Facultad de Ciencias
Universidad Zaragoza

Memoria Trabajo Fin de Máster

Multi-modal study of the effectiveness of several antibiotics and construction of predictive algorithm

Author:
Tellería Serrano, Oriol [†]
Academic Tutors:
Gracia Tabuenca, Zeus [‡]
Asín Lafuente, Jesús [§]

November 21, 2024

[†]758771@unizar.es
[‡]zeus@unizar.es
[§]jasin@unizar.es

Contents

1 Acknowledgements

In first place thank to both of my project supervisors, Zeus Gracia Tabuenca, Jesús Asín Lafuente and David Ojeda Aure, for their guidance and encouragement throughout the course of this project. Their expertise and thoughtful advice have been crucial in shaping both the direction and the execution of this research.

Special appreciation goes to my colleagues in the work group of NTT Data, especially Laura Miralles Miras whose guiding added greatly to my own understanding of the project. Also to my fellow project partner Roberto Garcia Peña with whom without their participation and help this project would not have been able to carry out.

Also, I want to thank to José Ramón Paño Pardo and Gabriel Tirado Anglés this project would not have been possible without the contributions them, they have helped me to understand many biological aspects related to this work and I am grateful for their support.

Lastly, I wanted to acknowledge my family. Because they are the ones who at the end of the day have been helping me throughout my career. Without this support I would not have been able to get to where I am today.

2 Structure of the memory

During the first chapter 3 we will give some context to the reader in order to better understand our work. In the this section we will oversee our objectives 3.3 and we will give a resume of the medical framework in which this project is developed, focusing on the explanation and operation of antibiograms 3.1.

Over the next section 4.1 which is the most extensive of this work, we will be focusing overall in the explanation of the methods and tools used to carry this project as well as the methodology that we have build to develop it.

Next, in the chapter 5 we give the expose the results obtained after running our algorithms with their correspondent analysis. Finally , during the last section 6 the conclusions derived from the analysis are presented along a little section with some conclusions of the project and future work.

It is also possible to find an appendix's, where it can be found the graphics and tables which have not been shown to ease the reading of the memory.

3 Introduction

Nowadays clinicians are facing a growing threat at hospitals and other healthcare facilities, called Antimicrobial Resistance (AMR) which threatens patients on intensive healthcare units (UCIs). These patients can become infected by Healthcare-associated infections (HAIs) that are not present or incubating at the time of admission or they could become infected before being admitted in the hospitals. Hence, complicating the stage of the patient in the hospital [1]. These infections are mostly of bacterial origin. So, they are treated by using antibiotics and if the appropriate antibiotic isn't selected a resistance can be developed to the antibiotic which if it's not taken carefully can be spread out to other subjects turning it to a health risk for the society.

Thus, patients carry a risk of being infected by different bacteria under their stance at services in the hospitals plus the same bacteria can be resistant to several antibiotics arising a challenge for clinicians which must treat them. In the current state of art the way to tackle this challenge is by experience meaning at first is tried a general antibiotic to see if the infection doesn't spread but if the patient gets worse or it doesn't recovers then an antibiogram is made to select the optimal antibiotic to use.

The criteria of which antibiotic is used it is made by measuring the minimum inhibitory concentration (MIC) [2] which is the minimal concentration of the antibiotic to stop bacterial proliferation. Often this process takes hours, even days. It takes cost in laboratory equipment such as antibiogram dishes, pipettes...etc. Also, taking blood samples of the patient and clinician labor is required. Another problem associated to this is we can develop certain resistance to antibiotics because at the beginning the general antibiotic used is not selected regarding the patient data so the bacteria could develop resistance to the antibiotic which can later on be reach to the rest of the population.

It is imperative to ensure that the aim of this work is to replace the work or the criteria of a health professional. We want to implement statistical classifiers which could benefit clinicians in their decision making by supporting to their decisions[3]. By implementing machine learning we are able to process large amount of data and detect hidden patterns otherwise that would otherwise go unnoticed [4]. This processing of data allow for historical surveillance which would help in the prevention and development of antibiotic resistance. Moreover, to minimize the risk of introducing antibiotic resistance by individualizing the response for each patient in terms of antropometrics, etiology, genetics [5].

Secondly, the costs associated with conducting an antibiogram for each patient can be reduced. This is because it can accelerate the response of the healthcare system to a certain threat, thereby increasing the survival expectancy of patients.

Finally, help to reduce costs of materials which can then help to optimize the resources of our healthcare network but also, work of clinicians making their work easier and saving

them time to carry out other important activities.

By introducing an algorithm that can help to made a decision on which is the optimal antibiotic according to antibiograms data stored and biological information from the patient. To do so, we will test several statistical methods such as logistic regression, decision tree, and simple neural networks. By introducing data mining we hope to lay a foundation which later on could be used by medics in their work. For example, to consult in a database via similar patients and their diagnostics or to survey high-risk populations.

3.1 Antibiograms

While our work is not to take samples of patients in order to construct the database is mandatory to give a short insight to the reader on the why, how and what of the antibiograms. For that reason this section will cover the three basic questions.

3.1.1 What are antibiograms?

Antibiograms are microbiological medical tests designed to determine the most effective antibiotic to use against a specific organism that has caused or may cause an infection in a patient. The primary goal is to assess the susceptibility, meaning the resistance, of the bacteria/organism to a group of antibiotics.

The MIC (or CMI in Spanish) of a microorganism for a specific antibiotic is the lowest concentration of that antibiotic necessary to inhibit its growth under standardized conditions. It is not an absolute value and cannot be compared across different antibiotics and/or microorganisms; therefore, its higher or lower value relative to the MIC of another antimicrobial is not a useful guide in choosing targeted antibiotic therapy. This is because certain MIC levels of a specific antibiotic may be toxic to a patient, depending on their characteristics (weight, age, etc.). Consequently, this is an individualized test that requires interpretation by a healthcare professional [2].

3.1.2 Why antibiograms are done

The basic utility of an antibiogram is to establish the correct antibiotic treatment for the patient. It is essential to determine whether the microorganism responsible for the infection has mechanisms that confer resistance to any antibiotic, to avoid including it as part of the therapy [2].

Regarding treatment, the antibiogram is not only necessary at the initiation of therapy, but also useful in monitoring and even confirming empirical treatments. In some cases, the infectious disease is severe, and treatment is started before the sensitivity data of the strain is known. The antibiogram must confirm, or if necessary, correct the treatment [6].

Another application of resistance study techniques is in epidemiology. It is crucial to detect the increase in resistance levels in clinical isolates to implement corrective measures.

Moreover, it can also have diagnostic utility because the resistance profile can sometimes guide bacterial identification.

The Centers for Disease Control and Prevention (CDC) [7] define the appropriate use of antibiotics as a practice that maximizes therapeutic impact while minimizing toxicity and the development of resistance. Inappropriate use of antibiotics has significant consequences: it increases costs, toxicity, and microbial resistance.

The widespread use of antibiotics has been correlated in multiple studies with the development of resistance, including cross-resistance to other drug families. The increase in resistance, in turn, raises treatment costs and promotes therapeutic failure. Additionally, some resistant strains may result in higher morbidity and mortality, as is the case with methicillin-resistant Staphylococcus aureus (MRSA)[8].

The most important factors to consider when prescribing antibiotics can be summarized as (although there are more factors):

- The most common etiology of each infection is largely predictable based on location, age, sex, and weight.

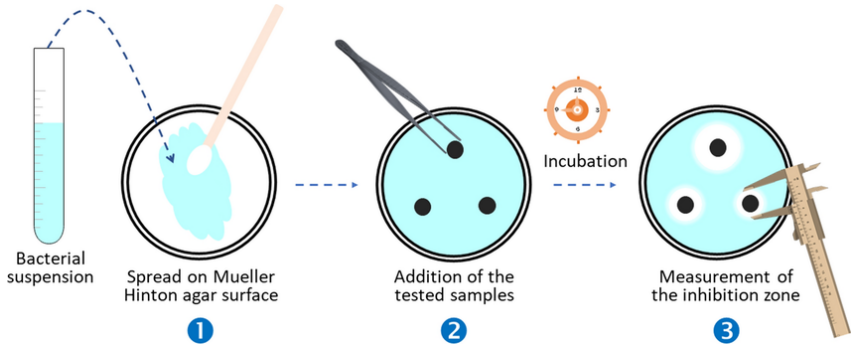
In recent years, due to the considerable increase in antimicrobial resistance, there has been a need to develop new antibiotics and combinations of them to treat multidrug-resistant bacteria that pose a significant public health risk. Therefore, it is essential to tailor antibiotic therapy in each situation to minimize the development of resistance and prevent the emergence of new multidrug-resistant strains [9] [10].

3.1.3 How are done?

There are various methods to perform this type of medical test, each with its own advantages and disadvantages. Here, we present the most common methods and list additional methods for estimating MIC and sensitivity [6] [11].

3.1.3.1 Diffusion Methods

1. **Disk diffusion method**: In this test, absorbent paper disks impregnated with antibiotics are placed on the surface of the agar in a Petri dish where the microorganism has previously been cultured. The antibiotic diffuses into the medium, creating concentration gradients that inhibit the growth of the microorganism at a certain concentration of the antimicrobial agent, that is, at a certain distance from the disk (see Figure 1). Depending on the diameter of the inhibition zone, the microorganism is categorized as Sensitive or Resistant by comparing it to standardized data. This method is advantageous because it is versatile, simple, and



h

Figure 1: Schema of a Kirby-Bauer Test or disk diffusion test. Available from: https://www.researchgate.net/figure/Schematic-representation-of-the-disk-diffusion-method-of-Kirby-and-Bauer_fig5_358212661 by Abdelqader El G. et El Arbi B., Polymer Bulletin 79(12):1-24. January 2022. Copyright by Springer Nature.

cost-effective, but its drawback is that it does not allow for a direct reading of the MIC [12].

2. **Gradient strip**: This method involves a non-porous plastic strip where the antibiotic diffuses into the medium, creating a concentration gradient along the strip (see Figure 2). The growth is inhibited at a certain point on the strip, which determines the antibiotic concentration. This method is simple and allows for a direct measurement of the MIC, but the strips are expensive as they require pre-treatment to predefine the antibiotic gradient [2].

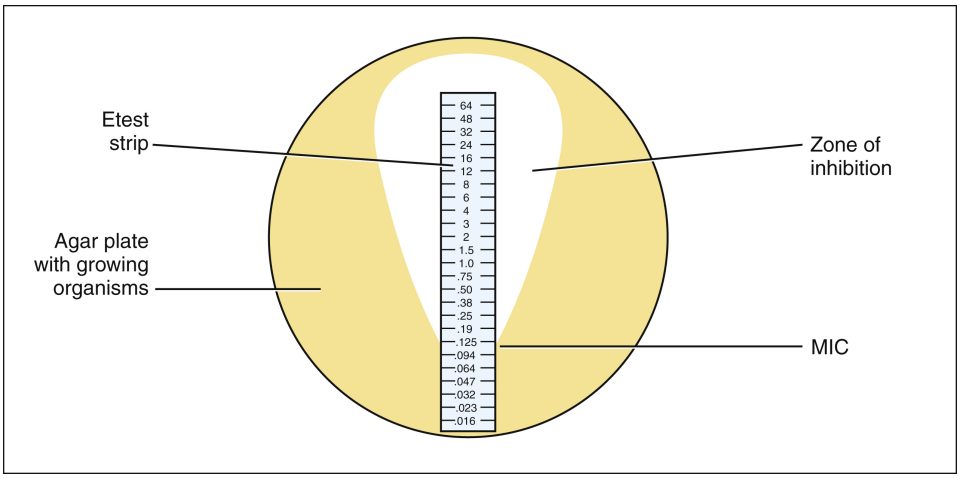


Figure 2: Schema of a E-test or Gradient strip test. Available from: https://rr-americas.woah.org/app/uploads/2022/12/sp_metodo-de-difusion-2022_albornoz_compr.pdf by Ezequiel A. Juny 2022. Copyright by WOAH .

3.1.3.2 Dilution Methods

1. **Agar Dilution**: Plates are prepared with agar containing increasing concentrations of the antibiotic diluted in the medium, and the microorganism is added (see Figure

3). The plate with the lowest concentration of antibiotic that inhibits the microorganism is considered the MIC. This method provides a direct reference measure of the MIC, but it is labor-intensive as it requires more preparation [2].

2. **Broth Dilution**: Similar to the agar dilution method, but instead of using agar, a liquid culture medium with increasing concentrations of the antibiotic is used (see Figure 4). This method offers the same advantages and disadvantages as the previous method [2].

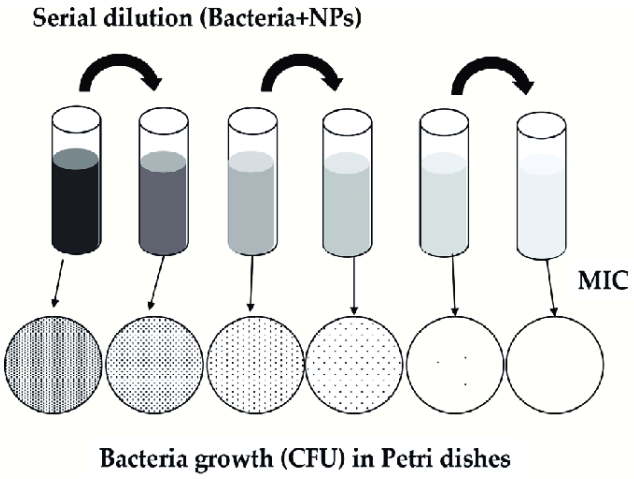


Figure 3: Schema of the agar test. Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Agar-dilution-method-with-NPs_fig3_332522731 by Alejandro L. Vega-Jiménez. CC-BY 3.0

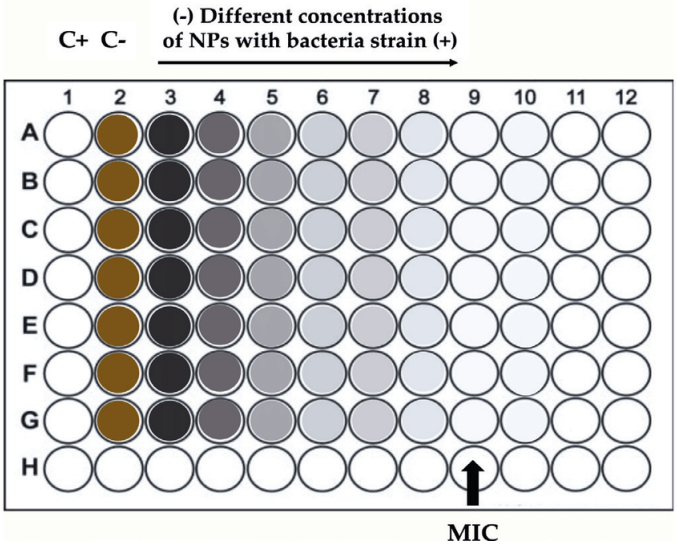


Figure 4: Schema of a broth dilution test. Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Broth-dilution-method-with-NPs_fig4_332522731 by Alejandro L. Vega-Jiménez. CC-BY 3.0

They exist other alternative Methods for determining resistance of microorganisms to antibiotics. We list and summarize a few of the most common ones here (see Figures 5, 6, 7 and 8) [13] :

1. **Molecular Biology**: Polymer chain reaction (PCR) for the detection of genes associated with specific bacterial resistances.

