Advanced Data Analysis Project

Spectral Dataset

Week 2

LUT University

Alejandro Lopez Vargas
Sina Saeedi
Pau Ventura Rodríguez

Visit our GitHub to check the script.

Dataset introduction

Hyperspectral Soft Sensor's dataset originates from a comprehensive multi-sensor study that amalgamated spectral data and vegetation characteristics from a total of 42 datasets gathered across various continents, climates, and vegetation types.

Our project will focus on Predicting traits: selecting a proper technique, comparing the performance of the models in case of MLR, PCR, PLS and k-PLS models. Selecting which are the strengths and the weaknesses of each model following criteria such as the test partition prediction performance, the interpretability of the model or the time elapsed for training and prediction.

Dataset description

The hyperspectral data, which serves as the input variables, covers wavelengths spanning from 450 to 2500 nm, with increments of 1 nm, although some of them are missing (from 1351 to 1430 and from 1801 to 2050), leading to a total of 1721 inputs.

The leaf and canopy traits, representing the response variables, encompass various attributes such as leaf pigments, leaf area index, equivalent water thickness and more, which adds up to a total of 20 target variables.

Although the features have a linear correlation (i.e. the feature "1 nm" corresponds to half the wavelength of "2 nm"), the dataset cannot be considered a time-series dataset, as the features do not have a time context and all the features are extracted at the same time (in contrast with time series, which we only get a subset of features at each time step).

The project goals

The primary goal of this project is to utilize spectral data from various wavelengths to predict specific traits. To achieve this objective, we plan to employ various regression models, including MLR, PCR, PLS, and k-PLS models. As part of the process, we will perform several pretreatment steps, such as addressing missing values (NaNs), normalizing the data, handling outliers, and more. Ultimately, our aim is to compare the results of these models from various perspectives.

A brief explanation about the spectral soft sensor

A soft sensor, also known as a virtual sensor, is a computational or mathematical model that estimates or predicts a physical process or property in a system or industrial process. Unlike traditional sensors, which directly measure a physical parameter (e.g., temperature, pressure, or flow rate) through physical means, soft sensors use data from various sources, such as other sensors, historical data, or mathematical models, to infer or estimate the desired information.

A spectral soft sensor is a type of soft sensor that specifically deals with estimating or predicting information related to a system's spectral data. Spectral data typically involves measurements across a range of wavelengths or frequencies, and spectral soft sensors are designed to analyze and interpret this data to make predictions or estimations about certain properties or traits.

Challenges of the data

Missing data: Although input data have no missing values, the response variable has a high null value percentage, which might be caused due to the fact that the dataset is a combination of 42 different dataset, each of which might be focused in a different aspect. If performing a supervised training, the model will require the response variable to be non-null to adjust the parameters.

Splitted Dataset: The dataset itself consists of two different datasets with different numbers of columns. The first has 12.180 observations and 1.741 features, while the second one has many less observations (1.115) but includes extra features (1.758 in total). The extra features that second dataset includes are:

'Anthocyanin concentration (mg/g)', 'Boron concentration (mg/g)', 'C concentration (mg/g)', 'Ca concentration (mg/g)', 'Carotenoid concentration (mg/g)', 'Cellulose (mg/g)',
'Chlorophyll concentration (mg/g)', 'Copper concentration (mg/g)', 'Fiber (mg/g)', 'Lignin (mg/g)', 'Magnesium concentration (mg/g)', 'Manganese concentration (mg/g)',
'N concentration (mg/g)', 'NSC (mg/g)', 'P concentration (mg/g)', 'Potassium concentration (mg/g)', 'Sulfur concentration (mg/g)'

