IDIAP COMMUNICATION REPORT

MACHINE LEARNING FOR ADVERSE EVENT DETECTION IN LATENT TUBERCULOSIS INFECTION TREATMENT

Colombine Verzat

Idiap-Com-02-2020

AUGUST 2020

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Machine learning for adverse event detection in latent tuberculosis infection treatment

Thesis performed in the

Biosignal Processing Group
Idiap Research Institute
programme of Artificial Intelligence
UniDistance

to obtain the degree of

Master of science in Artificial Intelligence

by

Colombine VERZAT

under the supervision of:

Dr. André Anjos, project supervisor
Flavio Tarsetti, company supervisor

Martigny, Idiap, 2020

Written by Colombine Verzat <colombine.verzat@idiap.ch>

Acknowledgements

First, I would like to thank Dr. André Anjos for his excellent supervision throughout the thesis and for having made this thesis a great learning experience. I also want to thank my second supervisor Flavio Tarsetti, who was a great help to define the project and motivation. I extend my thanks to our Brazilian and Canadian collaborators, Dr. Dick Menzies, Dr. Anete Trajman, Dr. Jonathon Campbell and Mayara Bastos, who shared the dataset with us and kindly answered my questions about their experiments.
Next, I would like to thank all teachers and assistants at Idiap, who taught me elements which were crucial to this thesis. I am also very grateful to the people of the Biometrics lab, especially Vedrana, Pavel, Amir, Zohreh, Alex, Tiago and Sushil, who welcomed me and let me join their coffee breaks and beer hangouts.
I would like to thank the other master AI students, with whom I have developed a great friendship, in particular Antoine, Yannick, Jérémy, Jonathan, Nicholas and Daniel.
Finally, I would like to thank Giezi, who supported me and kept me motivated throughout the entire project.

C. V.

Abstract

One quarter of the world's population has latent tuberculosis infection (LTBI). In this form of the disease, the bacteria has a 10 to 15% chance to start replicating and cause the patient to develop active tuberculosis. In those cases, preventive therapy is thus essential to limit the spread of the disease. Unfortunately, the treatment for LTBI can cause severe adverse events which discourages patients. Predicting which patients are most at risk of developing adverse events could thus improve treatment efficacy and help achieving WHO goals of TB elimination by 2050.

The goal of this study is to identify whether it is possible to predict the occurrence of adverse events in patients based on their clinical data.

To address this, we disposed of a clinical dataset of 6485 patients who had LTBI and went through treatment. A small part of these patients developed adverse events associated to the treatment. First, we reproduced a study by Campbell et al. [1] performed on this dataset using a logistic regression model. We then investigated the predictive power of this model using generalization and established a baseline from the resulting model. Finally, we explored how non-linear machine learning models could improve the performance compared to the baseline.

We found that multivariate logistic regression yielded a classifier with the following performance: AUC= 0.65 ± 0.04. Although non-linear techniques matched the baseline performance, they failed to significantly improve the prediction further.

These findings suggest that part of the data is linearly separable, while some isolated points in the dataset cannot be easily generalized. Patients with and without adverse events seem to overlap in the variable space, which suggests that an efficient detection of adverse events is difficult to achieve with this dataset. The improvement of the model may require a larger

and less imbalanced dataset, possibly along further explanatory variables permitting a better characterization of the patient.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Active Tuberculosis and Latent Tuberculosis Infection

Tuberculosis (TB) is an infectious disease which primarily affects the lungs, by deposition of bacterium *Mycobacterium Tuberculosis* onto the lung alveolar surfaces [3]. These bacteria are spread from one person to another through aerosol droplets released into the air via coughs and sneezes. The progression of the disease mainly depends on the response of the host immune system: if the host response is efficient enough, the bacteria remain in the body in an inactive state and cause no symptoms. This is referred to as "latent TB infection" (LTBI). On the contrary, if the host immune response cannot contain the initial infection in the lungs, the bacteria replicate in an uncontrolled manner and the disease is referred to as "active TB". The patient is therefore infectious and can spread the disease to others.

The most common symptoms of active TB are persistent cough occasionally accompanied by sputum (thick fluid produced in the lungs) or blood, chest pain, weight loss, fever and night sweats. In later stages, TB can affect other parts of the body such as the spine or the kidneys, with various symptoms depending on the organ [4]. As opposed to active TB, LTBI is asymptomatic and non contagious. However, it can unpredictably develop into active TB. Some factors such as co-morbidities (*e.g.* HIV or diabetes) or smoking can aggravate the risk of falling ill, as these may compromise the immune system.

Active TB can be treated with a standard 6 months treatment of four antimicrobial drugs. However, many strains of TB have become resistant to the drugs that are most used to treat the disease, in which case the patient must take second line drugs [4]. In the case of LTBI, taking the treatment might increase the risk of developing active TB, however the risk is significantly reduced with an effective chemopreventive treatment [5].

Tuberculosis (TB) remains one of the ten most frequent causes of death worldwide and the

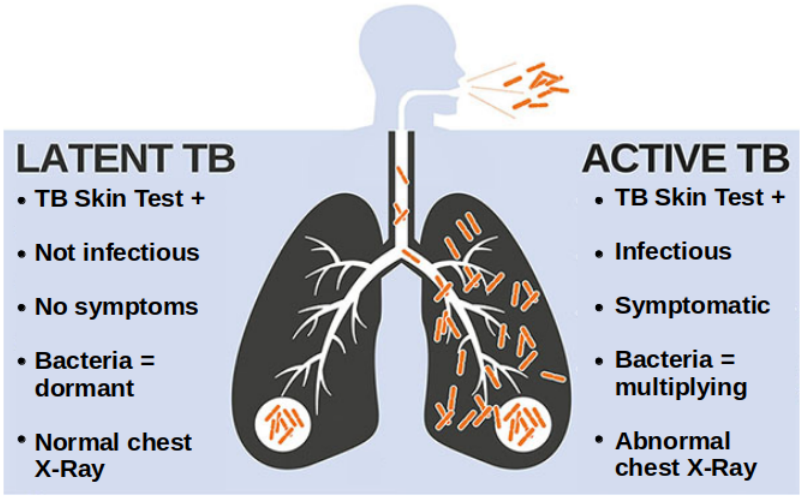

Figure 1.1 – Comparison between active and latent TB. Adapted from http://www.bccdc.ca/about/news-stories/stories/its-time-to-end-tb. Both latent and active TB show positive TB skin test but only active TB displays abnormal chest X-Ray. Latent TB is not infectious and the patients show no symptom since the bacterium is dormant in the lungs. On the other hand, patients with active TB can contaminate others and show symptoms.

World Health Organisation (WHO) estimates that one quarter of the world population is carrier of LTBI, with a 10 to 15% chance of developing active TB [6] (Figure 1.2). Together with the United Nations, the WHO has devised a set of milestones to eliminate TB[1] by 2050. This includes the prevention of active TB through treatment of LTBI, which has important potential individual and public health benefits, and hence is a cornerstone for achieving TB elimination [7]. Yet, only a small proportion of those who would benefit from LTBI treatment will complete it. Firstly, diagnosis of LTBI is still a challenge. Two types of tests are currently used to detect LTBI, both with several limitations [8]: tuberculin skin testing (TST) and interferon-gamma release assays. Both tests may over-diagnose LTBI (false positive results) or miss cases (false negative results), and there is currently no golden standard test. Importantly, before treating LTBI, it is mandatory to rule out active TB. This is achieved through symptom screening and chest X-rays. Distinguishing radiologic signs of active TB from a normal X-ray in primary care facilities is a challenge. Finally, since treating one quarter of the world's population represents a difficult endeavor to tackle, and only a minority will progress to an active disease, high-risk groups should be focused on.

[1] Annual incidence of less than 1 TB case per million population.

Figure 1.2 – Latent Tuberculosis Infection in the world. Adapted from https://www.paho.org/en/documents/world-tuberculosis-day-2020-infographic-jpg-treatment-tb-infection-latent-tb. 1/4 of the world's population represents roughly 2 billion people, and 10% of those results in approximately 200 million people developing active TB

1.2 Adverse Events

Safety is also a major concern for LTBI treatment which is often the cause for adverse events such as hepatotoxicity (drug-induced liver disease), rash, gastrointestinal intolerance, dizziness, and other drug-related side effects [9]. The most feared adverse event of LTBI treatment is hepatotoxicity: it may be fatal or require hospitalization and liver transplant, which is a highly complex and costly procedure with high morbidity and mortality rates. Recognition of characteristics of individuals that increase the risk of adverse events would allow a more careful follow up. Currently for LTBI treatment, blood tests and liver function monitoring are not routinely done because they are not cost-effective: the event is severe or even fatal but uncommon. However, blood testing could be cost-effective in a subset of persons with pre-identified risks for this complication. Fear of adverse events has reduced the acceptability of LTBI treatment both by healthcare workers (who fail to prescribe) and patients (who fail to initiate the treatment). Prediction of those who are more likely to develop adverse events would be of great value to reduce fear and intensify follow-up of high-risk patients.

The main goal of this work is to identify patients who are more likely to develop adverse events during LTBI treatment, using a *predictive model* based on patient clinical data.

1.3 Related Work

The prediction of health outcomes from clinical data is an important problem in health research. It is usually assessed by computing scores for risk stratification, based on statistical models such as logistic regression. Most applications are based on the belief that there is a

relatively small number of important risk factors and that careful selection of those variables increases the model performance for outcome prediction. However, it can be argued that risk factors typically interact with each other in a complicated and generally unknown way, and therefore often are eliminated from predictive models, when they could potentially improve model performance (and therefore should be incorporated as well). More recently, machine learning techniques have become available, based on inductive inference rather than on classical statistics. These techniques allow a proper validation scheme and allow to investigate the non-linear relationship between all available variables and the outcome.

We review here how clinical data is used to predict the risk of adverse events after surgery, and how machine learning models have been shown to increase model performance compared to traditional statistical models. We then review how such models have been developed in the context of TB and LTBI treatment, and explain why machine learning could be a useful tool for our predictive model of adverse events.

1.3.1 Prediction of adverse events after surgery

Using patient clinical data to assess the risk of adverse events is quite common for hospitalized patients and in particular, patients who went through surgery [10, 11, 12]. Indeed, this allows to target high-risk patients who may benefit from post-operative interventions, which leads to a better utilization of hospital resources. The review of Falconer et al. [11], whose purpose is to analyze models developed for predicting adverse drug events in hospitalized patients, identifies similarities between these models. What comes out of their analysis of the models' development is that most models use binary logistic regression, and pre-selected candidate predictor variables such as patient demographics, medications, medical conditions and a variety of laboratory tests. These variables are first tested using univariate logistic regression, and only statistically significant ones are included in multivariate analysis. Model performance, defined as the ability to discriminate patients with or without an adverse event, is often evaluated using the area under the Receiver Operative Characteristic curve. The authors from the review conclude that no perfect model was identified, and the developed models lacked a proper validation scheme. This is where machine learning could add some insight, by dividing the dataset into a train set, to train the model, and a separate test set, to evaluate the model. A study comparing machine learning techniques with classical statistical models in predicting health outcomes [13] finds that, while ROC curves are quite close, the multi-layer perceptron (MLP) consistently shows best performance in all data sets. More recent studies also explore the use of machine learning, for example with Markov Chain Models (MCM) in order to capture the temporal sequence and timing of adverse events [14], and find that these models outperform baseline models based on a risk index. However, one study by Han et al. [15], where logistic regression is combined with a machine learning approach to predict

adverse events following spine surgery, concludes that this approach does not perform better than a standard generalized regression.

Machine learning in the development of predictive models is being investigated, and sometimes shows better predictive performance than standard statistical models, such as logistic regression, which is why our predictive model for adverse events should investigate both strategies and compare their performance.

1.3.2 Prediction of adverse events during LTBI treatment

In the case of LBTI treatment, the identification of risk factors for adverse events has been widely explored, in order to improve treatment and prognosis. The review of Resende and dos Santos-Neto [16], reports age, gender, treatment regimen, alcoholism, HIV co-infection, genetic factors, and nutritional deficiencies as risk factors related to antituberculosis drugs. Another study by Castro et al. [17] reports a correlation between age and hepatotoxicity, and another between diabetes mellitus and adverse events, due to antituberculosis drugs. These studies help to determine which risk factors should be included in a predictive model for adverse events. Many studies also compare the occurrence of adverse events between different drug regimen, making it an important risk factor to take into account in our analysis [18].

Multivariate logistic regression has been used to determine risk factors associated with the occurrence of adverse events during LTBI treatment [19, 1], or associated with the practitioner's decision not to prescribe LTBI treatment, partially based on the risk to develop adverse events [20]. While these statistical analyses clearly identify important risk factors for adverse events, no predictive model per se was developed. To the best of our knowledge, no study using machine learning has explored the prediction of adverse events during LTBI treatment yet.

Logistic regression models based on prior medical knowledge (selecting only risk factors known to be associated with adverse events) have the advantage of being simple and easy to interpret. In contrast, machine learning models are more complex and have less intuitive interpretation. However, they might provide additional information compared to simple explanatory models, by characterizing non-linear relationship between clinical variables and the outcome. Sauer et al. [21] compared the performance of different machine learning models in their ability to predict active TB treatment failure based on patient clinical characteristics, and found high predictive performance (AUC= 0.74). Another study applied Artificial Neural Networks (ANN) to predict TB disease based on TB suspect clinical data such as gender, age, HIV-status, previous TB history, sample type, and signs and symptoms of TB [22], and found a predictive accuracy of about 94%. These studies demonstrate that machine learning models can efficiently predict specific outcomes of TB treatment based on patient clinical data and

should be investigated for the detection of adverse events in LTBI treatment.

1.4 Outline

This thesis is organized as follows: Chapter 2 describes the materials available and methods used in this study. The clinical dataset of LTBI patients is described, highlighting the different methods and tools used in this project. In particular, this chapter introduces the work of the collaborators who provided the dataset (Campbell et al. [1]), whose aim was to provide a rough estimate of odds of developing adverse events for certain categories of patients via logistic regression.

Our first contribution, which constitutes the first part of Chapter 3, was to process the data by creating protocols and formulating the question as a machine learning problem. To this end, we set up a reproducible research framework and established figures of merit to analyze the results of this work.

Chapter 3 then continues by introducing the experiments performed, starting from a reproduction of Campbell et al. [1]'s work, and expanding their analysis to create an evaluation framework for the target prediction task. This chapter concludes by improving on the predictions obtained through the use of non-linear machine learning techniques.

Chapter 4 analyzes the performances of the models presented in the experiments. Finally, Chapter 5 sums up the important findings of this work and suggests directions for future research.

2 Methods

This chapter first depicts the clinical variables and outcomes present in the dataset of LTBI patients supporting this thesis. In a second part, the different statistical methods implemented to detect adverse events are described. Finally, the third part details how this work was made reproducible, using the Bob framework [1].

2.1 Dataset

This work on the prediction of adverse events during LTBI treatment relies on a clinical dataset obtained from the Canadian Institute of Health Research (CIHR). The dataset was collected in the context of a multicenter open-label trial in TB clinics affiliated to universities in Canada, Brazil and Saudi Arabia [23]. The reason for this trial is that the recommended standard therapy for LTBI in most countries (9 months of isoniazid) has severe disadvantages such as poor completion rates and serious adverse events, and this has stimulated interest in finding shorter and safer regimens for LTBI treatment, such as 4 months of rifampin. The trial was designed to compare the frequency of adverse events and the treatment completion rates in patients given 4 months of daily rifampin or 9 months of daily isoniazid for LTBI.

The dataset combines results of Phase II [23] and Phase III [24] international randomized controlled trials in consenting adults with a positive LTBI diagnostic test. It consists of two separate databanks in Microsoft Excel format with respectively 847 and 5992 adult participants.

Information includes baseline demographic and clinical characteristics, concurrent morbidities and medication use, habits (smoking, alcohol and illicit substance use), type of treatment prescribed (4 months of rifampin (4RIF) or 9 months of isoniazid (9INH)), number of doses taken and occurrence of adverse events, classified as grades 1-2 (minor) or 3-5 (major), accord-

[1] https://www.idiap.ch/software/bob/

ing to the National Cancer Institute guidelines[2]. The study personnel in each center enrolled and registered participants verifying eligibility criteria, obtained consent, verified assignment and administered treatment. Patients were followed by their usual treating physician, who made all management decisions including discontinuation of therapy.

At each follow-up visit, participants were questioned about and examined for adverse events [24]. Adverse event reporting, assessment and grading was performed according to a standardized protocol described in Campbell et al. [1]. A panel of clinical-epidemiologic experts then independently categorized the severity of adverse events according to the following scale: an adverse event that was not related to a trial drug; an adverse event of grade 1 or 2 that was related to a trial drug (not serious); an adverse event of grade 3 or 4 that was related to a trial drug (generally considered to lead to trial-drug discontinuation if related to a trial drug); or a grade 5 event (death) that was related to a trial drug. The adverse events were also categorized by the experts into one of ten types: drug interaction, rash, hepatotoxicity, gastrointestinal (GI) intolerance, hematologic, pregnancy, dizziness, drug-induced pancreatitis, seizure, and other.

2.1.1 Clinical variables

The clinical variables available in the dataset are common factors known to be potentially related to adverse events in such treatment, or predictors selected by the principle that their identification could permit clinical action. The clinical variables are sometimes continuous (*e.g.* age) or categorical (*e.g.* gender). The distribution of those variables between the two treatments (4RIF and 9INH) is displayed in Figures 2.1 and 2.2. Table 2.1 displays the clinical variables involved in Campbell et al. [1] as well as their distribution into categories and treatment arms. Here is a summary of the variables:

- Age: varying from 18 to 90 years old,
- Gender: categorized as male or female,
- Body mass index (BMI): weight in kilograms divided by the square of height in meters,
- HIV status: HIV-positive or negative,
- Immunosuppression: having another immunosuppressing condition than HIV (e.g diabetes),
- Alcohol use or consumption categorized as: never drinks, less than one drink per week, one drink or more per week
- Smoking history: smoking habits dichotomized as: never smoked, current or ex-smoker,

[2] https://www.eortc.be/services/doc/ctc/CTCAE_4.03_2010-06-14_QuickReference_5x7.pdf

- Medication consistency: number of pills taken during the treatment divided by number of days of the treatment. This is also referred as *treatment adherence*.
- Use of concomitant medication: indicates if the patient is taking other medications while on LTBI treatment,
- Alanine aminotransferase (ALT) measures (ALT is an biomarker for liver health and thus can be linked to adverse events categorized as hepatotoxicity),
- White blood cells (WBC) count (can be linked to hematologic adverse events),
- Platelet count (can also be linked to hematologic adverse events)

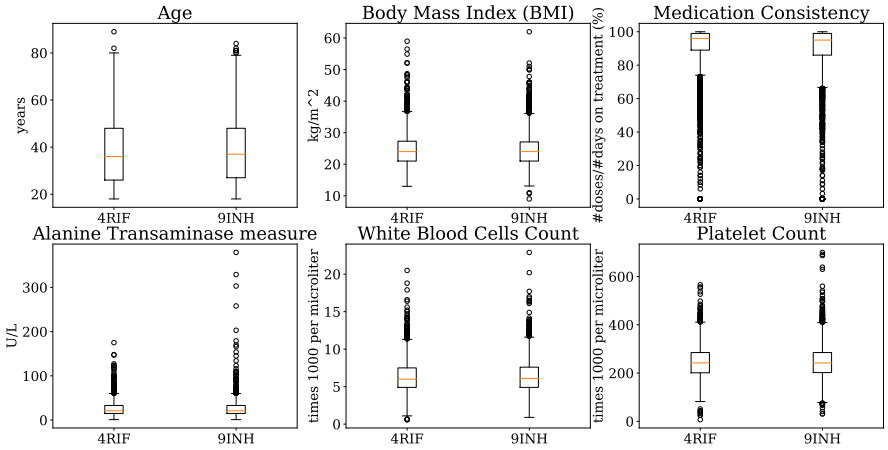

Figure 2.1 – Distributions of continuous clinical variables. All variables are similarly distributed between the rifampin and the isoniazid treatments. Most patients are rather young, with average BMI of 25 and high medication consistency.

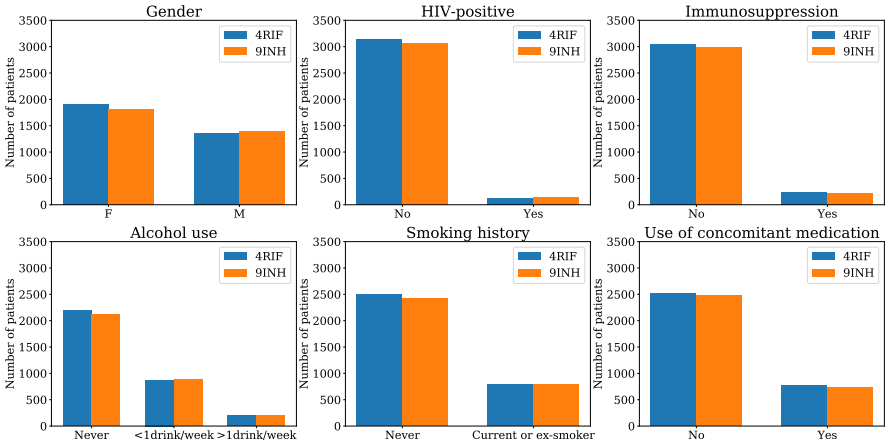

Figure 2.2 – Distributions of of categorical clinical variables. Both treatments have a similar of number of patients in each category. There are slightly more females than males, and most patients don't have any kind of immunosuppressing condition.

Table 2.1 – Baseline characteristics. Other immune suppression corresponds to non-HIV-related immune suppression, such as diabetes or renal failure. Medication consistency corresponds to the proportion of days a patient took its medication while on LTBI treatment. This table is adapted from Table 1 in [1]

		4 RIF (N=3280)	**9 INH (N=3205)**
Age	18-34	1489	1436
	35-64	1661	1642
	65-90	130	127
Sex	Male	1364	1394
	Female	1916	1811
BMI (Body Mass Index)	Underweight	216	222
	Normal	1674	1646
	Overweight	916	907
	Obese	474	430
Immune Suppression	HIV positive	130	138
	Other immune suppr.	221	196
Alcohol use	Never drinks	2200	2112
	≤ 1 drink per week	873	891
	> 1 drink per week	207	202
Smoking history	Never smoked	2496	2421
	Currently or has smoked	784	784
Medication consistency	< 90%	840	1054
	≥ 90%	2440	2151
Concomitant Medication	Any	763	735
	None	2517	2473
ALT levels	Normal	2984	2972
	Above normal	184	196
WBC count	Normal	2796	2810
	Below normal	438	424
Platelet count	Normal	3085	3084
	Below normal	145	146

2.1.2 Outcome

For the detection of adverse events, we followed the same rules indicated in Campbell et al. [1] to select the adverse events in the dataset. This meant including only adverse events resulting in permanent discontinuation of study medication and judged possibly or probably related to study drug (as opposed to non-related or unlikely related to study drug). This selection resulted in a total of 199 adverse events over a total of 6485 patients. Table 2.2 displays the number of adverse events per grade and per type.

Table 2.2 – Adverse Events judged possibly or probably related to therapy by the adverse event panel - by study drug, grade and type [1]. There is a total of 199 adverse events in the database, and more events are occurring in the 9INH treatment compared to the 4RIF treatment. There are roughly as many grade 1-2 events as grade 3-4 events. Only one death (grade 5 event) occurred. Most grade 3-4 events are hepatotoxicity and are much more present in the 9INH treatment.

