0,9978

Table 4.2: Error on the training set

Model	RMSE	MAE	R^2	RMSE in years	MAE in years
Linear Regression	0,2808	0,2030	0,924	2,8179	2,0367
KNN	0,1536	0,1038	0,9772	1,5414	1,0417
Random Forest	0,0706	0,0453	0,9951	0,7084	0,4551
Neural Network	0,0469	0,0294	0,9978	0,4709	0,2952

Table 4.3: Error on the test set

We can observe that the training error of the *KNN Regressor* model is extraordinarily low. However, the error obtained on the test set, using data the model has never seen before, is much higher than the training error. This means that the model is overfitting on the training data, thereby losing its generalization capability.

Overfitting is defined as the behavior of a model that learns the training data excessively, obtaining a very precise value for this data but being unable

to make good predictions for new data since it lacks generalization capability, having over-adjusted to the given training data.

This is a relatively common behavior in this type of model, as it is instance-based, meaning it memorizes the training set. However, despite its overfitting, the model performs better than linear regression.

On the other hand, something similar happens with the results from the neural network and the random forest regressor. While the *Random Forest Regressor* obtains slightly better results for the training set, the neural network outperforms it significantly on the test set. This helps us understand that the generalization capability of the neural network is superior to that of the random forest regressor.

Finally, it is concluded that, as initially anticipated, the more complex model, the neural network, has obtained the best results.

4.4. Error Analysis

By using the different error metrics, we can get an idea of the error we can assume when using the model. However, it is very interesting to study the error to discern which values tend to fail more. This will allow us to predict the probability of error on a sample to be predicted.

Since the best results were obtained with the neural network, the study has only been applied to the predictions from this model.

For this analysis, the error has been calculated as the difference between the expected value and the value obtained by the neural network.

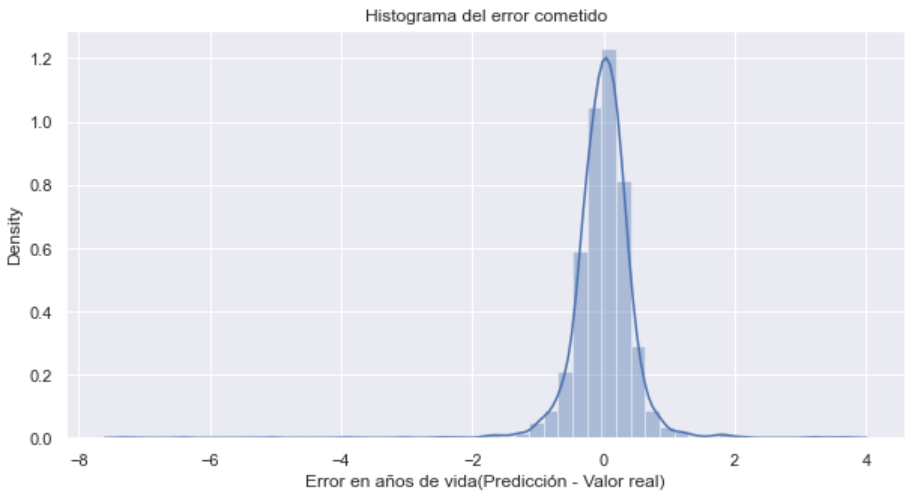

Figura 4.1: Distribution of the obtained error

Figure 4.1 shows the distribution of the error. This is divisible into two types.

The **overestimation error** is when the calculated value is greater than the obtained value, while the **underestimation error** is the opposite, errors where the obtained value is lower than expected.

According to the graph in figure 4.1, positive errors are classified as overestimation, and negative values as underestimation.

To analyze the most significant errors, the 2.5 % of the highest errors from both overestimation and underestimation have been selected using quantiles and labeled accordingly, leaving the rest as *Middle*.

- **Average error** due to overestimation in years: 1.2760 years
- The **largest error** due to overestimation in years: 3.7217 years
- **Quantile 0.975** to determine overestimation: 0.7938 years
- **Average error** due to underestimation in years: -1.4538 years
- The **largest error** due to underestimation in years: -7.3143 years
- **Quantile 0.025** to determine underestimation: -0.8443 years

By comparing figure 4.2, it is observed that, while the highest overestimation errors occur for life expectancies between 60 and 65 years, the most notable underestimation errors occur for low life expectancies.

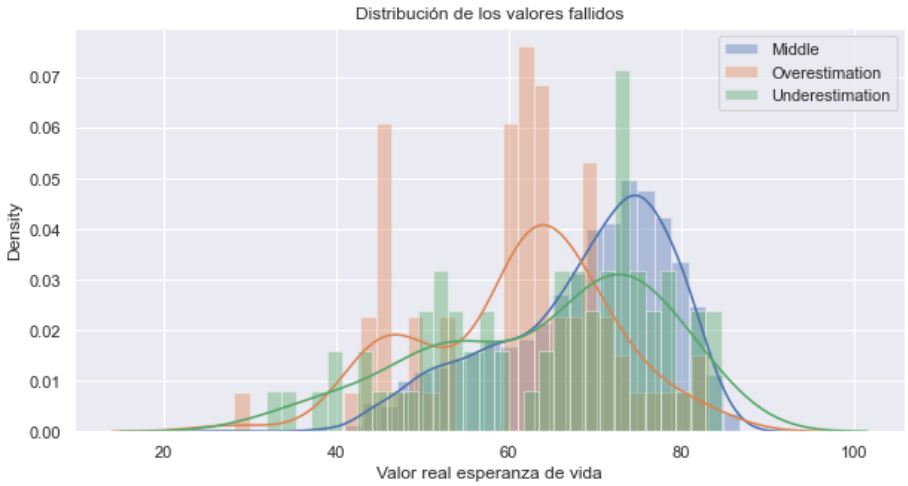

Figura 4.2: Distribution of actual life expectancy by error type

Next, each type of error has been analyzed separately to determine the features that characterize the records with the highest error rates.

Overestimation

By subtracting the mean of each feature from the entire set with the set of records that produce overestimations, it was detected that the feature *Diet Composition Oils And Fats* exceeds the mean by 0.705614 and *Low CI Value % Death Cardiovascular* is 0.649977 below the mean.

From the graphs in figure 4.2 and 4.3, we can conclude that the model tends to overestimate when the value to predict is very low, around 45-55 or 60-75 years of life, and the value of *Diet Composition Oils And Fats* is lower than the average and/or when the value of *(Low CI Value) % Death Cardiovascular* is higher than the average.

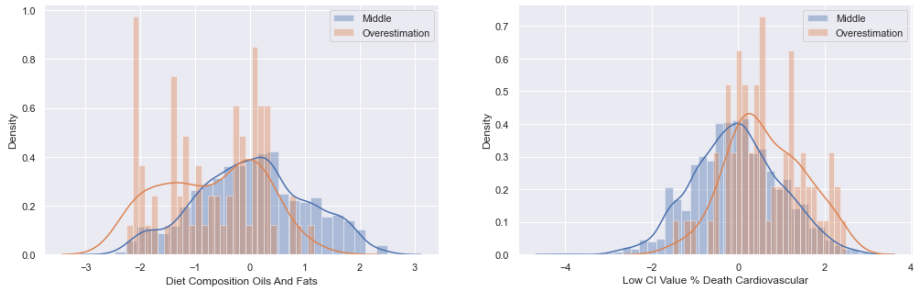

Figura 4.3: Distributions of *Diet Composition Oils And Fats* and *Low CI Value % Death Cardiovascular* for overestimated and non-overestimated values

Underestimation

The features that deviate most above the mean in underestimation are *Conflict and Terrorism Deaths %* and *Diet Calories Fat*.

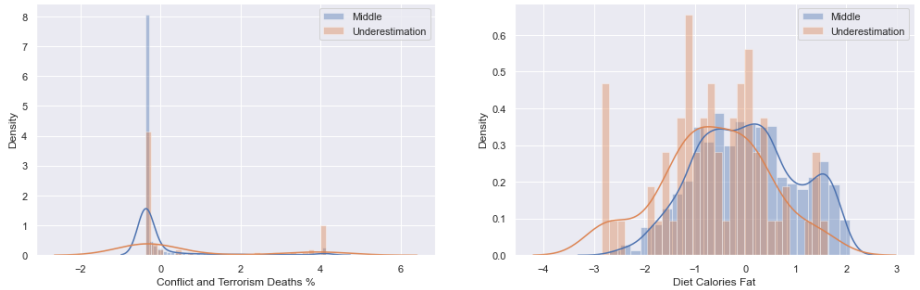

Figura 4.4: Distributions of *Conflict and Terrorism Deaths %* and *Diet Calories Fat* for underestimated and non-underestimated values

After analyzing the graphs in figures 4.2 and 4.4, it was concluded that the model tends to underestimate when the value to predict is very low, when *Conflict and Terrorism Deaths %* is very high, and/or when *Diet Calories Fat* is very low.

Error Probability

After studying which input values tend to cause the highest error in the model's prediction, an analysis was conducted to observe the probability of what predictions might be more wrong.

By studying the graph in figure 4.5, the following conclusions were drawn:

- The probability of high error by overestimation will be higher if the prediction is between 45-50 years, 60-65, or very low values.
- The probability of high error by underestimation will be higher if the prediction is a low value (less than 40 years).
- We can be more certain that the prediction is more accurate when it is greater than 70 years.

Figura 4.5: Distribution of life expectancy prediction by error type

4.5. Interpretación del comportamiento de los modelos

Una vez construidos y evaluados los modelos de *machine learning*, se ha llevado a cabo un análisis de interpretación de comportamiento de los modelos. Este se ha aplicado según diferentes enfoques dependiendo del modelo. El principal objetivo era determinar qué indicadores de desarrollo son los que más influencian en el cálculo de la esperanza de vida del país.