2. **Micro-arrays**: Simultaneous detection of multiple resistance genes through specific hybridization with labeled probes.

3. **Inmuno-chromatography**: For detecting mutations in the antibiotic target or microbial enzymes that hydrolyze the antibiotic.

4. **Colorimetric methods**: A change in the pH of the medium due to the hydrolysis of the antibiotic by a bacterial enzyme produces a color change in the indicator.

Others methods are fluorescent in situ hybridization, mass spectrometry, flow cytometry, nephelometry, chemiluminescence [2].

Figure 5: Doctor performing a PCR test. Available from: https://www.freepik.es.

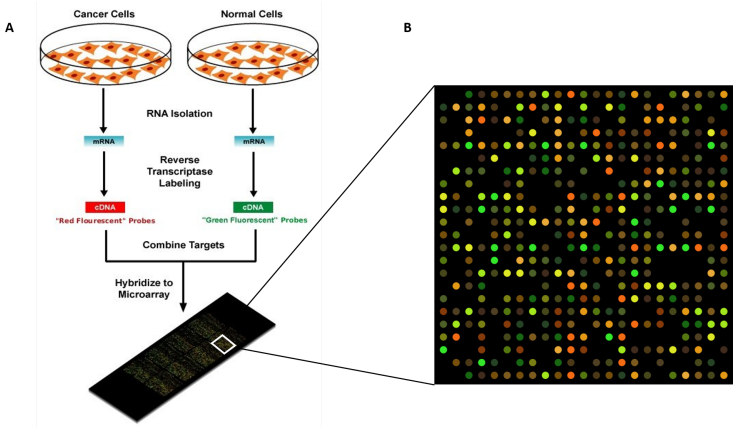


Figure 6: Cartoon schematic of a typical micro-array experiment. Available from https://w.wiki/BxA8 by Larssono 20 September 2007.

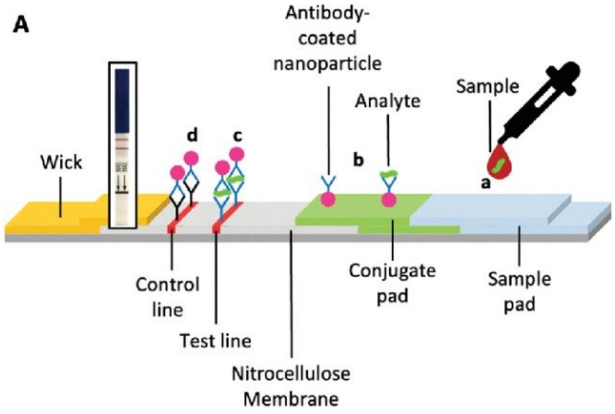


Figure 7: Schema of Inmunochromotography. Available from: https://www.researchgate.net/figure/Schematic-of-a-prototypical-lateral-flow-device-A-Device-constituents-in-a-successful_fig1_362309043 by Alexander N. Baker July 2022. CC-BY 3.0

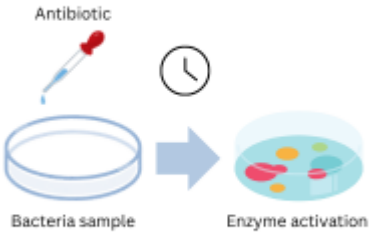


Figure 8: Schema of a calorimetric test. Own work

3.2 State of Art

While we have provided the reader with some introduction on the firs section 3 and an overview on antibiograms 3.1. We must contextualize this work with the current situation in this area of knowledge. The line of work which involves this project is embedded with the growing tendency of implementation complex statistical analysis to solve or give

answer biological problems. We must remark the goal of this work it isn't to substitute professional criteria. We want to lend bring these tools in order to start a new way of approach to fight AMR.

It is well known that resistance to antimicrobial is a theme of global awareness and in various countries it has risen suddenly [14] [8]. The quality standards and control of infections linked to health attention have put into effect a continuous watch over sensibility patterns and resistance in bacteria of high virulence and mortality. The common guideline is to watch over the resistance through descriptive statistical methods[15]. The descriptive statistics is useful and we cannot negate the advantages which derive from it's use. But the next "natural/logical step would be to apply more complex methods that can unravel mechanisms which are masked or to allow us for a more selective response.

Also, these descriptive methods have limitations when it comes to detecting patterns when the volume of data is large. In addition, this generation of data produces data whose exploitation could be a useful tool in the fight against antimicrobial resistance. In some healthcare centers, classificatory models have already implemented in order to detect and identify patterns [16]. Allowing to discover early these patterns and to take measures to avoid major problems [17].

In the bibliography we encounter several studies which have already tackled similar problems. But, most articles emphasize on the fact that *"the current body of work on machine learning models for AMR prediction is largely focused on genomic data models"* [1]. Which is natural since the volume data generated on DNA analysis is huge therefore requires of specific tools. Hence, we have an opportunity to bring our knowledge and lend it to the society.

Some recent articles of the bibliography have discussed what we describe. As a matter of fact, in the article [1] they implemented machine learning algorithm. Showing improved classification performance (area under the receiver operating characteristic curve 0.88-0.89) vs a naive model (AUC 0.86) for 6 different organisms and 10 antibiotics using the Philips eICU Research Institute (eRI) database. In other article [18] using different records from several data sources (from Stanford emergency departments and from Massachusetts General Hospital and Brigham & Women's Hospital) they achieved a coverage rate (fraction of infections covered by treatment) of 85.9% for Stanford data and for Boston dataset achieved a 90.4% coverage rate by using different machine learning models. Another article using 19.538 blood samples and 16.765 urine samples [19] tested several machine learning methods. In the article using an RF algorithm they achieved an AUC 0.61 for the blood samples and 0.68 for urine samples. The PRAUC values were slightly higher reaching 0.97 for blood and 0.88 for urine.

[20] In these articles we abandon the idea of trying to characterize the AMR via the genomic dynamics. Their objective is to develop predictive models to identify variables when genomic information is scarce or it cannot be obtained. Instead, the variables of

study are from patient information (as Age, Gender, CMI, Ethnicity, Race,...etc) and microbiology test that are easier to acquire. Thence, achieving the goal of reducing the usage of unnecessary or ineffective antibiotics while improving patient outcomes.

These articles perfectly complement our objectives. Their results show that even though this line of research is still in its early stages, we can study which variables are important and may be useful for future studies. Although, our project is in the same line of research, it does not have the resources described in these articles, so our results must be understood in that context. Even if there are similarities, the results will be subject to the context. While previous studies use extensive and individual datasets, in our case, we are limited by this fact and have had to artificially generate more data to train our models. Additionally, our models are simpler. Specifically, the idea is to create a toy model with the possibility of scaling it *a posteriori*.

The reviewed literature demonstrates the increasing significance of applying advanced statistical methods to AMR [3]. Current studies emphasize a shift from solely relying on descriptive statistics to employing predictive models capable of uncovering complex patterns masked within large-scale data. While most existing research focuses on genomic data due to its rich and voluminous nature, there is a growing body of work that pivots towards using clinical and microbiological data when genomic information is limited. Such approaches have shown promising results, as evidenced by studies that highlight improved predictive accuracy and early detection capabilities, aiding in timely clinical decision-making and enhancing patient outcomes.

Our work seeks to contribute to this evolving field by applying these methodologies within a constrained context. We work within limitations that require us to artificially expand data for model training and simplify model structures. Nonetheless, our aim is to develop a foundational predictive model that can be scaled and improved in future research. This approach not only aligns with current efforts to optimize AMR prediction in diverse data environments but also underscores the potential for deploying practical, adaptable solutions in real-world healthcare settings where data resources may be scarce. Such efforts pave the way for more efficient use of antibiotics, better patient management, and a more strategic response to the global challenge of AMR.

3.3 Objectives

Although in the previous section we have advanced some objectives in this section, we will present contextualized objectives that have been aimed for in the completion of this work. These objectives are divided into two categories. The first category are general objectives:

1. Design a pipeline for selecting an efficient classification algorithm.

2. Integrate and study statistical tools to support medical analysis.

And the second category are specific objectives:

1. Develop a data storage criterion for antibiogram test results to ensure proper algorithm functionality.
2. Create an algorithm that is easily scalable.
3. Highlight advantages and disadvantages of each algorithm.
4. Facilitate the understanding of results from classification algorithms by providing a simple and visual presentation of outcomes.

In summary, this work primarily focuses on managing our databases to format them for optimal efficiency in subsequent algorithmic use (i.e. cleaning data and modifying certain formats). Next, these same data will be employed in data mining to extract relevant information that can support analysis. Finally, the information will be presented visually to enable evaluation and comprehension of the results, thereby promoting the integration and use of the data in future practical applications.

4 Methods and Methodology

During this section our main focus will be to explain in detail the arguments on which we have constructed the algorithm, going from the initial assessments of selecting and classifying the variables from our data to the selection of the optimal method of classification for our work.

4.1 Methodology

Our methodology as stated before is purely computational since our approach is to use the results from previous analysis already done and saved in data bases from laboratories and posterior simulation to ensure statistical relevance. The objective is to build a pipeline which will be the basis of an application/process that will be called by clinicians.

Therefore, the methodology of this work can be divided in three main chapters:

1. Discrimination criteria.
2. Classifiers.
3. Evaluation criteria.

Each chapter has it's characteristics and particular processes which are related to other previous steps from the other sections constructing a general pipeline which that can be checked on the Appendix (see on the GitHub [21]) where the code is found and commented.

4.1.1 Discrimination criteria

As first step in this work will be to study our set of data in order to obtain which of the attributes inside the data are the variables which we must use in our different methods to construct the model for our algorithm.

Therefore, we need to incorporate a set of filters in order to avoid over-parametrization and to rank the attributes of the dataset regarding their relevance which can be checked in the code which is in a Github repository [22].

These so called "filters" are no more than code commands to strip/clean our dataset of variables that will not be used in order to optimize the computation time and the memory of the system. Also, to these filters we must assure that we have sufficient data to propose a model for each antibiotic and description. So, inside each loop to compute the model we will set conditions regarding these two categories, assuring the number of patients for a given set of condition-antibiotic is enough to compute a model that can be relevant. In other words, if the number of patients is minimal we will skip the computation of the model for this case because we would be strongly biasing the model.

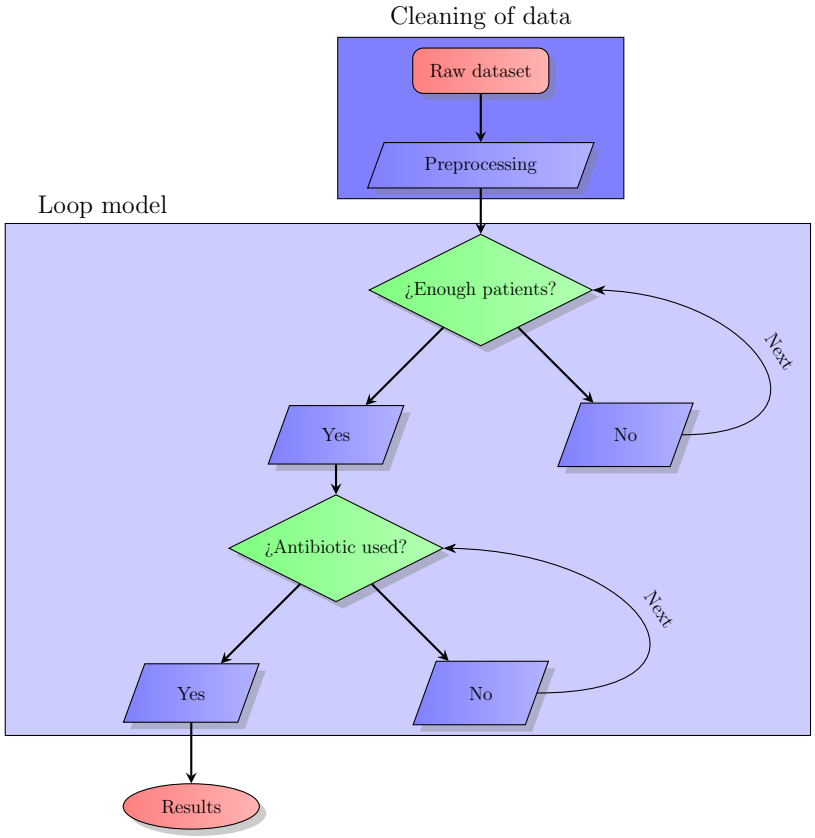


Figure 9: Flowchart of the model-loop.

4.2 Classifiers

To address our problem effectively, we have decided to explore and evaluate several classification methods to identify the most suitable one for our work. This process involves testing and evaluating each classifier using our data to determine which performs best in terms of the selected metrics 4.3. Over this section, we will provide an overview of three classification methods we have tested. By doing so, we aim to offer the reader a comprehensive understanding of the context behind our approach. Thus, facilitating the comprehension of the methodologies employed in this research and approaching them to the clinical specialists in the field.