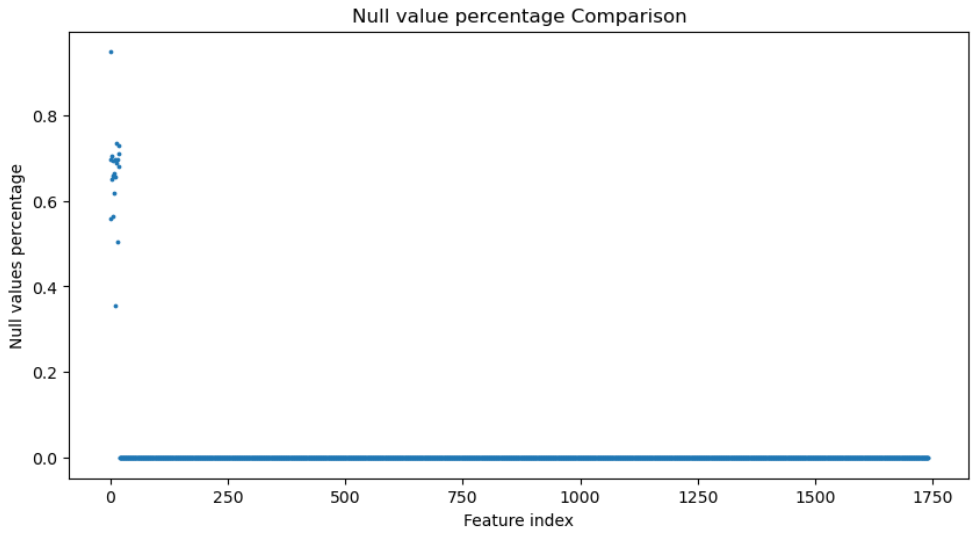
Note that those are concentrations for the targets that we have in both datasets.

If we combined both datasets, the concentrations would have a null percentage of over 90%. This combined with the fact that those targets do not appear in the dataset's description have led us to drop those columns.

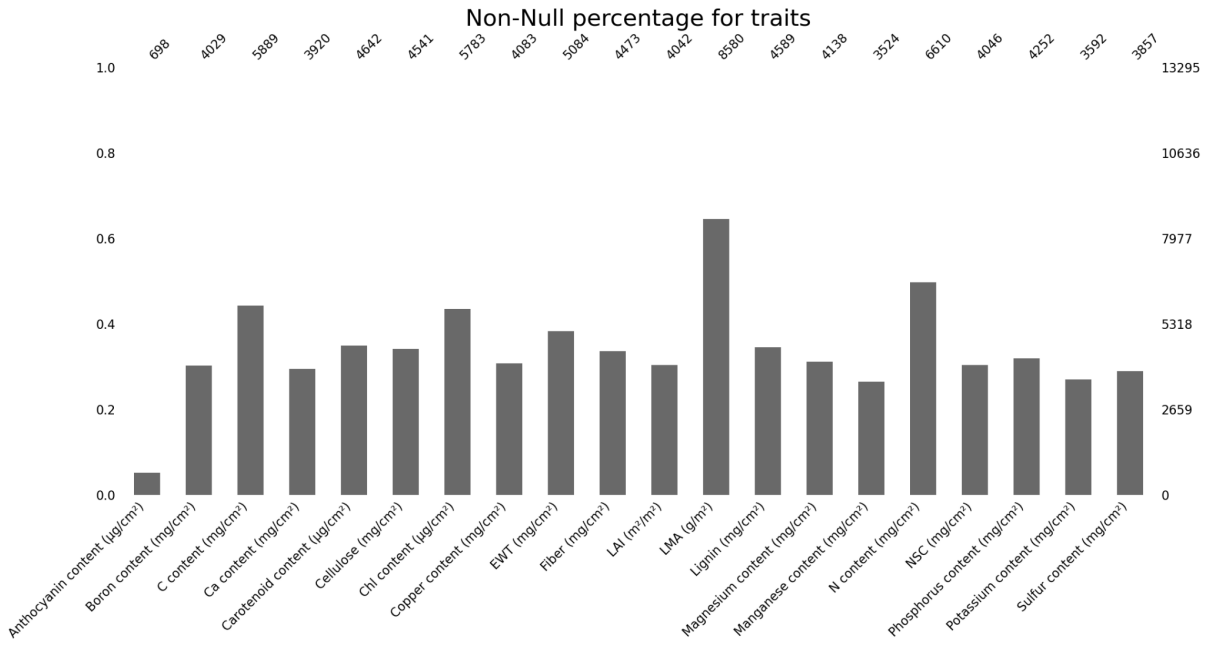
The full cleaned dataset is then made up of 13.295 observations and 1.741 columns.