Adverse events	4 RIF (N=3280)	9 INH (N=3205)	Total
All adverse events	68	131	199
Grade 1-2	37	56	93
Drug interaction	1	0	1
Rash	19	11	30
Hepatotoxicity	1	17	18
GI Intolerance	10	15	25
Hematologic	2	0	2
Dizziness	0	5	5
Other	4	8	12
Grade 3-4	31	74	105
Drug interaction	2	0	2
Rash	6	2	8
Hepatotoxicity	11	65	76
GI Intolerance	3	1	4
Hematologic	6	0	6
Pregnancy	2	2	4
Dizziness	1	2	3
Drug-induced Pancreatitis	0	1	1
Seizure	0	1	1
Grade 5: Death	0	1	1

Additionally, Campbell et al. [1] devised a specific set of outcomes for their statistical analysis. The primary outcome was defined as the combination of grade 1-2 rash (providers are usually hesitant to continue medications if a rash develops) and all grade 3-5 adverse events (more serious events). Secondary outcomes included grade 1-4 rash, grade 3-4 hepatotoxicity, grade 3-4 hematological events, and grade 3-5 non-hepatotoxic or non-rash adverse events. Table 2.3 summarizes the number of events for each outcome.

Table 2.3 – Outcomes of adverse events. The primary outcome (in bold) and secondary outcome are the main ones investigated in this study. This table is adapted from Table 2 in Campbell et al. [1]

Outcomes	4 RIF (N=3280)	9 INH (N=3205)	Total
all adverse events	68	131	199
grade 1-2 rash + all grade 3-5	**50**	**86**	**136**
grade 3-4 hepatotoxicity	11	65	76
grade 1-4 rash	25	13	38
grade 3-4 hematological	6	0	6
grade 3-5 non-rash and non-hepatotoxic	14	8	22

With this dataset, we explore the prediction of adverse events in LTBI, by using different algorithms which model the relationship between the covariates (patient clinical variables) and the outcome (occurrence of adverse event during the treatment).

2.2 Algorithms

This section introduces the classification problem, as well as the methods used in Campbell et al. [1], namely univariate and multivariate logistic regression. The machine learning techniques explored in this thesis are described as well.

2.2.1 Binary classification task

The task of classifying data is to decide class membership y' of an unknown data item x' based on a dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of data items x_i with known memberships y_i [25]. For binary classifications problems, the class labels of y are either 0 or 1. In our case, the class membership y corresponds to the occurrence of adverse events during the treatment. y_i is equal to 1 if patient i had an adverse event during the treatment ("positive" class), whereas y_i is equal to 0 if patient i had no adverse event during the treatment ("negative" class). The x_i are m-dimensional vectors, the components of which are called covariates (in statistics) or input variables (in machine learning). The covariates correspond to the clinical variables of the patient available from the dataset.

In most problems, there is no functional relationship $y = f(x)$ between y and x, and the relationship between x and y is described by a probability distribution $P(x, y)$. From statistical decision theory, the optimal class membership decision is to choose the class label y that maximizes the posterior distribution $P(y|x)$. Logistic regression and ANNs, the most widely

used models in biomedicine, build an approximation of $P(y|x)$, providing a function form f and a parameter vector β to express $P(y|x)$ as $P(y|x) = f(x,\beta)$. The parameters β are determined based on the data set D and the function form f differs for logistic regression and ANNs.

2.2.2 Logistic regression

Logistic regression analysis is a statistical technique to evaluate the relationship between various predictor variables (either categorical or continuous) and an outcome which is dichotomous [26]. The outcome or occurrence of the event is a binary variable: either the event occurs or it does not occur. In our case, either the patient has suffered from adverse events during treatment or he has not suffered from it. Therefore, event occurrence variables can be coded with 0 and 1:

- $Y_i = 1 \Leftrightarrow$ patient i had an adverse event during treatment,
- $Y_i = 0 \Leftrightarrow$ patient i had no adverse event during treatment.

To measure the probability of this event, three equivalent ways can be used: probability of the event, odds in favour of the event, log-odds in favour of the event. They are all equivalent since knowing the value of one measure for the event allows to compute the values of the two other measures for the same event.

- The probability of the event $Y_i = 1$ is a number p_i between 0 and 1 and we write $P(Y_i = 1) = p_i$. $p_i = 1$ means that the event is certain to occur and $p_i = 0$ means that the event is certain not to occur. Because Y is a binary variable, or Bernoulli random variable, $P(Y_i = 0) = 1 - p_i$.
- The odds in favor of the event is defined as the probability that the event occurs divided by the probability that the event does not occur. The odds in favor of $Y_i = 1$ is defined as:

$$\text{ODDS}(Y_i = 1) = \frac{P(Y_i = 1)}{P(Y_i \neq 1)} = \frac{P(Y_i = 1)}{P(Y_i = 0)} = \frac{p_i}{1 - p_i}. \tag{2.1}$$

  An odds number is between 0 and ∞. An odds of 0 means we are certain the event does not occur while an odds of ∞ corresponds to certainty that the event occurs. An increased odds corresponds to increased belief in the occurrence of the event.

- The log-odds in favor of an event is defined as the log of the odds in favor of the event:

$$\log\text{ODDS}(Y_i = 1) = \log \frac{P(Y_i = 1)}{P(Y_i = 0)} = \log \frac{p_i}{1 - p_i} = \text{logit}(p_i). \tag{2.2}$$

A log-odds is a number between $-\infty$ and ∞. A log-odds of $-\infty$ means that we are certain the event does not occur while a log-odds of ∞ corresponds to certainty that the event occurs. In general, an increased log-odds corresponds to an increased belief in the occurrence of the event.

Univariate logistic regression

Consider the predictor variable X to be any of the risk factors that might contribute to the occurrence of adverse event (*e.g.* immunosuppression, smoking habits, etc). Probability of "success" ($Y_i = 1$) will depend on levels of the risk factor. Let $X = (X_1, X_2, \ldots, X_k)$ be a set of explanatory variables. x_i is the observed value of the explanatory variables for observation i. In univariate analysis, we focus on a single variable X.

The logistic regression function is the log-odds of a success expressed linearly as the combination of all the covariates considered:

$$\text{logit}(p_i) = \text{logit}(P(Y_i = 1|X_i = x_i)) = \beta_0 + \beta_1 x_i. \tag{2.3}$$

On the probability scale, Equation 2.3 may be written:

$$p_i = \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}} = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_i)}}. \tag{2.4}$$

Multivariate logistic regression

Each of the univariate analyses assesses the association of a dichotomous variable with one predictor factor, whereas multivariate regression allows to study the simultaneous effect of multiple factors on a dichotomous outcome. The second step to build the multivariate regression model is to identify the best combination of explanatory variables to include in the model. The logistic regression equation now takes the following form:

$$\text{logit}(p_i) = \text{logit}(P(Y_i = 1|X_i = x_i)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k, \tag{2.5}$$

given k explanatory variables identified in univariate analysis. On the probability scale, the equation 2.5 may be written:

$$P(Y_i = 1) = \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_k x_k}}{1 + e^{\beta_0+\beta_1 x_1+\ldots+\beta_k X_k}} = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\ldots+\beta_k X_k)}}. \tag{2.6}$$

Equation 2.6 provides a model which can be used to predict the probability of an event happening for a particular individual given his/her profile of predictive factors.

Correction for rare events

Typically in logistic regression (and statistical software packages for logistic regression), the convergence of the model fitting algorithm is based on the maximum likelihood estimation (MLE) method. MLE is a method estimating the parameters β by maximizing a likelihood (or similarly, the log-likelihood) function so that under the assumed statistical model, the observed data is the most probable. The assumed statistical model here is the logistic regression model and its log-likelihood function is the following:

$$\log L(\beta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(h_\beta(x_i)) + (1-y_i) log(1-h_\beta(x_i))], \tag{2.7}$$

where β is the vector of logistic regression parameters and $h_\beta(x_i)$ is the hypothesis $P(Y_i = 1)$. ML estimate of each regression parameter β_r is usually obtained by solving the score equation $\frac{\partial log(L(\beta))}{\partial \beta_r} = U(\beta_r) = 0$. In general, there is no closed-form solution and the ML estimates are obtained using iterative algorithms such as Newton-Raphson (NR) or Iteratively Reweigthed Least Squares (IRLS).

Firth-penalized likelihood In situations where there is a "separation" in the data set, *i.e.* the data is sparse, finite ML estimates do not exist: the likelihood converges to a finite value while at least one parameter estimate diverges to $\pm\infty$. In our case, the frequency of adverse events in the database is very low and represents only 3% of the patients. Heinze and Schemper proposed a new procedure for logistic regression [27], which arrives at finite estimates for the parameters by a modification of the score function. This procedure was originally developed by Firth [28] to reduce the bias of ML estimates in generalised linear models, and has been extensively studied by Heinze and Schemper. In order to remove the bias from the parameter estimates, Firth suggested to maximize the penalized log likelihood defined as follows:

$$\log L^*(\beta) = \log L(\beta) + \frac{1}{2}\log(|I(\beta|), \tag{2.8}$$

where $|I(\beta)|$ is the determinant of the Fisher information matrix. It has been shown that parameter estimates from this approach are always finite and have lower small sample bias than ML estimates.

The R package `logistf`[3], used in Campbell et al. [1], provides a comprehensive tool to facilitate the application of Firth's modified score procedure in logistic regression analysis. In this package, the estimation of Firth-type logistic regression parameter estimates is based on a Newton-Raphson algorithm.

[3] https://cran.r-project.org/web/packages/logistf/

Weighted likelihood After having implemented Firth bias-reduced correction for rare events on Python, we realized that computation time was quite high and could be reduced by using another strategy to account for the small number of adverse events. This strategy consists in balancing the two classes *i.e.* patients who had an adverse event (positives) and patients who didn't have any adverse event (negatives), and is called class-balanced loss. It addresses the problem of training from imbalanced data by introducing a weighting factor that is inversely proportional to the effective number of samples. The weighting factor α was calculated as the proportion of adverse events (for a specific outcome): $\alpha = \frac{\#\text{adverse events}}{\text{total \# patients}}$. α is therefore a small number near 0, since adverse events are rare in our dataset.

The log-likelihood function of the weighted logistic regression model was rewritten as follows:

$$\log L(\beta) = -\frac{1}{n}\sum_{i=1}^{n}[(1-\alpha)y_i log(h_\beta(x_i)) + \alpha(1-y_i)log(1-h_\beta(x_i))]. \tag{2.9}$$

When $y_i = 1$, the patient belongs to the positives, and the term in Equation 2.9 is multiplied by $(1-\alpha)$ to increase the impact of this term on the whole function. On the other hand, when $y_i = 0$, the patient belongs to the negatives, and the term in Equation 2.9 is multiplied by α to diminish the impact of this term on the whole function. The idea is to encourage the classifier to correctly classify positives. The comparison between Firth and weighted corrections can be seen in Section 3.2.1.

2.2.3 Non-linear models

Logistic regression and artificial neural networks (ANN) share common root in statistical pattern recognition and the latter can be seen as a generalization of the former [25]. They both provide a function form f and parameter vector α as $P(y|x) = f(x,\alpha)$. Parameters α are determined based on the dataset, by maximum-likelihood estimation. The function form of f differs for logistic regression and ANNs. Logistic regression is considered a parametric method because the contribution of parameters (coefficients β) can be interpreted, whereas ANN is considered semi-parametric method, because parameters of a neural networks (*i.e.* weights) are often difficult to interpret.

The main weakness of linear predictors is their lack of capacity: for classification, the populations have to be linearly separable (*e.g.* XOR problem). However, machine learning models can solve non-linearly separable problems. For example, the multi-layer perceptron performs a non-linear mapping with the first layer so that the data is linearly separable for the second layer.

Multi-layer perceptron

The multi-layer perceptron (MLP) is the most common type of neural network used for supervised prediction [29]. It is a particular case of ANN, composed of multiple logistic regression units. The MLP consists of a network of processing elements arranged in layers: an input layer (receives external inputs), one or several hidden layers containing several hidden neurons or nodes (logistic regression units) and an output layer (produces classification results) (Figure 2.3). The input layer consists of a set of neurons representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation, followed by a non-linear activation function. The output layer receives the values from the last hidden layer and transforms them into output values. Binary classification for MLP is done by thresholding the output of the network.

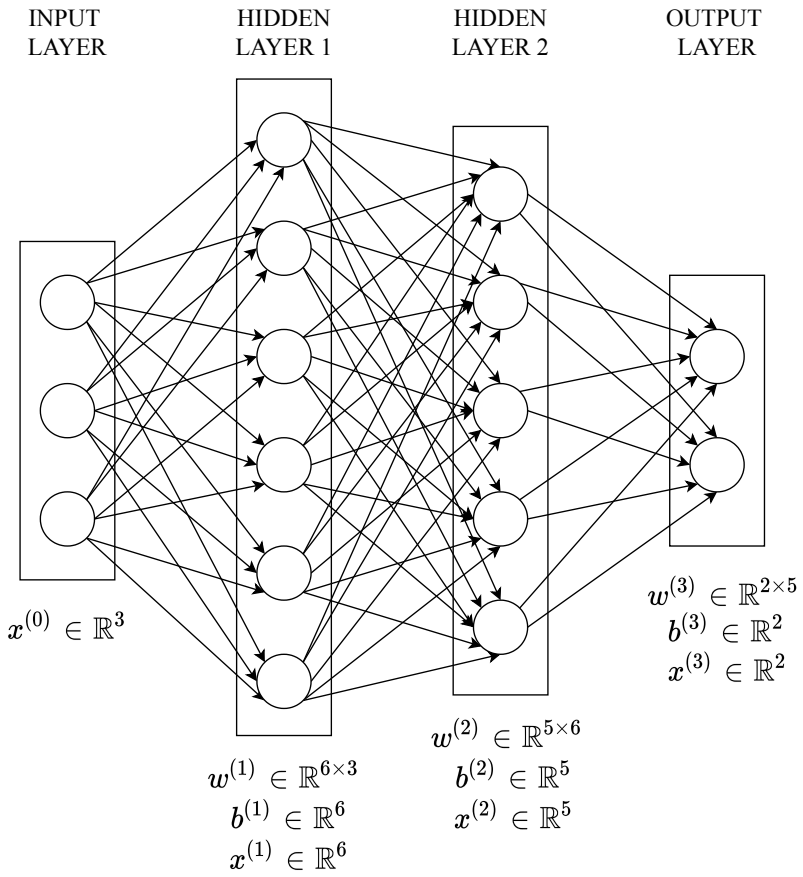

Figure 2.3 – Illustration of an MLP with 2 hidden layers. The network consists of an input layer with 3 features, a first hidden layer with 6 neurons, a second hidden layer with 5 neurons, and an output layer of 2 neurons. The input is noted $x^{(0)}$ and has the same number of dimensions than the number of features. Each hidden layer has an associated weight matrix w and a bias b. For each variable, the superscript indicates the layer.The network sequentially computes the output of each neuron when subjected to the input, following equation 2.10

More formally, the input is noted $x^{(0)} \in \mathbb{R}^{d+1}$, d being the number of input variables. Each layer receives the output of the previous layer $x^{(l-1)} \in \mathbb{R}^{N^{(l-1)}}$ and consists of $N^{(l)}$ neurons. In order to compute the output of the model, the MLP first performs a forward pass, processing the activations from the input to the output. If we denote L the number of layers, $w^{(l)} \in \mathbb{R}^{N^{(l)xN(l-1)}}$ the weight matrix, and $b \in \mathbb{R}^{N^{(l)}}$ the bias (one bias per neuron), the network sequentially computes the output of each neuron when subjected to the input, from left to right:

$$\forall 1 \leq l \geq L \begin{cases} s^{(l)} = w^{(l)} x^{(l-1)} + b^{(l)} \in \mathbb{R}^{N^{(l)}} \\ x^{(l)} = \sigma(s^{(l)}) \in \mathbb{R}^{N^{(l)}} \end{cases}, \tag{2.10}$$

where σ is a non-linear activation function, which allows to introduce non-linearity to the network. This function is responsible for mapping the input to the output. Different activation functions can be used, such as the sigmoid (*i.e.* logistic) or the hyperbolic tangent (*i.e.* tanh) functions, and the choice of activation depends on the task. The sigmoid function is a natural choice for the last layer of a network performing binary classification (the output labels are 0 or 1, and the sigmoid maps in the range [0, 1]). The MLP is then trained with gradient descent during the backward pass, by minimizing a loss function over the training set:

$$\mathscr{L}(w, b) = \sum_n l_n = \sum_n l(f(x_n; w, b), y_n), \tag{2.11}$$

where:

- $f(x_n; w, b)$ is the output of the network on sample n,
- y_n is the label of sample n,
- l is the loss on each of the individual samples,
- w represents the weights in all layers,
- b represents the bias in all layers

To do that, the partial derivatives of the loss function with respect to the loss parameters are computed, and then used to update the parameters using the following update rule:

$$w_{t+1} = w_t - \lambda \nabla \mathscr{L}(w_t), \tag{2.12}$$

where λ is a parameter known as the learning rate. In a binary classification task, it is standard to use a neural network architecture with a single logistic output unit and the cross-entropy loss function. With this combination, the output prediction is always between 0 and 1, and is interpreted as a probability.

Implementation We decided to use a package called `scikit-learn`[4], which provides a standard MLP implementation. As explained above, the activation function was set as the sigmoid function, and the loss function as the cross entropy loss. From there, other hyperparameters have to be initialized in `scikit-learn` to properly define the MLP:

- the number of layers and hidden neurons in each layer,
- the solver used to minimize the loss function: limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), stochastic gradient descent (SGD)
- a regularization parameter α, to prevent overfitting,
- the maximum number of iterations (for L-BFGS) or epochs (for SGD)
- the size of minibatches for stochastic optimizers,
- the learning rate λ for stochastic optimizers

These hyperparameters are fine-tuned in the experiments' section, in order to find the best predictive model of adverse events.

Optimizers L-BFGS belongs to the family of quasi-Newton methods and solves large nonlinear optimization problems with simple bounds on the variables [30]. It is a solver that approximates the inverse of the Hessian matrix to perform parameter updates. SGD is an optimization technique which minimizes a loss function in a stochastic fashion, performing a gradient descent step sample by sample. With mini-batch SGD, SGD divides the data into some batches, and optimizes one batch at each iteration. Using SGD thus introduces some new hyperparameters to fine tune: maximum number of iterations of gradient descent (*i.e* number of epochs), batch size and learning rate for gradient descent.

Regularization and early stopping There are several strategies to prevent overfitting, and two of them were investigated in this project. They either involve either using a regularizing parameter, or stopping the training before the model overfits. The first option corresponds to weight decay in the context of MLP. Weight decay may fix overfitting by limiting the magnitude of the weights. By default, `scikit-learn` classifier uses a regularization parameter $\alpha = 0.0001$. Increasing α may fix high variance (sign of overfitting) by encouraging smaller weights, while decreasing α may fix high bias (sign of underfitting) by encouraging larger weights, resulting in a more complicated boundary. Regularization is realized by adding a term to the cost function

[4] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

that penalizes larger weights:

$$E(w) = E_0(w) + \frac{1}{2}\alpha \sum_i w_i^2 \tag{2.13}$$

Where E_0 is the error measure used (in our case, cross-entropy loss), w is the weight vector and α the parameter governing how strongly large weights are penalized [31].

The second option is called early stopping, and consists in only partially adapting the model to the data set, which allows to restrict the model complexity. A subset of the training data is used as a validation set, to terminate training when the validation score is not improving. This option cannot be applied when using the L-BFGS solver, since it is auto-stopped. In `scikit-learn`, this is implemented by automatically setting aside 10% of training data as a validation set and terminating training when validation score is not improving by at least a given tolerance for a given number of consecutive epochs. Given the small size of our dataset, taking 10% of the training set as a validation set might result in a too small subset to ensure that the stopping mechanism functions correctly. Our strategy was to use the test set as a validation set, to ensure that the training is stopped optimally to prevent overfitting.

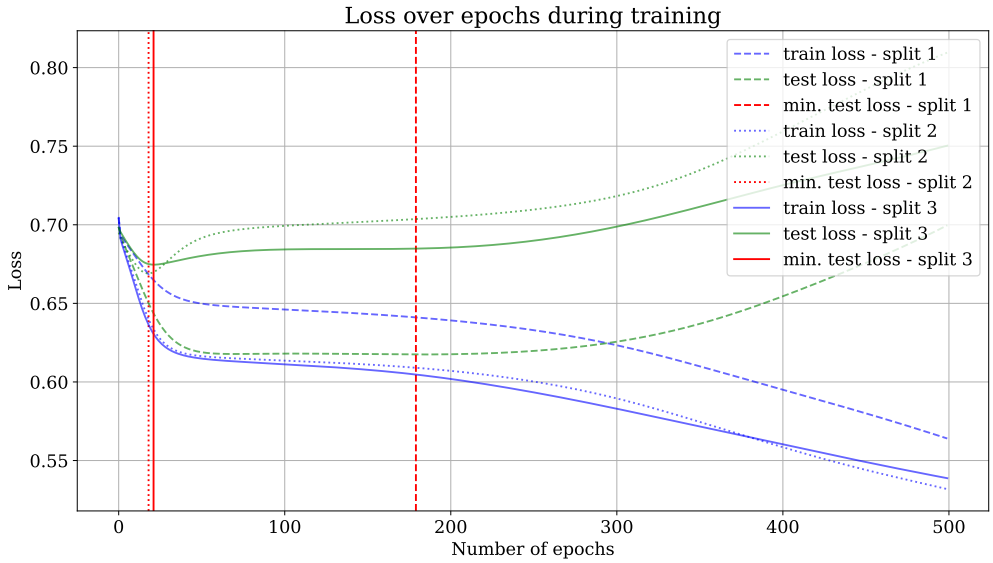

Figure 2.4 – Evolution of train and test loss curves during training for 3 splits. All the train loss curves (in blue) are decreasing over time. The test loss curves (in green) initially follow the same trend, and at some epoch, they start increasing again and this corresponds to overfitting. The epoch where the behaviour of these curves changes corresponds to the minimum of the test loss (in red) and can vary from one train/test split to another.

Concretely, we evaluated the train loss and test loss at each epoch during training, until the maximum number of epochs was reached. The trained model at each epoch was also

saved. At the end of training, the epoch where test loss was minimum was extracted, and the corresponding trained model was returned. This process, illustrated in Figure 2.4, ensured that training was stopped when test loss was at its minimum.

Correction for rare events The learning phase and the subsequent prediction of machine learning algorithms can be affected by the problem of imbalanced data set. The balancing issue corresponds to the difference between the number of samples in the different classes. With a greater imbalance ratio, the decision function favors the class with the larger number of samples, usually referred to as the majority class.

Since we cannot modify the loss function of the MLP classifier in `scikit-learn`, we cannot use the same technique than for logistic regression to restore the balance between classes. Another technique consists in oversampling the samples from the minority class (*i.e.* in our case, the positives), that is, replicating minority instances to increase their population. This was implemented using a python library called `imbalanced-learn`[5].

Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm which maps a vector of predictors into a higher dimensional plane through either linear or non-linear kernel functions [32]. In a binary classification problem, the objective of the SVM algorithm is to find a hyperplane in an N-dimensional space, N being the number of features, that distinctly classifies the data points in one of two groups, say $\{-1\}$ and $\{+1\}$. A decision hyperplane can be defined by an intercept term b, a decision hyperplane normal vector w which is perpendicular to the hyperplane and a non-linear function ϕ which maps the predictors into a higher dimension feature space [33]:

$$y(x) = w^T \phi(x) + b. \tag{2.14}$$

The classification function is then:

$$f(x) = \text{Sign}(w^T \phi(x) + b). \tag{2.15}$$

The objective of SVM is to find the plane that has the maximum "margin", which means the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement which ensures that future data points can be classified with more confidence. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Support vectors form supporting planes,

[5] https://imbalanced-learn.readthedocs.io/en/stable/

respectively $w^T\phi(x) + b \geq +1$ for the $\{+1\}$ class, and $w^T\phi(x) + b \leq -1$ for the $\{-1\}$ class. The classification is achieved by maximizing the margin of separation r between the two planes given by $r = 2/||w||$. This is equivalent to minimizing the cost function:

$$C(w) = \frac{||w||^2}{2} + c\sum_{i=1}^{n}\xi_i, \tag{2.16}$$

subject to the linear equality constraints:

$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \tag{2.17}$$

where $C > 0$ is analogous to the inverse of a regularisation coefficient and controls the trade-off between classification errors and model complexity (the margin), and ξ_i the slack-variable. This variable is the penalty of a misclassified observation that controls how far on the wrong side of the hyperplane a point can lie when the training data cannot be classified without error, that is when the objects are not linearly separable and a soft separating non-linear margin is required [33].

The nonlinear mapping by the feature function ϕ is computed through special nonlinear semi-positive definite K functions called kernels. Thus the minimization described in Equation 2.16 is generally solved through a dual formulation problem:

$$\min\frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{n}\alpha_i, \tag{2.18}$$

subjected to the following linear constraints:

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \text{ and } 0 \leq \alpha_i \leq C, \tag{2.19}$$

where $\alpha_i (i = 1, \ldots, n)$ are nonnegative Lagrange multipliers and K is a kernel function. For non-linear modeling one often employs an Radial Basis Function (RBF) kernel based on a Gaussian function, which is defined as:

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2). \tag{2.20}$$

The kernel coefficient γ defines the extent of the influence of a data point on the decision hyperplane. When γ is large the influence is local and the decision boundary is close as well. When γ is small many points are neighbors, even when they are from different classes. Compromise needs to be found based on local density of points of each class.

The solution of the classification problem is a weighted sum of kernels evaluated at the support

vectors.

Correction of rare events The SVM classifier from `scikit-learn` implements a `class_weight` parameter with a "balanced" mode which adjusts weights inversely proportional to class frequencies in the input data, similarly to the weighted likelihood system for logistic regression (c.f. Section 2.2.2).

2.2.4 Building a model

In machine learning, one of the main requirements is to build computational models with a high ability to generalize well the extracted knowledge. A classification result may be overly optimistic if performance cannot be measured on a data set that was not used for model training. In the ideal case, testing on a separate data set will provide an unbiased estimate of the generalization error. Generalization allows to evaluate how the system performs on unseen data, which means to see whether it has predictive probabilities for new patients. It corresponds to a typical machine learning approach: dividing the dataset in a training set and a test set. The machine is trained on the train set, and is then evaluated on the test set (Figure 2.5). The two sets are exclusive, meaning that the test set corresponds to unseen data for the model. The process of splitting the data in two sets is called a "protocol" and any new split defines a new protocol.

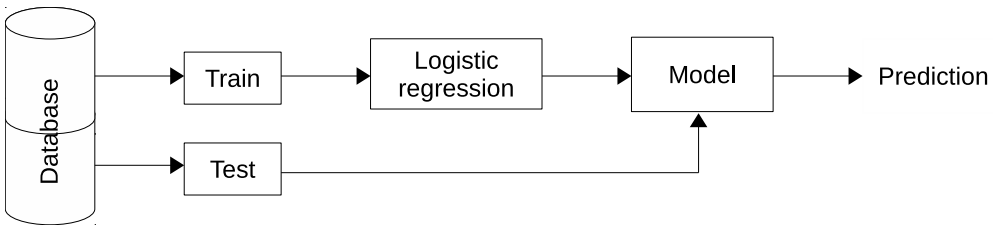

Figure 2.5 – Workflow of a typical machine learning approach. The data is split into two distinct subsets: the train and the test set. The model is trained using only the train set, and evaluated using the test set, which was not seen during training

Protocols

We divided the dataset in a train set consisting of 70% of the data, and a test set consisting of 30% of the data, which is customary in machine learning [34]. The separation was pseudo-random. Since the dataset contained a small number of positives, we wanted to have a similar number of positives in both subsets. Therefore, 70% of the positives were randomly selected and put in the train set, while the remaining 30% were put in the test set. The same thing was done for the negatives. This whole procedure was repeated 10 times at different seed

points, resulting in 10 protocols for each outcome. The seed points allow the procedure to be reproducible.

Another way to separate the data in a train and test set, which results in a less biased estimate of the model than simple train/test split, is k-fold cross-validation. It consists in randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a test set, and the method is fit on the remaining $k-1$ folds which constitute the train set [35]. In our experiments, we used both 5-fold and 10-fold cross-validation implemented with `StratifiedKFold`[6] in `scikit-learn`, which ensures that the folds are preserving the percentage of samples for each class.

The results of both the 70/30 split and k-fold cross-validation were reported using the mean model skill scores, as well as the standard deviation to include a measure of variance of the skill scores.

2.3 Metrics

This section presents the metrics which were used to report the results of our statistical models, and to create the figures of merit to compare the performance of the different models.

2.3.1 Odds ratio

The odds ratio metric applies to logistic regression, and allows to interpret the logistic regression estimates from the univariate and multivariate logistic regression analyses. The logistic regression coefficients β represent the logarithmic form of odds associated with each factor and are somewhat difficult to interpret by themselves. Odds ratios (OR) are used to compare the relative odds of the occurrence of the outcome of interest (patient has adverse event during treatment), given exposure to the variable of interest (*e.g.* smoking habit, concomitant medication, age, etc). The OR can also be used to determine whether a particular exposure is a risk factor for a particular outcome, and to compare the magnitude of various risk factors for that outcome. The OR represents the odds that an outcome will occur given a particular exposure, compared to the odds of the outcome occurring in the absence of that exposure.

- OR = 1: Exposure does not affect odds of outcome
- OR > 1: Exposure associated with higher odds of outcome
- OR < 1: Exposure associated with lower odds of outcome

As explained in the next paragraph, for categorical covariates, the OR is with respect to a

[6] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

reference category (exposure absent, odds ratio of 1.0). A variable representing the reference group will not be entered into the analysis; rather the other categories will be compared to this group. In multivariate analysis, the exponentials of the coefficients are adjusted OR (aOR) for having the outcome of interest, given that a particular exposure is present, while adjusting for the effect of other predictor factors. These aOR can be used to provide an alternative representation of the model. Contrary the OR or crude OR obtained when considering only one predictor variable, adjusted OR are obtained when taking into account the effect due to all the additional variables in the analysis.

Note on categorical variables When a logistic regression model is fitted to a data set which contains categorical explanatory variables, dummy variables are created to represent the different categories. A dummy variable, also known as an indicator variable, can take two values only, typically the values 0 or 1, to indicate the absence or presence of a characteristic.

Let X be the categorical predictor of age, which is divided in 3 categories: 18-34 years old, 35-64 years old, 65-90 years old. The number of dummy variables in a set that represents a nominal variable is equal to $K-1$, where K is the number of categories. Thus here, we define two dummy variables x_1 and x_2. For this age example, the regression model becomes:

$$\text{logit}(p_i) = \text{logit}(P(Y_i = 1|X_i = x_{i1}, x_{i2})) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}. \tag{2.21}$$

Which means that:

- If patient i is between 18 and 34 years old then $x_{i1} = 0$, $x_{i2} = 0$ (reference category) and $\text{logit}(p_i) = \beta_0$.
- If patient i is between 35 and 64 years old then $x_{i1} = 1$, $x_{i2} = 0$ and $\text{logit}(p_i) = \beta_0 + \beta_1$.
- If patient i is between 65 to 90 years old then $x_{i1} = 0$, $x_{i2} = 1$ and $\text{logit}(p_i) = \beta_0 + \beta_2$.

Therefore the intercept β_0 represents the log-odds for the reference category, and e^{β_0} is the baseline odds of having the outcome versus not having the outcome. The dummy variables β are the difference in log-odds compared to the reference category. For every unit increase in x_{i1}, the odds that the characteristic is present is multiplied by e^{β_1}. In other words, the exponential function of the regression coefficient $\exp(\beta_1)$ is the odds ratio associated with a one-unit increase in the exposure.

$$\frac{e^{(\beta_0+\beta_1(x_{i1}+1))}}{e^{(\beta_0+\beta_1 x_{i1})}} = e^{\beta_1}. \tag{2.22}$$

The first age group was chosen as the reference level and all results (and significant effects) presented are related to this reference level. A variable representing the reference group is not

entered into the analysis; rather the other categories are compared to this group. Investigators generally choose the reference category based on the main hypothesis being tested or on previous knowledge about the data. In the age example, we know that young people are usually less likely to develop adverse events and therefore they represent the most normative group.

2.3.2 Receiver Operating Characteristic (ROC) curve

ROC analysis investigates the accuracy of a model's ability to separate positive from negative cases [36] (*e.g.* predicting the presence or absence of adverse events). While classification accuracy depends on the prevalence of positive cases in the study population, ROC results do not, which makes them more accurate to evaluate our models, based on our imbalanced dataset. Binary classification outputs two discrete results (such as positive and negative) to infer an unknown, such as whether the patient has an adverse event or not. The accuracy of such a task is often assessed (in medicine) using measures of sensitivity SN and specificity SP, where:

$$SN = \frac{TP}{TP + FN}, \tag{2.23a}$$

$$SP = \frac{TN}{TN + FP}, \tag{2.23b}$$

and TP, TN, FP and FN are the counts of true positives, true negatives, false positives and false negatives, respectively. However, the output of our classifier is not directly a binary label, but rather a numeric value on a continuous scale, indicative of the likelihood of adverse events. Higher values indicate higher likelihood of the disease, while lower values indicate lower likelihood of the disease. To convert this value into a binary label, one must choose a threshold and compare the output value of that threshold, calling it positive if the value exceeds the threshold, and negative otherwise. Therefore, there is no particular value of sensitivity or specificity that characterizes the overall accuracy of the test, but rather an entire range of values, depending in the threshold used to discretize the test result. The ROC curve captures in a single graph the trade-off between a test's sensitivity and specificity over its entire range. Figure 2.6 illustrates how to analyze the performance of a classifier given its ROC curve.

The ROC curve is a good way to visualize a classifier's performance in order to select a suitable operating point (trade-off between FPR and TPR), or decision threshold. However, the operational point has not been determined in our case. Furthermore, it is often useful to have a single figure as a measure of classifier's performance when comparing different classification methods. The area under the ROC curve (AUC) has been shown to exhibit a number of desirable properties as a classification performance measure when compared to overall accuracy [37]. The ROC is a probability curve and the AUC represents the degree or measure

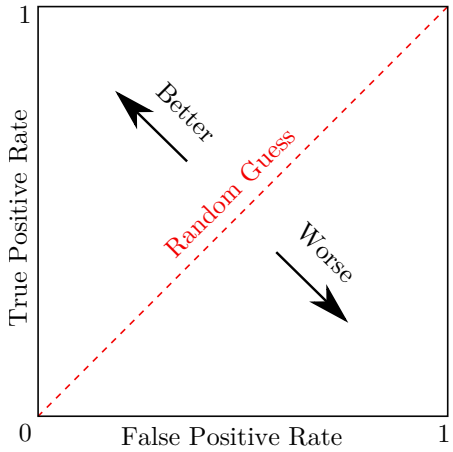

Figure 2.6 – Illustration of classifier performance on a ROC curve. A curve for a test with perfect accuracy would run vertically from the point (0, 0) to the point (0, 1) and then horizontally to (1, 1) at the top right of the graph. A curve for a test that performed no better than random guessing would run diagonally from (0, 0) to (1, 1) (red dashed line). Curves from real tests typically lie between these two extremes, in the upper left of the plot. If a test produces a curve that lies in the lower right, it means the test is incorrect more often than it is correct. The test could be improved by reversing its labels for positive and negative, which would reflect the ROC curve about the diagonal into the upper left of the plot. Any curve that lies completely above and to the left of another curve represents better test performance

of separability. The higher the AUC, the better the model is at distinguishing patients with adverse events and without adverse events.

2.4 The Bob framework

2.4.1 General presentation of Bob

Bob[7] is a signal-processing and machine learning toolkit designed to facilitate reproducibility in data science. It is meant to reduce development time and efficiently process data with reproducible research in mind. Reproducibility means that research should be repeatable, shareable, extensible and stable [2]. A strong emphasis is put on code clarity, documentation and unit testing. Concretely, Bob consists of a collection of tools and interfaces implemented in both C++ and Python, for researchers to prototype their ideas and algorithms.

A general workflow for pattern recognition or predictive tasks in Bob is represented in Figure 2.7. The first step consists in loading data from available raw samples and using a database

[7] https://www.idiap.ch/software/bob/

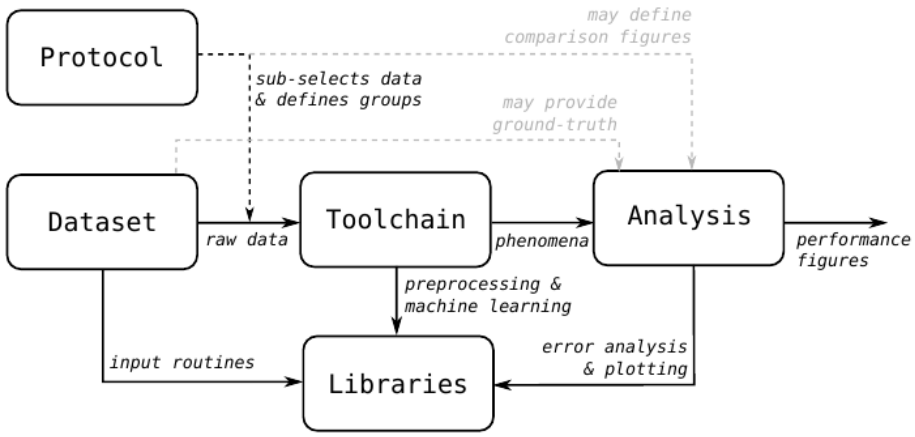

Figure 2.7 – Typical workflow in Machine Learning and Pattern Recognition [2]. The raw data is selected according to a given protocol and converted through a series of processing steps (toolchain). The protocol can also influence the comparison figures being used. Most components of the workflow can be shared via libraries.

protocol to specify how the contents of the database should be used for experiments. The data is then put through a series of processing steps (toolchain), which outputs a representation of the data for the analysis. Finally, the analysis yields figures of merit, which include measures of accuracy in detecting phenomena. All components are frequently re-used and shared via libraries. This workflow is encapsulated into a package, with unit tests for the implementation of the source code, and incorporated into the automatic testing framework of Bob (*i.e.* continuous integration system that is built into the git source code distribution platform).

2.4.2 Implementation using Bob

The Bob framework was used to implement the predictive model with two Bob packages (Figure 2.8).

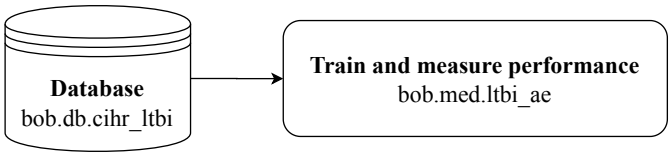

Figure 2.8 – Bob packages implemented in the project. The database interface is controlled by the first package, while the second package deals with the training and evaluation of the models

- `bob.db.cihr_ltbi` [8]: The package is called `bob.db` because it is a Bob database and `cihr_ltbi` because it is a database of LTBI patients collected by the Canadian Institutes of Health Research (CIHR). It acts as an interface to access the database, which cannot be uploaded on Bob, since it consists of sensible medical data. Concretely, this package enables the user to load clinical data (see Section 2.1) from its excel container, according to a given protocol.

- `bob.med.ltbi_ae` [9]: This package is responsible for training different models, evaluating them, and outputting figures of merit, such as Receiving Operating Characteristic (ROC) curves. In the case of the logistic regression model, this package estimates the beta coefficients in the logistic regression equation 2.3.

Both packages are hosted on the gitlab page of Bob (links in footnote) where their respective documentation can be found.

[8] https://gitlab.idiap.ch/bob/bob.db.cihr_ltbi
[9] https://gitlab.idiap.ch/bob/bob.med.ltbi_ae

3 Experiments

This chapter describes the experimentations done with the dataset in order to provide a predictive model of adverse events based on patient clinical data. In order to establish a baseline from the literature, we start by reproducing the results of Campbell et al. [1] who had access to the same clinical dataset. The second experiment compares different implementations to obtain the most efficient predictive model of adverse events with logistic regression. This best model is then generalized in the third experiment by splitting the dataset in a train and test set, to be able to evaluate the predictive performance of the model on unseen data. This model, based on logistic regression, is considered as a baseline to improve upon. The final experiment explores non-linear machine learning models (MLP and SVM) to improve the predictive power of the linear baseline.

3.1 Reproducing Campbell et al. (2019)

The article of Campbell et al. [1] consists in a post-hoc safety analysis, based on the clinical dataset described in Chapter 2. Their goal was to establish risk factors for adverse events during LTBI treatment, and to show that the isoniazid treatment (9INH) results in more adverse events than the rifampin treatment (4RIF). Although the second aspect of this research is less relevant for a general predictive model of adverse events, reproducing the statistical analysis of this article allows to confirm the influence of risk factors on the occurrence of adverse events during the treatment. With this experiment, we wanted to verify whether the clinical variables present in our dataset allowed to predict the occurrence of adverse events, and to have a reference for future experiments.

Here is a brief description of the statistical analysis carried out in Campbell et al. [1]: For each of the outcomes introduced in Section 2.1.2, an univariate analysis via logistic regression was carried out for previously selected potential risk factors for each treatment arm separately

(4RIF and 9INH). A multivariate logistic regression model was then created, for each outcome, including the covariate of age and all covariates that were significantly associated to the occurrence of the given outcome, which means they had a p-value < 0.1 in univariate analysis. The potential risk factors in univariate analysis were pre-selected according to literature and are displayed in Section 2.1.1 Table 2.1. For hepatotoxicity analysis (outcome 2), alanine transaminase (ALT) measures were also included as a potential risk factor, since ALT is an biomarker for liver health. In the hematologic analysis (outcome 4), white blood cell (WBC) and platelet measures were also included. For the first three outcomes, further analysis (again univariate and multivariate logistic regression) included the treatment arm (4RIF or 9INH) as potential risk factor.

3.1.1 Results

In Campbell et al. [1], the results from the logistic regression analyses are reported as tables displaying OR estimates for univariate and multivariate logistic regression. We reproduced those tables, and present here the ones reported in the main article, the tables from supplementary information being in appendix A. The tables of results reported in Campbell et al. [1] correspond to the systems predicting respectively the primary (Figure 3.1) and secondary (Figure 3.2) outcomes.

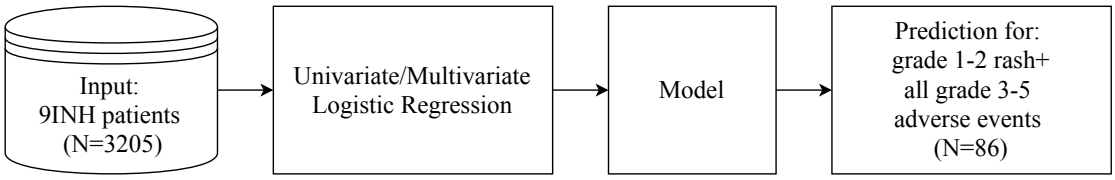

Figure 3.1 – Block diagram of logistic regression system of Table 3.1 to predict primary outcome from patients following isoniazid treatment. From 9INH patient clinical data, apply a univariate or multivariate logistic regression model to predict the occurrence of grade 1-2 rash combined to all grade 3-5 adverse events.

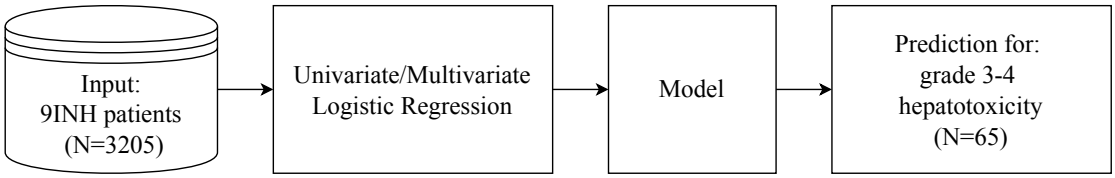

Figure 3.2 – Block diagram of logistic regression system of Table 3.2 to predict secondary outcome from patients following isoniazid treatment. From 9INH patient clinical data, apply a univariate or multivariate logistic regression model to predict the occurrence of grade 3-4 hepatotoxicity adverse events.

Table 3.1 displays the estimates for univariate and multivariate logistic regression, using clinical data of patients following the isoniazid treatment ($N = 3205$) to evaluate the risk of primary outcome (grade 1-2 rash + all grade 3-5 adverse events, $N = 86$). Each line of the table corresponds to one category of one clinical variable. The first column ("Number") is the number of patients in that category, and the second column ("Risk") is the number of patients within that category who are positive for the outcome. Univariate OR estimates are reported in the third column, and represent how more likely the patients belonging in a given category are to be positive for the outcome, compared to the reference category. For example, in Table 3.1, looking at the age variable, patients between 35 and 64 years old are 1.9 times more likely to have an adverse event (following primary outcome definition) compared to patients between 18 and 34 years old. Similarly, patients between 65 and 90 years old are 3.4 times more likely to have an adverse event than reference category patients. This kind of metric allows clinicians to deduce which patients are more at risk to develop adverse events (here older patients). In the fourth column, OR estimates correspond to logistic estimates for the multivariate analysis, performed using only variables that were considered significant from the univariate analysis. In Table 3.1, multivariate analysis uses age and concomitant medication and confirms that older patients are more likely to develop adverse events compared to young patients ($OR = 3$). Finally, the green values correspond to values that were equal to the ones reported in the article, while blue ones represent values that differed, with the exact value in parenthesis. As can be seen from Table 3.1, most values are exactly equal, with 0.1 variation for the remaining values.

Table 3.2 displays the estimates for univariate and multivariate logistic regression, using clinical data of patients following the isoniazid treatment ($N = 3205$) to evaluate the risk of secondary outcome (grade 3-4 hepatotoxicity, $N = 65$). For multivariate analysis, more risk factors were considered significant enough to be added as covariates. Old age remains an important risk factor for this outcome ($OR = 2.3, 5.3$), followed by immuno-suppressing conditions ($OR = 1.9, 1.7$) and pre-treatment ALT values ($OR = 2.5$). Again few values differ between the original article and our results, demonstrating robust results for this outcome as well.

Tables from supplementary information (see Appendix A) correspond to logistic regression estimates for other outcomes, as defined in Table 2.3, either for patients following the rifampin treatment, or the isoniazid treatment. Those analyses also include logistic regression combining patients for both treatment, and considering the treatment arm as a covariate.