4.2.1 Multinomial Logistic Regression

Since our problem is essentially a classification problem because our goal is to find the optimal antibiotic (*"C'est à dire"* search for the most suitable antibiotic regarding their effectiveness) to treat our patient. Therefore, the first method selected to carry out the selection/classification is using a Logistic Regression, in this specific case we will use a Multinomial Logistic Regression.

4.2.1.1 Theoretic frame

$$Pr(Y = k|X) = \frac{e^{\beta_{k0}+\beta_{k1}X_1+\beta_{k2}X_2+\cdots+\beta_{kp}X_p}}{1 + \sum_{j=1}^{K-1} e^{\beta_{j0}+\beta_{j1}X_1+\beta_{j2}X_2+\cdots+\beta_{jp}X_p}}$$

or if we take logarithms we can compute the log odds as:

$$\log\left(\frac{Pr(Y = k|X)}{Pr(Y = 0|X)}\right) = \beta_{k0} + \beta_{k1}X_1 + \beta_{k2}X_2 + \cdots + \beta_{kp}X_p$$

Where Pr is the probability that a case is in a particular category, β_{kj} is the intercept for category k, and the rest of the k's are the coefficients for the predictor variables. Each one of the β_{kj} indicates the change in the log odds of the dependent variable being in category k (compared to the baseline category) for a one-unit change in predictor X_i. The second equation is the odds ratio (OR), estimates the strength of the association between two events, in the example between category k and the baseline 0[23].

Multinomial logistic regression (MLR) is a statistical analysis method used to predict outcomes of a categorical dependent variable with more than two categories. It is an extension of binary logistic regression, suitable for scenarios where the response variable has three or more possible discrete outcomes [24]. We describe briefly here the primary advantages and disadvantages[25]:

- **Advantages**:

1. **Flexibility**: MLR is specifically designed to handle dependent variables with more than two categories, making it suitable for our problem.
2. **Non-Linearity**: MLR does not assume a linear relationship between the independent and dependent variables. This flexibility allows it to model complex relationships more accurately.
3. **Probabilistic interpretation**: MLR provides the probability of each possible outcome of the dependent variable, offering a clear and interpretative understanding of the likelihood of each category occurring given a set of predictor variables.
4. **Nature of predictors**: This method can include both continuous and categorical predictors, making it versatile for various types of data and allowing for more comprehensive models.

- **Disadvantages**:

1. **Large sample size requirements**: MLR can require a large sample size, especially when there are many predictors or categories. This is because the model needs enough data to reliably estimate the parameters for each category.
2. **Complexity in interpretation**: Interpreting the results of MLR can be more complex compared to other methods.
3. **Assumption of Independence of Irrelevant Alternatives (IIA)**: MLR assumes that the odds of preferring one category over another are independent of the presence or absence of other categories. This assumption can be unrealistic in many practical situations, potentially leading to biased estimates.

As cited at the beginning of this section, we will use a multinomial regression is because our variable to model is the sensitivity of each antibiotic which is a nominal (unordered) variable since it can take three exclusive categories (Sensible - S, Intermediate Sensible - SI and Resistant - R) [26].

Another useful advantage of this analysis is that it doesn't affect if they're certain independent variables that are statistically related which in our case is really helpful since in the current state of art we know little of which conditions of the patient affect the effectiveness of the antibiotics. But we're assuming that these variables have weak co-linearity.

Moreover, this type of analysis is widely implemented in the software (check the Scikit-Learn guide [27] or [28]) used for this work. Making really easy the construction of the commands to analyze the dataset and to fit the models in order to check effectiveness and discrimination. Our work will be to mechanize the process to compute the models and implement the filters explained in the beginning of this section. Then we will need to save the results for posterior analysis and fitting of the model.

4.2.2 Random forests

Random forest is a versatile and powerful machine learning method commonly used for either classification or regression problems. It's consists in constructing a large number of decision trees during training and outputs. The mode of the classes (for classification) or the mean prediction (for regression) of the individual trees. The idea behind RF is that a group of "weak learners" (decision trees) can combine to form a strong learner. Leading to improved performance and robustness compared to an individual decision tree.

4.2.2.1 Building the Forest

Each tree is built using the RF method by a process known as bootstrap aggregation (or bagging). This procedure uses random sampling to create several subsets of the initial training data. Then, a different decision tree is used to train, which enables each tree to be slightly different from the others. This variety among trees is essential as it reduces the over-fitting, which is a common problem in single decision trees.

In addition, the algorithm introduces another layer of randomization to each tree construction. At each split in the tree, a random subset of the features is chosen, and the optimal split is selected from this subset. Known as feature bagging, this method guarantees less correlation between trees. Further improving the robustness of the ensemble model. Thence the predictions of RF's lead to a more accurate model.

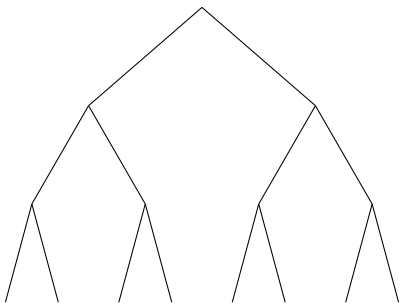


Figure 10: Example of the map of a tree, each node corresponds to an classification label associated to a question or evaluation of some certain value.

4.2.2.2 Prediction and Aggregation

Once the RF is has been constructed, predictions are gathered from the predictions of all the individual trees. In a classification task, each tree in the forest "votes" for a class label, and the class with the most votes is selected as the final prediction. For regression tasks, the predictions are averaged to produce the final output.

The key strength of RF's is their ensemble approach, which usually leads to better generalization on unseen data compared to a single decision tree. By combining predictions

from many trees, RF's resolves the problem of over-fitting. In particular when dealing with high-dimensional datasets of datasets with complex interactions among features.

4.2.2.3 Importance and Interpretation

One of the advantages of RF's is their ability to provide insights into the importance of features in the prediction process. After training, RF's can rank the features based on their contribution to the model's decision-making. This is typically done by measuring the average decrease in the Gini index (for classification) or mean squared error (for regression) across all trees in the forest when a particular feature is used. Features that contribute significantly to the reduction of impurity or error are deemed more important.

The ranking of feature importance is valuable tool in number of contexts, including feature selection, model interpretation, and gaining domain insights. However, while RF's are able to identify which influential features, they are often considered "black box" models due to complex of the individual decision-making process within the ensemble. Therefore, while RF's can provide an understanding of feature relevance, they may not offer a comprehensive insight insights into the relationships between variables or the rationale behind specific predictions.

4.2.2.4 Decision trees

Decision tree classifiers are a type of supervised learning algorithm used for both classification and regression tasks. They work by splitting the data into subsets based on the values of input features, creating a tree-like model of decisions and their possible consequences [29].

1. **Tree Structure**: A decision tree consists of nodes, branches, and leaves:
    - Nodes: Represent a feature or a split of data. It is were a branches starts.
    - Branches: Represent the outcome of a decision rule applied to the feature.
    - Leaves: Represent the final class labels (for classification) or continuous values (for regression).
2. **Splitting**: The process starts at the root node and splits the data into branches based on a feature that results in the most homogeneous sub-groups. Common criteria for splitting include Gini impurity, entropy (information gain).
3. **Recursion**: The splitting process is applied recursively to each branch, forming a tree structure. The recursion continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of samples per leaf.

4. **Pruning**: To avoid over-fitting, decision trees may be pruned by removing branches that have little importance or by setting constraints on tree growth. Pruning helps in maintaining the generalization capability of the model.

This method relies in a completely different type of mathematical hypothesis has it's own set of advantages and disadvantages since the algorithm is constructed by making decisions in each node that it encounters as if we're climbing a tree (we will give a more precise explanation on the criteria that builds the branches in following sections). Here we list the advantages and disadvantages that we consider most important for our work and for better comprehension of the reader[27].

- **Advantages**:

  1. **Interpretation**: Decision trees are easy to interpret and visualize see Figure 10. They clearly capture the hierarchy of features and the decision-making process.
  2. **Non-Linearity**: They can model relationships between features without requiring linear relationships.
  3. **Ease of Use**: They require little data pre-processing and can handle both numerical and categorical data.
  4. **Cheap resource**: In general, the run time cost to construct a balanced tree is $\mathcal{O}(n_{samples} n_{features} \log(n_{samples}))$. Although the tree construction algorithm attempts to generate balanced trees, they will not always be balanced. Assuming then that the sub-trees remain approximately balanced, the cost at each node consists of searching through $\mathcal{O}(n_{features})$ to find the feature that offers the largest information gain. This has a cost of $\mathcal{O}(n_{samples} n_{features} \log(n_{samples})$ at each node, by summing the cost at each node we find $\mathcal{O}(n_{samples} n^2_{features} \log(n_{samples}))$.

- **Disadvantages**:

  1. **Over-fitting**: Without proper pruning, decision trees can become overly complex and capture noise in the data.
  2. **Instability**: Small changes in the data can lead to significant changes in the structure of the tree.
  3. **Bias towards features with many levels**: Features with more levels or unique values might dominate the splitting process.

As we have seen decision trees have many advantages as disadvantages but we must also make a difference regarding the variables under study. We encounter in the bibliography types of trees:

- If the response variable is continuous then we can build **regression trees**.
- If the response variable is categorical then we can build **classification trees**.

As we mentioned in previous sections our dependent variable is categorical since it has three values that can take. So, we will select to work with classification trees as our variable to model (Sensitivity to the antibiotic) is categorical with three different levels or in some cases with two levels.

4.2.2.5 Classification criteria

Since we will be working with classification trees it is useful to give a general overview of how the trees are constructed and what criteria they follow in order to construct the classification. Given training vectors $\mathbf{x}_i$ and a label vector y, a decision tree recursively partitions the feature space such that the samples with the same labels or similar target values are grouped together[27].

Let the data at a certain node m be represented by Q_m with n_m samples. For each candidate split θ consisting of a feature j and threshold t_m, partition the data into and subsets Q_m^{left} and Q_m^{right}. The quality of a candidate split of node is then computed using an impurity function or loss function, the choice of which depends on the task being solved (in our case classification).

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta)) \tag{1}$$

The aim is to minimize the equation above by selecting the best parameters, to do so we use the measurements of impurity $H(Q_m)$ such as Gini:

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{2}$$

or Log Loss or Entropy:

$$H(Q_m) = -\sum_k p_{mk} \log(p_{mk}) \tag{3}$$

where p_{mk} is the proportion of class k observations in node m or what is similar, the number of k child nodes seen from the node m [30].

4.2.2.6 Post Pruning

Post-pruning is performed after the tree has been fully constructed. The idea climb back the tree and remove branches that provide little improvement in prediction performance [31]. The most widespread method is cost-complexity pruning which is a method that introduces a regularization term to penalize the complexity of the tree. The complexity

of a tree T is often penalized based on the number of terminal nodes (leaves) it contains, denoted as $|T|$. The cost-complexity criterion is defined as:

$$R_\alpha(T) = R(T) + \alpha|T| \tag{4}$$

Where:

- $R(T)$ is the empirical risk (e.g., the mis-classification rate or mean squared error) of the tree T on the training data.
- $|T|$ is the number of leaves (or terminal nodes) in the tree
- $\alpha \geq 0$ is a complexity parameter that controls the trade-off between the tree's complexity and its fit to the data.

Every time a tree is grown the pruning is performed following the next process [32]:

1. **Grown the full size tree**: Start by growing the tree until every leaf is pure or contains fewer than a minimum number of samples.
2. **Compute the cost-complexity** for each sub-tree:

$$\alpha_m = \min_{\text{subtree } T_m} \frac{R(T) - R(T_m)}{|T| - |T_m|} \tag{5}$$

   This formula calculates the value of α for which pruning the sub-tree T_m yields the smallest increase in the overall cost-complexity.
3. **Prune** the sub-tree that provides the smallest α_m.
4. **Repeat** the process, generating a sequence of pruned trees.

Finally, choose the pruned tree that minimizes the cross-validated error rate or another performance metric on validation data.

4.2.3 Neural Networks

On approach in machine learning is to use neural network, which is a way to tackle the problem of classification. A neural network is a graph of (artificial) neurons (i.e. weighted non linear function) linked with each other in some architecture (see figure 11). This defines a non-linear function that connect the input (the data) to the output (in classification: a decision function telling whether the NN outputs that it's in class 0, 1, 2,...). Hence, globally given data $x(t)$, the NN outputs $y = f_\theta(x(t))$ where y is the label of class, and f_θ is a non-linear function of some parameter θ.

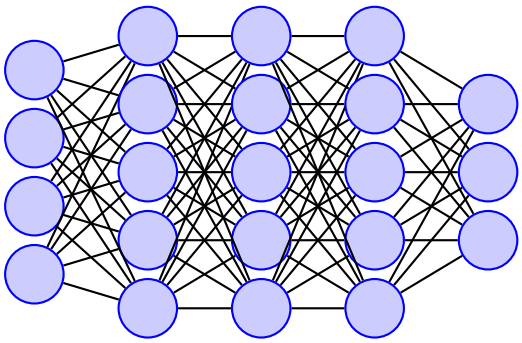


Figure 11: Example diagram of a fully connected neural network, in which the first column are the inputs and the last column are the outputs.

For a MLP as in Fig. 11, for just one layer, the output for tyhe k-th neuron is: $x_k^{(1)} = \sigma(\sum_i (\theta^{(1)})_i x_i + (\theta^{(1)})_0)$ where $(\theta^{(1)})_0$ is a bias term, σ a non linear function (e.g. a sigmoid like a tanh or the Rectified Linear Unit (ReLU) function). Then layer 2 is obtained by an analog combination from layer 1, and so on (here up to layer 4). Then, for a given input $x(t)$, the class of an example proposed by the NN is supposed to be the one of the neuron in the last layer with the highest value.

4.2.3.1 Stochastic Gradient descent algorithm

They are many different algorithms used in ML, such as simulated annealing, pass descent,... to could be used to fit properly the weights of the neurons. The most convenient and now standard approach is to use Stochastic Gradient Descent; that's the technique used here because it is the more extended in the bibliography these days.