Data Visualization

Missing values analysis:

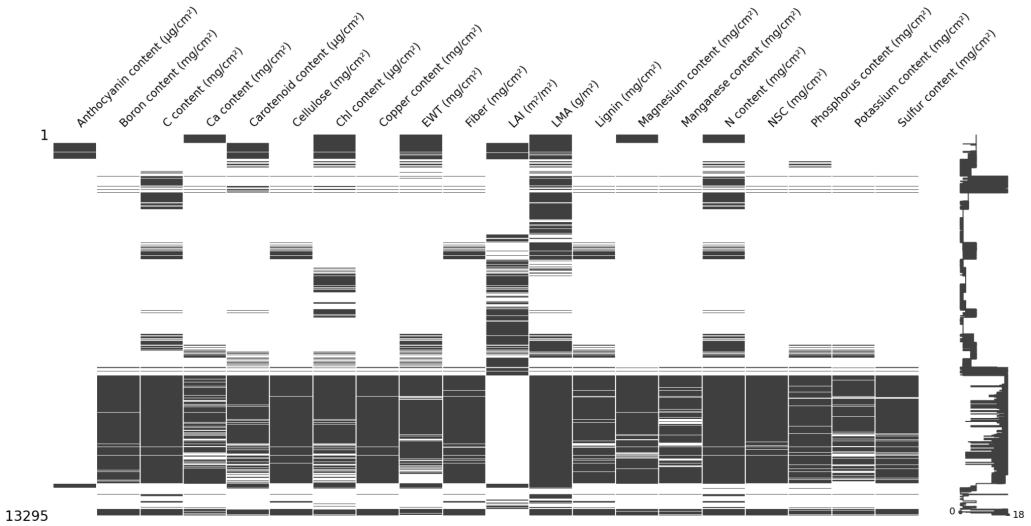


It's worth noting that the wavelengths in the hyperspectral data don't have any missing values, which is a positive aspect of the dataset as imputation won't be required. However, it's crucial to delve further into understanding and addressing missing values in the traits themselves, as this can significantly impact the modeling and prediction process.



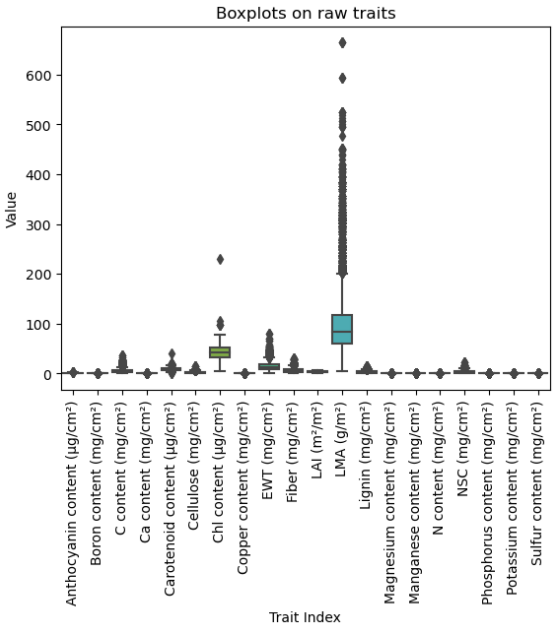
The dataset presents an interesting pattern regarding missing values in traits. The traits exhibit a mean of around 65% of null values, which is an alarming factor. Nevertheless, there are notable traits out of this trend, such as LMA with around 35% missing values, and Anthocyanin content with a substantial ~95% missing data. These outliers warrant specific attention and potential strategies for handling missing data.