From Table 3.1, we can conclude that old age and the use of concomitant medication increase the risk of grade 1-2 rash combined with all grade 3-5 adverse events. Results displayed in Table 3.2 show that old age, immunosuppressing conditions, frequent alcohol use, smoking and ALT levels above normal increase the risk of grade 3-4 hepatotoxic adverse events.

Table 3.1 – Results of univariate and multivariate model of risk factors for grade 1-2 rash + all grade 3-5 adverse events attributed to isoniazid. This table reproduces the left part of table 3 in Campbell et al. [1]. Most values are exactly equal to those from Campbell et al. [1] (in green) and only two values differ (in blue). In univariate analysis, age and concomitant medications significantly influence the occurrence of the primary outcome. Patients who are between 65 and 90 years old are 3.4 times more likely to suffer from this outcome compared to younger patients between 18 and 34 years old. Patients taking concomitant medications are 1.7 times more likely to develop this outcome compared to patients without any concomitant medications. In multivariate analysis, age remains the most influential risk factor.

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Age	18-34	1436	25	1 (ref)	1 (ref)
	35-64	1642	54	1.9	1.8
	65-90	127	7	3.4 (3.5)	3.0
Sex	Female	1811	48	1 (ref)	-
	Male	1394	38	1.0	-
BMI	Normal	1646	44	1 (ref)	-
	Underweight	222	5	0.9	-
	Overweight	907	26	1.1	-
	Obese	430	11	1.0	-
Immune Status	No Immune suppr.	2871	73	1 (ref)	-
	HIV-positive	138	5	1.6	-
	Other immune suppr.	196	8	1.7	-
Alcohol Use	Never drinks	2112	58	1 (ref)	-
	≤ 1 drink per week	891	20	0.8	-
	> 1 drink per week	202	8	1.5	-
Smoking history	Has never smoked	2421	60	1 (ref)	-
	Currently or has smoked	784	26	1.4	-
Medication Consistency	Consistency ≥ 90%	2151	57	1 (ref)	-
	Consistency < 90%	1054	29	1.0 (1.1)	-
Concomitant medications	None	2470	58	1 (ref)	1 (ref)
	Any	735	28	1.7	1.3

Table 3.2 – Results of univariate and multivariate model of risk factors for grade 3-4 hepatotoxicity attributed to isoniazid. This table reproduces the right part of table 3 in Campbell et al. [1]. Only one value differs from Campbell et al. [1] (in blue) while all others are exactly equal (in green). From univariate analysis, many risk factors are found significant enough to be included in multivariate analysis. In particular, age seems to have a large influence on this outcome: older patients are 5.7 times more likely to suffer from it compared to younger patients. Having an immunosuppressing condition or ALT levels above normal also strongly increases the risk to suffer from grade 3-4 hepatotoxicity. In multivariate analysis, age and ALT levels remain the most influential risk factors

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Age	18-34	1436	15	1.0 (ref)	1.0 (ref)
	35-64	1642	43	2.5	2.3
	65-90	127	7	5.7	5.3
Sex	Female	1811	33	1.0 (ref)	
	Male	1394	32	1.3	-
BMI	Normal	1646	34	1.0 (ref)	
	Underweight	222	4	1.0	-
	Overweight	907	19	1.0	-
	Obese	430	8	0.9	-
Immune Status	No Immune suppr.	2871	52	1.0 (ref)	1.0 (ref)
	HIV-positive	138	5	2.2	1.9
	Other immune suppr.	196	8	2.4	1.7
Alcohol Use	Never drinks	2112	40	1.0 (ref)	1.0 (ref)
	≤ 1 drink per week	891	17	1.0	0.9
	> 1 drink per week	202	8	2.2	1.8
Smoking history	Has never smoked	2421	42	1.0 (ref)	1.0 (ref)
	Currently or has smoked	784	23	1.7	1.4
Medication Consistency	Consistency ≥ 90%	2151	48	1.0 (ref)	
	Consistency < 90%	1054	17	0.7	-
Concomitant medications	None	2470	42	1.0 (ref)	1.0 (ref)
	Any	735	23	1.9	1.1
Pre-treatment ALT	Normal	2972	56	1.0 (ref)	1.0 (ref)
	Above normal	196	9	2.6	2.5 (2.6)

3.2 Building a logistic regression model

As a transition between the work of Campbell et al. [1] and the introduction of train/test separation to create a predictive model, we examined the performance of the logistic regression models presented in Tables 3.1 and 3.2 using ROC curves. To that end, we trained the logistic models on the whole dataset and evaluated them on the same dataset, which resulted in a bias that is corrected later when introducing a less biased evaluation protocol. The aim of this experiment was to choose which logistic regression model to select among those presented in the article. From the multiple logistic regression models introduced in Campbell et al. [1], our aim was to select the most conclusive implementation that we would later fine-tune and feed to an unbiased evaluation protocol.

In the following set of comparisons, the idea was to select the best implementation of logistic regression. More precisely, here were the different possible configurations:

- Correction for rare events: either Firth penalized log-likelihood or weighted log-likelihood
- Logistic regression model: either univariate or multivariate logistic regression
- Number of covariates included in multivariate model: either only significant ones or all available variables
- Type of covariates: either categorical or continuous (if possible)

For each comparison, in order to select the best model, we compared the model ROC curves. We changed one aspect at a time and each time kept the best configuration for the next comparison. The logistic models correspond to Figures 3.1 and 3.2. They were trained on clinical data from patients following the isoniazid treatment ($N = 3205$) and were evaluated on the same dataset. The outcome was either the primary outcome ($N = 86$) or the secondary outcome ($N = 65$).

3.2.1 Correction for rare events

As explained in Section 2.2.2, correction for rare events in Campbell et al. [1] was implemented using Firth penalized likelihood. However, a computationally more tractable method consists in balancing classes by adding proportional weights in the cross-entropy loss function. This section is a comparison between both methods, to verify that they yield similar results, making it acceptable to keep the weighted likelihood implementation for our models. Figures 3.3 and 3.4 display the ROC curves corresponding to Firth and weighted implementations respectively, of the univariate and multivariate logistic regression models in Table 3.1 (primary outcome for isoniazid patients).

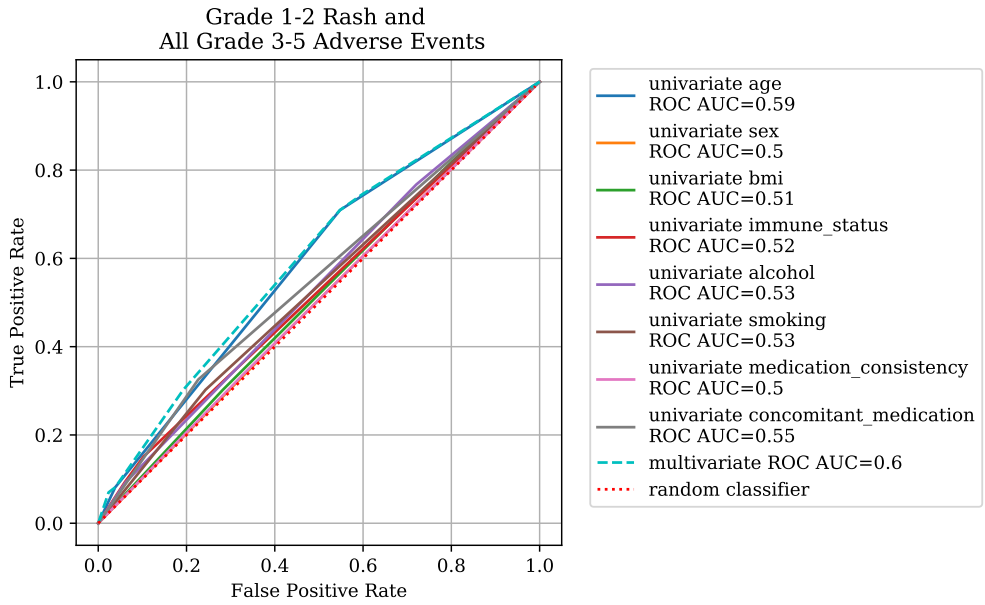

Figure 3.3 – ROC curves of univariate and multivariate logistic regression models, using Firth likelihood implementation, 9INH patients ($N = 3205$), primary outcome ($N = 86$). Most classifiers have a very poor performance close to random guessing (red dotted line), except age and the multivariate classifier.

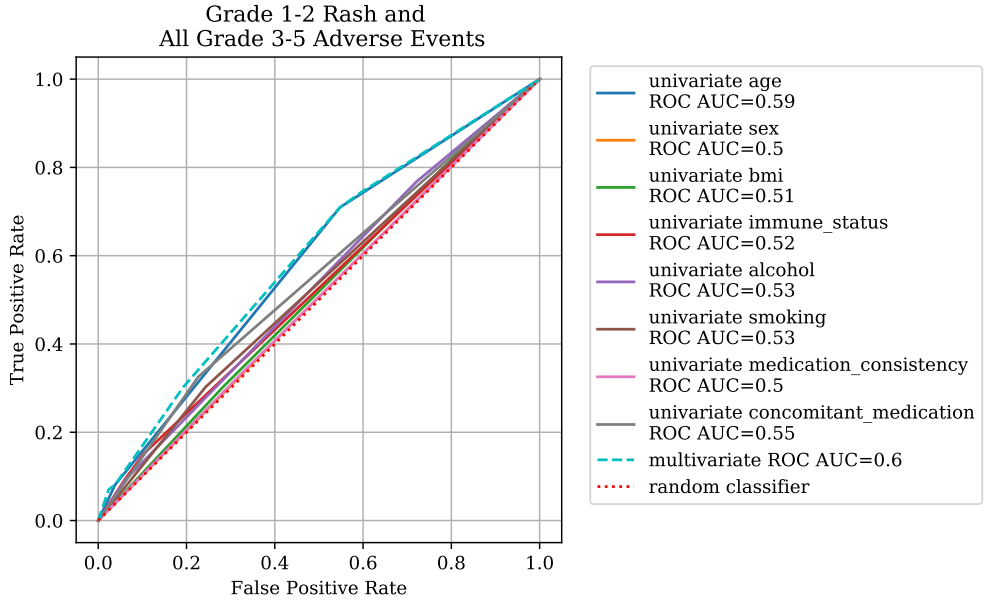

Figure 3.4 – ROC curves of univariate and multivariate logistic regression models, using weighted likelihood implementation, 9INH patients ($N = 3205$), primary outcome ($N = 86$). All ROC curves seem to be exactly the same than those in Figure 3.3 and their AUC also match perfectly

It appears that the ROC curves are very similar, if not indistinguishable (all AUC are equal), when comparing both implementations. We can also observe on these curves that the individual performance of each classifier is quite poor as it does not differ much from the random classifier (red dotted line), and the best AUC only equals 0.6. For this outcome, the best univariate classifier is the one using age as a risk factor (AUC= 0.59), which corresponds to the highest OR estimates observed in Table 3.1.

To have a better idea of the impact of the likelihood implementation on the ROC curves, we show the same comparison for the secondary outcome. Figures 3.5 and 3.6 display the ROC curves corresponding to firth and weighted implementations respectively, of the univariate and multivariate logistic regression models in Table 3.2 (secondary outcome for isoniazid patients). For these curves, we begin to see a small difference between firth and weighted implementations, in particular for the multivariate model (blue dashed line). However, the ROC curve for the multivariate model in Figure 3.6 is closer to the ideal point (0,1) than the ROC curve for the multivariate model using firth implementation. This indicates that the multivariate model is a slightly better classifier using weighted implementation. The actual difference based on AUC of the multivariate models is of 0.01 between the firth and weighted implementations. All the univariate ROC curves have exactly the same AUC using Firth or weighted implementation.

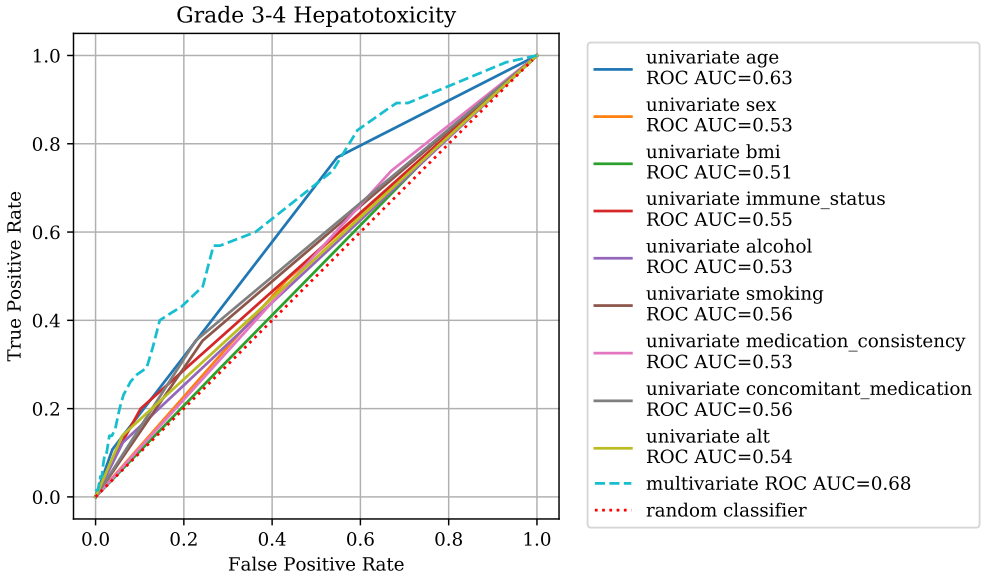

Figure 3.5 – ROC curves of univariate and multivariate logistic regression models, using firth likelihood implementation, 9INH patients ($N = 3205$), secondary outcome ($N = 65$). Again, all classifiers have ROC curve close to the random classifier, except age and the multivariate classifier, which performs better than the multivariate classifier for the primary outcome.

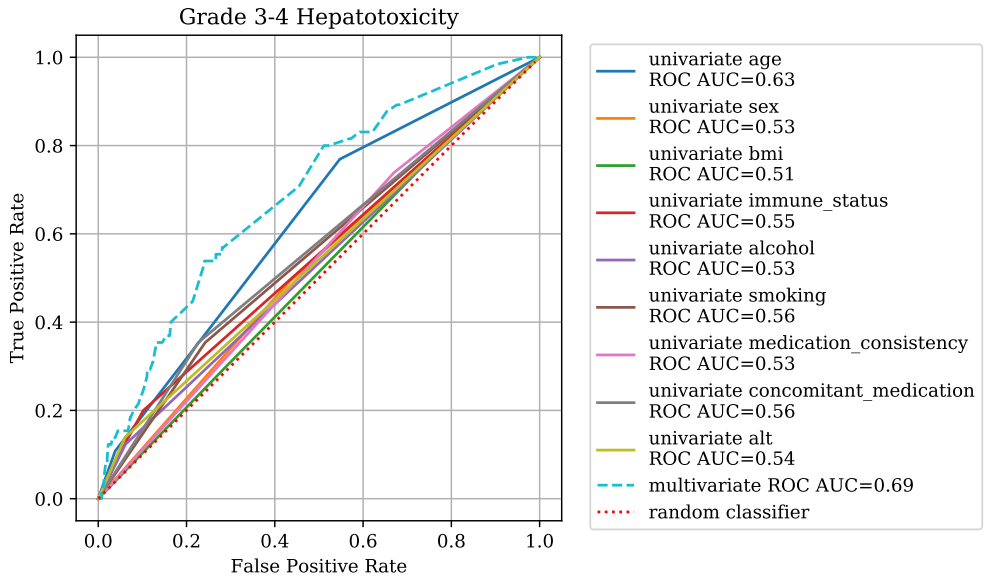

Figure 3.6 – ROC curves of univariate and multivariate logistic regression models, using weighted likelihood implementation, 9INH patients (N = 3205), secondary outcome (N = 65). All univariate classifiers are exactly the same than in Figure 3.5 and their AUC are also exactly equal. The only ROC curve that differs is the multivariate classifier one, with 0.01 difference in AUC.

Based on these two comparisons, we decided to keep weighted implementation of the log-likelihood as a correction for rare events, since it yielded similar if not better classifiers and allowed to perform computations more quickly.

3.2.2 Logistic regression model

In this second comparison, the goal was to choose the best logistic regression model between the univariate and multivariate models. As seen on figures from the previous experiment (Figures 3.4 and 3.6), the multivariate model was always the best classifier (AUC $\geq$ 0.6), compared to univariate classifiers (AUC $<$ 0.6). Thus, from here on, we only considered multivariate models. The best multivariate model for the primary outcome corresponds to the blue dashed line on Figure 3.4 with an AUC of 0.6. The best multivariate model for the secondary outcome corresponds to the blue dashed line on Figure 3.6 with an AUC of 0.68.

3.2.3 Number of covariates

To improve the multivariate model, other variables can be added as risk factors. In Campbell et al. [1], only the risk factors which were considered significant enough in the univariate analysis are incorporated in the multivariate analysis. However, using all available variables

might yield a better predictive performance, since risk factors might interact with each other in an unknown way. Figures 3.7 and 3.8 depict the ROC curves for multivariate models including either all clinical variables, or only significant covariates chosen from univariate analysis.

Judging from Figures 3.7 and 3.8, using all clinical variables resulted in a better classifier, although the AUC difference was not large (0.04 for the primary outcome, 0.01 for secondary outcome). Future models use all clinical variables as covariates in the multivariate logistic regression model.

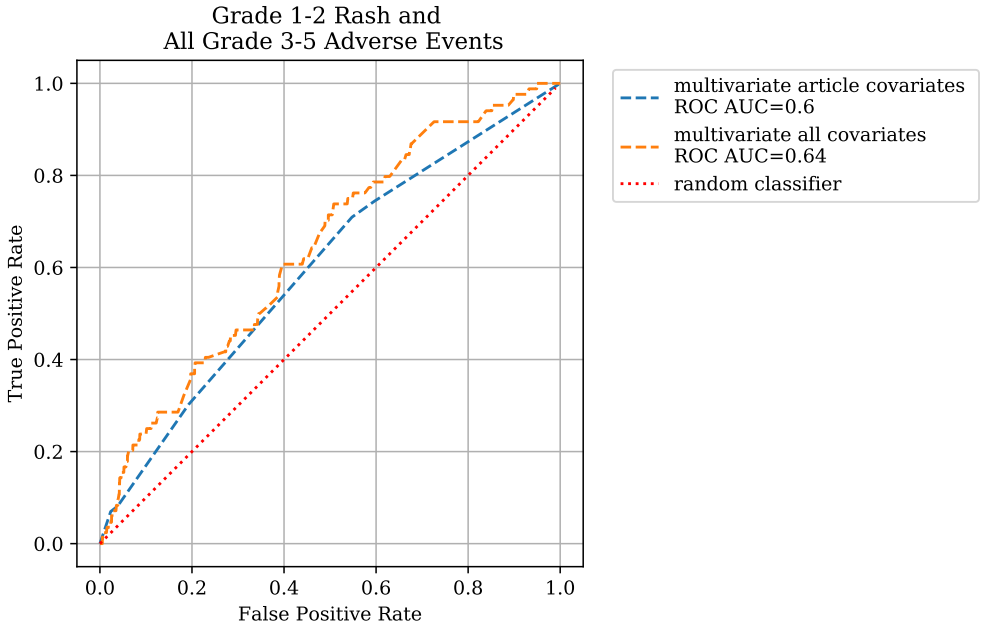

Figure 3.7 – ROC curves of multivariate logistic regression models, based on all or only significant covariates, 9INH patients ($N = 3205$), primary outcome ($N = 86$). Article covariates correspond to significant covariates determined in Campbell et al. [1]. For the primary outcome, they correspond to age and concomitant medication. The ROC curve of the multivariate classifier using all variables (orange curve) is higher than the other curve.

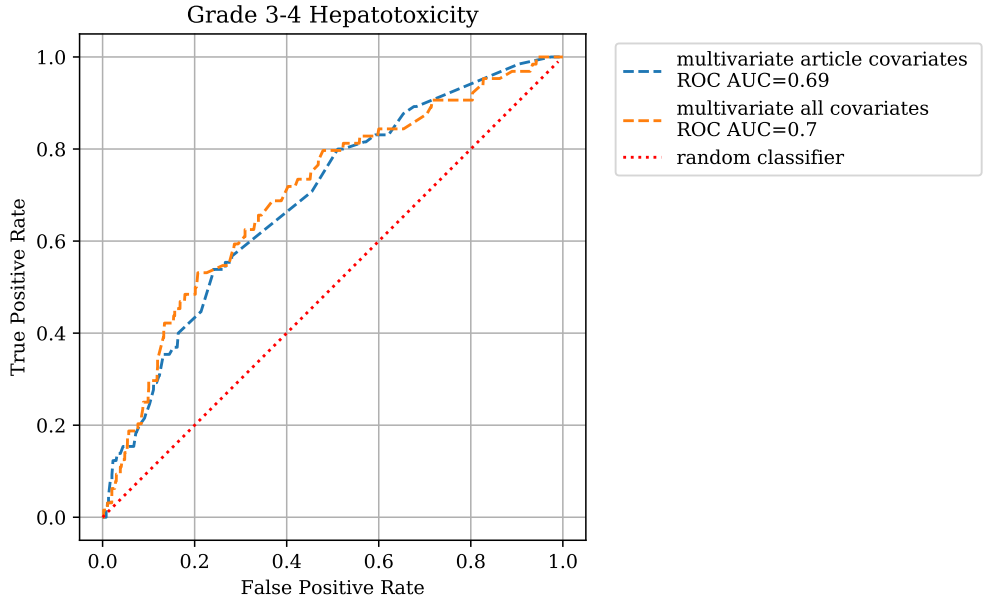

Figure 3.8 – ROC curves of multivariate logistic regression models, based on all or only significant covariates, 9INH patients ($N = 3205$), secondary outcome ($N = 65$). Article covariates correspond to significant covariates determined in Campbell et al. [1]. For the secondary outcome, they correspond to age, immune status, alcohol use, smoking history, concomitant medication and ALT. The ROC curves of both multivariate classifiers are quite similar.

3.2.4 Type of covariates

Finally, we investigated the type of clinical variables used in the multivariate model. As explained in Section 2.1.1, some of the covariates such age, bmi, etc, were originally continuous variables in the dataset and were then discretized in categorical variables. Using categorical covariates is useful to detect specific risk factors for clinicians. However, using continuous instead of categorical covariates might increase the predictive performance of the model, since it distinguishes more the patients. Therefore, we compared the multivariate model with all categorical variables versus the multivariate model with a mix of categorical and continuous variables, since not all variables were originally continuous. Only age, BMI, medication consistency and lab values (ALT, WBC, and platelet) were incorporated as continuous variables in the multivariate model. Since the values of these variables can have very different ranges, the covariates were normalized (which was not needed for categorical-only set of covariates). Figures 3.9 and 3.10 depict the ROC curves for multivariate models including either only categorical variables, or a mixed of categorical and continuous covariates.