This method is adapted to suppose the following hypotheses:

- We can compute the derivatives of the loss function $\mathcal{L}$ from with respect each neuron of our system (this is done using automatic differentiation tools)
- The space of solutions is not necessarily convex
- It's possible to fit (also known as train) the correct decision function (output) to the input data by changing the parameters of the model in following (minus) the gradient or the loss. This gradient is estimated on some selected data point (possibly randomly, hence the "stochastic" name).

As you may see here we have introduced another machine learning concept, which is the loss function $\mathcal{L}$. The definition of the loss function is not other that a function capable of estimate the quantity of loss/cost of making a change on our network, in a more mathematical approach is a function which associates a real number with a event. This means that a machine learning problem can be seen as a optimization problem, where we train our network in order to find an optimal solution of the variables.

The main objective then it is to minimize this loss function by modifying the parameters of the neural network, these parameters are the weights of the links between each neuron, the bias of each neuron and the activation functions.

The algorithm of gradient descent can be written as the next equation:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \vec{\nabla} \mathcal{L} \tag{6}$$

Where here the letter $\mathbf{w}$ makes reference of the weights (seen as a vector), the sub index makes reference to the time pass, η is a so called hyper-parameter which name is learning rate and is the gradient of the loss-function $\mathcal{L}$. As said before the objective is to minimize the loss function $\mathcal{L}$ meaning we want the derivative of the function to reach a minima (ideally the global minima) in order to vanish then we can state that the weights of the neural network have converged to a solution that it is the optimal for computing.

4.2.3.2 Loss-function $\mathcal{L}$

Computing the loss-function it is a very critical part of machine learning because the our method depends highly on computing the derivatives of $\mathcal{L}$. There are many models of loss functions already proposed on the bibliography [33], depending of the task to be solved. Here we will give a short overview on them.

They exist various types of Loss-function on the bibliography, we give a short overview of the two most common used.

Mean squared error

In statistics, the mean squared error (MSE) of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. In machine learning, MSE may refer to the average loss on an observed data set. It is computed with the next formula:

$$\text{Loss}_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2 \tag{7}$$

Where N is the size of the set, Y_i is the value of each data within the set and $\hat{Y}_i$ is the estimator of the i-th set (usually the mean). This is often the loss function used in regression.

Cross Entropy

Cross-entropy is the default loss function to use for multi-class classification problems. In this case, it is intended for use with multi-class classification where the target values are in the set { 0, 1, 2, ..., n }, where each class is assigned a unique integer value. The

cross entropy is computed following the next equation:

$$\text{Loss}_{CE} = -\sum_{i=1}^{N} Y_i \log\left(\hat{Y}_i\right) \tag{8}$$

Where N is the size of the sample/batch, Y_i is the value of each data within the set and $\hat{Y}_i$ is the estimator of the i-th set (usually the mean). This is the loss function of choice in supervised classification.

4.2.4 Architecture

For our work we will use a model of Multi Layer Perceptron which is an architecture rather simple. But the idea is to use an approach to later on study more intricate architectures. As these kind of neural network architectures have the next advantages and disadvantages:

- **Advantages of Neural Networks**

  1. **High Predictive Power**: NN, especially deep learning models, are known for their ability to capture complex, non-linear relationships in data. In the setting of antibiogram categorization, where the aim is selecting the most effective medicament for a patient, this competence is advantageous. The model can learn patterns that might be missed by traditional methods, such as the interactions between bacterial species, patient demographics (e.g., age, weight, sex), and antibiotic susceptibility profiles. **Automated Feature Extraction**: A huge advantage of NN is their ability to automatically extract the relevant features of the raw data provided to them. In an antibiogram context, this would equate to allowing the model to find patterns. Such a capability is extremely needed for situations when the basic relationships are very complicated and not easily caught by simpler models [34].
  2. **Scalability**: NN are easy to scale up, to increase the dataset size as and when more data becomes available, and hence, they can potentially increase their performance by learning from this updating of information, leading to much better and strong predictions in the future.

- **Disadvantages of Neural Networks**

  1. **Interpretation Issues:** A possible drawback of neural networks, is their interpretation. Neural networks work as "black boxes," so it becomes arduous to explain why a particular antibiotic was suggested on the basis of input features [35]. This lack of transparency can be a barrier to clinical adoption.

2. **Data Requirements and Over-fitting:** Neural networks require large amounts of data to perform well, particularly to avoid over-fitting. Over-fitting occurs when the model learns patterns that are specific to the training data but do not generalize well to new, unseen data. This could result in incorrect antibiotic recommendations, which could have serious clinical implications.
3. **Computational Complexity and Resource Intensity:** Training neural networks, is computationally expensive and requires more resources than other models, including extensive training time. Additionally,fine tuning and optimizing the network architecture would require expertise and time, making the implementation of NN's more complicated compared to simpler models.

While neural networks offer significant advantages in terms of predictive accuracy and feature extraction, their drawbacks, particularly related to interpretation and resource requirements, must be carefully considered in the context of antibiogram classification. The choice to use neural networks should be weighed against these factors, and in cases where interpretation is crucial, alternative models or methods to enhance the interpretation of neural networks should be explored. [5]

4.3 Evaluation criteria

In this section, we will discuss evaluation methods for building our application, as we need to select the most suitable algorithm. Therefore, we must assess the efficiency and effectiveness of the models, meaning their overall performance. To achieve this, we need to develop criteria that allow us to quantify these variables, enabling us to make an informed decision.

The most common approach in data science to evaluate these characteristics is through metrics such as precision, accuracy, recall, F1 score, and kappa score, which assess different aspects of the model. These metrics are computed from the confusion matrix that is constructed after running the model. Thence, we can compute the number of true positives, true negatives, false positives and false negatives. Below, we provide a brief definition of each metric[36].

- **Precision**: This is the fraction of correct predictions for each class. It is defined as the number of true positives divided by the sum of true positives and false positives.

$$\text{Pre} = \frac{TP}{TP + FP} \tag{9}$$

- **Accuracy**: Similar to precision but with a slightly different nuance, accuracy corresponds to the fraction of correctly classified instances. It is calculated as the

number of true positives divided by the sum of true positives, true negatives, false negatives, and false positives.

$$\text{Acc} = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

- **Recall**: This metric measures the fraction of instances of a class that were correctly predicted. It is defined as the number of true positives divided by the sum of true positives and false negatives.

$$\text{Rec} = \frac{TP}{TP + FN} \tag{11}$$

Additionally, many studies include another metric called the F1 score, which is defined as follows:

- **F1 Score**: The F1 score represents the harmonic mean of precision and recall.

$$\text{F1} = \frac{2 \cdot \text{Pre} \cdot \text{Rec}}{\text{Pre} + \text{Rec}} \tag{12}$$

- **Cohen's Kappa**: This metric is defined as the measure of agreement between predictions and the original/true classification.

$$\kappa = \frac{\text{Acc} - \text{expAcc}}{1 - \text{expAcc}} \tag{13}$$

Where expAcc is the expected accuracy computed as $\text{expAcc} = \sum p \cdot q$ where $p = rowsums/n$ which is the distribution of instances over the actual classes and $q = colsums/n$ is the distribution of instances over the predicted classes.

Since our model is multinomial, these metrics are not ideal as they are primarily designed for binary models. In our case, we need to introduce modifications to adapt them to our three-class scenario (multinomial). This adaptation is achieved through micro/macro-averaging and weighted averages. While it is true that for a given antibiotic, sensitivity might only have values (Sensitive-Resistant), it is common to encounter cases where the patient exhibits intermediate sensitivity (Sensitive EI). Therefore, these metrics must account for the class distribution within the dataset, particularly when a certain class is more prevalent than others.

In such cases, the dataset is termed imbalanced, and it is possible that the model's performance might appear poor due to a bias towards the majority class. To address this, the micro metric becomes crucial for evaluating the model's classification performance for each class, while the macro metric allows us to assess how well the model performs across all classes independently. Finally, the weighted average metric adjusts each metric

by weighting it according to the prevalence of each class in the dataset, summing the weighted metrics thereafter. This approach aims to balance the influence of each class on the metrics, thereby accounting for the presence of imbalanced dataset [37]

We now demonstrate how these modified metrics are calculated without including the definitions, as they have already been provided at the beginning of this section.

1. **Micro-Averaging**

   (a) Precision:

$$\text{Pre}_{\text{micro}} = \frac{\sum_{i=1}^{N} \text{TP}_i}{\sum_{i=1}^{N} (\text{TP}_i + \text{FP}_i)} \tag{14}$$

   (b) Recall:

$$\text{Rec}_{\text{micro}} = \frac{\sum_{i=1}^{N} \text{TP}_i}{\sum_{i=1}^{N} (\text{TP}_i + \text{FN}_i)} \tag{15}$$

2. **Macro-Averaging**

   (a) Precision:

$$\text{Pre}_{\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} \text{Pre}_i = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{16}$$

   (b) Recall:

$$\text{Rec}_{\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} \text{Rec}_i = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{17}$$

3. **Weighted Average**

   (a) Precision:

$$\text{Pre}_{\text{weighted}} = \sum_{i=1}^{N} w_i \cdot \text{Pre}_i = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} \cdot \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{18}$$

   (b) Recall:

$$\text{Rec}_{\text{weighted}} = \sum_{i=1}^{N} w_i \cdot \text{Rec}_i = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} \cdot \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{19}$$

Where N is the number of classes, i index referring to the i-th class, n_i is the number of instances in class i and w_i is the weight of class i, usually defined as:

$$w_i = \frac{n_i}{\sum_{j=1}^{N} n_j} \tag{20}$$

Finally, to complete our evaluation criteria for the classification models in this study, we will employ a widely used and popular tool across various fields of knowledge: ROC curves and the area under the curve (AUC-ROC).

The ROC curve is a graphical representation of sensitivity versus the false alarm rate, specifically described by the false positive rate on the x-axis and the true positive rate on the y-axis [38]. A random classifier would be represented by a diagonal line, that is, the TPR-FPR curve with a slope of 1. Therefore, a good classification model will have a convex ROC curve, indicating that the area under the curve is close to 1, whereas a poor model will have an area close to 0.5, and a bad model will have a low AUC and a concave curve [39].

With these metrics defined and graphical representations provided, we have useful tools to develop an analytical criterion capable of evaluating the proposed statistical models with clarity and rigor.

4.4 Materials and software

In order to achieve this work a Dell Latitude 5420 laptop was used to develop the codes for each statistical analysis with a processor 11th Gen Intel(R) Core(TM) i5-1145G7 @ 2.60GHz 1.50 GHz with a RAM memory of 8.00 GB.

Regarding the technologies used to make this work were used different sources. The list below show the details of each one and the purpose:

- **Visual Studio Code**: is a free editor of source-code which supports many programming languages such as Python, C, Java and many more [40]. It was used during the exploratory analysis as it can be seen on the appendix [22].
- **Rstudio**: this editor is based on R-language which is a powerful tool to carry on statistical analyses thanks to the fact that incorporates many of the functions and algorithms used in this work [28]. It was used to construct the pipeline to analyze the data-set as it can be see on the appendix [21].
- **H2O.ai**: is an open source library developed on Java and supported in Rstudio specialized that allows you to build machine learning models on big data [41].
- **Overleaf**: was used to write down this memory since is an open-source LateX compiler which allows the user to share the work on real time which eases collaboration [42].

4.4.1 Datasets

This project uses simulated data from an original dataset which is simulated and provided by NTT Data in which it's exposed the results of several antibiograms done to patients with infections. This dataset is fully anonymized because the entries doesn't allow to identify a patient. It is mandatory to describe the structure of our data and it's origin to give some context. Initially we had a .csv archive with the next variables/labels:

- Fecha:date of the test, in format (dd/mm/aa) where "dd" is day, "mm" is month and "aa" is year.
- Fecha de solicitud: Date and time in which the test was requested. With the same format as the label Date adding "hh" which is the hour in 24 hours format, and "mm" are minutes.
- Extracción: an empty column with out content.
- Fecha de activacionn: Date and time in which the test was activated to be done. Is presented in format (dd/mm/aa hh).
- Numero identificativo: An ID number unique for each entry (patient). Given to distinguish each patient in the system.
- Fecha de nacimiento: birth date of the patient.
- Sexo: sex of the patient, specified as masculine (M) or feminine (F).
- Edad: age of the patient at the time of the test.
- Tipo de paciente: Category of the patient, as an example, if a patient is UCI, hospitalize, ambulatory ... etc.
- Servicio: Department which requested the test, UCI medic, UCI quirurgic, UCI central.
- Prueba (HCUL): refer to an blood culture (HCUL is an acronym for "Hemo-culture/Blood culture").
- Muestra: the type of sample obtained for the test for example, blood, urine, tissue ...etc.
- Nº cultivo: number of the culture, if the test involves a microbiologic culture. Identifies the specific culture.
- Descripción: name of the organism causing the infection. Descries the infection detected in the culture for example, "Staphylococcus hominis", "Escherichia coli"... etc.
- Mecanismo de Resistencia: specific detail describing the resistance of the organism.
- Pool Antibioticos: name which encrypts the list of antibiotics used during the test.

Along these labels we find two columns for each antibiotic. One for the sensitivity (label which can take 3 values: Sensible, Sensible EI and Resistente) and one for the CMI which depending on the antibiotic can take certain arbitrary values. Therefore, we decided to train the models as category classification because the CMI have different range of values for each antibiotic making difficult to standardize the application of the algorithms against the label Sensibility. In addition for a certain CMI may it be possible that the concentration of the drug is toxic to be used which adds up a level of difficulty to laying the foundations of our project. For this reason we agreed on using the Sensibility as the variable to model and add these kind of considerations as improvements on the long run.

On the other hand, many of these variables are useless to us. Either because they are not etiologic variables to affect the test or we find null values. Thence, we must remove them. To do so in our project we used the script developed in python that it may be found on the appendix [22], also many entries in the original dataset were duplicated and had to be deleted.

From a dataset which contained originally 900 entries after applied the filtering we end up with a 211 patients dataset. To this we must add the fact that not all the patients have the same antibiogram which is due to the label organism. Depending on the organism a certain set of drugs are tested which it reduces the dimension of the partitions in which we wanted to test the models, as an example for the most populated case which is *Staphylococcus Hominis* we had 76 entries with antibiograms out of the 211 patients.