4.5.1. Interpretación de Random Forest Regressor

El análisis sobre *Random Forest*, se ha realizado mediante el atributo otorgado por este mismo modelo que provee la importancia de cada feature. La importancia de cada feature viene establecida por la media de la capacidad de disminución la impureza en la división de un nodo del árbol, es decir, cuánto reduce la varianza de la solución dicha feature al ser usada como nodo en el árbol de decisión. Este coeficiente, será un valor entre 0 y 1, teniendo en cuenta que la suma de las importancias de todas las features resultarán en 1. Por tanto, es una importancia relativa.

Primera iteración

En este trabajo se ha realizado una primera iteración, obteniendo unos resultados de error aceptables. No obstante, se estudió la importancia de las features a partir de la explicación que ofrece *Random Forest*. La importancia de las features según este enfoque que explicaron la toma de decisión de selección de features(3.2) fueron las especificadas en la figura 4.4.

Posición	Feature	Importancia
1	High CI Value Under 5 Mortality Rate	0,632327
2	Death Rate	0,138211
3	Low CI Value Under 5 Mortality Rate	0,087504
6	% Population Aged 80+	0,010222
8	Infant Mortality Rate	0,006056
13	% Population Aged 65+	0,004455
14	High CI Value Infant Mortality Rate	0,003856

Table 4.4: Features más importantes para la primera iteración

Segunda iteración

En la segunda iteración del proyecto se volvió a estudiar la importancia de las features tras la aplicación de *Random Forest*. Los resultados obtenidos fueron los descritos en el gráfico de la figura 4.6 para las features más significativas.

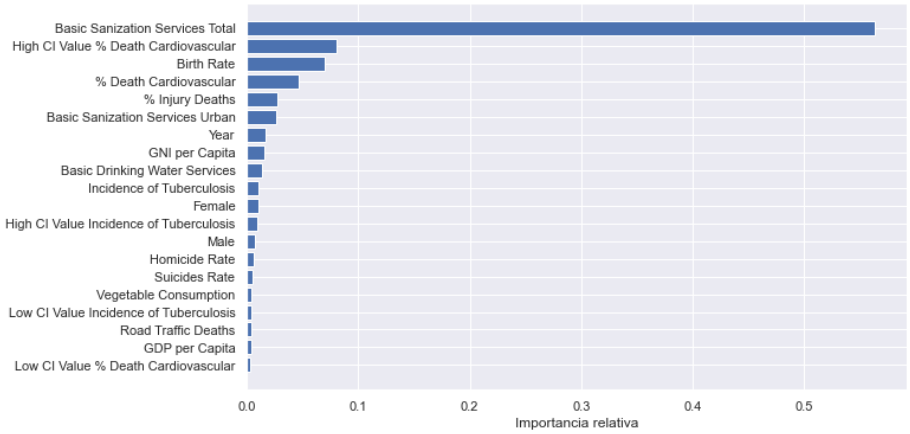

Figura 4.6: Importancia relativa de las features según *Random Forest*

Como podemos apreciar en dicho gráfico, la feature más importante para determinar la esperanza de vida, destacando ampliamente respecto al resto es *Basic Sanization Services Total*. Esto significa que este indicador de desarrollo será el que más distinga los casos, separándolos en dos grupos con una mayor reducción de la varianza de la esperanza de vida de dichas situaciones.

A continuación, destaca la importancia la probabilidad de morir por enfermedades cardiovasculares, la tasa de natalidad, el porcentaje de muertes por heridas, etc.

No obstante, este primer análisis de importancia de indicadores solo nos permite un limitado vistazo sobre el peso relativo de cada indicador, puesto que solo sabremos cual será más importante para el cálculo, pero no cómo afecta al resultado final. Es decir, mediante este análisis no podremos conocer que

impacto supone el aumento o disminución de los valores de estos indicadores que destacan sobre el resto.

Además, la importancia calculada mediante esta estrategia no es perfecta, puesto que da problemas para aquellas features que estén correladas, seleccionando una y renegando la importancia de la segunda, pudiendo inducir a conclusiones erróneas[56]. Por este motivo se ha decidido interpretar la importancia mediante otras estrategias, aunque teniendo en cuenta estos resultados para futuros análisis.

4.5.2. Interpretación de la Red de Neuronas

Las redes de neuronas son conocidas por su comportamiento de **caja negra**, es decir, son modelos que no explican su comportamiento, sino que sencillamente disponiendo de unas entradas podemos conocer su salida. La interpretación de este modelo es la más importante puesto que es el modelo que, como se expuso anteriormente, obtenía mejores resultados gracias a una mayor capacidad de generalización. Por tanto, el entender su comportamiento y la importancia relativa que da a cada indicador ha supuesto un reto, para el cual se han afrontado tres enfoques diferentes.

4.5.2.1. Wrapper Methods

Mediante los *wrapper methods* se ha analizado la evolución del error con un número incremental o decremental de indicadores siguiendo dos enfoques.

Backward Feature Selection

Se ha entrenado y validado la red de neuronas mediante la técnica de *backward feature selection* siguiendo el orden de eliminación de menos a más correlación con la esperanza de vida. Obteniendo así 87 redes de neuronas con un error incremental.