For both the primary and secondary outcome, the multivariate model containing a mix of categorical and continuous variables performed slightly better (AUC= 0.67, 0.73) than the

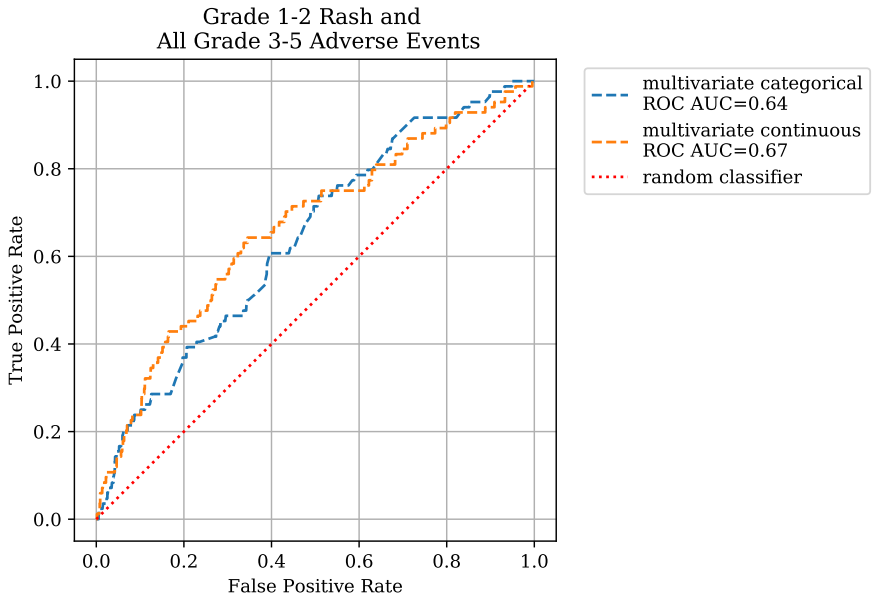

Figure 3.9 – ROC curve of multivariate logistic regression models, based on categorical or continuous covariates, 9INH patients ($N = 3205$)**, primary outcome** ($N = 85$)**.** The multivariate model using a mix of categorical and continuous covariates (orange curve) performs better than the model using only categorical variables (blue curve)

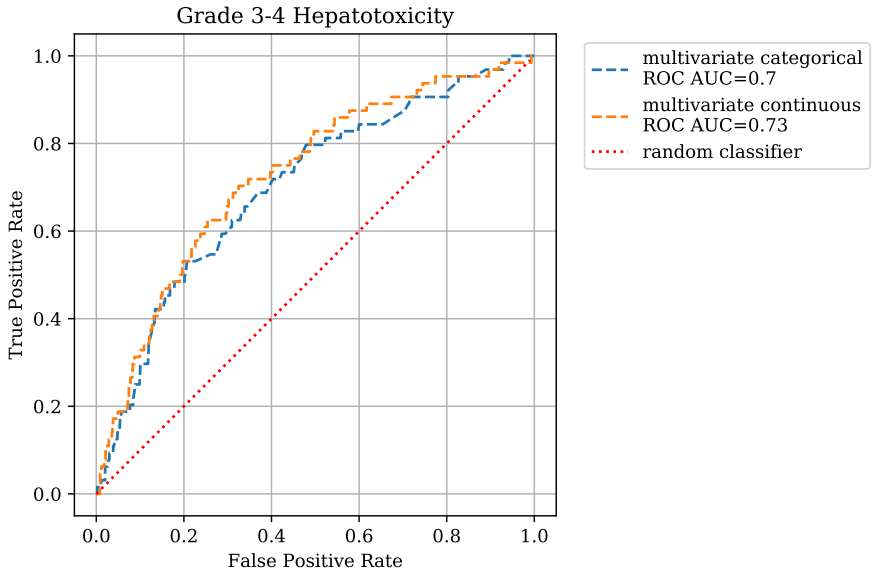

Figure 3.10 – ROC curves of multivariate logistic regression models, based on categorical or continuous covariates, 9INH patients ($N = 3205$)**, secondary outcome** ($N = 65$)**.** The multivariate model using a mix of categorical and continuous covariates (orange curve) performs better than the model using only categorical variables (blue curve)

multivariate model containing only categorical variables (AUC= 0.64, 0.7).

To sum up the results from these comparisons, the best logistic regression classifier for the primary and secondary outcomes had the following characteristics: weighted likelihood, multivariate logistic regression, all available clinical variables, mix of categorical and continuous variables.

3.2.5 Best logistic regression model

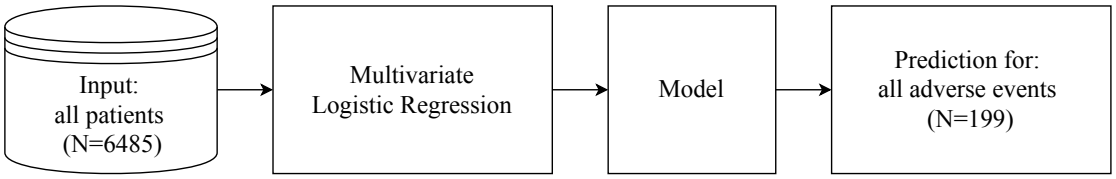

Figure 3.11 – Block diagram of logistic regression system to predict any adverse event from any patient. From all patient clinical data, apply a multivariate logistic regression model to predict the occurrence of any adverse event

The set of comparisons from the previous section was made using the primary and secondary outcomes defined by Campbell et al. [1] and only considering the isoniazid treatment. The ROC curves helped visualize the performance of logistic regression models to predict these specific outcomes in patients following the isoniazid treatment. However, the goal of this project was to create a predictive model of all adverse events for patient following either the rifampin or the isoniazid treatment. Therefore, we needed to see if those results were confirmed when considering all patients and all adverse events, which corresponds to the system displayed in Figure 3.11. The same comparisons were made using the whole dataset (N = 6485) and all adverse events(N = 199), and the resulting ROC curves are displayed in Figure 3.12. On this plot, the worse classifier was again the univariate one (AUC= 0.59). Multivariate models using either firth likelihood or penalized likelihood were very similar (0.01 AUC difference) and the same was true for using either categorical or continuous variables.

To sum up, here is the final logistic regression scheme, which was used in the generalization experiment:

- approach: train and evaluate on the same dataset
- protocol: all patients (N = 6485)
- outcome: all adverse events (N = 199)
- model: multivariate logistic regression
- covariates: all clinical variables, categorical and continuous when possible, normalized

- correction for rare events: weighted log-likelihood

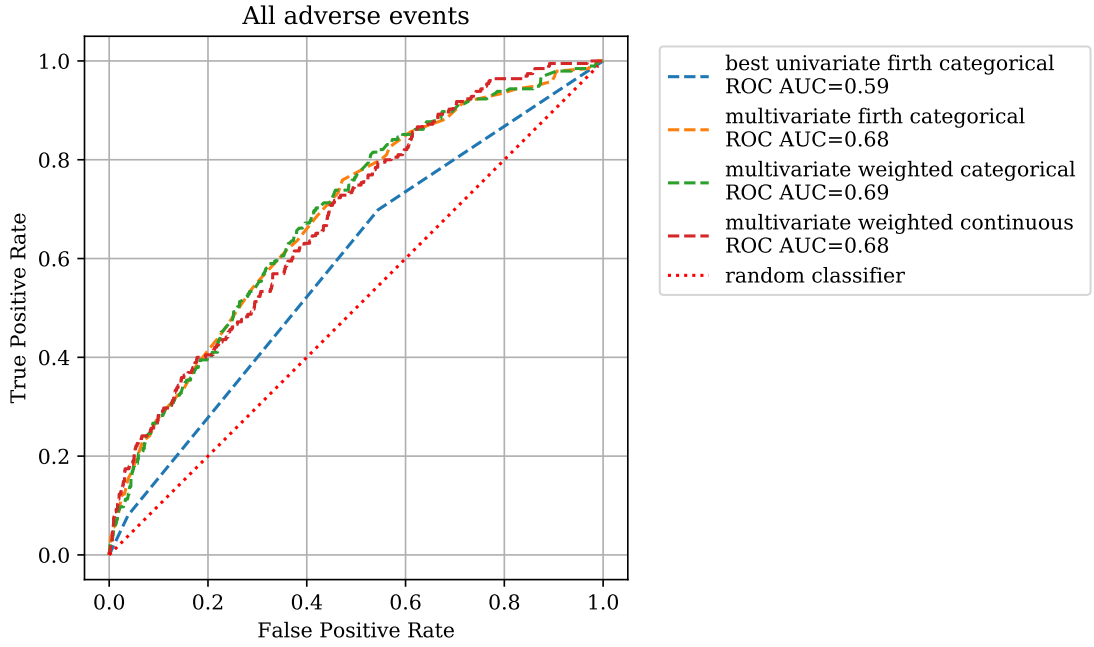

Figure 3.12 – Comparison of ROC curves of univariate and multivariate logistic regression models, all patients ($N = 6485$), all adverse events ($N = 199$). The best univariate classifier (in blue) is largely inferior to all 3 multivariate classifiers (orange: Firth implementation and categorical variables, green: weighted implementation and categorical variables, red: weighted implementation and a mix of categorical and continuous variables), which have a similar performance.

3.3 Generalization

As explained in Section 2.2.4, the unbiased evaluation of a predictive model requires to separate the dataset in a distinct train and a test set. This configuration, depicted in Figure 3.13, allows the model to be evaluated on data that was not seen during training, which prevents bias. This section introduces the predictive model based on the best multivariate logistic regression model (see Section 3.2.5), using train and test sets described in Section 2.2.4.

Figure 3.14, displays ROC curves of the multivariate logistic regression model, with a split between a train and a test set, and without split. We observe a small loss in predictive power (0.04 difference in AUC) between the train and test set for the model with train/test separation, which indicates that there is a bias when training and evaluating on the same dataset, without split. Both the training and testing performance were quite poor for this model (train AUC= 0.69 ± 0.02, test AUC= 0.65 ± 0.04), which indicates underfitting. The model does was not

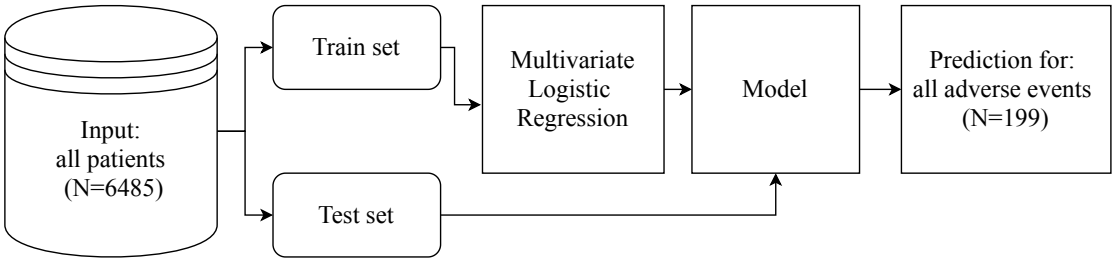

Figure 3.13 – Block diagram of logistic regression system with a train/test split to predict any adverse event from any patient. The input data is divided into a train set, used to train a multivariate logistic regression model, and a test set, used to evaluate the performance of the model.

able to clearly separate the data between positives and negatives. The model with train/test split performed equally well on the training set than the model without split, meaning that 70% of the dataset is enough to obtain this level of performance. As expected, there was more variability in the testing set (AUC std=0.04) compared to the training set (AUC std=0.02) performance between the different splitting protocols. The best performing multivariate logistic regression model had a test AUC of 0.65(±0.04), and constituted the baseline model to improve on.

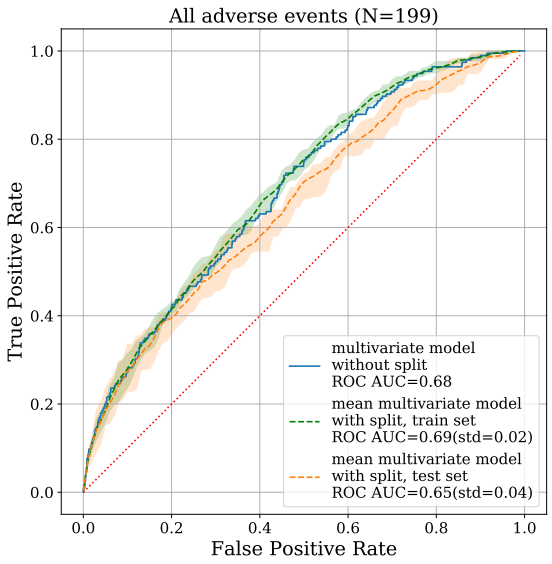

Figure 3.14 – ROC curves of multivariate logistic regression model, with and without separation between train and test set, all patients (N = 6485), all adverse events (N = 199). The multivariate model performance on the train set (in green) is similar to the multivariate performance on the whole dataset (in blue). There is a loss of performance when evaluating the model on the test set (in orange) indicating that there is a bias when training and evaluating on the same dataset.

3.4 Improving the model with non-linearity

Results from the best performing logistic regression model indicated a quite low predictive performance (AUC < 0.7), both on the train and test set, with the test performance being slightly lower. The aim of this section was to experiment whether we could recover the loss of performance between the train and the test set, but more importantly, whether we could improve the overall performance of the model. To do this, two non-linear machine learning models were investigated: MLP and SVM.

3.4.1 MLP

The first machine learning model explored was the MLP with a single layer with a varying number of hidden neurons. Increasing the number of hidden neurons increases the complexity of the model. The complexity of a neural network can be roughly estimated by calculating the total number of weights it contains. Such a number should be, in practice for an MLP, as small as possible such that it still performs well on both train and test sets. If this number is increased too much, it tends to overfit on the training data. As we have a rather small dataset (roughly 6000 samples with only 200 positives), we started with a small MLP with one layer containing from 2 to 10 neurons. The workflow of the system from the input features to the evaluation of the model is depicted in Figure 3.15.

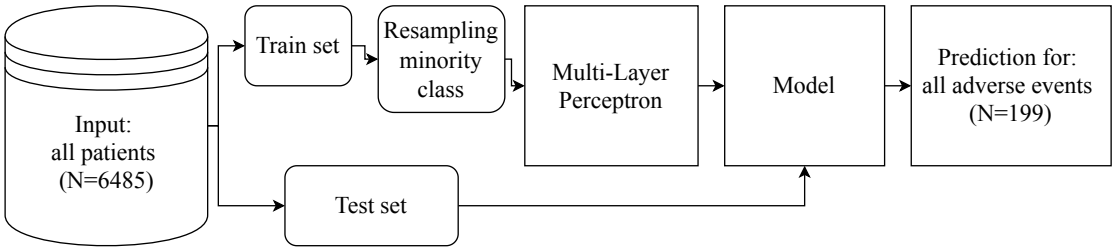

Figure 3.15 – Block diagram of MLP system with a train/test split and resampling on the train set to predict any adverse event from any patient. The input data is split into a train and a test set. The train set is first resampled to balance the dataset between positives and negatives, and then used to train the model. The test set is not resampled, and is used to evaluate the performance of the model.

We first used the MLP with the default hyperparameters suggested by `scikit-learn` for small datasets:

- activation: logistic,
- solver: L-BFGS,
- regularization: 0.0001,

- maximum number of iterations: 200

The resulting ROC curves are displayed in Figure 3.16. With an increasing number of neurons, the training performance greatly increased, as the ROC curve approached the top left point of the plot (best train AUC= 0.94). However, this did not translate into the testing performance, which remained quite low (best test AUC= 0.62) and inferior to the logistic regression performance (AUC= 0.65). This situation indicates overfitting: the model learns "by heart" the training data, which makes it inaccurate for unseen data. Once again, there was more variability in the testing set compared to the training set.

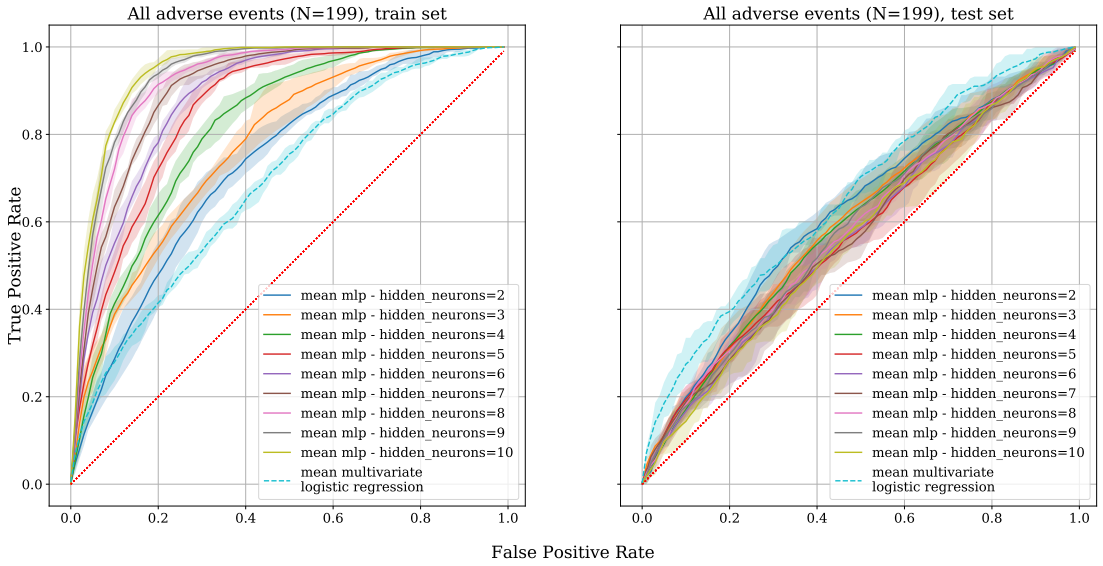

Figure 3.16 – ROC curves of MLP with hidden neurons varying from 2 to 10, L-BFGS optimizer, all patients ($N = 6485$), all adverse events ($N = 199$). On the left plot, the bigger number of neurons, the higher the train ROC curve and all MLP train ROC curves are higher than the multivariate logistic regression train ROC curve (blue dashed line). However, on the right plot, all MLP test ROC curves are lower than the multivariate logistic regression test ROC curve. As opposed to the left plot, the best MLP classifiers on the right plot seem to be the ones with the smaller number of neurons (blue, orange, and green curve).

Regularization

The reduce overfitting when using the L-BFGS optimizer, we increased the regularization hyperparameter α. With this experiment, we wanted to analyze whether increasing the regularization factor could reduce overfitting and thus improve performance. Figure 3.17 considers the MLP with 10 hidden neurons, which overfitted the most on Figure 3.16, and displays the evolution of the ROC curve when increasing α from 0.0001 to 100.

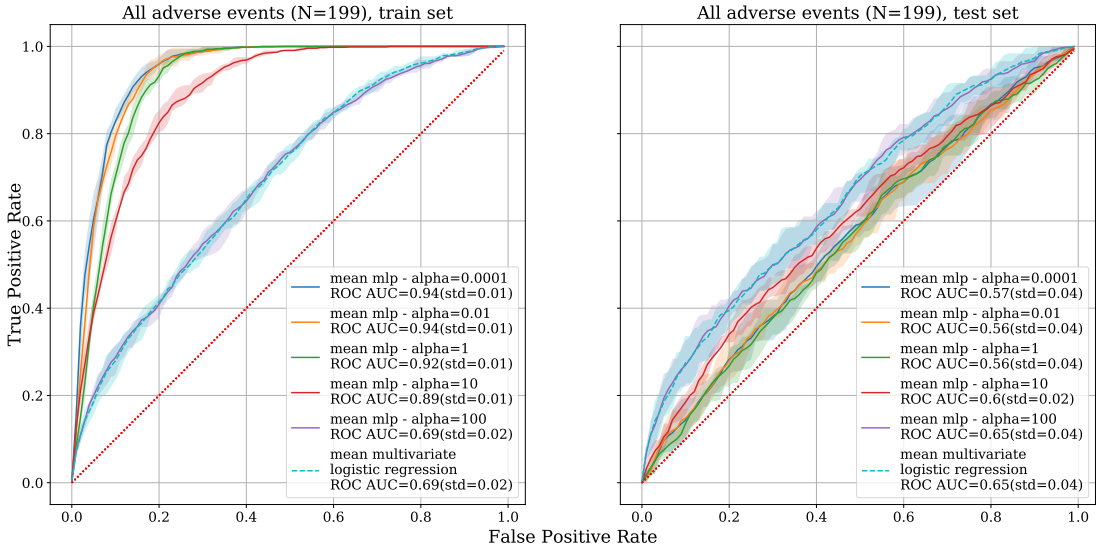

Figure 3.17 – ROC curves of MLP with 10 hidden neurons, regularization α varying from 0.0001 to 100, all patients ($N = 6485$), all adverse events ($N = 199$). On the left plot, the train ROC curves are still very high, except for $\alpha = 100$, where the performance is exactly the same as multivariate logistic regression (purple curve and blue dashed curve). On the right plot, all test ROC curves are still lower than multivariate logistic regression curve, except for $\alpha = 100$, which is again equal to multivariate logistic regression curve.

We observe that for $\alpha < 10$, regularization had a very limited effect on overfitting. The train curves were still much higher than the test curves. With $\alpha = 100$, there was no more overfitting since we had a similar curve for the train and test performance. However, this performance was equivalent to that of logistic regression. The best MLP classifier with the L-BFGS solver and 10 hidden neurons was obtained with $\alpha = 100$ and had an AUC of 0.65 ± 0.04.

Optimizer

Since we did not manage to improve the baseline performance with L-BFGS and regularization, we tried a different optimizer based on stochastic gradient descent (SGD). After some initial grid search to find the hyperparameters yielding the best AUC (not reported here), we chose the following parameters: 500 epochs, batch size of 50, constant learning rate of 0.1. The resulting ROC curves with number hidden neurons varying from 2 to 10 are displayed on Figure 3.18.

Similarly to Figure 3.16, all classifiers were overfitting and the best classifiers in terms of test performance were the ones with a small number of hidden neurons (best AUC= 0.61 ± 0.04). Just like the L-BFGS optimizer, SGD requires a mechanism to prevent overfitting. With SGD,

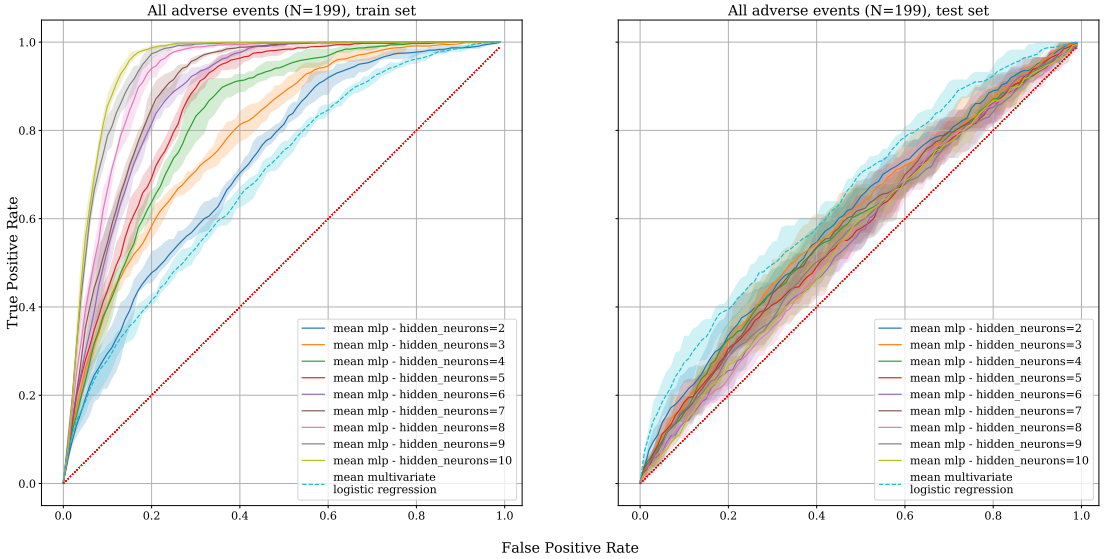

Figure 3.18 – ROC curves of MLP with hidden neurons varying from 2 to 10, SGD optimizer, all patients ($N = 6485$), all adverse events ($N = 199$). The pattern in this figure is very similar to the one in Figure 3.16 (L-BFGS without regularization): all MLP ROC curves are higher than multivariate logistic regression for the train set (left plot) but lower for the test set (right plot)

the strategy of early stopping described in Section 3.4.1 can be used. Figure 3.19 displays performance of the best classifier among the varying number of hidden neurons (AUC= 0.66 ± 0.04, 5 hidden neurons). This was only 0.01 better than our baseline model.