For this reason to obtain results with statistical consistency we agreed on enlarge our dataset. Thence, in this project we use those dataset constructed using the methods and methodology developed in this work [43].

After the indicated filtering work which formed the foundations of the structure of the database to be used in the future and the posterior simulation of the data by my comrade we have two datasets. First one to train our model with 10000 entries and a second one with 1000 entries which will serve to test the models. Both datasets have the next structure:

- Fecha.
- Sexo del Paciente.
- Edad del Paciente.
- Descripción.
- BMI: added via sampling following the distributions of weight in Spain [44].

Where the variables with same name as originally have the same meaning as they did originally and the dataset is lighter for the code editors to handle lowering the amount of computation time we need to extract information of the dataset.

5 Results

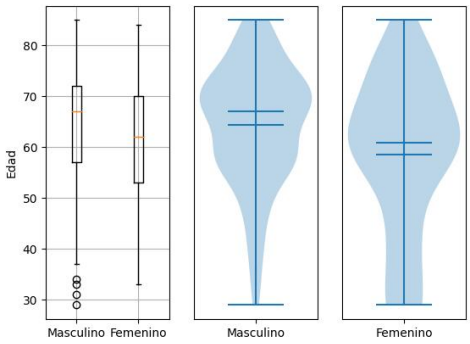
5.1 Original Dataset analysis

Before modeling our datasets we comprehend the data we have at our hands. That's why the first code developed to have a quick description of the original dataset was the [22]. This code have two sections, first one it cleans the dataset to be lighter formatting the columns that are useless for the modeling. And next part it builds some visual graphics to extract useful information as you may see in Figure 12.

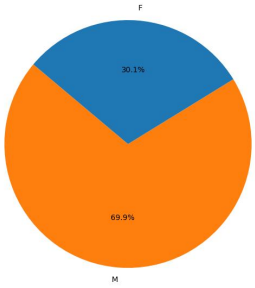
Of this picture we can extract first that the distribution of ages is quite old. As we can see in Figure 12a for both sexes the mean is between 60 to 70 years old which allow us to construct a profile of the type of patient which is more likely to be affected by bacterial infections upon admission in the healthcare system. Also, by looking to the violin plot we see a different distribution for sexes, males have more propensity to be infected as the age up but after passing the 70 years mark the proportion of males admitted decreases. While for women the proportion under 50 years admitted to the system is quite uniform only to be increased after passing this mark and after passing the 65 years the proportion starts decreasing.

From this we can extract that while in early ages the males have low propensity to be infected we have a regular proportion of women admitted. As both groups age up we can say the risk of being infected grows and decreases more likely due to the fact that the hope of life decreases.

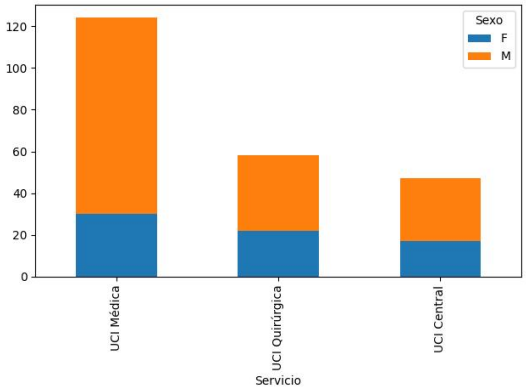
More information we can extract by looking at Figure 12b is the fact that the proportion of males infected is almost the double as the women. This is also covered on the next two graphs (see Figure 12c) where we see the proportion of women in some cases to be minimal while the proportion of males of males is prominent in the all services. In addition, in Figure 12d have plotted some the proportions vs the first 10 more populated cases of infection and we see that the infection caused by the *Staphylococcus's epidermis* is by far the most dominant in the dataset. This may be due to the fact of the bacteria is present in the most part of the normal microbiotic in the skin and in the mucous membrane of humans[45]. This bacteria is not harmful as we stated is present among humans but in the context of hospitals it becomes dangerous due to the capacity of the bacteria to form bio-films which protect the bacteria from antibiotics [46] which makes easy for patients to acquire a so called hospital-acquired infection [47] thus risking other patients to be infected and propagate the resistance to antibiotics perpetuating the problem of



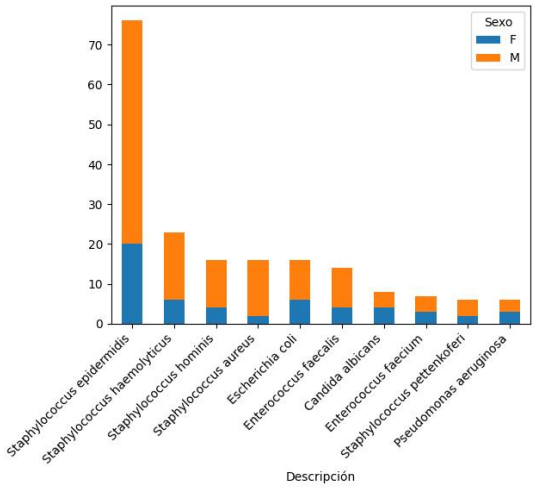
(a) Distribution of the Age for the sexes.



(b) Proportion of males vs females inside the original dataset.



(c) Distribution of sexes by Service that requested the test.



(d) Distribution of sexes for the most populated infections.

Figure 12: Results of computing the first code [22] for the original dataset.

AMR.

5.2 Model interpretation

Since in the previous section we've seen that the case of the pathogen *Staphylococcus Hominis* is the most populated pathogen of our data original dataset. Also, we've chosen the antibiotic Penicillin as example control for it's universality and for the same reason of the pathogen it contains the most entries in the original dataset. During this section we will be covering a more exhaustive analysis of the results from each algorithm. We will focus on the insights we can gather by using the different approaches such as generalized linear models, tree classification or neural networks.

In these analysis we will study the inference we can extract from each model. That is, what interpretation we can derive from running each algorithm on our datasets. Furthermore, we will examine the desired advantages and disadvantages of the model's, as

well as the extent to which these are beneficial or detrimental.

For all three algorithms we used the same conditions that were using the large dataset with 10000 entries as training set and to validate we used the 1000 entries dataset. We haven´t considered interaction terms among variables in the algorithms since Random Forests doesn't support this fact.

5.2.1 Multinomial Logistic Regression

After running the first algorithm selecting only those entries with *Staphylococcus Hominis* as pathogen. The following table is calculated for the antibiotic Penicillin:

	Estimate	Std. Error	t-value	Pr(>|t|)
Intercept	2.761	0.131	21.10	0.000
Sex	0.235	0.154	1.52	0.138
Age	-0.079	0.062	-1.28	0.21
BMI	0.038	0.069	0.56	0.577

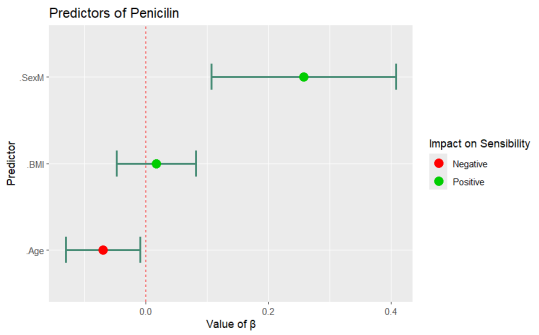


Figure 13: Coefficients estimated for the predictors of the Penicillin and their plot.

Regarding the results from table and the figure (see Figure 13) we can observe for the Sex (Male) predictor an estimated positive value. Therefore, meaning for fixed values of Age and BMI, the probability of being Resistant increases for Males as they are coded as 1 in our model. In terms of the log-odds, this would mean that changing from female to male (unit change) is associated to an increase 0.235 units [31].

For the case of the third predictor which is the BMI we obtain a similar conclusion for the log-odds but in this case the impact of this variable would be less as the value is one order of magnitude less than the Sex estimator. In terms of the probability we obtain similar conclusions, as the BMI is higher it would mean an increase in probability of being Resistant.

Finally, for the Age predictor it's the reverse because the value is negative which implies that a higher Age we have more probability of being Sensitive. Besides, we would have a -0.079 unit change in the log-odds for a unit change of 1 in the Age.

Although these results are interesting we must contextualize them. This means we have to study the rest of the statistical results we obtained form GLM. First let‘s focus on the standard error, for all the predictors the order of magnitude of the error is similar to the estimate. This means coefficient points have large imprecision.

Regarding the view of the statistics t-test and p-value we see best results for Age and Sex but they are not enough to provide evidence to reject the null hypothesis which means we cannot assure these variables govern the probability of the Sensitivity. For the

IMC the conclusion is similar because the values are in the case of t-value less than the counterpart of the other predictor and for the p-value is higher.

In the next figure 14 we've decided to represent the log of p-value vs the value of the estimator β obtaining a figure similar to a volcano eruption. We've included all the drugs for the given organism *Staphylococcus Hominis*.

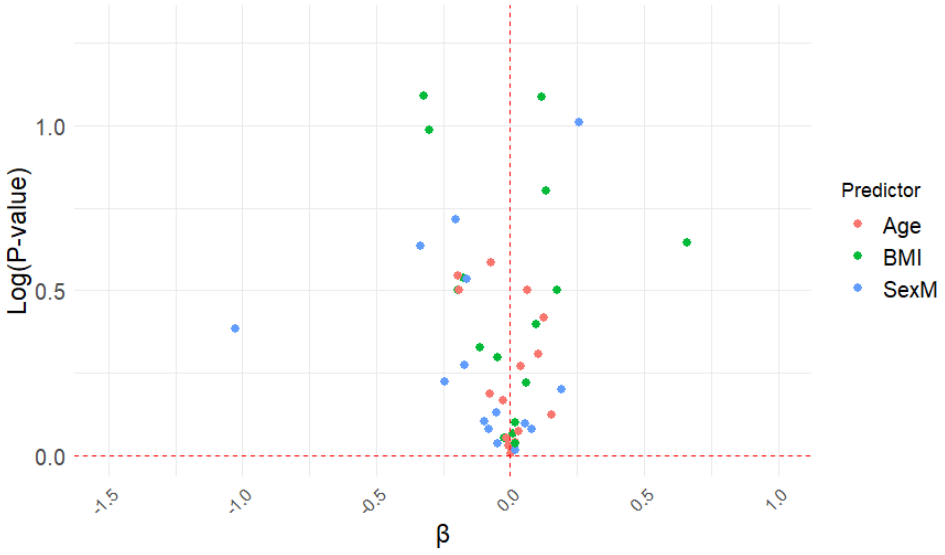


Figure 14: Volcano plot of all predictors for all the drugs using *Staphylococcus Hominis* data.

This type of graph is useful to observe which predictors are highly significant with respect to the other similar predictors because those with lower p-value have a higher position in the graph. Regarding the distribution of the value for each coefficient we see there's no clear clusters for each variable. In fact, the distribution for each variable seems to be random which it forbid us to throw further conclusions. Also, as we stated previously above, we encounter higher p-values (in the graph it means lower values of the logarithm). Thence for the vast majority of estimators we cannot reject the null hypothesis.

We can compute the metrics of the model (as we will do later for comparison). Thence we obtain the next table:

RMSE	AUC	PR-AUC	Gini
0.82	0.67	0.37	0.38

Table 1: Performance metrics for the GLM using sensibility to Penicillin.

An RMSE of 0.82 is rather high since our predictions take values between 1 and 3. This indicates a moderate error for our predictions. An AUC of 0.67 is moderate. Suggesting our GLM model have a decent performance but with enough window to improve. We can

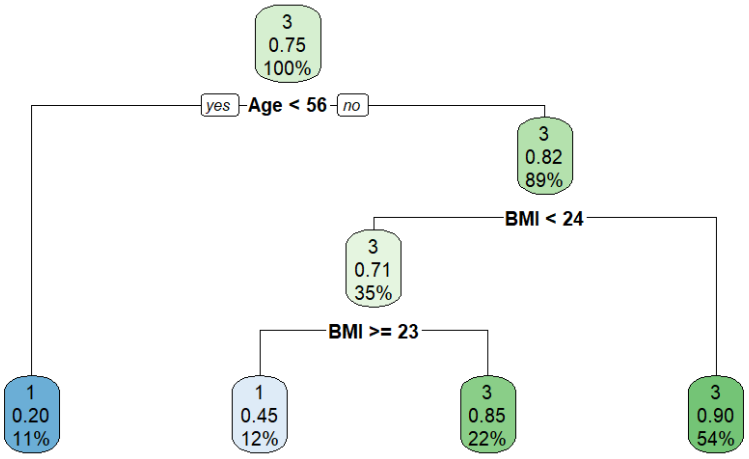


Figure 15: Plot of the decision tree.

conclude the same for the Gini coefficient. And lastly a 0.37 value of PRAUC suggests that our model is having difficulties to identify positive cases which in a un-balanced case this would be problematic.

5.2.2 Tree classification

Seen the results for the GLM algorithm we present in this section the analogous for the second algorithm. For this algorithm we choose a number of 50 trees with 15 depth and penalty coefficient of 0.01. In the first table 2 we see the relative importance's for each variable. As seen by the results from the table, the BMI is the most important variable

	Relative importance	Scaled importance	Percentage
Age	6.97	1.00	63
BMI	3.53	0.50	32
Sex	0.55	0.08	5

Table 2: Importance's computed via tree classification for Penicillin.

to predict sensibility, followed by Age and the less important variable we have the Sex.

So as to visualize better these results from classification we are able to plot the tree, obtaining the figure 15.

By inspecting the tree we can visualize the hierarchy really easy since in first node we encounter the Age followed by BMI in the child nodes. First let's remark that

RMSE	AUC	PR-AUC	Gini
0.35	0.79	0.88	0.58

Table 3: Performance metrics of the tree

First, for the RMSE metric which evaluates the root of the average of the squares of the errors. The lower the value, the better the classifier. Hence, the predictions from the tree classifier have a mean error of 0.35.

For a good classifier we would have values of AUC and PR-AUC closer to 1 since it would mean there's a good discrimination. Meaning the model is able to classify observations into the real classes. In our case, the value for the AUC is closer to 1/2 meaning the model, in the case of the Penicillin, is predicting randomly the classes. As for the PR-AUC, the value is better than his homologous, which means the classifier is better in distinguishing between two classes than to give the correct label to the observation.