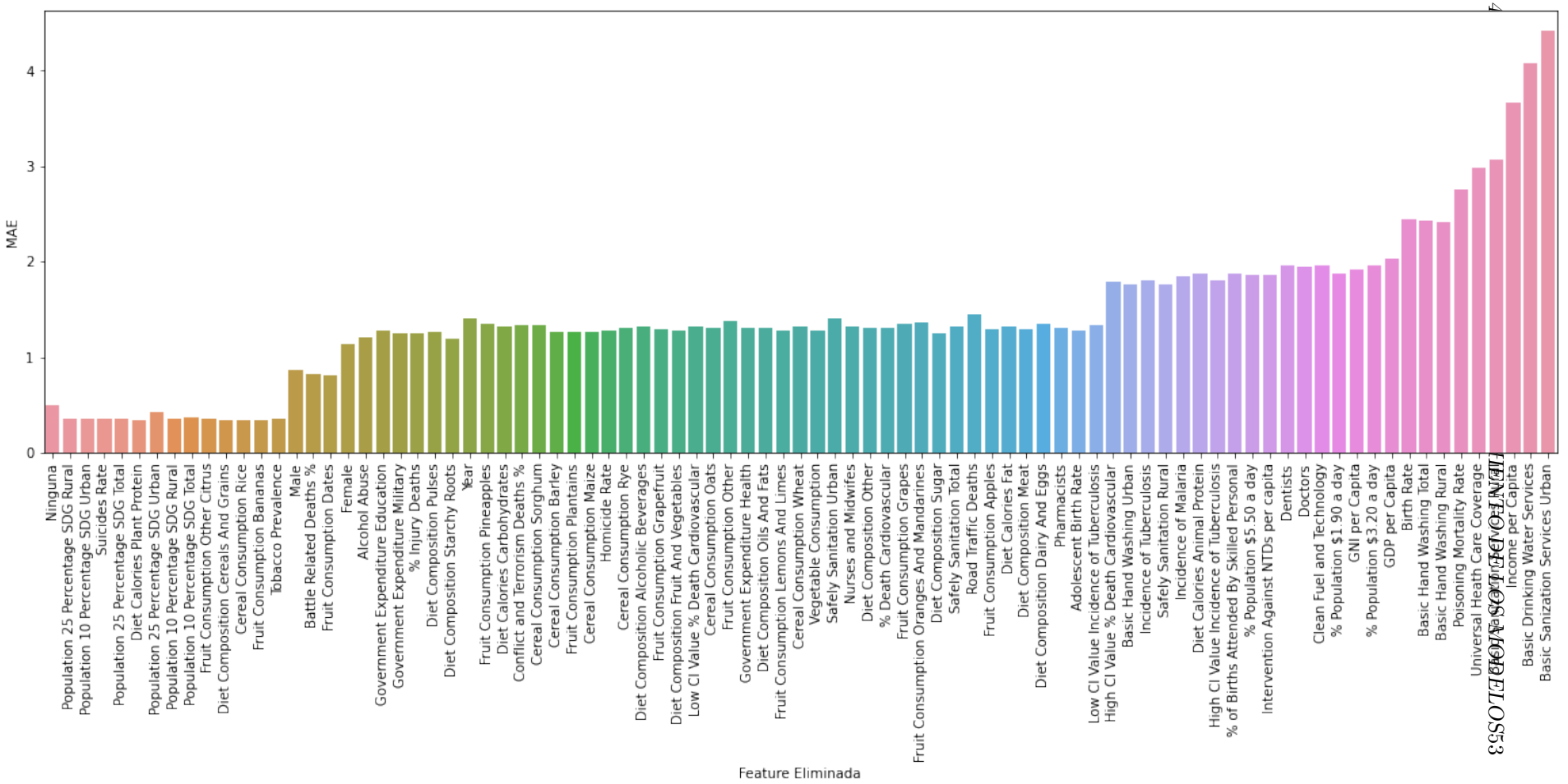

Figura 4.7: Evolución del MAE durante el *backward feature selection*

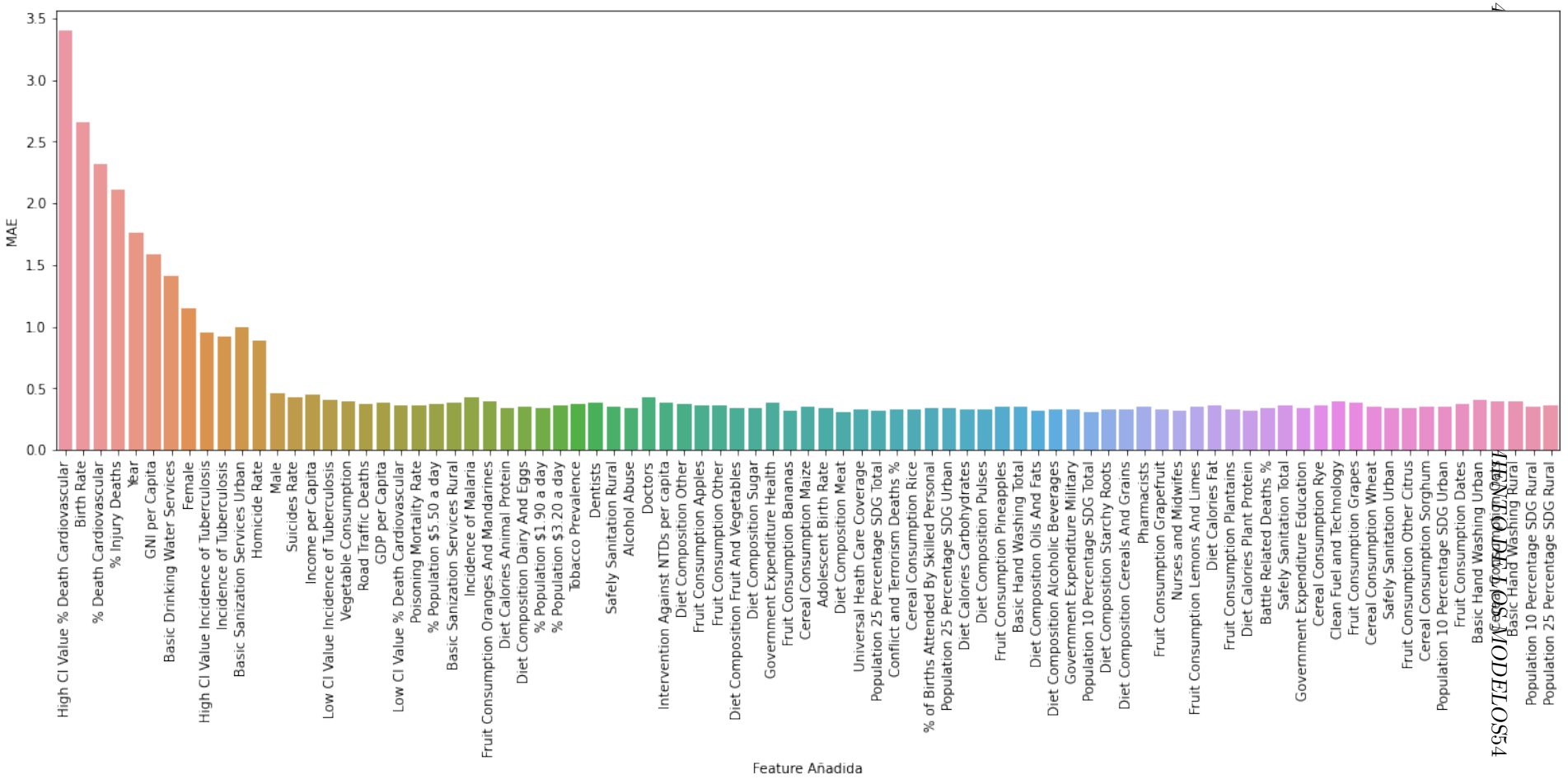

Figura 4.8: Evolución del MAE durante el *forward feature selection*

Se ha analizado en más detalle el error absoluto medio(MAE), como se muestra en la figura 4.7, donde se dibuja el error por cada feature eliminada, llegando a la conclusión de que los indicadores eliminados que más han aumentado el error y por tanto, más importantes son:

- Male
- Female
- High CI Value % Deaths Cardiovascular
- Birth Rate
- Poisoning Mortality Rate
- Universal Health Care Coverage
- Basic Sanization Services Rural
- Income per cápita
- Basic Drinking Water Services
- Basic Sanization Service Urban

Forward Feature Selection

El orden elegido para ir añadiendo cada feature ha sido la importancia según el modelo de *Random Forest Regressor*. En este caso la evolución del error para cada nueva red neuronal creada irá en decremento.

Tras el análisis de la evolución del MAE de la figura 4.8, se ha concluido que las features más relevantes según esta técnica de selección de features son:

- High CI Value % Death Cardiovascular
- Birth Rate
- % Death Cardiovascular
- % Injury Deaths
- Year
- GNI per cápita
- Basic Drinking Water Services
- Female
- Male

As we can observe, both strategies yield common results, highlighting, for example, the importance of the probability of dying from cardiovascular diseases and gender. However, the influence of the order in which subsets are created is very apparent in these decisions. Additionally, due to the high execution time required to apply these techniques, the implemented neural network has been trained with 100 *epochs*, although better results could have been achieved with

a different number of iterations (either more in case of *underfitting* or fewer in case of *overfitting*). Lastly, it is important to note that cross-validation has not been applied in these trainings, meaning that small increases or decreases in error are not interpretable. For these reasons, it was decided to study the neural network with other approaches.

However, this approach has helped us gain a preliminary view of the importance of the features, allowing us to highlight and predict the weight of some of them on the implemented neural network.

4.5.2.2. Genetic Algorithm

Due to the slow execution of the optimization, this strategy has been applied to only three cases, where life expectancy is low, medium, and high. For all cases, a maneuvering margin (*margin_input*) of 5 has been chosen. This value will allow the solution to differ from the original state enough to produce a significant increase in life expectancy.

Case of low life expectancy

The first chosen case, corresponding to low life expectancy, was Afghanistan in 1990 for both genders. After more than 12,500 generations, the best result obtained was an increase in life expectancy of 4.09 years, from the original 50.3 to 54.4 according to the neural network.

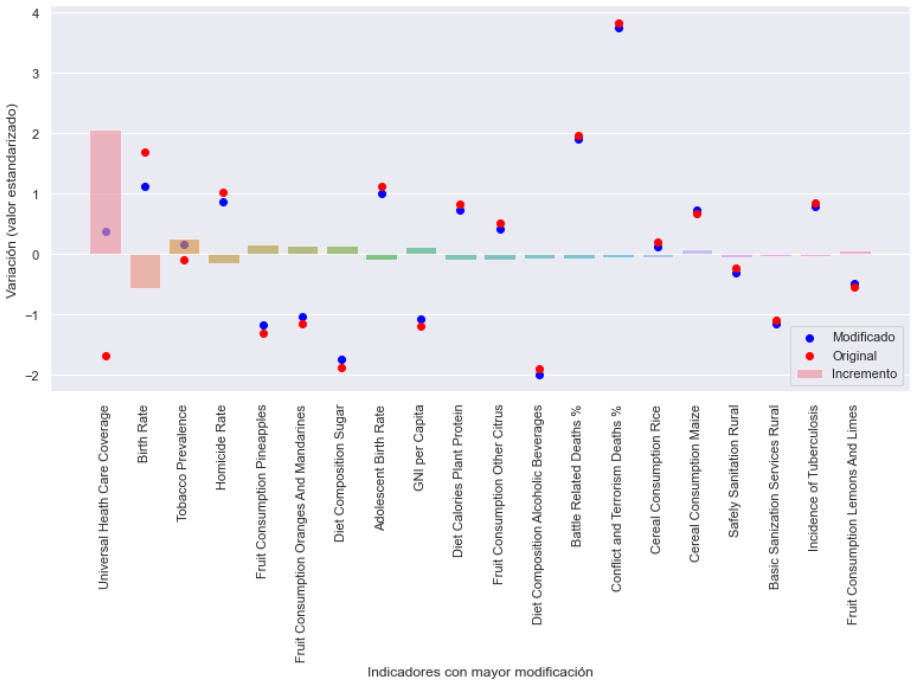

Figura 4.9: Modifications to the indicators of Afghanistan in 1990 for both genders to maximize life expectancy

The modification made to the indicators, with standardized values, is represented

in figure 4.9. The bars represent how much the value has been modified, the red points mark the original value of the feature, and the blue points mark the final value.

For this specific case, the indicator whose modification maximizes life expectancy, with a highly differentiating weight, is *Universal Health Care Coverage*, that is, access to basic health services. This value was originally well below the average (zero), and it was increased above it.

The next two most modified indicators were the birth rate and tobacco use. Contrary to what might be expected, this solution proposes a decrease in the birth rate and an increase in tobacco consumption to increase life expectancy.

This solution thus indicates that in countries with lower birth rates and higher tobacco consumption, there is a higher life expectancy. Therefore, our neural network considers these two factors to increase it. This means that our dataset is correlated with the target variable. However, we discover that correlation does not imply causality, meaning that just because two things are correlated (like in this case), it does not mean one causes the other.

Case of medium life expectancy

The second case chosen to apply the genetic algorithm to maximize life expectancy was Lithuania in 2005 for the male gender.