Missing values distribution:

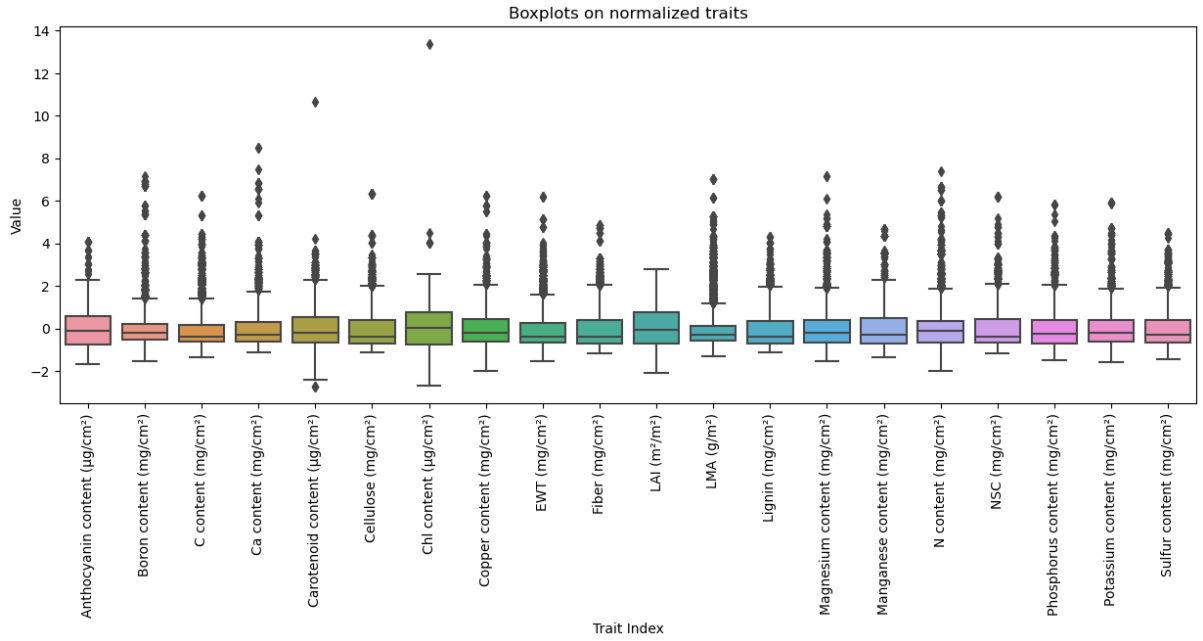


It would be insightful to investigate whether there are any discernible clusters or patterns among the missing values in traits. Identifying clusters of missing data could offer valuable insights into why certain traits have higher percentages of missing values and help in devising targeted strategies for data imputation using other traits. Visually, it can be seen that there is a big cluster of samples with almost all variables, but having LAI as missing values (this can cause a lower correlation as we will discuss later). Other samples outside this cluster have all of the traits but LAI and LMA with a high percentage of Nan.

Distribution of the target variables:



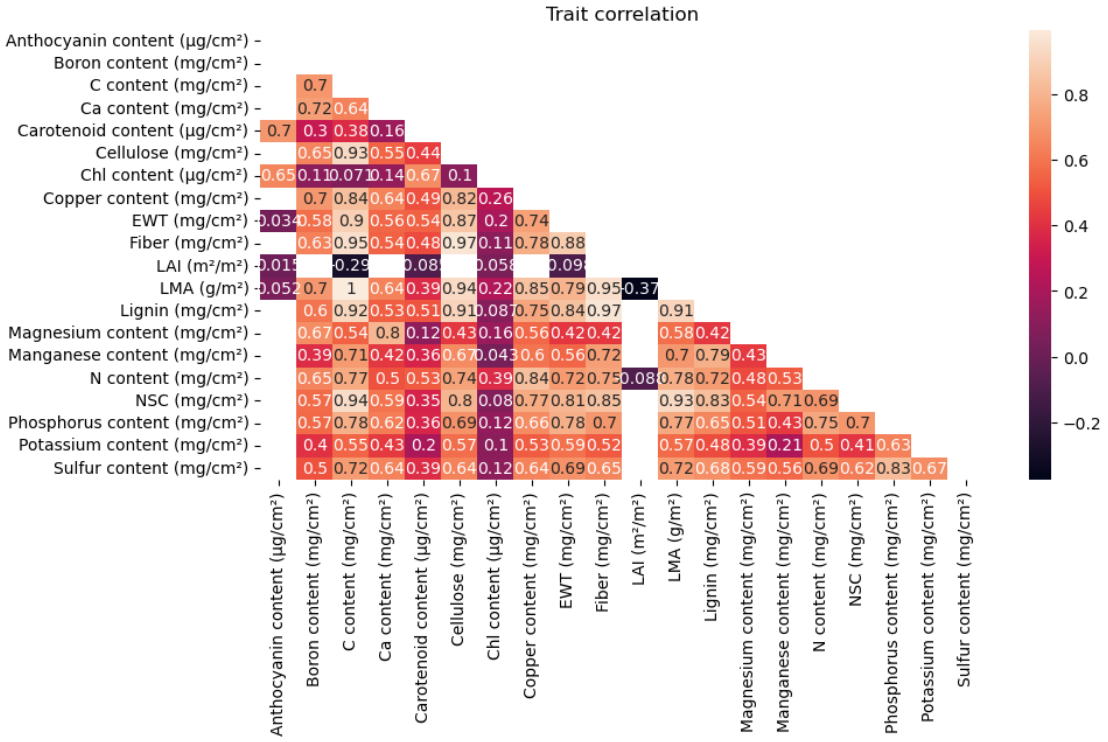
This boxplot guides us on visualizing the values that the traits take. For example, note that LMA reaches values of 600, while the second highest (Chl content) only reaches values of 250. All other features do not reach 100. To view single variable outliers, it's better to scale our data.