Lastly, for the Gini Coefficient we encounter a value close to null. Since, Gini index quantifies the inequality among values a Gini closer to 0 would express perfect equality of cases. Therefore, our model have a rather bad capability of discrimination but with enough space to improve. Since a optimal model would have a value closer to 1.

5.2.3 Neural network classification

As done in previous subsections we present the results for the Penicillin sub-set upon running a neural network of 200 neurons and 100 where the link activation function was the Rectifier function. And the algorithm was computed for 50 epochs with no stopping rounds. Similar to the Tree algorithm we have an importance's table:

	Relative importance	Scaled importance	Percentage
Sex(M)	0.97	0.87	0.39
BMI	0.36	0.36	0.14
Age	0.34	0.34	0.13

Table 4: Importance's computed via neural network for Penicillin.

An advantage from neural networks is that we can plot the partial dependency of the probability of being labeled in the class Resistant by the neural network. Obtaining the next figures:

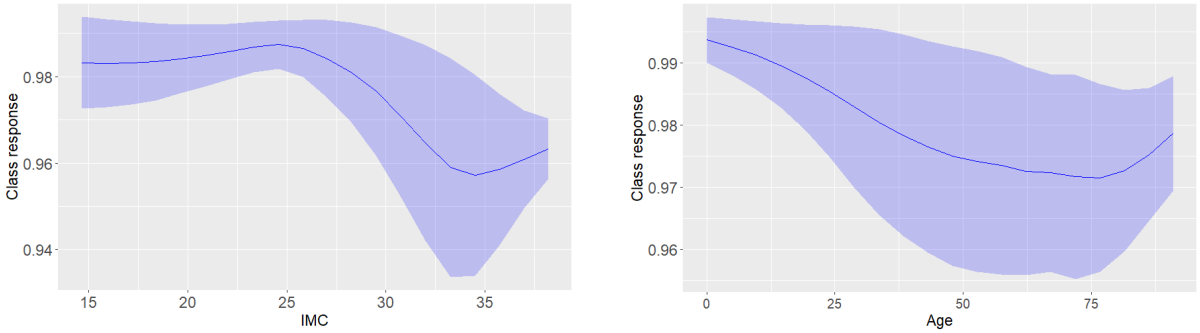


Figure 16: Partial dependency plots for IMC on the left and for Age on the right.

On these figures we can observe a common dependency of the decrease of probability to be labeled Sensible as both variables Age and BMI increase. Then we can gather from these plots that as a person is older have higher probability of being Resistant upon infection. While for BMI the overall behavior is similar but as BMI surpass the threshold of 25 it suffers another decrease in response.

Now as before, the algorithm have metrics to evaluate it's particular performance. We've selected the same metrics as the tree classifier. Also, due to the particular case of the Neural-Networks we add to the table the Log-Loss or Cross Entropy.

RMSE	AUC	PRAUC	Gini	Log-loss
0.25	0.58	0.77	0.16	0.16

Table 5: Metrics of the neural networks Penicillin.

By visual inspection of the table we encounter a value of RMSE less than the previous one. As for the rest of the metrics, their values have increased. Meaning this model is able to have better precision. But the values from AUC, Gini index indicate that it doesn't discriminate as well as the previous. Which indicates that it is giving labels almost randomly. The PR-AUC states that although is giving classes almost randomly it is good at distinguishing between two different classes.

For the last metric, the Cross-entropy value. It's value is give us information on how good are the probability predictions, in this case since is a 2 case prediction. These predictions are near real values.

As a resume we can conclude that our neural network is performing decent in terms of precision and quality of probability generated. Although since the values of Gini index and AUC are bad we are encountering difficulties to separate the classes. Which is a problem on how the model is capable to discriminate.

5.3 Model Performance comparison

During this section we will be covering the information we extracted after running the algorithm [21] over both synthetic datasets. The large dataset contained 10000 entries was used as train-data while the test of the models where done using the 1000 entries dataset.

For sake of the reader the major part of images are in the appendix where they can be consulted, here we have selected 6 images (see Figure 17) to characterize the functioning of the 3 algorithms used to model our data. They correspond to 2 antibiotics (Tetracycline and Trimetoprim). We have selected this curves among the all generated because for this 2 antibiotics we are sure we can encounter the three classes of the Sensibilidad (Sensible which is 1, Sensible EI corresponding to 2 and Resistente which is class 3). In many other

of the antibiotics we encounter a major presence of one of the classes over the remaining two classes.

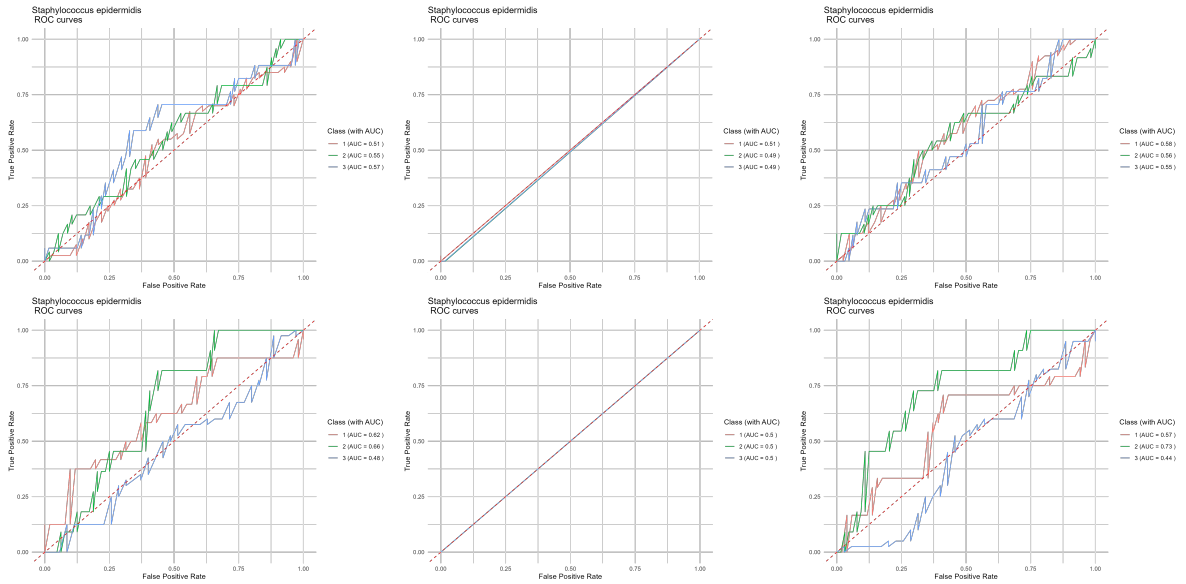


Figure 17: From left to right GLM - Tree - NN comparative view of the ROC curves. First row corresponds to Tetraciclina antibiotic and second row is for Trimetoprim Sulfametoxazol.

By inspecting the figure the most shocking thing is the performance of the Tree classifier with regard to the remaining classifiers. The ROC curve in this case is diagonal. As we explained in previous sections (see sub -section4.3) when the ROC curve (which is the representation of the false positive rate vs true positive rate) is diagonal the classifier is random, meaning it is classifying classes with equal probability, or what we interest us more, it doesn't adapt to the problem.

Inspecting furthermore the figures we see that for the GLM and NN algorithms the ROC curves seem similar in the case for the first antibiotic, this may be due to the distribution of classes in the dataset. Both curves are smother and close to the diagonal which means in this case the algorithms are performing quite randomly therefore the models haven't capability to discriminate well between classes. This is also supported by the AUC which is presented along each graph.

For the second antibiotic the performance of the GLM and NN models is slightly better because the curves have higher value of AUC under them in two out of three cases which means the classifiers are good to distinguish classes. As a matter of fact both algorithms have good capability to predict a Sensible EI case.
But to fully characterize the 3 methods of classification we need more strict tools than a visual analysis. Plus since the amount of graphs provided by the code is huge (see the A) we computed the metrics presented in section 4.3. Below we provide here the results of the analysis resumed in the Table 6 where we can find the results for the metrics computed by doing the mean overall the antibiotics processed by each model.

Metric	GLM Model			Tree Model			Neural Networks		
	Macro	Micro	Weighted	Macro	Micro	Weighted	Macro	Micro	Weighted
Precision	0.332 ± 0.007	0.436 ± 0.016	0.679 ± 0.046	0.208 ± 0.062	0.178 ± 0.062	0.177 ± 0.071	0.695 ± 0.008	0.689 ± 0.012	0.579 ± 0.047
Recall	0.434 ± 0.031	0.436 ± 0.015	0.446 ± 0.016	0.324 ± 0.014	0.178 ± 0.062	0.178 ± 0.062	0.569 ± 0.02	0.691 ± 0.014	0.695 ± 0.015
F1 Score	0.385 ± 0.011	0.436 ± 0.017	0.514 ± 0.025	0.321 ± 0.054	0.200 ± 0.062	0.130 ± 0.070	0.781 ± 0.022	0.691 ± 0.015	0.600 ± 0.026
Accuracy		0.436 ± 0.016			0.178 ± 0.062			0.691 ± 0.015	
Cohen's Kappa		-0.014 ± 0.010			0.008 ± 0.000			-0.001 ± 0.009	
AUC		0.543 ± 0.014			0.516 ± 0.003			0.677 ± 0.013	

Table 6: Comparison of Performance Metrics across Models

As before the difference between the metrics the Tree classifier with regard to rest of metrics is quite remarkable being the model with poorest values for all metrics. In this first case the model fails to perform well in the classification for each class as well to perform across all the classes. Also if it's the case of highly imbalanced datasets. Thence we can conclude this is the worst model for our problem which would be taken in the future.

Inspecting the section of the table which corresponds to the GLM model we see a better performance at micro level than at macro level which would mean it would be a good classifier to perform independently. Still the values of the metrics are really low and under 50 percent (except for AUC) of cases we would have wrong predictions which means this classification algorithm is not performing well with our dataset. In case of weighted average we encounter better values of metrics but this fact is likely to be produce by the fact that the average is shifted towards the most populated class which also have more probability to be better estimated. Hence this "good· result should be taken with caution.

Finally, in the case of NN we have the better metrics over the three models. Regarding the Macro-point of view (and micro-point of view) we see moderate good precision, F1 score and AUC meaning the model is classifying moderately well each class. While having a less Recall meaning the predictions have more error. Looking at the third column we see similar values for the metrics. Hence, the NN in global performs in a similar way as if it were a unique problem.

Overall, from the three models the best results come of with the Neural Networks, this may be due to the fact that they have flexibility and doesn't relay on linear-relationships to predict relations.

6 Conclusions

Following the structure of the previous section, we will divide the conclusions into two subsections. First, we will summarize the conclusions on the Penicillin and it's contextualization. Secondly, we will examine the results from the global performance.

6.1 Penicillin comparative

What we have seen on the interpretation 5.2 for each model with regard to the subset Penicillin is that each model have certain advantages and disadvantages. Which now we will discuss during this section. First let's state that a common feature among our data is that is rather simple and it didn't required of a lot of pre-processing to adapt for each model. For that we will consider all models flexible enough to handle this kind of data.

We will start talking trough what advantages have arise for each model. In case of the GLM what we can highlight is that it allow us for a more individualized analysis for each variable. This is useful because for each estimator we have also statistical tests that give us insights on null hypothesis. Which then allow us to study falsifiability of the data. Another good feature from GLM is the For tree classification we've gathered that is an visual and quick method which is good when we want to see the hierarchy. And another feature is their good performance with respect to the metrics. Third for Neural Networks we haven't encountered any advantages that we listed on previous sections. This is due to the limitations encountered in our work. In small batches NN doesn't perform better than other methods.

When discussing disadvantages we see that for GLM the downside of this method is the failure on rejecting the null hypothesis. Which make us hesitate to take as good conclusions what we can gather from the predicted values for the estimators. Moreover, the performance values are rather low when analyzing Sensibility for the Penicillin. We encounter a model with low precision and moderate discrimination with room to improve. As for the tree classification there are no important disadvantages rather apart from the fact that is a type of method which doesn't support interaction between variables. Thus is more rigid than the other two. Finally for NN we encounter a method which is complex than their counterpart since to obtain visual results we have to transform the outcome. In addition, we have a methods with lower discrimination than fellow methods but with higher precision than the GLM. Thence, as we deduced for the GLM the conclusions from analyzing NN outcomes must be taken with hesitation.

Let us now contextualize our findings with those of the bibliography. We will compare our results from the penicillin section with the studies we have already cited along this work. In the articles, researchers had similar aims to ours, although we differ in our evaluation techniques. Hence, we will compare similar metrics from each study to avoid misleading conclusions.

Looking at the table 7 we can see that not all studies evaluated the same metrics. In fact, the evaluation methodology varies between studies. There are no clear benchmarks for this type of research. To contextualize our work we've selected the common metrics of AUC and PR-AUC from the available studies. With these two variables we're able to capture how precise is a model and how well can it discriminate between classes.

	Project			**Wang** [1]			**Farooq** [16]		**Corbin** [19]	**Aguero** [20]		
Metric	GLM	RF	NN	GLM	RF	NN	GLM	RF	RF	GLM	RF	NN
AUC	0.67	0.79	0.57	0.89	0.89	0.88	0.90	0.86	0.61	0.78	0.82	0.87
PR-AUC	0.37	0.88	0.77	-	-	-	-	-	0.97	-	-	-

Table 7: Comparative table of results from different articles.

Looking at the comparison table, we can draw the following conclusions from the models in relation to those in the bibliography. First, our models (GLM, RF, NN) generally have lower AUC values than those present in the Wang [1] and Farooq [16] research. This would mean that these studies are better optimized and have access to datasets that are more appropriate for predicting AMR. Our RF model achieved an AUC of 0.79, which is remarkable but still lower than other studies.

For PR-AUC values, we find a reasonable performance when comparing RF and NN. This is important when analyzing scenarios with unbalanced classes. However, these values have room for improvement, as in Farooq's report a PR-AUC of 0.97 was achieved.