Different protocols

The protocol used for previous analyses followed a 70-30 split between the train and the test set, with 10 random separations for variability. However, other strategies can be used to separate the data, and in particular, one can use k-fold cross-validation (Section 2.2.4). Cross-validation generally results in a less biased estimate of the model skill than other methods, such as a simple train/test split. We used 5-fold and 10-fold cross-validation protocols, to evaluate if those configurations yielded a better performance of MLP compared to logistic regression. We evaluated to best MLP classifier found so far, SGD with 5 hidden neurons and early stopping. Figure 3.20 displays ROC curves of logistic and MLP classifier for the 5-fold validation protocol, and Figure 3.21 displays the same ROC curves for the 10-fold validation protocol. In both cases, the performance of the logistic regression classifier was stable (AUC= 0.66) although there was more variability in the 10-fold validation scheme, since there were 10 folds instead of 5. The performance of the MLP classifier also did not change much between the 5-fold and 10-fold protocol, although slightly superior than logistic regression in the 10-fold (AUC= 0.67 ± 0.07).

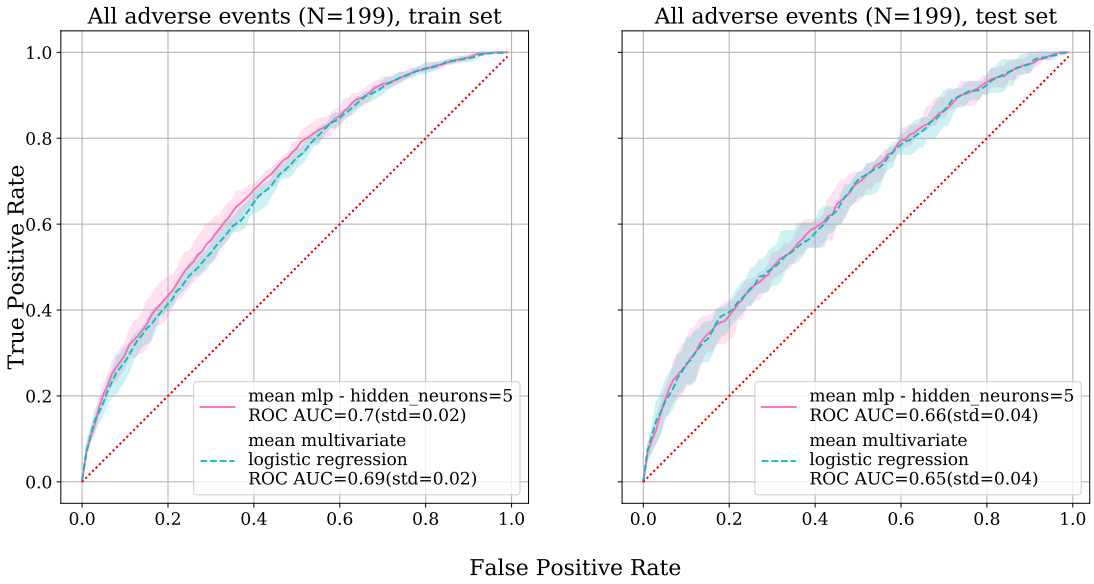

Figure 3.19 – ROC curve of MLP with 5 hidden neurons, SGD optimizer and early stopping, all patients ($N = 6485$), all adverse events ($N = 199$). With early stopping , both the train and test ROC curves of the MLP are very similar to the train and test ROC curves of the multivariate logistic regression classifier, respectively.

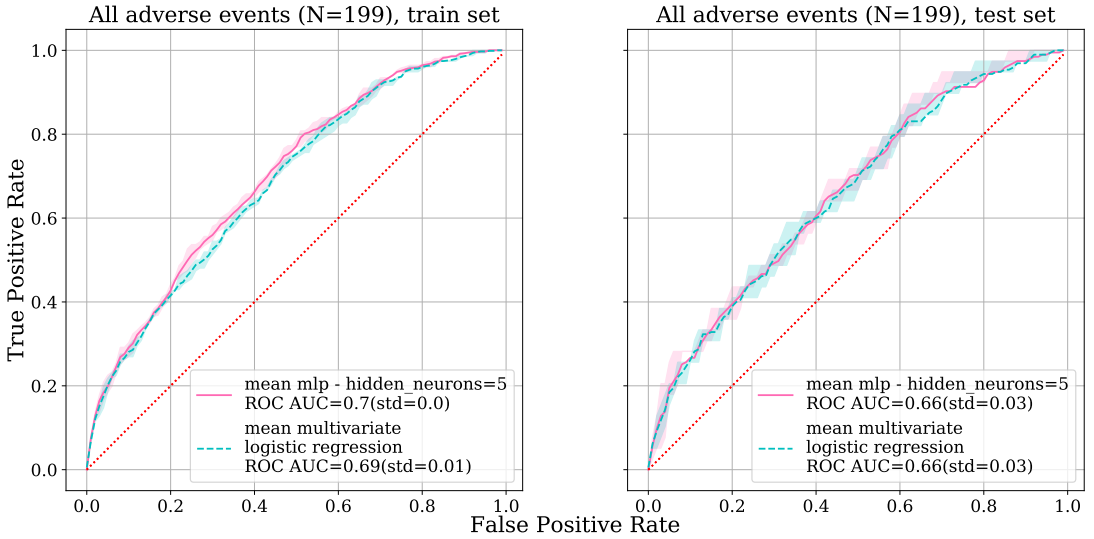

Figure 3.20 – ROC curve of MLP with 5 hidden neurons, SGD optimizer and early stopping, 5-fold cross-validation, all patients ($N = 6485$), all adverse events ($N = 199$). With 5-fold cross-validation protocol, both multivariate logistic regression and MLP had a test AUC of 0.66 ± 0.03

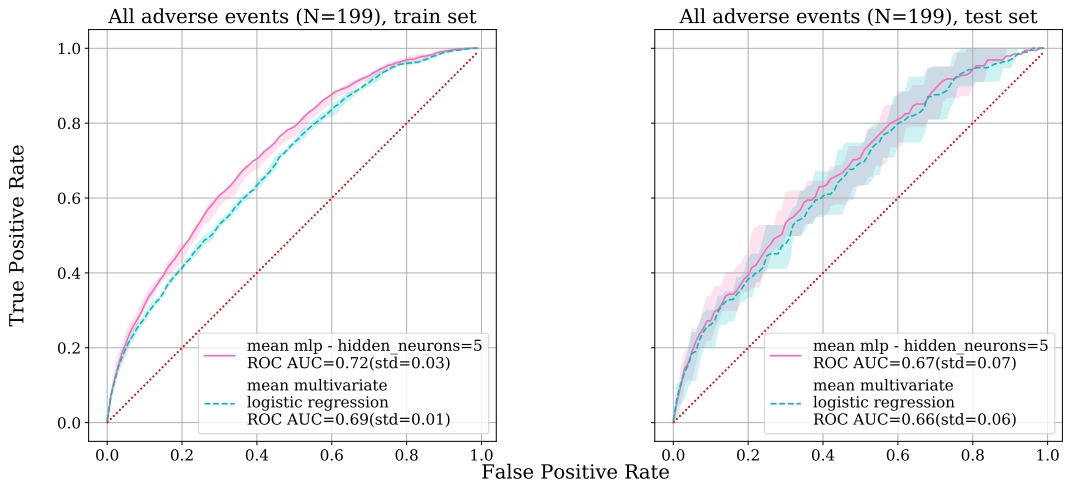

Figure 3.21 – ROC curve of MLP with 5 hidden neurons, SGD optimizer and early stopping, 10-fold cross-validation, all patients ($N = 6485$), all adverse events ($N = 199$). With 10-fold cross validation, the MLP had both train and test ROC curves slightly higher than those of the multivariate logistic regression classifier

The best MLP classifier consisted in applying a 5-hidden neurons MLP optimized with mini-batch SGD and early stopping to the dataset using a 10-fold cross-validation protocol, which resulted in a AUC ROC= 0.67 ± 0.07.

3.4.2 SVM

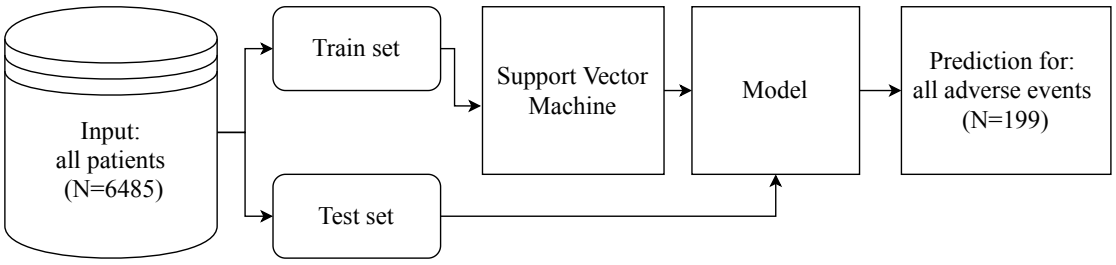

Figure 3.22 – Block diagram of SVM system with a train/test split to predict any adverse event from any patient. The input data is split into a train and a test set. The train set is used to train the SVM while the test set is used to evaluate the performance of the model.

In this final experiment, we evaluated another non-linear model based on SVM to improve the performance of our predictive model of adverse events and see whether the failure to improve predictive power is linked to MLP or to the non-linearity. We used the RBF kernel and performed a grid search to find the best combination of regularization parameter C and kernel coefficient γ. Table 3.3 displays the grid search results using ROC AUC as a scoring function and a 5-fold cross-validation protocol, when varying hyperparameters C and γ. The best

combination of hyperparameters from this table, $C = 1$ and $\gamma = 0.001$, yielded a performance equivalent to that of logistic regression (AUC= 0.66 ± 0.05). For this combination, we also examined the number of support vectors belonging to each class. We obtained 4140 ± 35 support vectors from the negatives and 135 ± 1 support vectors from the positive class. This represents respectively 84% of the negatives and 87% of the positives.

Table 3.3 – Grid search results for SVM classifier, 5-fold cross-validation. When C increases, the resulting AUC does not change much for a given γ. However, for a given value of C, increasing γ decreases the resulting AUC. The best AUC is observed with the lowest C and the lowest γ.

	$C = 1$	$C = 5$	$C = 10$	$C = 50$	$C = 100$
$\gamma = 0.001$	0.66 ± 0.05	0.65 ± 0.05	0.65 ± 0.04	0.65 ± 0.04	0.65 ± 0.04
$\gamma = 0.01$	0.65 ± 0.04	0.63 ± 0.03	0.62 ± 0.03	0.60 ± 0.03	0.59 ± 0.04
$\gamma = 0.1$	0.58 ± 0.03	0.57 ± 0.03	0.57 ± 0.03	0.57 ± 0.04	0.57 ± 0.04
$\gamma = 1$	0.57 ± 0.04	0.58 ± 0.05	0.57 ± 0.06	0.57 ± 0.07	0.57 ± 0.06
$\gamma = 10$	0.56 ± 0.03	0.56 ± 0.03	0.56 ± 0.03	0.56 ± 0.03	0.56 ± 0.03

To wrap up the experiments, Figure 3.23 displays the ROC curves of the best classifier obtained for each method (logistic regression, MLP, SVM) for the prediction of adverse event for any patient, using a split between train and test set of 70/30.

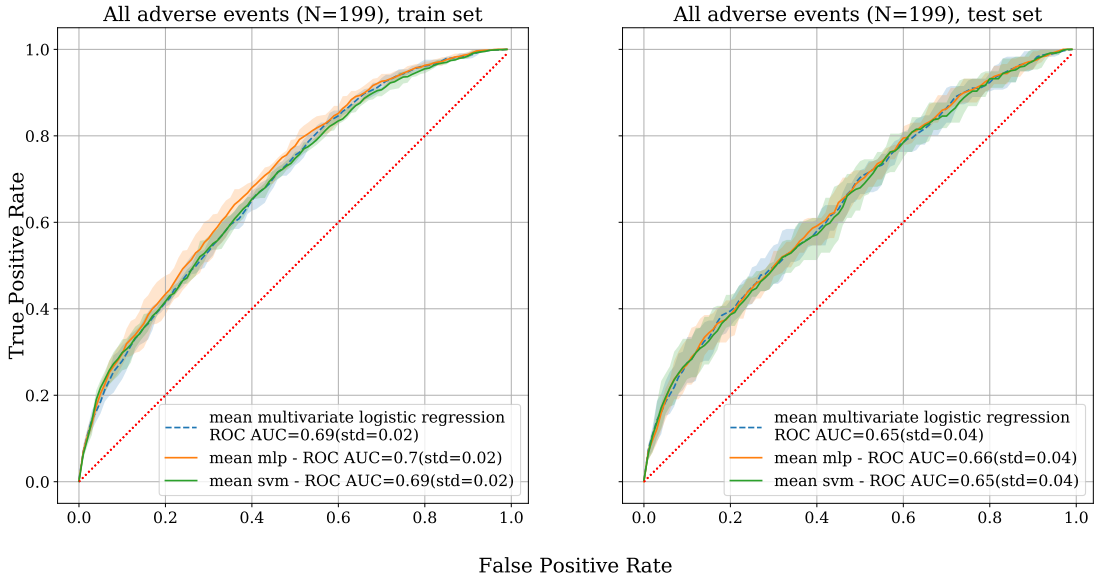

Figure 3.23 – ROC curves of best classifiers for multivariate logistic regression, MLP and SVM, all patients ($N = 6485$), all adverse events ($N = 199$). The MLP classifier has a slightly higher ROC curve than SVM and multivariate logistic regression, both for the train and test sets, but the difference in AUC is very small.

4 Analysis

In this study, we explored the predictive capabilities of patient clinical data regarding the occurrence of adverse events during LTBI adverse events, using linear and non-linear models. We first reproduced the logistic model from Campbell et al. [1] and found that it had some predictive capability when using multivariate logistic regression, weighted likelihood as a correction for rare events and all available clinical variables, whether categorical or continuous. Secondly, we generalized the multivariate logistic regression model by training and evaluating it on separate subsets of data. This resulted in a baseline model with a test AUC of 0.65 ± 0.04, corresponding the best classification performance using a linear classifier. Finally, we tried to improve the predictive power of the model by exploring two non-linear machine learning methods, MLP and SVM. MLP was investigated using L-BFGS optimizer with regularization, and mini-batch SGD with early stopping, but didn't show any improvement on the predictive performance. SVM with RBF kernel was explored with a grid search varying regularization and kernel coefficient, and also had similar performance than multivariate logistic regression. To sum up, we found that multivariate logistic regression, MLP and SVM have similar performances for our predictive task.

In this section, we discuss the predictive capabilities of the clinical dataset and then focus on the main contribution of this work: to build a predictive model of adverse events. We then analyze why non-linear models were not able to improve the predictive performance and introduce other possible strategies.

4.1 Patient clinical data is correlated with risk of adverse events during LTBI treatment

The study from Campbell et al. [1] shows that the presence or absence of some risk factors influences the risk of adverse events during LTBI treatment. In Section 3.1.1, the different

tables reported OR estimates for the different clinical variables in the dataset. These estimates can easily be interpreted and correspond to how much the exposure to a given factor increases the risk of adverse events, compared to when there is no exposure to this risk factor. Depending on the outcome being evaluated, different risk factors proved to be important predictors of adverse events. This can be explained by the fact that adverse events have different degrees of severity and belong to different types, and thus are not related to the same combination of risk factors. This will be important when considering the predictive model of all adverse events. More generally, the article confirms that there is a correlation between patient clinical data and the occurrence of adverse events.

However, the authors of the study did not investigate the predictive capabilities of clinical data *per se*. In Campbell et al. [1], there was no measure of the performance of the logistic regression models, when trying to classify patients positive or negative for the outcome. Logistic regression models introduced in that work were numerous, with different possible implementations and there was a need to evaluate and select the best configuration by comparing the different configurations in terms of predictive power on the whole dataset.

Moreover, there is a clear imbalance in the dataset between the number of patients who developed an adverse event (so-called positives) and the number of patients who had no adverse event during the treatment (so-called negatives). This imbalance between the two classes needs to be corrected with some mechanism in order to underline the importance of classifying correctly the positives, which are considered as rare in the dataset.

Based on the comparison between Firth and weighted likelihood for the correction of rare events (Figures 3.3, 3.4, 3.5 and 3.6), we decided to keep weighted implementation of the log-likelihood, since it yielded similar, if not better, classifiers and allowed to perform computations more efficiently. From this comparison, it also appeared that the multivariate model gave better performance (best AUC= 0.69) than each individual univariate model (AUC$\leq$ 0.63). The fact that multivariate classifiers resulted in much better performance than univariate classifiers can easily be explained by the fact that using more information makes it easier to separate patients with adverse event from patients without adverse event. The prediction of health outcomes from available data is often based on the belief that there exists a small number of important risk factors and that careful selection of those variables is the key to successful performance of the models for outcome prediction [13]. However, each of the variables may contribute in its own way to the final outcome and thus should not be eliminated from predictive models.

The second comparison evaluated the difference between pre-selecting variables from univariate analysis to incorporate into multivariate analysis, and using all available variables as covariates regardless of their individual significance in univariate analysis (Figures 3.7 and 3.8).

Taking all clinical variables into account yielded a slightly better performance (AUC= 0.64 > AUC= 0.6 and AUC= 0.7 > AUC= 0.69) for logistic regression and is also more relevant when using non-linear machine learning models, which can potentially draw more sophisticated separation hyperplanes. Since we present the same data to both models and compare their performance, we need to present the same features to both systems for a fair comparison. Therefore, our follow-up experiments use all clinical variables as covariates in the multivariate logistic regression model. Using all available clinical variables instead of selecting only significant ones for multivariate analysis, is a way to reduce model complexity, but it may cause a loss in flexibility. Any characteristic can contribute to the overall picture of an individual's health, which explains the better performance of the approach that predicts the occurrence of adverse events in terms of as many variables as possible.

Finally, using categorical variables allows clinicians to clearly distinguish groups of high-risk patients and makes it easier to interpret the logistic regression estimates. However, using continuous variables might be more useful in the context of a predictive model, since one loses information when discretizing a variable. In our experiments (Figures 3.9 and 3.10), using continuous variables resulted in a better performance than using only categorical variables (AUC= 0.67 > AUC= 0.64 and AUC= 0.73 > AUC= 0.7). The motivation to use continuous variables also comes from the fact that using categorical variables introduces an additional operation on the data which may add bias. Therefore, we chose to keep the mixed-variable multivariate logistic regression model for further experiments.

To sum up the results from these comparisons, the best logistic regression classifier for the primary and secondary outcomes had the following characteristics: weighted likelihood, multivariate logistic regression, all available clinical variables, mix of categorical and continuous variables, and resulted in the ROC displayed in Figure 4.1.

Reproducing the work of Campbell et al. [1] was also an achievement in terms of reproducible research, demonstrating that using the same dataset and algorithms could result in very similar OR estimates.

4.2 Building a predictive model

Building a predictive model of adverse events means that given a patient's clinical data, the model predicts whether the patient will develop an adverse event during the treatment. This goal differs from Campbell et al. [1], which mainly evaluated categories of high-risk patients given their treatment and for specific outcomes of adverse events.

To build our general predictive model, the first step was to extend the outcome to all adverse events, and to include patients from both treatments (Figure 3.11). The comparisons

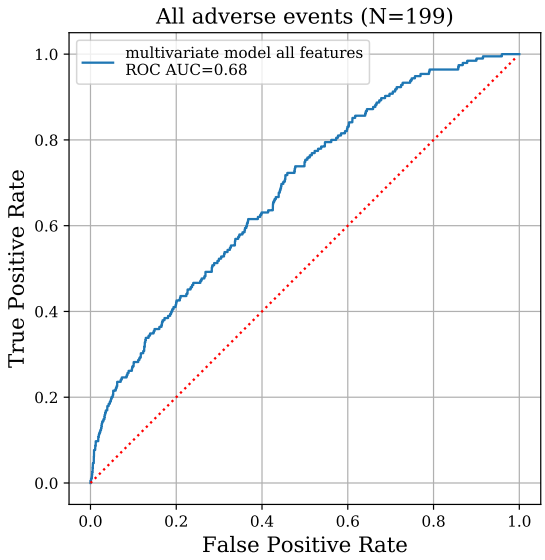

Figure 4.1 – ROC curve of multivariate logistic regression, all patients ($N = 6485$), all adverse events ($N = 199$).

made in the previous experiment (firth vs weighted, univariate vs multivariate, categorical vs continuous), to select the best implementation of logistic regression, were repeated to verify that our conclusions held when evaluating all adverse events for patients regardless of their treatment. Looking at Figure 3.12, the multivariate models were again much better than the best univariate model (~0.1 AUC difference). However, there was not much difference between firth and weighted implementation (~0.01 AUC difference) and the same was true for using either categorical or continuous variables. The three multivariate models on this figure had a very similar performance so any one of them could be used. However, as explained before, using weighted likelihood to correct for rare events is computationally less complex. Furthermore, using continuous covariates (when possible) rather than categorical ones may reduce bias although the resulting AUC is about 0.01 smaller.

Secondly, the selection of the best logistic regression classifier was done by training and evaluating the model on the same dataset, which resulted in a large bias. There is no guarantee that the model will predict adverse events efficiently when presented with unseen data. Introducing a separation of the dataset in a train and a test set with no patient in common tends to reduce that bias. With this configuration, the performance of the multivariate logistic regression classifier on the test set constituted a new baseline benchmark to improve upon. Since the model was presented to unseen data when evaluated, the test performance (AUC= 0.65 ± 0.04) was slightly lower than train performance (AUC= 0.69 ± 0.02) and exhibited higher variability between the different train/test splits, as can be seen on Figure 3.14. Both train and test performances were quite poor (AUC< 0.7), indicating that increasing the model

complexity with non-linear models could possibly improve the performance.