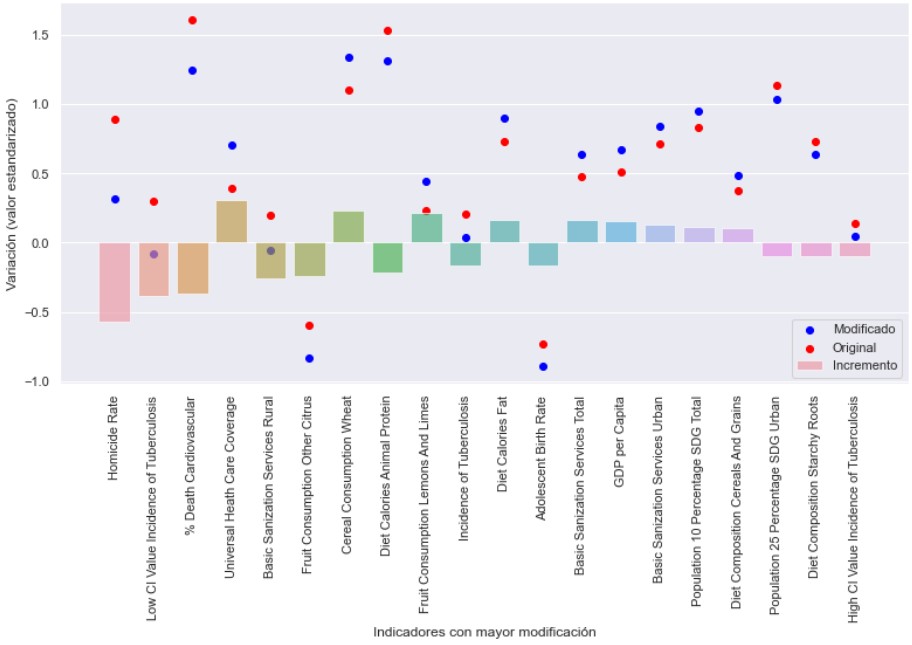

Figura 4.10: Modifications to the indicators of Lithuania in 2005 for the male gender to maximize life expectancy

In this situation, a life expectancy increase of 3.4 years was achieved, from 65.6 to 69.

In this case, shown in figure 4.10, the most notable indicator to modify was the homicide rate, which decreased from a point initially well above the average.

A similar situation occurred for the following indicators: tuberculosis incidence and the probability of dying from cardiovascular diseases.

Case of high life expectancy

The third and last case applied was the one closest to our situation, that is, Spain in 2019 for both genders. Starting from a life expectancy of 83.4 years, the changes led to 87.1, increasing it by 3.63 years.

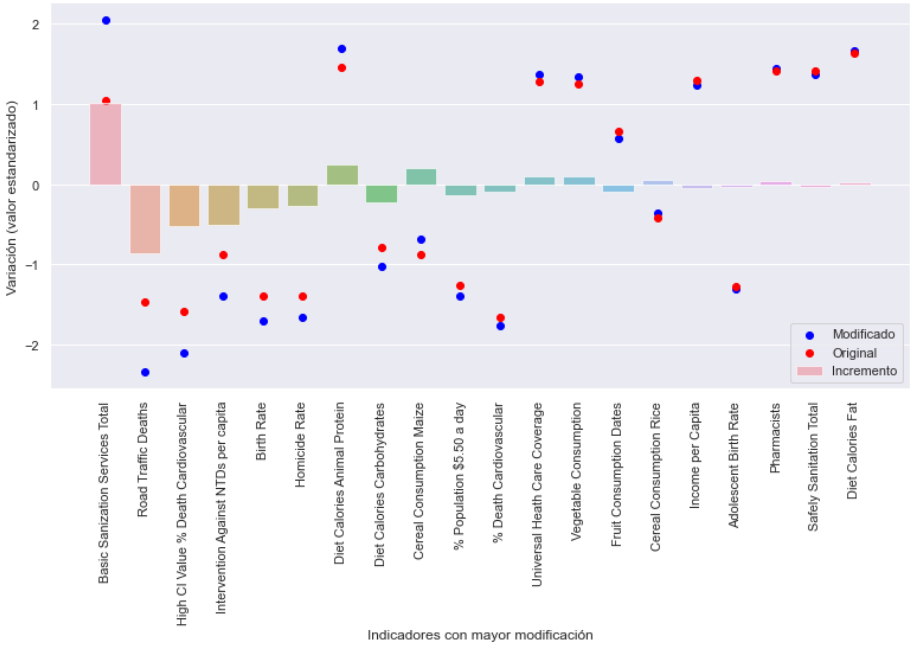

Figura 4.11: Modifications to the indicators of Spain in 2019 for both genders to maximize life expectancy

Figure 4.11 shows that for countries with high life expectancy, the feature requiring the most modification is, curiously, *Basic Sanitation Services*, which started from a value notably above the average. This indicates that this value is directly proportional to life expectancy, meaning that the more this indicator increases, the higher the life expectancy will be. The opposite happens with the features that follow: traffic deaths, probability of dying from cardiovascular diseases, number of interventions for neglected tropical diseases, birth rate, and homicide rate. All of these start from a value much below the average, and the lower they are, the more they maximize life expectancy.

Although the individual analysis has focused on the most important features, it is necessary to highlight the importance of food-related indicators, which, although not at the top, are still very present in the changes made.

From the results obtained in the different cases, we interpret that the operation and decision-making of the model rely heavily on cases with high life expectancy, seeking to maximize all the values of their features (we observe that the initial points are always farther from the average), while for those cases with low life expectancy, the values tend to be brought closer to the average.

One advantage of this approach for model interpretation is that it not only shows us the features that have the most influence on the studied case but also helps us understand what changes need to be promoted to increase life expectancy. However, it has two major disadvantages: one is the limit of some features, meaning that those related to percentages will have a maximum and minimum threshold, so the genetic algorithm would need to be modified to penalize exceeding those limits. The second weakness is the model itself, as mentioned earlier, correlation does not imply causality, and characteristics of countries, such as a low birth rate, will not necessarily lead to an increase in life expectancy.

4.5.2.3. Shapley Additive Explanations (SHAP)

Using this technique, various graphs have been obtained to understand the influence of the features on the results given by the neural network, both at a general level and in specific cases.

General Analysis

In figure 4.12, the 15 most influential features on the model's prediction are shown, ordered from highest to lowest. The influence of the features is calculated by the mean of the absolute values of their *SHAP values*.

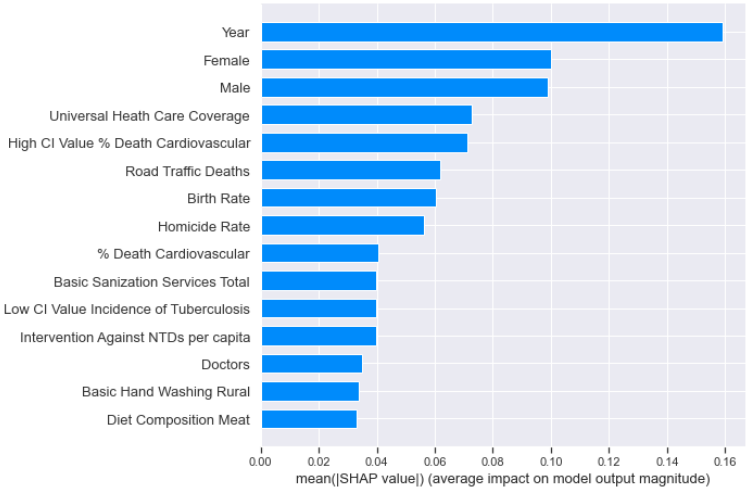

Figura 4.12: Average contribution of each feature to the final result

According to this black-box model interpretation strategy, the features whose values have the most impact on average are year and gender, with a large difference from the others. These are two features that we can classify as non-modifiable, meaning they cannot be altered with the goal of increasing life expectancy. However, they are highly necessary for calculating life expectancy, as the SHAP values clearly indicate.

The other most determining indicators are related to various areas such as healthcare, birth rate, accidents, safety, and food. We will analyze their behavior later.

To interpret the SHAP values more clearly, it is important to consider that they add and subtract from the final value. However, these figures are not interpretable because the neural network's output is standardized. To interpret both figure 4.12 and the others, we can perform the conversion shown in figure 4.13, where the standardized and unstandardized life expectancy values are compared.

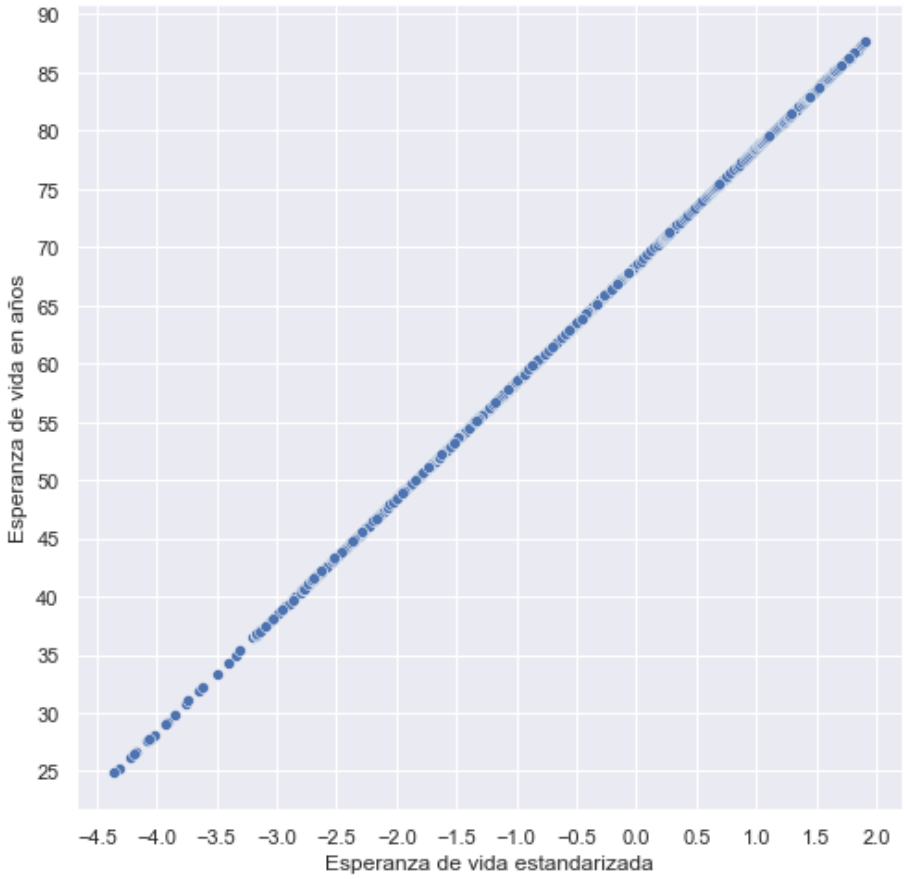

Figura 4.13: Comparison of standardized and unstandardized life expectancy values

Once this conversion is presented, it is worth noting that the feature with the greatest influence on the data has an average of 0.16 standardized years, which is approximately 1.6 years. This means that the influence of a single feature on the final prediction will not be excessively significant; instead, it will be the sum of all the indicators that will gradually determine the result.