The best model in our project is RF with a competitive performance in terms of PR-AUC but surpassed in AUC by other studies. As for the GLM, we find significantly lower performance values compared to those of the studies, suggesting a need for adaptation and improvement. Finally, for NN we have a rather low AUC (0.57), which indicates that it is not as well optimized as other models.

In conclusion, we can say that there is room for improvement. This could involve refining the choice of features, adjusting the hyper-parameters on the models. Or implement techniques of regularization of unbalanced classes [18]. Nonetheless, we must take in count with what type of data we were working. While on the research's they have access to repositories of antibiograms we did not have such capacity. Our data was simulated which can alter the results as the data could be not fully capturing the reality. Thus, with limited and simulated data it is possible to develop models with decent performance.

6.2 Global performance

The fact of the poor global performance of GLM and Tree classifiers can be due to the fact that the datasets are quite small and biased towards one class in many antibiotics. This is where NN regains its flexibility over the other models. In general, our metrics have lower values than those encountered in the bibliography [20] [16] for such cases.

As mentioned in the previous section, this could be due to the simulation of the train/test datasets. These datasets where built synthetically meaning they're not a construction of many antibiograms from test done. They were simulated using tools, software and databases (for more details on the process consult [43]) with relation to our problem.

In this process of building these dataset we can perform errors due to the lack of information in this field of knowledge or simply to the inability to fully capture the nature of the problem. Thence, not capturing the complexity and reality of antibiograms. Related to this is the lack of access to repositories with antibiograms [18]. All the articles cited in this work to compare the results had access to data banks with information related to patients and results of antibiogram tests. In order for us to achieve similar results we would need to access to real data. This would allow us to optimize the simulation of the data and refine the parameters of the models.

6.3 Challenges and opportunities

As discussed in the previous section, our models need improvements. First we would need to asses the problem of the quality and origin of the data. A real source would allow us to extract useful information and inference which later on would be useful for clinicians to construct strategies to watch and fight AMR. For example, In that way we could see which variables govern the resistance to certain drugs avoiding the use of generic antibiotics or which features make more prominent a patient/pathogen to be susceptible to an antibiotic.

Another improvement to take in count to implement methods to avoid effects on models of un-balanced classes which can alter the performance of the classifier. There are certain methods in the literature [3] covering such problem. Since our data had the restrictions listed previously in this work these methods weren't included. Also, as a next logical step of improvement to this work would be extending the analysis to other classification methods such as K-nearest neighbors, Least Discriminant Analysis, Least Absolute Shrinkage and Selection Operator...etc.

On a larger time scale, it would be important to implement features from genetic studies [4]. This would allow a better understanding of how AMR dynamics work. To bridge the gap between genetic ML and interpretative ML. To provide a more complete answer. In this direction, coordination between research and current surveillance programs would also be necessary. In this way, the feedback generated by these studies would make it possible to accurately label the data and generate useful reports for medical staff. Managing and minimizing the threat of AMR.

A Images and tables

For the sake of the reader in this section we will put the Images and tables of the analysis due to the amount of volume that the algorithm it generates.

A.1 GLM data

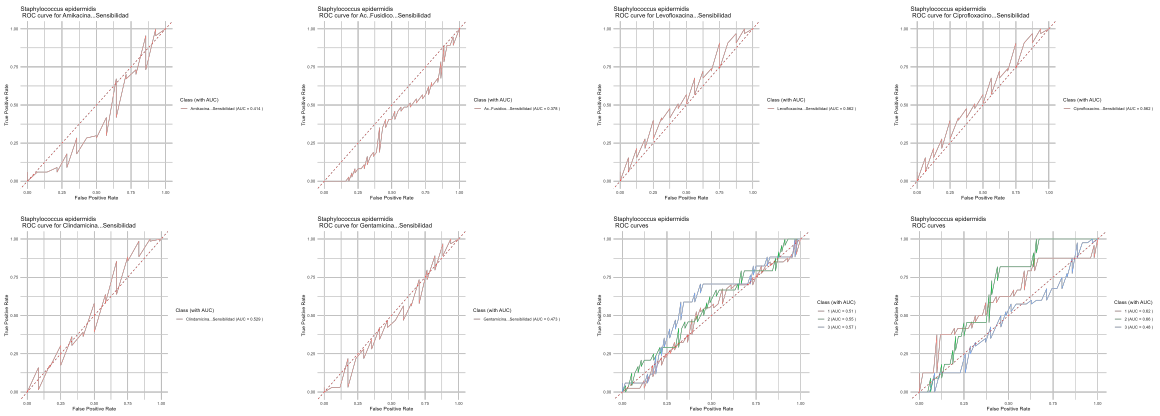


Figure 18: Result of ROC curves for the glm.

A.2 Tree classifier data

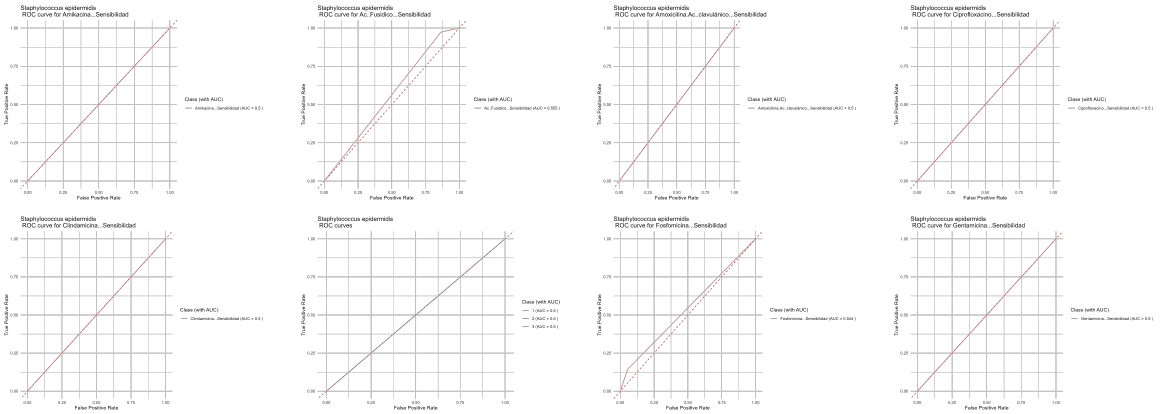


Figure 19: Result of ROC curves for the tree gradient machine.

A.3 Neural Network results

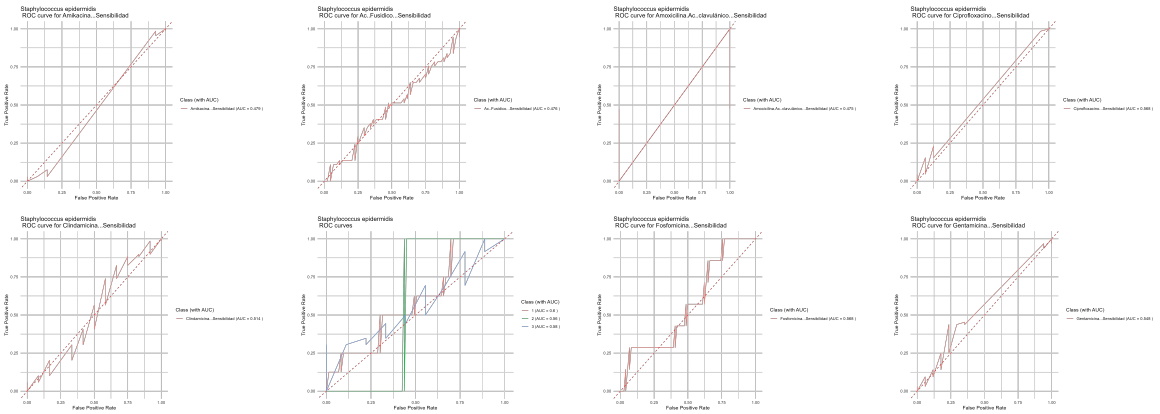


Figure 20: Result of ROC curves for the NN.

References

[1] Taiyao Wang et al. *Predicting Antimicrobial Resistance in the Intensive Care Unit*. 2021. arXiv: 2111.03575 [stat.AP].

[2] Porras González A. Martínez Campos L. *Lectura interpretada del antibiograma*. 2021. URL: https://www.guia-abe.es/generalidades-lectura-interpretada-del-antibiograma#_parampkpd.

[3] Eyad Elyan and Amir et alli Hussain. "Antimicrobial Resistance and Machine Learning: Challenges and Opportunities". In: *IEEE Access* 10 (2022), pp. 31561–31577. DOI: 10.1109/ACCESS.2022.3160213.

[4] Jee In Kim and Finlay Maguire et alli. "Machine Learning for Antimicrobial Resistance Prediction: Current Practice, Limitations, and Clinical Perspective". In: *Clinical Microbiology Reviews* 35.3 (2022), e00179–21. DOI: 10.1128/cmr.00179-21. eprint: https://journals.asm.org/doi/pdf/10.1128/cmr.00179-21. URL: https://journals.asm.org/doi/abs/10.1128/cmr.00179-21.

[5] D. et al Ching T. Himmelstein. "Opportunities and obstacles for deep learning in biology and medicine." In: *Journal of the Royal Society Interface* (2018). DOI: https://doi.org/10.1098/rsif.2017.0387.

[6] Saavedra Lozano J. Hernanz Lobo A. *Generalidades sobre antibioticoterapia. Bases para un tratamiento empírico racional*. 2018. URL: https://www.guia-abe.es/generalidades-generalidades-sobre-antibioticoterapia-bases-para-un-tratamiento-empirico-racional-.

[7] CDC. *Antibiotic Resistance Threats in the United States*. Tech. rep. U.S. Department of Health and Human Services, Atlanta, 2019.

[8] European Centre for Disease Prevention, Control, and WorldHealth Organization. "Antimicrobial resistance surveillance in Europe 2023 - 2021 data." In: (2023).

[9] Rafael Cantón J. y Elías García Sánchez José A. García Rodríguez. *Métodos Básicos de estudio de la sensibilidad a antimicrobianos. Recomendaciones de la SEIMC.* 2016. URL: https://www.seimc.org/contenidos/documentoscientificos/procedimientosmicrobiologia/seimc-procedimientomicrobiologia11.pdf.

[10] Esparza Olcina MJ Obando Pacheco P Suárez-Arrabal MC. *Descripción general de los principales grupos de fármacos antimicrobianos. Antibióticos.* 2020. URL: https://www.guia-abe.es/generalidades-descripcion-general-de-los-principales-grupos-de-farmacos-antimicrobianos-antibioticos-.

[11] Jesús Cercenado Emilia y Saavedra-Lozano. "El antibiograma. Interpretación del antibiograma: conceptos generales (I)". In: *Anales de Pediatría Continuada* (2009). DOI: 10.1016/S1696-2818(09)71927-4. URL: https://www.elsevier.es/es-revista-anales-pediatria-continuada-51-articulo-el-antibiograma-interpretacion-del-antibiograma-S1696281809719274.

[12] Maria T. Vazquez-Pertejo. *Pruebas de sensibilidad o antibiogramas.* Web Page. 2022/10 2022. URL: https://www.msdmanuals.com/es/professional/enfermedades-infecciosas/diagn%C3%B3stico-de-laboratorio-de-las-enfermedades-infecciosas/pruebas-de-sensibilidad-o-antibiogramas.

[13] Miguel Ángel March Rosselló Gabriel Alberto Bratos Pérez. "Antiograma rápido en Microbiología Clínica". In: *Enfermedades Infecciosas y Microbiología Clínica* (2015). DOI: 10.1016/j.eimc.2014.11.014. URL: https://www.elsevier.es/es-revista-enfermedades-infecciosas-microbiologia-clinica-28-articulo-antibiograma-rapido-microbiologia-clinica-S0213005X14003966.

[14] Organización Mundial de la Salud. *Plan de acción mundial sobre la resistencia a los antimicrobianos.* Organización Mundial de la Salud, 2016, 30 p.

[15] Canut Blasco A. et alli. Calvo Montes J. "Preparación de informes acumulados de sensibilidad a los antimicrobianos". In: *Sociedad Española de Enfermedades Infecciosas y Microbiología Clínica (SEIMC).* (2014).

[16] Muhammad Shoaib Farooq and Mehreen Ilyas. *Predicting environment effects on breast cancer by implementing machine learning.* 2023. arXiv: 2309.14397 [cs.LG].

[17] Nicolas Ayala-Aldana and Leticia Gonzalez-Valdés. "Metodo de random forest para el reconocimiento de patrones de sensibilidad y resistencia en antibiogramas". es. In: *Revista chilena de infectología* 40 (Feb. 2023), pp. 76–77. ISSN: 0716-1018. URL: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0716-10182023000100076&nrm=iso.

[18] Basmadjian R et alli Weaver C. "Reporting of Model Performance and Statistical Methods in Studies That Use Machine Learning to Develop Clinical Prediction Models: Protocol for a Systematic Review". In: *JMIR Res Protoc* (2022). DOI: 10.2196/30956.

[19] Medford RJ. Corbin CK. "Personalized Antibiograms: Machine Learning for Precision Selection of Empiric Antibiotics." In: *AMIA Jt Summits Transl Sci Proc* (2020), pp. 108–115.

[20] Sergio Martínez-Agüero and Mora-Jiménez et alli. "Machine Learning Techniques to Identify Antimicrobial Resistance in the Intensive Care Unit". In: *Entropy* 21.6 (2019). ISSN: 1099-4300. DOI: 10.3390/e21060603. URL: https://www.mdpi.com/1099-4300/21/6/603.

[21] Oriol Tellería Serrano. *Code developed for modeling.* URL: https://github.com/oriolsj/Analisis-Logistico/blob/661eb795027cc5f63714258ace811edd6646d37a/Analisisv2.0.Rmd.

[22] Oriol Tellería Serrano. *Code developed for cleaning datasets.* URL: https://github.com/oriolsj/Analisis-Logistico/blob/e5619e4e9b05d351930691cafa846c35d35337a7/Explorarotyan.py.

[23] A. Agresti. *An Introduction to Categorical Data Analysis.* Wiley Series in Probability and Statistics. Wiley, 2018. ISBN: 9781119405283. URL: https://books.google.es/books?id=pHZyDwAAQBAJ.