4.3 Linear and non-linear models resulted in similar predictive power

Two non-linear machine models, MLP and SVM, were investigated to improve the predictive power of the baseline model. MLP was explored with two different optimizers: L-BFGS and SGD. In both cases, there was clear evidence of overfitting because the train ROC curves were far superior to the test ROC curves (Figures 3.16 and 3.18). Overfitting occurs when the model captures too closely the training data statistics, and starts to model the noise of the data rather than its structure. Overfitting can be reduced with the following (among others) regularization techniques: explicit regularization (applied to L-BFGS) and early stopping (applied to SGD). In the case of L-BFGS, increasing the strength of regularization managed to reduce overfitting, and to obtain the same performance than multivariate logistic regression (AUC= 0.65 ± 0.04) with 10 hidden neurons and regularization parameter α set to 100, as can be seen on Figure 3.17. The fact that the resulting train ROC curve was close to the best multivariate logistic regression system suggests that further increasing α would only result in a worse performance on the train set, and thus on the test set as well. In the case of SGD and early-stopping, the best MLP classifier had 5 hidden neurons and early stopping managed again to reduce overfitting and reach an almost equal performance than logistic regression (AUC= 0.66 ± 0.04), as can be seen on Figure 3.19. However neither optimizer showed any significant improvement on the predictive performance. SVM with an RBF kernel performed best with $C = 1$ and $\gamma = 0.001$ after grid search (Table 3.3), but also had similar performance than multivariate logistic regression on the test set (AUC= 0.65 ± 0.04).

The non-linear machine learning models yielded similar performance compared to logistic regression (Figure 4.2). If non-linear methods cannot improve the performance of our model, it suggests that there is a part of the patient population that can be classified with a linear separator, and another part which corresponds to possibly overlapping isolated cases in a multi-dimensional space. These are very specific cases, from which we cannot learn. We cannot generalize from those specific points which correspond to combinations of features specific to theses patients. Further analysis of misclassified patients might help to understand those specific cases, in collaboration with clinicians to explain those particular combinations.

4.3.1 Overlap between classes

One way to visualize the separability of the data is to perform a t-Distributed Stochastic Neighbor Embedding (t-SNE) analysis. t-SNE is a dimensionality reduction technique used to represent high-dimensional dataset in a low-dimensional space of 2 or 3 dimensions so that we can visualize it. In contrast to other dimensionality reduction algorithms, t-SNE creates a

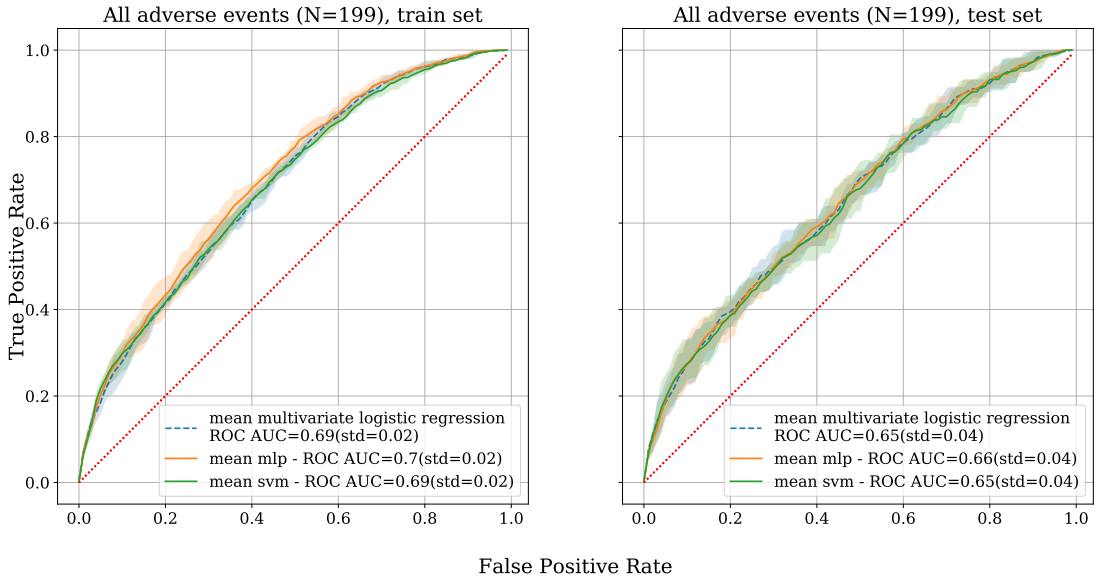

Figure 4.2 – ROC curves of best classifiers for multivariate logistic regression, MLP and SVM, all patients (N = 6485), all adverse events (N = 199). The MLP classifier has a slightly higher ROC curve than SVM and multivariate logistic regression, both for the train and test sets, but the difference in AUC is very small.

reduced feature space where similar samples are modeled by nearby points and dissimilar samples are modeled by distant points with high probability [38]. This means that t-SNE preserves the structure of the original dataset and the 2D visualization allows us to look for patterns in the dataset and in particular, to apprehend the degree of data separability. Figures 4.3, 4.4 and 4.5 represent the t-SNE embeddings with 2 components on the whole dataset, the train set (70% of dataset) and test set (30% of dataset), respectively.

On these plots, we can see that positives and negatives are completely superposed, and we fail to see any linear separation between classes. All positives seem to overlap negatives clusters. Even on the whole dataset (Figure 4.3), there is no clear evidence of linear separability and isolated cases. This does not confirm the hypothesis that there is a part of the population that can be linearly classified, while some isolated cases fail to follow the trend. However, this way of verifying the hypothesis leads to a premature rejection of our hypothesis because this is a 2D representation of a 11-dimensional space (11 features in the dataset) which may not be readily discriminative. Therefore, we might miss something in this representation. The t-SNE analysis provides a possible explanation as to why most classifiers have quite low predictive performance (AUC$\leq$ 0.7): the data of patients with and without adverse events seems to be completely overlapped. However, this does not explain why the classifiers still perform better than random guess.

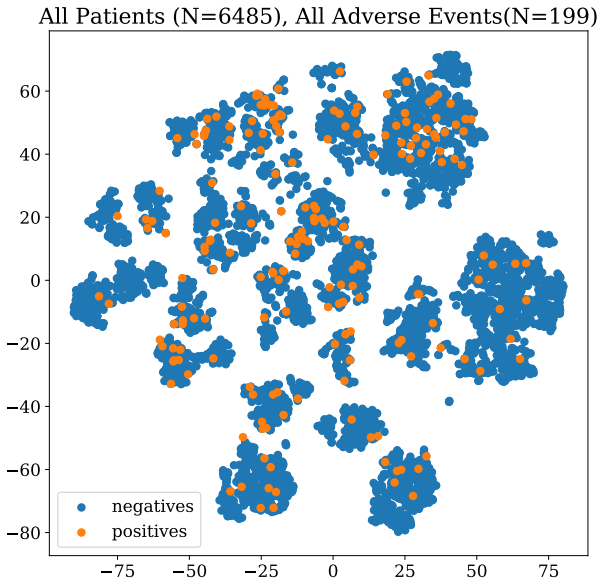

Figure 4.3 – t-SNE with 2 components on the whole dataset. The negatives (patients without adverse events) form several clusters and positives (patients with adverse events) are superposed on these clusters. This plot also allows to visualize the imbalance in the dataset between positives and negatives.

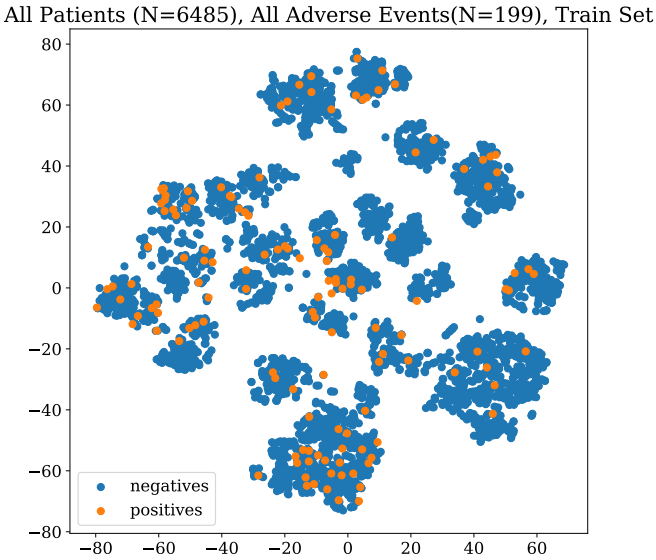

Figure 4.4 – t-SNE with 2 components on the train set (70% of dataset). Negatives and positives are overlapped in clusters

This overlap between positives and negatives is consistent with the number of support vectors from SVM results. We obtained a very high number of support vectors, representing 80% of the points in the dataset. This means that most points are within the margin of separation between positives and negatives and are very close to each other. A high number support

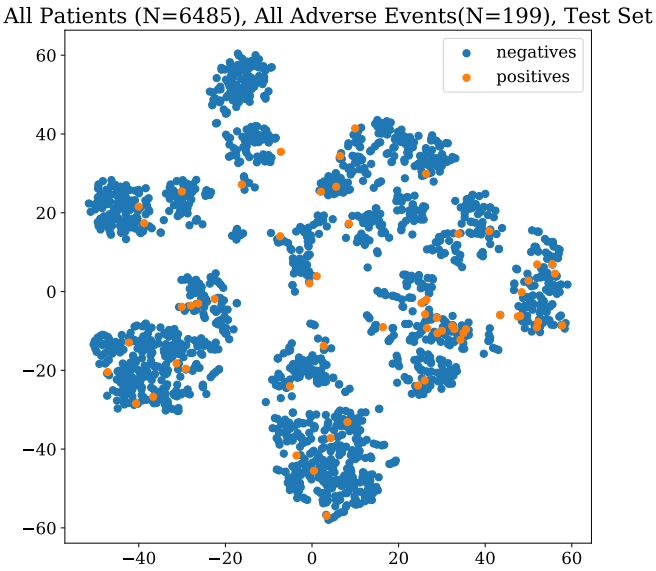

Figure 4.5 – t-SNE with 2 components on the test set (30% of dataset). Negatives and positives are overlapped in clusters

vectors usually indicates overfitting, however in this case, the train performance is only slightly superior to the test performance. Therefore, we could rather interpret this high number of support vectors as an inherent problem in the data: the positives and negatives are so overlapped that it makes it difficult to separate them, even in a non-linear way.

4.3.2 MLP embedding

This section investigates the data separability in MLP with 2 hidden neurons, to visualize how MLP performs better than a random classifier. We plot the output of the 2 hidden neurons (before last layer), which projects the input (11-dimensional) in a 2D space to perform a linear separation. The 2D space can be visualized and represents the input to the last neuron. The points are computed by feeding train and test data to MLP equation 4.1 using the values of trained weights and intercepts:

$$f(w^T x + b) \text{ with } f(x) = \frac{1}{1+e^{-x}} \tag{4.1}$$

We also plotted the linear decision boundary at the last neuron to understand how the separation is performed. To do this, we solved the following equation:

$$w_1 x_1 + w_2 x_2 + b = t \Leftrightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{b-t}{w_2} \tag{4.2}$$

where w_1, w_2 correspond to the weights assigned to the 2 hidden neurons, b to the bias

term, and t to the best separation threshold. This threshold is evaluated on ROC plot and corresponds to a trade-off between a low FPR and a high TPR. The resulting plots are displayed on Figure 4.6 for the train set and Figure 4.7 for the test set.

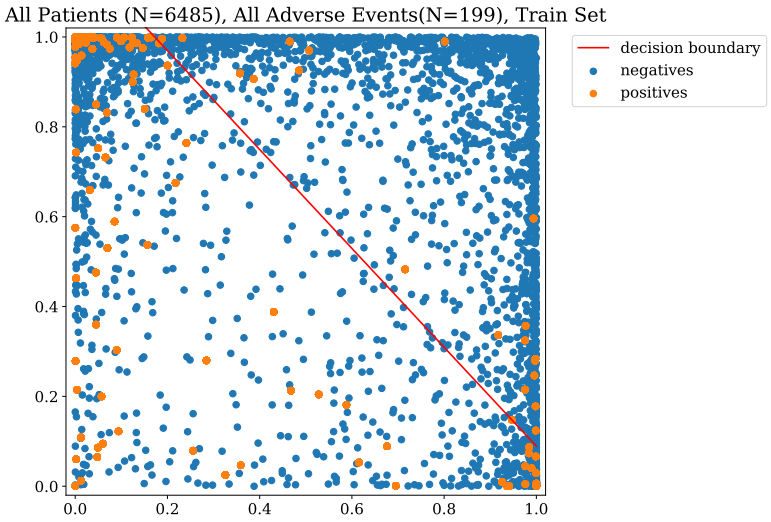

Figure 4.6 – MLP embedding of train set for 2 hidden neurons. The negatives fill the 2D space and accumulate at the upper and right border of the plot. The positives accumulate at two corners in the plot: upper left and lower right. The MLP classifier is bound to make a lot of false negatives since the negatives are everywhere and cannot be separated by a line. The classification of positives is rather good since the decision boundary encompasses both the upper left and lower right corners, and there are only a few isolated cases on the other side.

On these plots, we can see that MLP manages to project the input in a 2D space where positives form clusters together. These clusters still overlap the negatives but MLP does better than a random classifier by maximizing the number of positives on one side of the boundary, despite the negatives present on that side.

To sum up, the different visualizations (t-SNE and MLP embeddings) show that positives and negatives are largely overlapped and suggest that an efficient classification of patients with and without adverse events is difficult to achieve with this dataset.

4.4 Limitations and Outlook

Our predictive task consisted in a binary classification task: either the adverse event occurs or it does not. However, all adverse events have different characteristics and maybe different properties. A multiclass classifier might have a better performance since specific features might be associated with specific type/grade of adverse events.

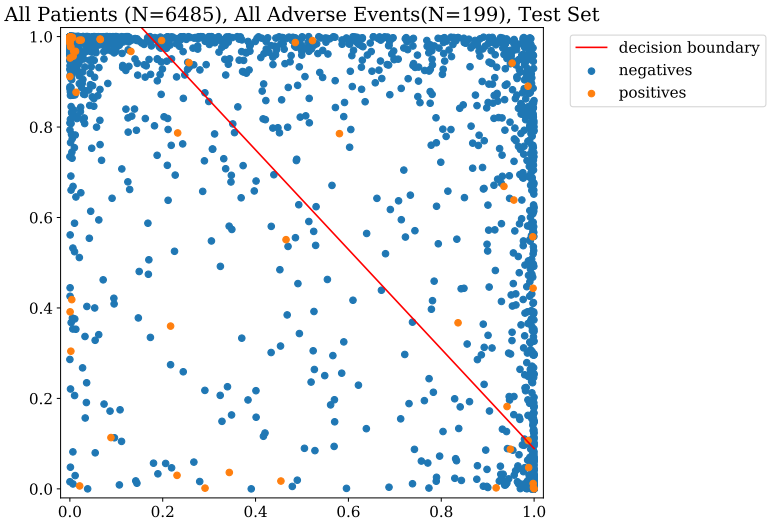

Figure 4.7 – MLP embedding of test set for 2 hidden neurons. The distribution of negatives and positives follows quite closely the pattern from the train set (Figure 4.6): negatives at the borders, positives in the corners. The points are sparser than the train set since the test set is smaller. The linear decision boundary includes the corner with the most positives but there are still a lot of negatives on that side.

Another limitation of this work may be linked to the size and imbalance of the dataset. Being consistently successful at predicting adverse events requires sufficient sample size and this might not be the case here. Small datasets exhibit more variation and may be subject to more noise, which makes it harder to detect patterns linked to occurrence of adverse events. Using a larger dataset, by continuing the data collection on other clinical trials with LTBI treatment, could help get better results.

One could also imagine using additional clinical features, which were also collected during the clinical trials. Although not used in the original study by Campbell et al. [1], those variables might have an importance in a non-linear model. This would require collaboration with clinicians to determine which variables might have predictive power in this context.

Finally, using other libraries with more freedom on the implementation side could have an impact on the results. For example, `scikit-learn` MLP classifier does not have a mechanism to adjust class weights to compensate for the small number of positives, which is why we resorted to resampling. Using libraries such as `Pytorch` or `Keras` allows more control on the training phase.

5 Conclusion

In this project, we managed to establish a correlation between patient clinical data and adverse events, based on the work of Campbell et al. [1]. Then we built a baseline predictive model of adverse events using multivariate logistic regression and an unbiased evaluation protocol. Finally we implemented two non-linear machine learning models which had similar performance than the baseline model. The whole project was conducted with reproducibility in mind. Two bob packages were implemented: a database interface and a package to train and evaluate the different models in order to obtain the same figures of merit for each model.

The failure to improve the predictive power of the model might be inherently due to the dataset, which is highly imbalanced with very few adverse events, and whose clinical data of patients with adverse events appear not to stand out compared to patients without adverse events. Strategies such as increasing the size of the dataset or changing implementations might result in better performances. In the continuity of this work, one could investigate the performance of these classifiers when predicting specific type of adverse events or adverse events with specific grades in a multiclass classification task.

A An appendix

Table A.1 – Results of univariate and multivariate model of risk factors for grade 1-2 rash + all grade 3-5 adverse events attributed to rifampin. This table reproduces the left part of table 4 in Campbell et al. [1]

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Age	18-34	1489	18	1 (ref)	1 (ref)
	35-64	1661	28	1.4	1.1
	65-90	130	4	2.8	1.6(1.7)
Sex	Female	1916	34	1 (ref)	-
	Male	1364	16	0.7	-
BMI	Normal	1674	24	1 (ref)	-
	Underweight	216	3	1.1	-
	Overweight	916	16	1.2	-
	Obese	474	7	1.1	-
Immune Status	No Immune suppr.	2929	42	1 (ref)	1 (ref)
	HIV-positive	130	1	0.8	0.5
	Other immune suppr.	221	7	2.4	1.3
Alcohol Use	Never drinks	2200	31	1 (ref)	-
	≤ 1 drink per week	873	17	1.4	-
	> 1 drink per week	207	2	0.8	-
Smoking history	Has never smoked	2496	42	1 (ref)	-
	Currently or has smoked	784	8	0.6	-
Medication Consistency	Consistency ≥ 90%	2440	30	1 (ref)	1 (ref)
	Consistency < 90%	840	20	2.0	2.0
Concomitant medications	None	2157	27	1 (ref)	1 (ref)
	Any	763	23	2.9	2.8

Table A.2 – Results of univariate and multivariate model of risk factors for grade 1-4 rash adverse events attributed to rifampin. This table reproduces the right part of table 4 in Campbell et al. [1]

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Age	18-34	1489	6	1 (ref)	1 (ref)
	35-64	1661	15	2.1(2.2)	1.6
	65-90	130	4	8.1	4.4
Sex	Female	1916	18	1 (ref)	-
	Male	1364	7	0.6	-
BMI	Normal	1674	14	1 (ref)	-
	Underweight	216	0	0.3	-
	Overweight	916	9	1.2	-
	Obese	474	2	0.6	-
Immune Status	No Immune suppr.	2929	21	1 (ref)	1 (ref)
	HIV-positive	130	0	0.5	0.3
	Other immune suppr.	221	4	2.8	1.2
Alcohol Use	Never drinks	2200	17	1 (ref)	-
	≤ 1 drink per week	873	7	1.1	-
	> 1 drink per week	207	1	0.9	-
Smoking history	Has never smoked	2496	21	1 (ref)	-
	Currently or has smoked	784	4	0.7	-
Medication Consistency	Consistency ≥ 90%	2440	16	1 (ref)	-
	Consistency < 90%	840	9	1.7	-
Concomitant medications	None	2157	12	1 (ref)	1 (ref)
	Any	763	13	3.6	2.9

Table A.3 – Results of univariate and multivariate model of risk factors for grade 3-4 hepatotoxicity adverse events attributed to rifampin (N=11 events).. This table reproduces table 7 in Campbell et al. [1] (SI). Multivariate models not created due to no significant covariates

		Number	Risk N	Univariate OR Estimate
Age	18-34	1489	4	1 (ref)
	35-64	1661	7	1.5
	65-90	130	0	1.3
Sex	Female	1916	7	1 (ref)
	Male	1364	4	0.8
BMI	Normal	1674	7	1 (ref)
	Underweight	216	0	0.5
	Overweight	916	3	0.9
	Obese	474	1	0.7
Immune Status	No Immune suppr.	2929	9	1 (ref)
	HIV-positive	130	0	1.2
	Other immune suppr.	221	2	3.5
Alcohol Use	Never drinks	2200	8	1 (ref)
	≤ 1 drink per week	873	2	0.7
	> 1 drink per week	207	1	1.9
Smoking history	Has never smoked	2496	9	1 (ref)
	Currently or has smoked	784	2	0.8
Medication Consistency	Consistency ≥ 90%	2440	7	1 (ref)
	Consistency < 90%	840	4	1.7(1.8)
Concomitant medications	None	2157	8	1 (ref)
	Any	763	3	1.4
Pre-treatment ALT	Normal	3060	10	1.0 (ref)
	Above normal	184	1	2.4

Table A.4 – Results of univariate and multivariate model of risk factors for grade 1-4 rash attributed to isoniazid (N=13 events). This table reproduces table 8 in Campbell et al. [1] (SI)

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Age	18-34	1436	4	1.0 (ref)	1.0 (ref)
	35-64	1642	9	1.9	2.1
	65-90	127	0	1.2(1.3)	1.3
Sex	Female	1811	9	1.0 (ref)	
	Male	1394	4	0.6	-
BMI	Normal	1646	6	1.0 (ref)	
	Underweight	222	0	0.6	-
	Overweight	907	5	1.5	-
	Obese	430	2	1.5	-
Immune Status	No Immune suppr.	2871	13	1.0 (ref)	-
	HIV-positive	138	0	0.8	-
	Other immune suppr.	196	0	0.5	-
Alcohol Use	Never drinks	2112	10	1.0 (ref)	-
	≤ 1 drink per week	891	3	0.8	-
	> 1 drink per week	202	0	0.5	-
Smoking history	Has never smoked	2421	11	1.0 (ref)	-
	Currently or has smoked	784	2	0.7	-
Medication Consistency	Consistency ≥ 90%	2151	5	1.0 (ref)	1(ref)
	Consistency < 90%	1054	8	3.2	3.4
Concomitant medications	None	2470	10	1.0 (ref)	-
	Any	735	3	1.1	-

Table A.5 – Results of univariate and multivariate model of risk factors for grade 3-4 hematologic adverse events attributed to rifampin (N=6 events). This table reproduces table 9 in Campbell et al. [1] (SI). Multivariate analysis was not performed due to the scarcity of events.

		Number	Risk N	Univariate OR Estimate
Age	18-34	1489	4	1 (ref)
	35-64	1661	2	0.5
	65-90	130	0	1.3
Sex	Female	1916	4	1 (ref)
	Male	1364	2	0.8
BMI	Normal	1674	2	1 (ref)
	Underweight	216	2	7.8
	Overweight	916	1	1.1
	Obese	474	1	2.1
Immune Status	No Immune suppr.	2929	6	1 (ref)
	HIV-positive	130	0	1.7
	Other immune suppr.	221	0	1.0
Alcohol Use	Never drinks	2200	3	1 (ref)
	≤ 1 drink per week	873	3	2.5
	> 1 drink per week	207	0	1.5
Smoking history	Has never smoked	2496	6	1 (ref)
	Currently or has smoked	784	0	0.2
Medication Consistency	Consistency ≥ 90%	2440	3	1 (ref)
	Consistency < 90%	840	3	2.9
Concomitant medications	None	2157	4	1 (ref)
	Any	763	2	1.8
Pre-treatment WBC	Normal	2972	4	1.0 (ref)
	Below normal	196	2	3.6
Pre-treatment Platelets	Normal	2972	5	1.0 (ref)
	Below normal	196	1	5.8

Table A.6 – Results of univariate and multivariate model of risk factors for grade 3-4 non-rash and non-hepatotoxic adverse events attributed to isoniazid (N=8 events). This table reproduces table 12 in Campbell et al. [1] (SI). Multivariate analysis was not performed due to the scarcity of events.