The graph that provides the best insight for interpreting the behavior and influence of each feature on the neural network's predictions is shown in figures 4.14 and 4.15. These are a set of violin plots, one for each feature, displaying the *SHAP values* for different samples of a feature. The vertical thickness of the figure indicates the number of cases with that SHAP value for that feature. Lastly, the color indicates the original value of that feature, being bluer for a

lower value and redder for a higher value.

For example, the graph shows that for *Year*, a high value of the feature will result in a high SHAP value, increasing the life expectancy outcome. The opposite happens for a low value of the same feature.

Through this graph, we can understand the positive or negative influence of increasing or decreasing a particular indicator. In most cases, the behavior is either directly or inversely proportional, meaning a higher feature value will correspond to a higher (or lower, in the case of an inverse relationship) SHAP value. However, the SHAP value of a feature depends on the values of the other features. Therefore, some features' influence will vary positively or negatively depending on the values of the others. We can observe this behavior, for example, in *Vegetable Consumption*.

Finally, for many features, we observe that the value corresponding to the mean (where the blue and red colors meet in purple) tends to reduce the prediction value.

Individual Feature Analysis

To analyze the progression of the SHAP value for a specific feature as its input value varies, in addition to the correlation and influence of some features on others in the calculation of this value, what are known as dependence plots have been used.

In dependence plots, we can observe on the axes the original input value of the feature and its corresponding SHAP value. Additionally, we can see how they vary depending on the value of another feature through the color of the points. This feature is automatically chosen by the algorithm, selecting the one that most influences (or depends on) the analyzed feature.

We analyze the behavior of the most relevant feature according to this strategy, *Year*, in the dependence plot in Figure 4.16. This image confirms what was mentioned earlier: the higher the year, the greater the positive influence on life expectancy and vice versa. However, the SHAP values vary even with the same input, due to the influence of the other inputs. The feature that *Year* most depends on when establishing its SHAP values is *% of Births Attended By Skilled Personal*. The red color on the points represents a higher value for this feature, and gradually, the blue represents the opposite.

Therefore, it is concluded that the year will be a much more determining factor (since it will have a higher *SHAP value* in absolute value) for those cases where the percentage of births attended by skilled personnel is lower. This means that the year will have much more influence on the final outcome in countries with poorer healthcare.

For other features, on the other hand, the positive or negative influence on the final result does not follow a direct logic on the input value, as in the previous case. Instead, they depend much more on the values of the other features. This is the case for the *Population 10 Percentage SDG Total* feature, i.e., the percentage of the population whose healthcare spending exceeds 10 % of their income. In its dependence plot in Figure 4.17, it can be seen that the same input value for this feature can influence life expectancy both positively and negatively. This difference is directly related to the value of interventions against neglected tropical diseases per capita.

Figura 4.14: Violin plots of the *SHAP values* for each feature (Part 1)

Figura 4.15: Violin plots of the *SHAP values* for each feature (Part 2)

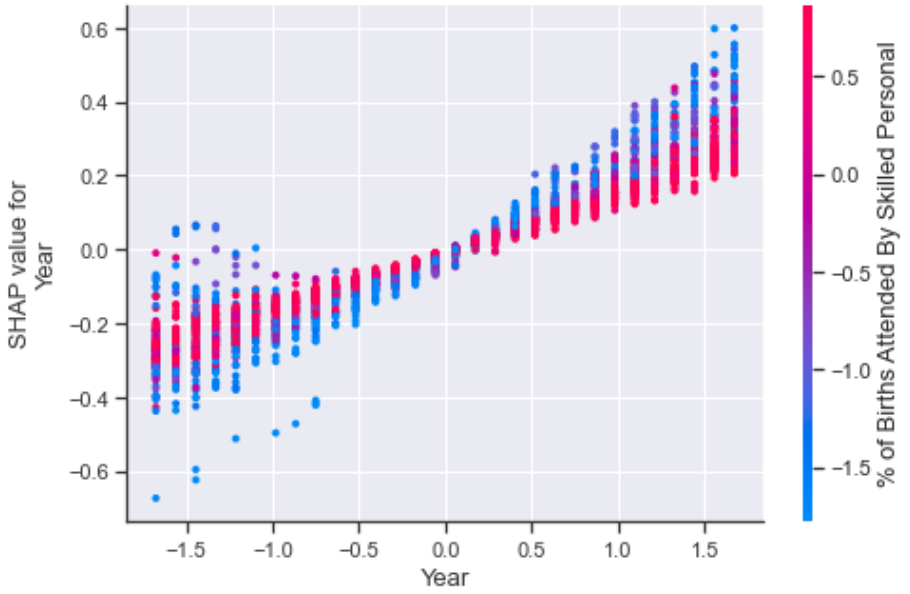

Figura 4.16: Dependence plot for the *Year* feature

High values of *Population 10 Percentage SDG Total* will have a positive influence on the result if *Interventions Against NTDs per Capita* is low, and a negative influence when the value is high. Conversely, low values will influence positively if per capita interventions are high and vice versa.

Thanks to these types of graphs, unusual or unexpected behaviors have been detected, as happens with the *Income per Capita* feature. It can be intuited that higher income will contribute more positively to life expectancy. However, as seen in its dependence diagram, presented in Figure 4.18, these expectations do not hold.

Instead, low values of *Income per Capita* positively influence the model, while high values may slightly decrease the calculated life expectancy value. Additionally, we observe that this contrary influence on the final result becomes more extreme depending on the value of dairy and egg consumption.

Individual Case Analysis

The best functionality provided by this strategy for model analysis and interpretation is the explanation of individual cases. For a specific case, it is possible to see how and to what extent each feature contributes to the final result.

For example, we analyze the case of Spain in 2005 for both genders. The life expectancy in this year is 80.553, the neural network gives a very similar result, 80.603 years, which standardized is 1.21. In the graph shown as Figure 4.19, we can appreciate the influence of the most relevant features in the calculation. We can highlight the high positive influence of *Road Traffic Deaths* and the probability of dying from cardiovascular diseases on the result.

To visualize the contribution of each feature to the prediction, we use waterfall

Figura 4.17: Dependence plot for the *Population 10 Percentage SDG Total* feature

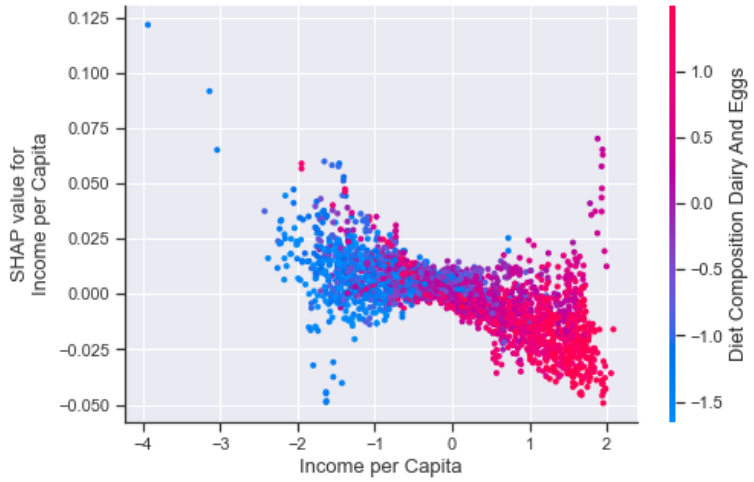

Figura 4.18: Dependence plot for the *Income per Capita* feature

charts, like the one in Figure 4.20, which shows each feature individually with its SHAP value. In the waterfall chart, starting from the mean result value (in our dataset, 0 due to standardization), each feature contributes its *SHAP value* until reaching the value obtained by the neural network. In this case, the prediction for Afghanistan in 1990 for the female gender is studied, where the real life expectancy is 51.442 years, and the model's calculated life expectancy is 51.573 years.

For this case, we can see the participation of the feature *Conflict and Terrorism Deaths %*, which plays a negative role in the final result. It is remarkable that, despite the high number of features that make up the input, almost all of them contribute to the output, either more or less influentially.

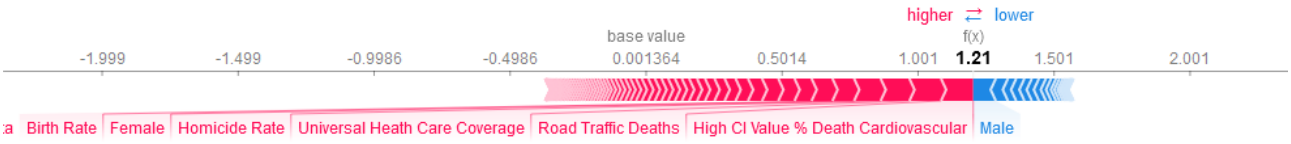

Figura 4.19: Explanation of the influence of each feature on the final prediction of the neural network for the case of Spain in 2005 for both genders

With these graphs, we can explain exactly how our model makes decisions to determine an output.