[24] Qingzhou Shi Cheng Hua Dr. Youn-Jeng Choi. *Companion to BER 642: Advanced Regression Methods.* 2021. URL: https://bookdown.org/chua/ber642_advanced_regression/.

[25] Robert Malouf. *A comparison of algorithms for maximum entropy parameter estimation.* 2002. DOI: 10.3115/1118853.1118871. URL: https://doi.org/10.3115/1118853.1118871.

[26] Celina K Gehringer et al. *How to develop, externally validate, and update multinomial prediction models.* 2023. arXiv: 2312.12008 [stat.ME].

[27] F. Pedregosa et al. *Scikit-learn: Machine Learning in Python.* 2011.

[28] Posit. *Rstudio.* 2024. URL: https://posit.co/products/open-source/rstudio/.

[29] L. Breiman and J. Friedman. *Classification and Regression Trees.* Chapman and Hall/CRC., 1984. DOI: https://doi.org/10.1201/9781315139470.

[30] J.R. Quinlan. *programs for machine learning.* Morgan Kaufmann, 1993.

[31] Tibshirani R. Hastie T. and Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2009.

[32] J.R. Quinlan. "Simplifying decision trees". In: *International Journal of Man-Machine Studies* (1987). DOI: https://doi.org/10.1016/S0020-7373(87)80053-6.

[33] Aston Zhang et al. "Dive into Deep Learning". In: *arXiv preprint arXiv:2106.11342* (2021).

[34] Bengio Y. LeCun Y. and Hinton G. "Deep learning". In: *Nature* (2015).

[35] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.

[36] Said Bleik. *Computing Classification Evaluation Metrics in R*. 2016. URL: https://blog.revolutionanalytics.com/2016/03/com_class_eval_metrics_r.html.

[37] Ajitesh Kumar. *Micro-average, Macro-average, Weighting: Precision, Recall, F1-Score*. 2023. URL: https://vitalflux.com/micro-average-macro-average-scoring-metrics-multi-class-classification-python/.

[38] Ivo Grosse Jan Grau and Jens Keilwagen. "PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R". In: *Bioinformatics, Volume 31, Issue 15* (2015). DOI: https://doi.org/10.1093/bioinformatics/btv153. URL: https://cran.r-project.org/web/packages/PRROC/vignettes/PRROC.pdf.

[39] Blaise Hanczar et al. "Small-sample precision of ROC-related estimates". In: *Bioinformatics* (2010). DOI: 10.1093/bioinformatics/btq037. URL: https://doi.org/10.1093/bioinformatics/btq037.

[40] Microsoft. *Visual Studio Code*. 2024. URL: https://code.visualstudio.com/.

[41] Joaquín Amat Rodrigo. *Machine Learning con H2O y R*. 2020. URL: https://cienciadedatos.net/documentos/44_machine_learning_con_h2o_y_r.

[42] Digital Science UK Ltd. *Overleaf*. 2024. URL: https://www.overleaf.com/.

[43] Roberto García Peña. "Gestión y análisis de datos de resultados de medicamentos en pacientes con infecciones". PhD thesis. Escuela de Ingeniería y Arquitectura. Universidad de Zaragoza, 2024.

[44] Instituto Nacional de Estadística. *Encuesta Europea de Salud 2020. Determinantes de salud: Cifras absolutas*. 2010. URL: https://www.ine.es/jaxi/Tabla.htm?path=/t15/p420/a2019/p03/l0/&file=01004.px&L=0.

[45] Norma Fariña and Letizia Carpinelli. "Staphylococcus coagulasa-negativa clínicamente significativos: Especies mís frecuentes y factores de virulencia". In: *Revista chilena de infectología* (2013). URL: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0716-10182013000500003&nrm=iso.

[46] Kathie L. Rogers, Paul D. Fey, and Mark E. Rupp. "Coagulase-Negative Staphylococcal Infections". In: *Infectious Disease Clinics of North America* (2009). Staphylococcal Infections. DOI: https://doi.org/10.1016/j.idc.2008.10.001. URL: https://www.sciencedirect.com/science/article/pii/S0891552008000871.

[47] Bijie H et alli Rosenthal VD. "International Nosocomial infection control Report". In: *American Journal of Infection Control.* (2009). DOI: doi:10.1016/j.ajic.2011.05.020.. URL: https://eprints.ugd.edu.mk/5983/1/infection%20control%202004-2009.pdf.

Table 8: Performance Metrics by Drug of the glm.

Metric	Macro	Micro	Weighted	Drug
Precision	0.35	0.52	0.94	Penicilina
Recall	0.59	0.52	0.52	
F1 Score	0.40	0.52	0.65	
Accuracy		0.52		
Cohen's Kappa		0.04		
Precision	0.34	0.52	0.99	Amoxicilina Ac. clavulánico
Recall	0.76	0.52	0.52	
F1 Score	0.37	0.52	0.67	
Accuracy		0.52		
Cohen's Kappa		0.03		
Precision	0.35	0.52	0.94	Oxacilina
Recall	0.59	0.52	0.52	
F1 Score	0.40	0.52	0.65	
Accuracy		0.52		
Cohen's Kappa		0.04		
Precision	0.30	0.36	0.50	Linezolid
Recall	0.38	0.36	0.36	
F1 Score	0.38	0.36	0.39	
Accuracy		0.36		
Cohen's Kappa		-0.07		
Precision	0.36	0.51	0.83	Eritromicina
Recall	0.59	0.51	0.51	
F1 Score	0.31	0.51	0.61	
Accuracy		0.51		
Cohen's Kappa		0.04		
Precision	0.32	0.47	0.74	Clindamicina
Recall	0.38	0.47	0.47	
F1 Score	0.39	0.47	0.56	
Accuracy		0.47		
Cohen's Kappa		-0.03		
Precision	0.36	0.32	0.41	Tetraciclina
Recall	0.32	0.32	0.32	
F1 Score	0.31	0.32	0.34	
Accuracy		0.32		
Cohen's Kappa		0.00		
Precision	0.31	0.44	0.65	Gentamicina
Recall	0.37	0.44	0.44	
F1 Score	0.39	0.44	0.52	
Accuracy		0.44		
Cohen's Kappa		-0.05		
Precision	0.29	0.42	0.64	Tobramicina
Recall	0.34	0.42	0.42	
F1 Score	0.36	0.42	0.50	
Accuracy		0.42		
Cohen's Kappa		-0.11		

Table 10: Performance Metrics by Drug of the glm (2).

Metric	Macro	Micro	Weighted	Drug
Precision	0.29	0.42	0.64	Amikacina
Recall	0.34	0.42	0.42	
F1 Score	0.36	0.42	0.50	
Accuracy		0.42		
Cohen's Kappa		-0.11		
Precision	0.26	0.31	0.46	Rifampicina
Recall	0.32	0.31	0.31	
F1 Score	0.33	0.31	0.35	
Accuracy		0.31		
Cohen's Kappa		-0.14		
Precision	0.34	0.37	0.86	Fosfomicina
Recall	0.46	0.37	0.37	
F1 Score	0.34	0.37	0.48	
Accuracy		0.37		
Cohen's Kappa		0.02		
Precision	0.36	0.43	0.43	Trimetoprim Sulfametoxazol
Recall	0.36	0.43	0.43	
F1 Score	0.36	0.43	0.43	
Accuracy		0.43		
Cohen's Kappa		0.04		
Precision	0.33	0.42	0.50	Ac. Fusidico
Recall	0.43	0.42	0.42	
F1 Score	0.45	0.42	0.45	
Accuracy		0.42		
Cohen's Kappa		0.00		
Precision	0.32	0.44	0.56	Mupirocina
Recall	0.42	0.44	0.44	
F1 Score	0.44	0.44	0.49	
Accuracy		0.44		
Cohen's Kappa		-0.03		
Precision	0.40	0.48	0.75	Ciprofloxacino
Recall	0.39	0.48	0.48	
F1 Score	0.47	0.48	0.59	
Accuracy		0.48		
Cohen's Kappa		0.08		
Precision	0.40	0.48	0.75	Levofloxacina
Recall	0.39	0.48	0.48	
F1 Score	0.47	0.48	0.59	
Accuracy		0.48		
Cohen's Kappa		0.08		
Precision	0.30	0.43	0.62	Screening de Cefoxitina
Recall	0.38	0.43	0.43	
F1 Score	0.40	0.43	0.50	
Accuracy		0.43		
Cohen's Kappa		-0.08		

Table 12: Performance Metrics by drug of the tree clasiffier (2).

Metric	Macro	Micro	Weighted	Drug
Precision	0.35	0.52	0.94	Penicilina
Recall	0.59	0.52	0.52	
F1 Score	0.40	0.52	0.65	
Accuracy		0.52		
Cohen's Kappa		0.04		
Precision	0.34	0.52	0.99	Amoxicilina Ac. clavulánico
Recall	0.76	0.52	0.52	
F1 Score	0.37	0.52	0.67	
Accuracy		0.52		
Cohen's Kappa		0.03		
Precision	0.35	0.52	0.94	Oxacilina
Recall	0.59	0.52	0.52	
F1 Score	0.40	0.52	0.65	
Accuracy		0.52		
Cohen's Kappa		0.04		
Precision	0.33	0.40	0.54	Linezolid
Recall	0.43	0.40	0.40	
F1 Score	0.42	0.40	0.43	
Accuracy		0.40		
Cohen's Kappa		-0.01		
Precision	0.32	0.48	0.83	Eritromicina
Recall	0.22	0.48	0.48	
F1 Score	0.36	0.48	0.60	
Accuracy		0.48		
Cohen's Kappa		-0.01		
Precision	0.33	0.48	0.75	Clindamicina
Recall	0.42	0.48	0.48	
F1 Score	0.42	0.48	0.57	
Accuracy		0.48		
Cohen's Kappa		-0.00		
Precision	0.24	0.23	0.29	Tetraciclina
Recall	0.23	0.23	0.23	
F1 Score	0.21	0.23	0.24	
Accuracy		0.23		
Cohen's Kappa		-0.13		
Precision	0.31	0.44	0.64	Gentamicina
Recall	0.39	0.44	0.44	
F1 Score	0.41	0.44	0.51	
Accuracy		0.44		
Cohen's Kappa		-0.05		

Table 14: Performance metrics by drug (Part 2)

Metric	Macro	Micro	Weighted	Drug
Precision	0.17	0.17	0.03	Tobramicina
Recall	0.50	0.17	0.17	
F1 Score	0.29	0.17	0.05	
Accuracy		0.17		
Cohen's Kappa		0.00		
Precision	0.17	0.17	0.03	Amikacina
Recall	0.50	0.17	0.17	
F1 Score	0.29	0.17	0.05	
Accuracy		0.17		
Cohen's Kappa		0.00		
Precision	0.41	0.22	0.57	Rifampicina
Recall	0.16	0.22	0.22	
F1 Score	0.46	0.22	0.32	
Accuracy		0.22		
Cohen's Kappa		0.04		
Precision	0.46	0.86	0.84	Fosfomicina
Recall	0.47	0.86	0.86	
F1 Score	0.93	0.86	0.85	
Accuracy		0.86		
Cohen's Kappa		0.05		
Precision	0.32	0.32	0.10	Trimetoprim. Sulfametoxazol
Recall	0.33	0.32	0.32	
F1 Score	0.48	0.32	0.15	
Accuracy		0.32		
Cohen's Kappa		0.00		
Precision	0.43	0.07	0.47	Ac. Fusidico
Recall	0.07	0.07	0.07	
F1 Score	0.23	0.07	0.13	
Accuracy		0.07		
Cohen's Kappa		0.03		
Precision	0.33	0.03	0.19	Mupirocina
Recall	0.05	0.03	0.03	
F1 Score	0.18	0.03	0.05	
Accuracy		0.03		
Cohen's Kappa		0.02		
Precision	0.24	0.19	0.28	Levofloxacina
Recall	0.44	0.19	0.19	
F1 Score	0.36	0.19	0.18	
Accuracy		0.19		
Cohen's Kappa		0.01		

Table 15: Performance Metrics by drug using neural networks (Part 1)

Metric	Macro	Micro	Weighted	Drug
Precision	0.96	0.96	0.93	Penicilina
Recall	0.50	0.96	0.96	
F1 Score	0.98	0.96	0.94	
Accuracy		0.96		
Cohen's Kappa		0.00		
Precision	0.99	0.99	0.97	Amoxicilina Ac. clavulánico
Recall	0.50	0.99	0.99	
F1 Score	0.99	0.99	0.98	
Accuracy		0.99		
Cohen's Kappa		0.00		
Precision	0.96	0.96	0.93	Oxacilina
Recall	0.50	0.96	0.96	
F1 Score	0.98	0.96	0.94	
Accuracy		0.96		
Cohen's Kappa		0.00		
Precision	0.33	0.33	0.11	Linezolid
Recall	0.50	0.33	0.33	
F1 Score	0.50	0.33	0.17	
Accuracy		0.33		
Cohen's Kappa		0.00		
Precision	0.89	0.89	0.79	Eritromicina
Recall	0.33	0.89	0.89	
F1 Score	0.94	0.89	0.84	
Accuracy		0.89		
Cohen's Kappa		0.00		

Table 16: Performance Metrics by drug using neural networks (Part 2)

Metric	Macro	Micro	Weighted	Drug
Precision	0.85	0.85	0.73	Clindamicina
Recall	0.50	0.85	0.85	
F1 Score	0.92	0.85	0.78	
Accuracy		0.85		
Cohen's Kappa		0.00		
Precision	0.24	0.48	0.24	Tetraciclina
Recall	0.32	0.48	0.48	
F1 Score	0.65	0.48	0.32	
Accuracy		0.48		
Cohen's Kappa		-0.02		
Precision	0.79	0.79	0.62	Gentamicina
Recall	0.50	0.79	0.79	
F1 Score	0.88	0.79	0.70	
Accuracy		0.79		
Cohen's Kappa		0.00		
Precision	0.83	0.83	0.68	Tobramicina
Recall	0.50	0.83	0.83	
F1 Score	0.90	0.83	0.75	
Accuracy		0.83		
Cohen's Kappa		0.00		