		Number	Risk N	Univariate OR Estimate
Age	18-34	1436	6	1.0 (ref)
	35-64	1642	2	0.3
	65-90	127	0	0.9
Sex	Female	1811	6	1.0 (ref)
	Male	1394	2	0.5
BMI	Normal	1646	4	1.0 (ref)
	Underweight	222	1	2.5
	Overweight	907	2	1.0
	Obese	430	1	1.3
Immune Status	No Immune suppr.	2871	8	1.0 (ref)
	HIV-positive	138	0	1.2
	Other immune suppr.	196	0	0.9
Alcohol Use	Never drinks	2112	8	1.0 (ref)
	≤ 1 drink per week	891	0	0.1
	> 1 drink per week	202	0	0.6
Smoking history	Has never smoked	2421	7	1.0 (ref)
	Currently or has smoked	784	1	0.6
Medication Consistency	Consistency ≥ 90%	2151	4	1.0 (ref)
	Consistency < 90%	1054	4	2.0
Concomitant medications	None	2470	6	1.0 (ref)
	Any	735	2	1.3

Table A.7 – Results of univariate and multivariate model of risk factors for grade 3-4 non-rash and non-hepatotoxic adverse events attributed to rifampin (N=14 events). This table reproduces table 13 in Campbell et al. [1]

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Age	18-34	1489	8	1 (ref)	1 (ref)
	35-64	1661	6	0.7	0.5
	65-90	130	0	0.7	0.4
Sex	Female	1916	9	1 (ref)	-
	Male	1364	5	0.8	-
BMI	Normal	1674	3	1 (ref)	1 (ref)
	Underweight	216	3	7.8	9.5
	Overweight	916	4	2.4	2.4
	Obese	474	4	4.6	3.6
Immune Status	No Immune suppr.	2929	12	1 (ref)	-
	HIV-positive	130	1	2.7	-
	Other immune suppr.	221	1	1.6	-
Alcohol Use	Never drinks	2200	6	1 (ref)	1 (ref)
	≤ 1 drink per week	873	8	3.3	3.0
	> 1 drink per week	207	0	0.8	0.7
Smoking history	Has never smoked	2496	7	1 (ref)	-
	Currently or has smoked	784	7	0.6	-
Medication Consistency	Consistency ≥ 90%	2440	7	1 (ref)	1 (ref)
	Consistency < 90%	840	7	2.9	2.9
Concomitant medications	None	2157	7	1 (ref)	1 (ref)
	Any	763	7	3.3	4.8

Table A.8 – Results of univariate and multivariate model of risk factors for combined outcome of grade 1-2 rash and all grade 3-5 adverse events with study drug as a predictor (N=86 events isoniazid; N=50 events rifampin). This table reproduces table 14 in Campbell et al. [1].

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Treatment Arm	Isoniazid	3205	50	1.0 (ref)	1.0 (ref)
	Rifampin	3280	86	0.6	0.6
Age	18-34	2925	43	1.0 (ref)	1.0 (ref)
	35-64	3303	82	1.7	1.5
	65-90	257	11	3.1	2.3
Sex	Female	3727	82	1.0 (ref)	
	Male	2758	54	0.9	-
BMI	Normal	3320	68	1.0 (ref)	
	Underweight	438	8	0.9	-
	Overweight	1823	42	1.1	-
	Obese	904	18	1.0	-
Immune Status	No Immune suppr.	5800	115	1.0 (ref)	1.0 (ref)
	HIV-positive	268	6	1.2	0.9
	Other immune suppr.	417	15	1.9	1.2
Alcohol Use	Never drinks	4312	89	1.0 (ref)	-
	≤ 1 drink per week	1764	37	1.0	-
	> 1 drink per week	409	10	1.2	-
Smoking history	Has never smoked	4917	102	1.0 (ref)	-
	Currently or has smoked	1568	34	1.1	-
Medication Consistency	Consistency ≥ 90%	4591	87	1.0 (ref)	1.0 (ref)
	Consistency < 90%	1894	49	1.4	1.4
Concomitant medications	None	4987	85	1.0 (ref)	1.0 (ref)
	Any	1498	51	2.0	1.7

Table A.9 – Results of univariate and multivariate model of risk factors for grade 3-4 hepatotoxicity with study drug as a predictor (N=65 events isoniazid; N=11 events rifampin). This table reproduces table 15 in Campbell et al. [1].

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Treatment Arm	Isoniazid	3205	65	1.0 (ref)	1.0 (ref)
	Rifampin	3280	11	0.2	0.2
Age	18-34	2925	19	1.0 (ref)	1.0 (ref)
	35-64	3303	50	2.3	2.1
	65-90	257	7	4.5	4.0
Sex	Female	3727	40	1.0 (ref)	
	Male	2758	36	1.2	-
BMI	Normal	3320	41	1.0 (ref)	
	Underweight	438	4	0.8	-
	Overweight	1823	22	1.0	-
	Obese	904	9	0.8	-
Immune Status	No Immune suppr.	5800	61	1.0 (ref)	1.0 (ref)
	HIV-positive	268	5	1.9(2.0)	1.7
	Other immune suppr.	417	10	2.4	1.8
Alcohol Use	Never drinks	4312	48	1.0 (ref)	1 (ref)
	≤ 1 drink per week	1764	19	1.0	0.9
	> 1 drink per week	409	9	2.1	1.7
Smoking history	Has never smoked	4917	51	1.0 (ref)	1 (ref)
	Currently or has smoked	1568	25	1.6	1.2
Medication Consistency	Consistency ≥ 90%	4591	55	1.0 (ref)	-
	Consistency < 90%	1894	21	0.9	-
Concomitant medications	None	4987	50	1.0 (ref)	1.0 (ref)
	Any	1498	26	1.8	1.1
Pre-treatment ALT	Normal	6032	66	1.0 (ref)	1.0 (ref)
	Above normal	380	10	2.5	2.4

Table A.10 – Results of univariate and multivariate model of risk factors for grade 1-4 rash with study drug as a predictor (N=13 events isoniazid; N=25 events rifampin). This table reproduces table 16 in Campbell et al. [1].

		Number	Risk N	Univariate OR Estimate	Multivariate OR Estimate
Treatment Arm	Isoniazid	3205	13	1.0 (ref)	1.0 (ref)
	Rifampin	3280	25	1.9	2.0
Age	18-34	2925	10	1.0 (ref)	1.0 (ref)
	35-64	3303	24	2.1	1.8
	65-90	257	4	4.9	3.6
Sex	Female	3727	27	1.0 (ref)	1(ref)
	Male	2758	11	0.6	0.6
BMI	Normal	3320	20	1.0 (ref)	-
	Underweight	438	0	0.2	-
	Overweight	1823	14	1.3	-
	Obese	904	4	1.8	-
Immune Status	No Immune suppr.	5800	34	1.0 (ref)	-
	HIV-positive	268	0	0.3	-
	Other immune suppr.	417	4	1.8	-
Alcohol Use	Never drinks	4312	27	1.0 (ref)	-
	≤ 1 drink per week	1764	10	0.9	-
	> 1 drink per week	409	1	0.6	-
Smoking history	Has never smoked	4917	32	1.0 (ref)	-
	Currently or has smoked	1568	6	0.6	-
Medication Consistency	Consistency ≥ 90%	4591	21	1.0 (ref)	1 (ref)
	Consistency < 90%	1894	17	2.0	2.2
Concomitant medications	None	4987	22	1.0 (ref)	1.0 (ref)
	Any	1498	16	2.5	1.9

Bibliography

[1] Jonathon R Campbell, Anete Trajman, Victoria J Cook, James C Johnston, Menonli Adjobimey, Rovina Ruslami, Lisa Eisenbeis, Federica Fregonese, Chantal Valiquette, Andrea Benedetti, and Dick Menzies. Adverse events in adults with latent tuberculosis infection receiving daily rifampicin or isoniazid: post-hoc safety analysis of two randomised controlled trials. *The Lancet. Infectious diseases*, 3099(19):1–12, 2019. ISSN 1474-4457. doi: 10.1016/S1473-3099(19)30575-4. URL http://www.ncbi.nlm.nih.gov/pubmed/31866327.

[2] A Anjos, T Pereira, P Korshunov, A Mohammadi, and S Marcel. Continuously Reproducing Toolchains in Pattern Recognition and Machine Learning Experiments. *International Conference on Machine Learning (ICML)*, (31), 2017. URL http://publications.idiap.ch/downloads/papers/2017/Anjos{_}ICML2017-2{_}2017.pdf.

[3] Issar Smith. Mycobacterium tuberculosis Pathogenesis and Molecular Determinants of Virulence. *Clinical Microbiology Reviews*, 16(3):463–496, July 2003. ISSN 0893-8512. doi: 10.1128/CMR.16.3.463-496.2003. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC164219/.

[4] Tuberculosis - Symptoms and causes, . URL https://www.mayoclinic.org/diseases-conditions/tuberculosis/symptoms-causes/syc-20351250.

[5] Patrick Tang and James Johnston. Treatment of Latent Tuberculosis Infection. *Current Treatment Options in Infectious Diseases*, 9(4):371–379, 2017. ISSN 1523-3820. doi: 10.1007/s40506-017-0135-7. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5719124/.

[6] Tuberculosis (TB), 2019. URL https://www.who.int/news-room/fact-sheets/detail/tuberculosis.

[7] WHO | Latent TB Infection : Updated and consolidated guidelines for programmatic management, . URL http://www.who.int/tb/publications/2018/latent-tuberculosis-infection/en/.

[8] A. Trajman, R. E. Steffen, and D. Menzies. Interferon-Gamma Release Assays versus Tuberculin Skin Testing for the Diagnosis of Latent Tuberculosis Infection: An Overview

of the Evidence. *Pulmonary Medicine*, 2013, 2013. ISSN 2090-1836. doi: 10.1155/2013/601737. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3582085/.

[9] Kathleen R. Page, Frangiscos Sifakis, Ruben Montes de Oca, Wendy A. Cronin, Meg C. Doherty, Lynn Federline, Sarah Bur, Thomas Walsh, Walter Karney, James Milman, Nancy Baruch, Akintoye Adelakun, and Susan E. Dorman. Improved Adherence and Less Toxicity With Rifampin vs Isoniazid for Treatment of Latent Tuberculosis. *American Medical Association*, 166:1860–1870, 2006.

[10] Jacqueline M. Bos, Gerard A. Kalkman, Hans Groenewoud, Patricia M.L.A. Van Den Bemt, Peter A.G.M. De Smet, J. Elsbeth Nagtegaal, Andre Wieringa, Gert Jan Van Der Wilt, and Cornelis Kramers. Prediction of clinically relevant adverse drug events in surgical patients. *PLoS ONE*, 13(8):1–12, 2018. ISSN 19326203. doi: 10.1371/journal.pone.0201645.

[11] Nazanin Falconer, Michael Barras, and Neil Cottrell. Systematic review of predictive risk models for adverse drug events in hospitalized patients. *British Journal of Clinical Pharmacology*, 84(5):846–864, 2018. ISSN 13652125. doi: 10.1111/bcp.13514.

[12] Daniel D. Bohl, Alexander J. Idarraga, George B. Holmes, Kamran S. Hamid, Johnny Lin, and Simon Lee. Validated Risk-Stratification System for Prediction of Early Adverse Events Following Open Reduction and Internal Fixation of Closed Ankle Fractures. *Journal of Bone and Joint Surgery - American Volume*, 101(19):1768–1774, 2019. ISSN 15351386. doi: 10.2106/JBJS.19.00203.

[13] Xiaowei Song, Arnold Mitnitski, Jafna Cox, and Kenneth Rockwood. Comparison of machine learning techniques with classical statistical models in predicting health outcomes. *Studies in Health Technology and Informatics*, 107:736–740, 2004. ISSN 18798365. doi: 10.3233/978-1-60750-949-3-736.

[14] Senjuti Basu Roy, Moushumi Maria, Tina Wang, Anne Ehlers, and David Flum. Predicting Adverse Events After Surgery. *Big Data Research*, 13:29–37, 2018. ISSN 22145796. doi: 10.1016/j.bdr.2018.03.003. URL https://doi.org/10.1016/j.bdr.2018.03.003.

[15] Summer S. Han, Tej D. Azad, Paola A. Suarez, and John K. Ratliff. A machine learning approach for predictive models of adverse events following spine surgery. *Spine Journal*, 19(11):1772–1781, 2019. ISSN 18781632. doi: 10.1016/j.spinee.2019.06.018.

[16] Laíse Soares Oliveira Resende and Edson Theodoro dos Santos-Neto. Risk factors associated with adverse reactions to antituberculosis drugs. *Jornal Brasileiro de Pneumologia*, 41(1):77–89, 2015. doi: 10.1590/s1806-37132015000100010.

[17] Ana Tavares e. Castro, Mariana Mendes, Sara Freitas, and Paulo Cravo Roxo. Incidence and risk factors of major toxicity associated to first-line antituberculosis drugs for latent

and active tuberculosis during a period of 10 years. *Revista Portuguesa de Pneumologia*, 21(3):144–150, 2015. ISSN 08732159. doi: 10.1016/j.rppnen.2014.08.004. URL http://dx.doi.org/10.1016/j.rppnen.2014.08.004.

[18] Philip Lobue and Dick Menzies. Treatment of latent tuberculosis infection: An update. *Respirology*, 15(4):603–622, 2010. ISSN 14401843. doi: 10.1111/j.1440-1843.2010.01751.x.

[19] Benjamin M. Smith, Kevin Schwartzman, Gillian Bartlett, and Dick Menzies. Adverse events associated with treatment of latent tuberculosis in the general population. *Cmaj*, 183(3):173–179, 2011. ISSN 14882329. doi: 10.1503/cmaj.091824.

[20] Claudia C. Dobler, Queenie Luu, and Guy B. Marks. What Patient Factors Predict Physicians' Decision Not to Treat Latent Tuberculosis Infection in Tuberculosis Contacts? *PLoS ONE*, 8(9):5–10, 2013. ISSN 19326203. doi: 10.1371/journal.pone.0076552.

[21] Christopher Martin Sauer, David Sasson, Kenneth E. Paik, Ned McCague, Leo Anthony Celi, Iván Sánchez Fernández, and Ben M.W. Illigens. Feature selection and prediction of treatment failure in tuberculosis. *PLoS ONE*, 13(11):1–14, 2018. ISSN 19326203. doi: 10.1371/journal.pone.0207491.

[22] Muhammad Tahir Khan, Aman Chandra Kaushik, Linxiang Ji, Shaukat Iqbal Malik, Sajid Ali, and Dong-Qing Wei. Artificial Neural Networks for Prediction of Tuberculosis Disease. *Frontiers in Microbiology*, 10(March):1–9, 2019. ISSN 1664-302X. doi: 10.3389/fmicb.2019.00395. URL https://www.frontiersin.org/article/10.3389/fmicb.2019.00395/full.

[23] Dick Menzies, Richard Long, Anete Trajman, Marie-Josée Dion, Jae Yang, Hamdan Al-Jahdali, Ziad Memish, Kamran Khan, Michael Gardam, Vernon Hoeppner, Andrea Benedetti, and Kevin Schwartzman. Adverse Events with 4 Months of Rifampin Therapy or 9 Months of Isoniazid Therapy for Latent Tuberculosis Infection. *Annals of Internal Medicine*, 149, 2008.

[24] Dick Menzies, Menonli Adjobimey, Rovina Ruslami, Anete Trajman, Oumou Sow, Heejin Kim, Joseph Obeng Baah, Guy B Marks, Richard Long, Vernon Hoeppner, Kevin Elwood, Hamdan Al-Jahdali, Martin Gninafon, Lika Apriani, Raspati C Koesoemadinata, Afranio Kritski, Valeria Rolla, Boubacar Bah, Alioune Camara, Isaac Boakye, Victoria J Cook, Hazel Goldberg, Chantal Valiquette, Karen Hornby, Marie-Josée Dion, Pei-Zhi Li, Philip C Hill, Kevin Schwartzman, and Andrea Benedetti. Four Months of Rifampin or Nine Months of Isoniazid for Latent Tuberculosis in Adults. *New England Journal of Medicine*, 379(5):440–453, may 2018. ISSN 0028-4793. doi: 10.1056/NEJMoa1714283. URL https://doi.org/10.1056/NEJMoa1714283http://files/70/Menziesetal.-2018-FourMonthsofRifampinorNineMonthsofIsoniazi.pdfhttp://files/69/NEJMoa1714283.html.

[25] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics*, 35(5-6):352–359, 2002. ISSN 15320464. doi: 10.1016/S1532-0464(03)00034-0.

[26] Priya Ranganathan, C. S. Pramesh, and Rakesh Aggarwal. Common pitfalls in statistical analysis: Logistic regression. *Perspectives in Clinical Research*, 8:107–112, 2017. doi: 10.4103/picr.PICR.

[27] Georg Heinze and Michael Schemper. A solution to the problem of separation in logistic regression. *Statistics in Medicine*, 21(16):2409–2419, 2002. ISSN 02776715. doi: 10.1002/sim.1047.

[28] David Firth. Bias Reduction of Maximum Likelihood Estimates. *Biometrika*, 80(1):27–38, 1993.

[29] Dookie Kim, Sanjiban Sekhar Roy, Tim Länsivaara, Ravinesh Deo, and Pijush Samui. *Handbook of research on predictive modeling and optimization methods in science and engineering*. IGI Global, 2018.

[30] Richard Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrain optimization. *Journal of Scientific Computing*, 16(5):1190–1208, 1995. doi: 10.1137/0916069.

[31] Anders Krogh and John A. Hertz. A Simple Weight Decay Can Improve Generalization. *Proceedings of the 4th International Conference on Neural Information Processing Systems*, pages 950–957, 1991. ISSN 00406090.

[32] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20: 273–297, 1995. doi: https://doi.org/10.1007/BF00994018.

[33] J Maroco, D Silva, A Rodrigues BMC . . . , and Undefined 2011. Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic. *Bmcresnotes.Biomedcentral.Com*, 2011.

[34] Max Bramer. *Principles of data mining*, volume 180. Springer, 2007.

[35] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[36] Thomas A. Lasko, Jui G. Bhagwat, Kelly H. Zou, and Lucila Ohno-Machado. The use of receiver operating characteristic curves in biomedical informatics. *Journal of Biomedical Informatics*, 38(5):404–415, 2005. ISSN 15320464. doi: 10.1016/j.jbi.2005.02.008.

[37] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. ISSN 00313203. doi: 10.1016/S0031-3203(96)00142-2.

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Strengths:

- EPFL bioengineer with strong analytical skills
- Machine Learning professional experience
- MATLAB, C++, Python

Colombine Verzat

Education

2018–present **Dual Master in Artificial Intelligence**, Idiap, Switzerland.
State-of-the-art theoretical courses in machine learning and professional activity at Idiap using machine learning for the detection of adverse events during Latent Tuberculosis Infection treatment

2015–2018 **Master of Science (MSc) in Bioengineering**, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
Neuroengineering education including neuroscience, biomechanics, biophysics, neuroprosthetics, brain computer interaction, programming in Python and functional brain imaging

2012–2015 **Bachelor in Life Sciences**, EPFL.
Engineering education combining advanced training in basic science (algebra, calculus, physics, computer programming, signal processing, electronics) with chemistry, biology, bioinformatics and neurosciences

2012 **Baccalauréat Section scientifique**, *Lycée de la Légion d'Honneur*, Saint-Denis, France.
Mention très bien, option européenne (*reinforcement in english including study trips in the UK and USA*)

Core Experience

Sept-17–March-18 **Institute of Behavioural Neuroscience, University College London**, *Master thesis*, Dr. C. Perrodin, Pr. C. Sandi, London, UK.
"Identifying perceptually informative acoustic cues in mouse social communication"
Designed and ran behavioural experiments in mice, analysed the resulting video-tracking data in Matlab

Spring 2017 **Laboratory of Behavioral Genetics, EPFL**, *Semester project*, Dr. J. Rodrigues, Pr. C. Sandi.
"Analytical approaches to data from human virtual reality and neuro-physiology studies"
Ran virtual reality experiments on human subjects and analysed gait data from wearable sensors in Matlab

Autumn 2016 **Second Sight medical products**, *4-month internship*, M. Flourence, F. Merlini.
"Challenge of integrating artificial vision provided by retinal neuroprosthetic device Argus II into residual peripheral vision in Age-related Macular Degeneration patients"
Designed and ran behavioural tasks for AMD patients in Manchester to assess the benefice of the Argus II device

Spring 2015 **Biodesign for a real world, Hackuarium and EPFL**, *S. Hirosue*.
"Acoustic feedback of arsenic levels in contaminated water in *Biodesign for the real world* project"
Part of a local team in a large-scale open source project, added a new auditory modality to the prototype bioreporter using Arduino
Poster at the Salon des Technologies et de l'Innovation de Lausanne (STIL) 2015

Rue du Collège 6 – 1920 Martigny
078 988 78 28 • *colombine.verzat@laposte.net*
https://www.linkedin.com/in/colombineverzat/

2014–2017 **Teaching assistant in calculus and programming**, EPFL.
Tutored first-year EPFL students for practical exercises, corrected exams, supervised group projects

Additional Experience

Spring 2017 **Festival Balélec**, *Electricity manager*, EPFL.
Co-planned the electricity distribution of a large music festival (15'000 people), led a team of 10 staffs, regular meetings with electricity collaborator of EPFL and electricity distributors

Nov-15–Nov-16 **EPFL Integration Week**, *Head of Animation*, EPFL.
Designed and implemented creative activities, drew a budget for activities' material, coordinated the organization of the week with a committee, led a team of staff

2015–2017 **AGEPoly**, *Team animation*, EPFL.
Part of a team to organize entertaining and social bonding activities on campus for students and staff

Scouting, *9 years*.
Including 3 years as a chieftain in Lausanne responsible for 20 girls (8 to 12 years old), planned and coordinated outdoor activities

Technical skills

Machine Learning Linear Algebra, Datastructures and algorithms, Foundations in statistics for AI, Signal processing

IT Basics of Arduino and Blender, good knowledge of Java and LaTeX, advanced level in MATLAB, C++ and Python

Engineering skills Mathematical and Physical calculations, Data analysis using probabilities and statistics, Programming, Computational models in biology, Analysis of electrical circuits and sensors in medical instrumentation

Biotechnology Strong knowledge in Molecular and Cellular Biology, Biochemistry, Physiology, Microbiology, Neurosciences, Oncology, Genomics, Biomechanics, Biomaterials, Tissue engineering, Biomicroscopy, BioMEMS, Stem cells, Bioethics, Neuroengineering, Brain Computer Interaction, Functional Brain Imaging

Research methods Experimental Procedures Design, Statistical analysis of research data, Writing scientific reports

Languages

French Native language

English Fluent: lived 7 months in the UK

German Basic understanding

Interests

Badminton Badminton Club Martigny, Rueil Athletic Club Badminton, Sobell Badminton Club

Board/card games Friendly competitive spirit, analytical mind

Cooking Occasional cooking lessons, discovering new recipes

Rue du Collège 6 – 1920 Martigny
078 988 78 28 • *colombine.verzat@laposte.net*
https://www.linkedin.com/in/colombineverzat/