Case Comparisons

To compare several cases, the decision plot is very interesting. This graph shows, as a vertical line, the progress of the calculated value after applying the SHAP value feature by feature, starting from zero until reaching the obtained result.

In the decision diagram of Figure 4.21, we can observe the contribution of some input features on the life expectancy predictions for Algeria with the **year progression**. The features are ordered from least to greatest deviation in the SHAP values to observe which features have the greatest difference and how much they contribute.

It is noticeable that, for this case, the most significant factors in the prediction from year to year, aside from the most commonly influential features, are *Population 25 Percentage SDG Rural*, *Diet Calories Plant Protein*, and *Diet Composition Alcoholic Beverages*, among others visible in the decision plot.

Next, the model's behavior was studied over **different groups divided by life expectancy**. Three groups were formed: one with low life expectancy, another medium, and another high, to see which features differentiate these groups.

The groups were selected based on defined life expectancy ranges for a single year for both genders so that these last unchanging features do not dominate the study graphs. The range values affect the actual value, not the predicted value by the neural network.

For **countries with high life expectancy**, a range of more than 82 years for 2015 was chosen. The results of the features with the most disparate *SHAP values* are shown in the decision plot in Figure 4.22.

In this graph, we can observe the high influence of initially less relevant features such as date consumption, tuberculosis incidence, or calories consumed from animal protein.

In the case of **cases with an average life expectancy**, the range of 66 to 68 years in the year 2000 was chosen. In these situations, there is a much larger disparity of origin, as can be seen in figure 4.23. The contribution of the features to the final result is much greater and more relevant. These features are *Basic Sanitation Services, Doctors, % Deaths Cardiovascular, Universal Health Care Coverage, Interventions Against NTDs, Homicide Rate*, and *Birth Rate*.

Finally, we analyze the group of **cases with low life expectancy**, placing

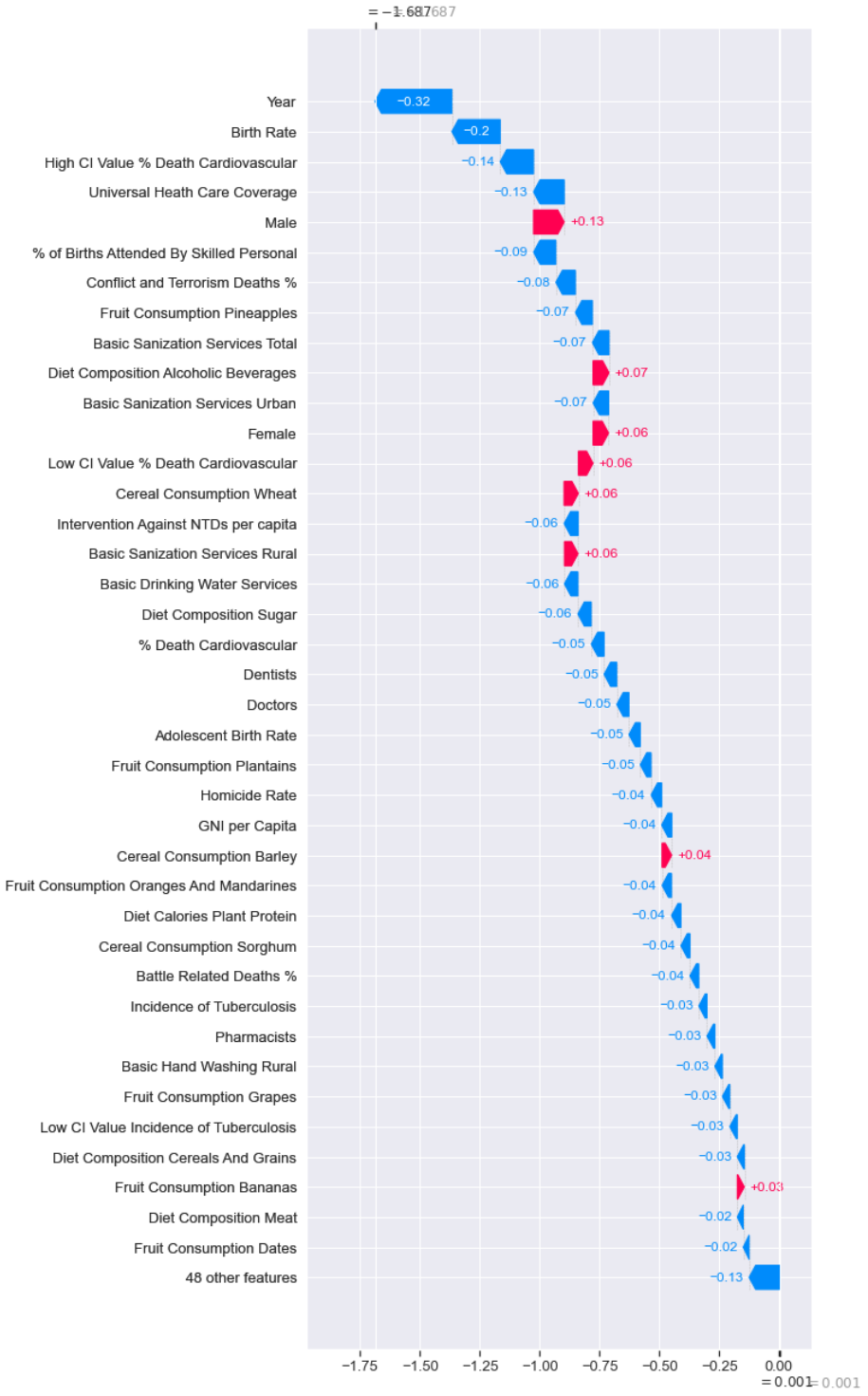

Figura 4.20: Explanation of the influence of each feature on the final prediction of the neural network in a waterfall chart for the case of Afghanistan in 1990 for the female gender

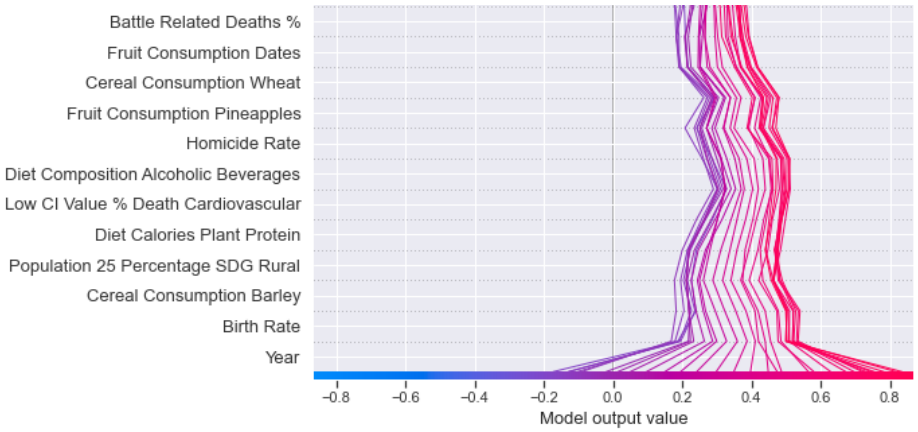

Figura 4.21: Decision plot comparing life expectancy predictions for Algeria for both genders year by year, from 1990 to 2019

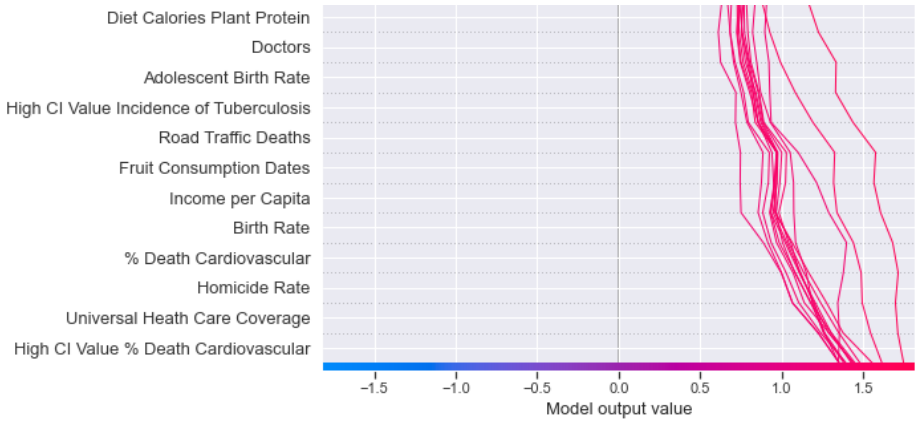

Figura 4.22: Decision plot for countries with life expectancy greater than 82 years in 2015

the range as below 50 years for the year 1990. In this case, we can also observe the most outlier values, allowing us to determine the origin of this behavior. Based on figure 4.24, it was concluded that the indicators most affecting the creation of these outliers are mainly the percentage of deaths due to conflicts and terrorism, followed by food consumption.

Finally, through the analysis with SHAP, we can conclude that, although the features that most affect the result are those present in figure 4.12, and their effect on the final result is observable in figures 4.14 and 4.15, in the end, in each particular case, certain features will have a greater influence, especially depending on the life expectancy predicted by our neural network.