An interesting insight of our data is that the outliers seem to be in the upper range of the values. This might be due to the fact that traits only take positive values. The most extreme outlier is located at Chl content,

reaching a value of 14. Other features with important outliers seem to be Carotenoid and Ca.

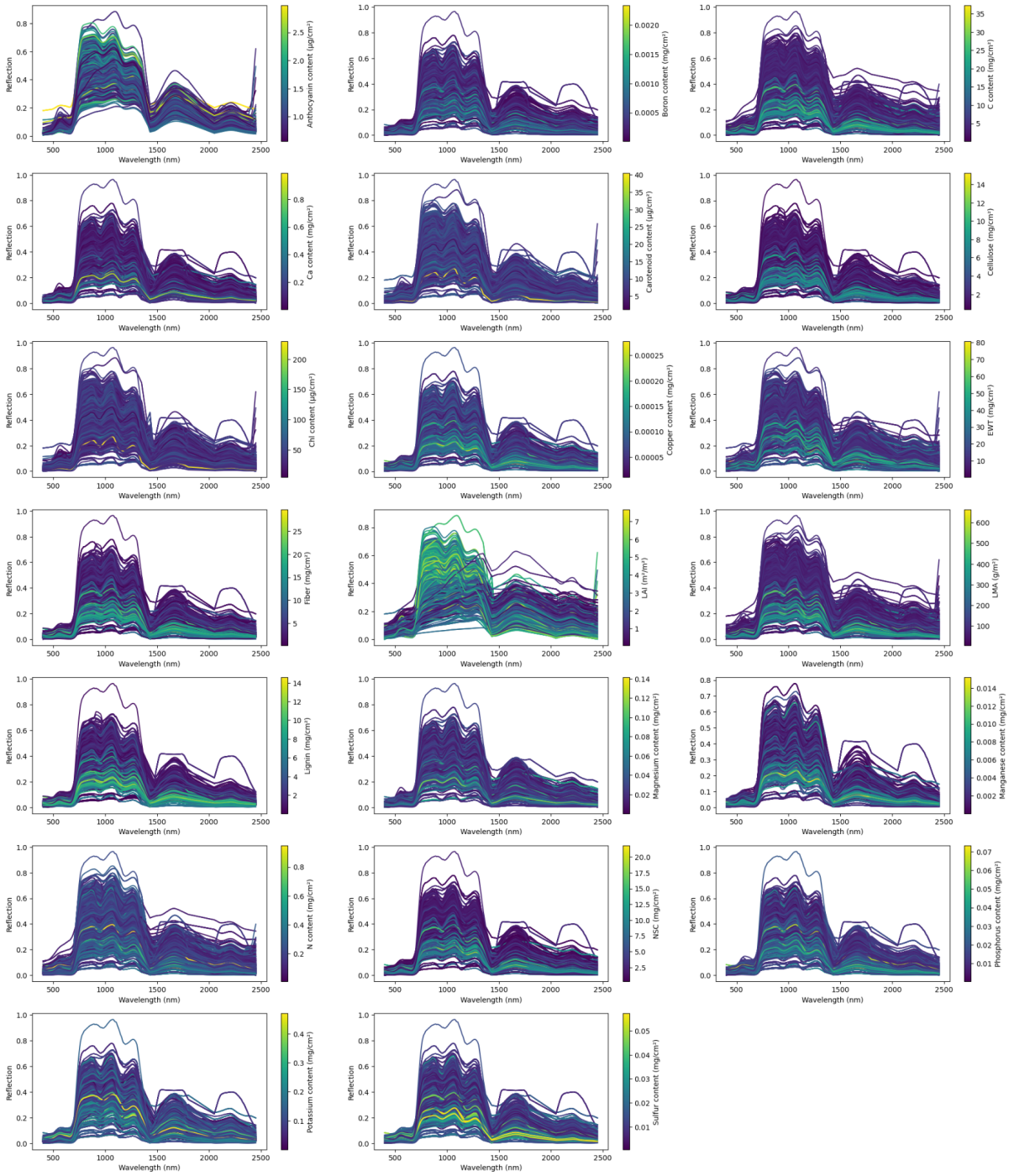
Correlation between target variables:



The correlation analysis between traits is crucial and reveals interesting insights. Traits showing high correlation likely share common spectral features, indicating that predicting one trait might provide valuable information for predicting another trait that is missing or poorly represented in the data. This interrelationship among traits can be leveraged not only for improved trait prediction but also for data imputation, where information from correlated traits can be used to estimate missing values in others. On the other hand, traits with low or no correlation may require different modeling approaches or data sources for accurate prediction. Note here that LAI trait is, as expected, not correlated with any other trait, while others share a correlation of at least 70% with other traits. Note also that all correlations are positive, which indicates that there is no pair of traits that affects each other inversely. Some traits even

have a correlation higher than 90%, like LMA and Cellulose or EWT with C content, among others. There are even two variables that are perfectly correlated, which are LMA and C consent.

Target-Input correlation:



It's interesting to observe the heterogeneous color distribution in certain graphs, for example those related to LMA and Lignin, where low reflection

values indicate a higher trait value (this trend actually is repeated for most of the traits). This heterogeneity could signify that these traits exhibit substantial variability across the dataset. Regarding the value ranges of the

wavelengths, it would be beneficial to explore the specific ranges that are most informative for predicting these traits, as this information can guide the selection of relevant spectral bands for modeling and feature extraction. Visually it can be observed that the first 700 wavelengths only reflect less than 0.2, while wavelengths from 700 to 1400 have the highest reflection value, reaching 0.8 or getting close to 1 in some samples. Then, in the 1400 to 1500 range, the reflection drastically falls to being less than 0.25, and then spikes again until 0.4, ending up slowly decreasing to 0.2 again.

Pretreatment Steps

Normalization: Most Machine Learning models require the features to be normalized, as we will focus on trying and comparing different models, it is necessary to make sure the data is in the correct format. From the input-response comparison graphs we can see the different values that the wavelengths take. Here we can notice that all values range from 0 to 1, but some ranges of wavelength tend to have a higher reflection value. If normalization was not applied, some wavelength values (specially the 700-1300 range) would receive a higher importance than the others. As this is not what we aim for, normalization will probably (after further discussion) be applied.

Imputation: Some models do not admit null values on the data and, although our input data have zero null values, our response does. Having null values on response data is a huge problem as we cannot tell the model which one should be the output for a given input. This can be targeted in two ways:

- Performing an imputation algorithm (probably iterative imputer) using all target trait information), thus ending up with zero null values on the target variables, but might end up including some bias.
- Building a ML model for each response variable, which would result in less samples when training the models.

Encoding: We do not have categorical features, so encoding will not be needed.

Outlier detection: The data has been scanned with devices. Those devices might include some error, or there can be a human factor that might cause some samples to have incorrect values. Outlier detection algorithms will help getting rid of those samples.

PCA: As a first sight into PCA, we have seen that we can maintain around 90% variance only by keeping two components, which will surely help us in visualizing the data and training the models.

Train-Validation-Test division: To train, validate and test models, we need to perform a split on the dataset.