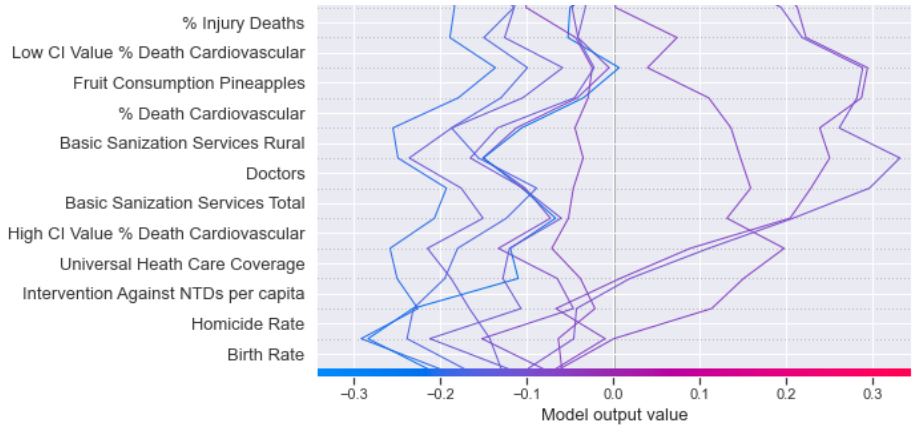

Figura 4.23: Decision graph for countries with a life expectancy between 66 and 68 years in the year 2000

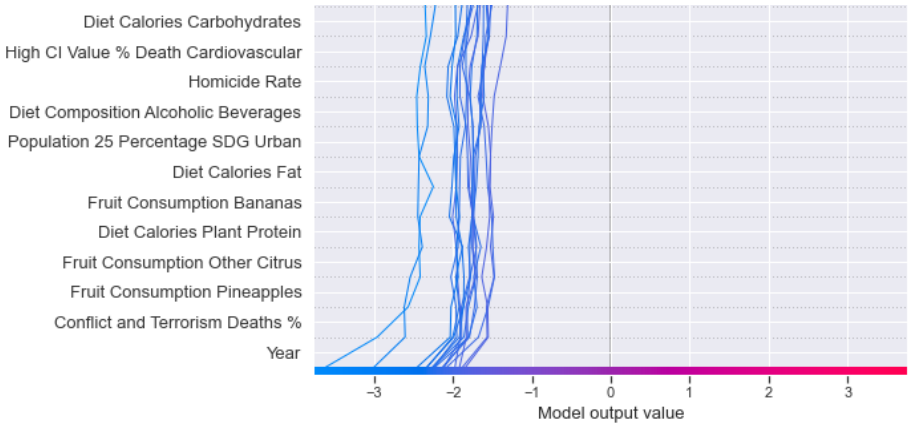

Figura 4.24: Decision graph for countries with a life expectancy lower than 50 years in 1990

4.6. Problems encountered

Throughout the preparation of this work, several obstacles have been encountered that needed to be overcome. To address these, various decisions and approaches have been adopted to find solutions.

Unified dataset

Since data was extracted from various sources, we did not have a single dataset but rather multiple files containing the desired information. Therefore, an ETL process had to be carried out to unify all the data into a single dataset that contained development indicators organized by country and year. Additionally, since certain indicators refer to gender, it was necessary to organize the data by this category, resulting in a dataset with a row for each country, year, and gender.

Data quality

In addition to the necessary analysis and cleaning of erroneous data required for the dataset, the main problem encountered with the dataset was the unknown values. The amount of empty data was overwhelmingly high. Therefore, a plan was devised to address this issue. The plan relied on techniques for imputing unknown values, such as interpolation and KNN, as well as eliminating both rows and indicators.

Feature selection

After applying an initial iteration with machine learning models on the original dataset, certain indicators were found to be very important. However, after analyzing them, it was concluded that their calculation was too closely related to life expectancy, not being causal factors but rather caused by life expectancy. Therefore, the decision was made to remove a set of indicators from the dataset, while extracting and adding another group that was more aligned with the project's objectives.

Model analysis

The machine learning model that provided the best results was the neural network. However, this model works as a black box, meaning that it is not possible to understand the decision-making process that was used to obtain the calculated result. Three solutions were proposed to this problem, three strategies for interpreting black-box models: *wrapper methods*, genetic algorithm, and SHAP.

Computational limitation

Computational limitation was a recurring problem in this project for the analysis of the neural network. The *wrapper methods* approach was highly limited by the order of the features for its execution, while the optimal solution would have been to test all possible feature combinations or at least all possible combinations of 10 features. Both cases were completely unfeasible due to execution time. Therefore, two approaches were chosen for ordering the features in the application of this strategy: by correlation with the target variable and based on the results from *Random Forest*.

We encountered this problem again when applying the genetic algorithm. Due to its functioning, during the execution of this machine learning technique, the neural network must be executed once for each individual (in our case, 100) per generation to calculate its quality. The execution of the neural network is computationally expensive, which significantly slowed down the learning process of the model. To speed up execution and avoid running the network for each individual, the logic for the individual's quality was modified so that, in cases where it did not closely resemble the input case, the life expectancy was not calculated, and the quality for that part was minimal. Despite these changes, the execution time for each case remained high, preventing us from performing a more complete and thorough analysis of the neural network using this strategy.

Correlation and causality

The results obtained from the interpretation of the neural network's operation show that for certain development indicators, the algorithm makes decisions that do not correspond to reality. This is the case, for example, with the feature referring to the birth rate. The value of this indicator might seem like it would not influence life expectancy. However, after the analysis, it was observed that for cases with a high birth rate, life expectancy is lower than when the birth rate is low. This happens because countries with a high birth rate are generally developed countries with a high life expectancy, whereas less developed countries have a higher birth rate. In other words, birth rate and life expectancy are correlated, so the model will use the first to calculate the second. However, this correlation does not imply causality.

As with birth rate, this may happen with other indicators. Therefore, we cannot conclude that modifying the studied factors will lead to a change in life expectancy. However, we can suggest that a country with certain characteristics given by the indicators will have a certain life expectancy associated with it.

Capítulo 5

Conclusions and Future Work

5.1. Conclusions

After completing the entire workflow and obtaining results, conclusions have been drawn regarding the initial objectives set out in this project.

The first point to highlight is that the unified dataset, which aimed to include development, food, economic, political, and health indicators for a wide number of countries organized by year and gender, has been successfully obtained. However, this dataset contained a high number of missing values that had to be imputed, which is a factor to consider.

A machine learning model based on a multilayer perceptron has been successfully developed to predict life expectancy from the provided factors. This type of regression model has clearly been the best of all those proposed, achieving a mean absolute error (MAE) of less than one-third of a year.

Analyzing a neural network is a complex process, in which different approaches have been proposed, allowing us to draw a broader range of conclusions.

However, the first conclusion regarding the analysis of which factors most influence a country's life expectancy using a machine learning model is undoubtedly the relationship between **correlation and causality**. From the results obtained in the different interpretations of the proposed black-box model, the significant influence of certain factors, such as the birth rate, became evident. These factors, which initially may seem unrelated to the outcome calculation, actually have a substantial weight in the model. This is because they show a high correlation with the target variable. However, this does not mean they cause the value, i.e., they are not the cause. For example, according to our model, a higher birth rate results in lower life expectancy and vice versa. This is clearly because countries with low birth rates are usually developed, with higher life expectancy, while countries with higher birth rates are often underdeveloped, with lower life expectancy.

Therefore, the factors that most influence the model have been interpreted as those that best characterize life expectancy.

Thanks to the application of ***wrapper methods***, it has been concluded that the key features that allow for an acceptable prediction error are only the nine listed below:

- Probability of dying from cardiovascular diseases, cancer, diabetes, or chronic respiratory diseases between the ages of 30 and 70, with a 95 % confidence interval
- Birth rate
- Percentage of deaths caused by injuries
- Year

- GNI per capita
- Percentage of the population with access to basic drinking water services
- Gender
- Tuberculosis incidence
- Homicide rate

Just with these features from the dataset, an acceptable error can be achieved. It is also likely that another set could achieve similar results, but the computational cost of testing and finding another set with this number of factors that yield a good error is very high.

Through the **genetic algorithm**, a model has been developed that, by adjusting a change margin as a parameter for the initial values of a country, year, and gender, is capable of optimizing those variations to maximize life expectancy. By interpreting the changes proposed by this algorithm for each case, it has been concluded that the neural network is based on the values of the cases with high life expectancy. Therefore, the closer the feature values are to those cases, the higher the life expectancy. For the features with high absolute values in those cases, the more extreme the value, the greater the increase in life expectancy. These indicators are:

- Percentage of the population with access to basic sanitation and hygiene services
- Deaths due to traffic accidents
- Probability of dying from cardiovascular diseases, cancer, diabetes, or chronic respiratory diseases between the ages of 30 and 70, with a 95 % confidence interval
- Interventions against neglected tropical diseases (NTDs)
- Birth rate
- Homicide rate
- Macronutrient diet

Thanks to the last approach proposed for interpreting the neural network, the great usefulness of SHAP for black-box models has been discovered. SHAP allows us to easily understand the influence of each feature on the final result, as well as how the values of certain features influence others.

This approach has allowed us to conclude the type of influence, whether positive or negative, on the final result of each factor using violin plots, as shown in Figures 4.14 and 4.15. Furthermore, it is concluded that the indicators that have the greatest influence on the calculation are, in order:

- Basic healthcare services covered
- Probability of dying from cardiovascular diseases, cancer, diabetes, or chronic respiratory diseases between the ages of 30 and 70, with a 95 % confidence interval

- Deaths due to traffic accidents
- Birth rate
- Homicide rate
- Percentage of the population with access to basic sanitation and hygiene services
- Tuberculosis incidence
- Interventions against neglected tropical diseases (NTDs)
- Number of doctors per 10,000 inhabitants
- Percentage of the population with access to soap and water handwashing services at home
- Meat consumption

However, although these factors are more relevant at a general level, different factors may influence individual cases or comparisons. For example, when tracking the evolution of life expectancy year after year, the most influential indicators vary, with those related to diet having a much greater weight, as shown in the case illustrated in Figure 4.21.

If the comparison is made by groups of countries with similar life expectancy, it is concluded that while the factors that make the difference for countries with high life expectancy are those that have the most influence at a general level, the most influential factors for countries with low life expectancy are deaths from armed conflicts and terrorism and those related to diet, such as:

- Pineapple consumption
- Fruit consumption: other citrus fruits
- Plant-based protein consumption
- Banana consumption
- Fat consumption
- Percentage of the population whose healthcare spending exceeds 25 % of their income
- Alcohol consumption

Finally, thanks to the analysis of individual cases provided by SHAP, it has been concluded that, although some factors have a greater weight on life expectancy, nearly all of the features in the dataset have, at least, a small influence on the final result.

5.2. Future Work

This project offers a series of possible extensions and improvements to be made.

Deployment

A web application could be developed with an interface that allows users to experiment with the implemented model, observing the influence of factors on the final result. The study could include hypothetical cases with user-inputted values, modifications to existing cases to see the impact of changes, or comparisons between different cases to observe the influence and differences of the input indicators.

The implementation of this proposal could be done, for example, through the *Azure Function* cloud service [57], which provides the infrastructure to perform it and offers a serverless service.

Different Dataset

This approach could be applied to a different dataset, either by expanding or reducing the number of input factors. Whether with different indicators or extracted from other sources, with fewer missing values to avoid conditioning the model too much.

Additionally, it could be separated into several different models, for example, for developed, underdeveloped, and developing countries. In this way, indicators like birth rate might have a more significant influence.

Another possibility would be to select a reduced set of especially interesting factors, eliminating features of greater influence, such as year, to study how accurately they can predict life expectancy.

Modification of the Genetic Algorithm

With new approaches, the genetic algorithm could be applied, but this time taking into account the maximum and minimum possible values of each indicator, penalizing values that exceed those limits, which could be determined by the maximum and minimum values within the dataset.

Bibliografía

[1] bismart, "¿qué hacemos? - etl." https://blog.bismart.com/es/que-hacemos-etl.

[2] P. Rodó, "Distribución normal." https://economipedia.com/definiciones/distribucion-normal.html, 2019.

[3] Wikipedia, "Distribución exponencial." https://es.wikipedia.org/wiki/Distribuci%C3%B3n_exponencial, 2021.

[4] P. N. Roldán, "Modelo de regresión." https://economipedia.com/definiciones/modelo-de-regresion.html, 2016.

[5] C. E. Ouyang, "Introduction to machine learning with python - chapter 2 - datasets and knn." https://elvinouyang.github.io/study%20notes/python-datasets-and-knn/, 2017.

[6] J. M. Heras, "Árboles de decisión con ejemplos en python." https://www.iartificial.net/arboles-de-decision-con-ejemplos-en-python/#Arboles_de_Decision_para_Regresion, 2020.

[7] J. M. Alvarez, "El perceptrón como neurona artificial." http://blog.josemarianoalvarez.com/2018/06/10/el-perceptron-como-neurona-artificial/, 2018.

[8] Atria Innovation, "Qué son las redes neuronales y sus funciones." https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/, 2019.

[9] M. Li, "Tectonic discrimination of olivine in basalt using data mining techniques based on major elements." https://www.researchgate.net/figure/K-fold-cross-validation-method_fig2_331209203.

[10] D. Nieves, "¿qué es el escenario de entrenamiento, validación y prueba de conjuntos de datos en aprendizaje automático?." https://es.quora.com/Qu%C3%A9-es-el-escenario-de-entrenamiento-validaci%C3%B3n-y-prueba-de-conjuntos-de-datos-en-aprendizaje-autom%C3%A1tico, 2021.

[11] S. Mazzanti, "Shap values explained exactly how you wished someone explained to you." https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30, 2020.

[12] E. O.-O. Max Roser and H. Ritchie, "Life expectancy," *Our World in Data*, 2013. https://ourworldindata.org/life-expectancy.

[13] M. Evaluation, "Lesson 3: Life tables." https://www.measureevaluation.org/resources/training/online-courses-and-resources/non-certificate-courses-and-mini-tutorials/multiple-decrement-life-tables/lesson-3.

[14] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[15] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (F. Loizides and B. Schmidt, eds.), pp. 87 – 90, IOS Press, 2016.

[16] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, 2010.

[17] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[18] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[19] F. Serredilla, "Salga."

[20] S. Lundberg., "Documentación shap en python." `https://shap.readthedocs.io/en/latest/index.html`, 2018.

[21] C. W. Hansen, "The effect of life expectancy on schooling: Evidence from the international health transition," *University of Southern Denmark*, 2012.

[22] M. S. Md. Nazrul Islam MONDAL, "Impact of socio-health factors on life expectancy in the low and lower middle income countries," *Iran J Public Health*, 2013. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4441932/.

[23] C. G. Marco Tulio Ribeiro, Sameer Singh, ""why should i trust you?": Explaining the predictions of any classifier." `https://arxiv.org/abs/1602.04938`, 2016.

[24] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting modelpredictions." `https://arxiv.org/abs/1705.07874`, 2017.

[25] D. Vorotyntsev, "What's wrong with lime." `https://towardsdatascience.com/whats-wrong-with-lime-86b335f34612`, 2020.

[26] F. Piccinini, "Interpreting black-box machine learning models with genetic algorithms." https://towardsdatascience.com/interpreting-black-box-machine-learning-models-with-genetic-algorithms-a803bfd134cb, 2020.

[27] World Bank, "World development indicators." https://databank.worldbank.org/source/world-development-indicators, 2021.

[28] World Health Organization, "World health organization datasets." https://www.who.int/data/collections, 2021.

[29] UNICEF, "Dataset archives." https://data.unicef.org/resources/resource-type/datasets/, 2021.

[30] Our World In Data, "World indicators." https://ourworldindata.org/, 2021.

[31] FAOSTAT, "Food and agriculture organization of the united nations data." http://www.fao.org/faostat/en/#data, 2021.

[32] U. Sharma, "World health statistics 2020 complete geo-analysis." https://www.kaggle.com/utkarshxy/who-worldhealth-statistics-2020-complete, Enero 2021.

[33] Naciones Unidas, "Estados miembro." https://www.un.org/es/about-us/member-states, 2021.

[34] Greelane, "Detecte la presencia de valores atípicos con la regla del rango intercuartílico." https://www.greelane.com/es/ciencia-tecnolog%C3%ADa-matem%C3%A1ticas/mates/what-is-the-interquartile-range-rule-3126244, 2018.

[35] N. N. Taleb, *The Black Swan: The Impact of the Highly Improbable*. Random House, 2007.

[36] A. Barai, "Normal distribution and machine learning." https://medium.com/analytics-vidhya/normal-distribution-and-machine-learning-ec9d3ca05070, 2020.

[37] ICHI.PRO, "Distribución normal y aprendizaje automático." https://ichi.pro/es/distribucion-normal-y-aprendizaje-automatico-260160071159920.

[38] B. AI, "Funciones de probabilidad." https://bootcampai.medium.com/funciones-de-probabilidad-fbd59eb55b59, 2020.

[39] J. L. Cano, "Ajuste e interpolación unidimensionales básicos en python con scipy," *Pybonacci*, 2013.

[40] J. Brownlee, "knn imputation for missing values in machine learning," *Machine Learning Mastery*, 2020.

[41] Mahomet, "Reverse a get dummies encoding in pandas," *Stack Overflow*, 2020.

[42] J. Brownlee, "How to use data scaling improve deep learning model stability and performance." https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/, 2019.

[43] B. Roy, "All about feature scaling." https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35, 2020.

[44] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the influence of normalization/transformation process on the accuracy of supervised classification," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 729–735, 2020.

[45] G. Box and D. R. Cox, "An analysis of transformations, journal of the royal statistical society." https://www.ime.usp.br/~abe/lista/pdfQWaCMboK68.pdf, 1964.

[46] Yeo and R. Johnson, "A new family of power transformations to improve normality or symmetry," 2000.

[47] J. Brownlee, "How to use power transforms for machine learning." https://machinelearningmastery.com/power-transforms-with-scikit-learn/, 2020.

[48] R. L. y Fernando Ortega, "Material de la asignatura de "machine learning"del grado de ingeniería del software en la escuela técnica superior de sistemas informáticos de la universidad politécnica de madrid," 2020.

[49] F. Serradilla, "Material de la asignatura de .ªgentes inteligentes"del grado de ingeniería del software en la escuela técnica superior de sistemas informáticos de la universidad politécnica de madrid," 2021.

[50] G. Perez, "¿por qué es relu la función de activación más común utilizada en redes neuronales?." https://es.quora.com/Por-qu%C3%A9-es-ReLU-la-funci%C3%B3n-de-activaci%C3%B3n-m%C3%A1s-com%C3%BAn-utilizada-en-redes-neuronales, 2021.

[51] J. B. Diederik P. Kingma, "Adam: A method for stochastic optimization." https://arxiv.org/abs/1412.6980, 2014.

[52] J. Brownlee, "A gentle introduction to early stopping to avoid overtraining neural networks." https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/, 2018.

[53] J. Holland, "Adaptation in natural and artificial systems," 1975.

[54] C. Darwin, *On the Origin of Species*. John Murray, 1859.

[55] L. Shapley, "Shapley values," 1953.

[56] P. Płoński, "Random forest feature importance computed in 3 ways with python." https://mljar.com/blog/feature-importance-in-random-forest/, 2020.

[57] Microsoft, "Documentación de azure functions." `https://docs.microsoft.com/es-es/azure/azure-functions/`.