Università degli Studi di Pavia
Università della Svizzera Italiana

Joint Ph.D. program in Computational Mathematics and Decision Sciences
XXXIV Cycle

EXPLAINABLE ARTIFICIAL INTELLIGENCE: AN APPLICATION TO COMPLEX GENETIC DISEASES

Advisor:
Prof. Marco GORI

Co-Advisors:
Prof. Simone FURINI
Prof. Alessandra RENIERI

PhD Dissertation of:
Nicola PICCHIOTTI
matr. 468987

Academic year 2020 - 2021

To
Teresa
and Elena
who fight against themselves,
and love.

Acknowledgements

Thanks to Marco Gori for his ideas, and for welcoming me. Thanks to Monica, Francesco, and Elena, coauthors as well as friends, and thanks to Matteo, Francesca, and Leonardo, friends as well as colleagues. Thanks to Sergio, Elisa, Kristina, and Chiara: good friends and good coauthors. Thanks to Luca Spadafora, Andrea Spuntarelli, and Paolo Fiorini, for being good bosses. Thanks to Alessandra Renieri, for the opportunity to contribute to this great research, and for teaching me what it means to be strong. Thanks to Simone Furini for his brilliant reasoning and humility. Thanks to Luca Pavarino for his availability. Thanks to Monica Bianchini and Marco Tanfoni. Thanks to Maurizio Sanarico, for his openness of mind. Thanks to the SAILAB people, Pavia's Ph.D. students, all the people of Genetica Medica of the University of Siena, and everyone participating in Thursday's evening meetings. Great thanks to the GEN-COVID Multicenter Study, `https://sites.google.com/dbm.unisi.it/gen-covid`. Thanks to the voluntary patients and the research staff in every hospital, not only for contributing to the data of the study. Finally, thanks to my mother, my siblings, my father, and Teresa, for being there for me.

Brescia
28/11/2021

Contents

List of Figures

List of Tables

Introduction

Ludwig Wittgenstein concluded the *Tractatus Logico-Philosophicus* with the well-known sentence: *"Whereof one cannot speak, thereof one must be silent"*. We believe that, when there is nothing to say, statistical models are supposed to be silent, just like humans. The challenge, both for humans and for models, is to understand when there is nothing to say. For instance, according to Wittgenstein, humans should not give an answer to a philosophical question, ill-defined by a misunderstanding language. Then, what about models? When is a machine learning model trustworthy enough to give reliable predictions in different contexts? In this thesis, we try to define and implement a framework in which machine learning models are as much reliable as possible. This topic is particularly relevant for high-stakes applications, such as the predictive model describing the COVID-19 severity from host genetic, which we will tackle in Chapters 3 and 4.

In particular, the "black box" nature of deep neural network models is often a limit for safe applications, since the reliability of the model predictions can be affected by the incompleteness[1] in the optimization problem's formalization [5]. Doshi et al. have recently shown that an adequate level of interpretability could increase the neural network trustworthiness [6]. However, this is generally difficult to achieve without altering the mechanism of deep learning [4]. For these reasons, we are mainly concerned with the need to make the model predictions explainable to humans with a by-design approach. It is reasonable to think that, when the model is doing some explainable task, it is under the control of humans, who live the automatic model's decisions much more comfortably.

Following the *fil rouge* of explainability, we will introduce the theoretical aspects of the problem (Chapter 1); we will continue with presenting novel methodologies in the interpretable Artificial Intelligence (AI) field (Chapter 2) and conclude in Chapters 3 and 4 with present-day applications of interpretable machine learning models to the genetic component of COVID-19 severity.

In order to frame and systematize the topic, in Chapter 1 we define a set of requirements that an AI model has to satisfy in order to be considered reliable. Concretely, we propose three main areas to be assessed:

1. the correctness of the model architecture, both from the point of view of data

[1] For instance, incompleteness happens whenever the model is solving a proxy of the assigned task, e.g., the well-known snow detector, instead of the desired wolf Vs husky classifier, described in [4].

and from the one of the algorithm;

2. the adequacy and reliability of functional requirements related to the model predictions; and

3. the fulfillment of the policy guidelines, including privacy, safety, fairness, and accountability.

In the same chapter, we provide a checklist with practical requisites in the form of specific questions, in order to bridge the gap between abstract requirements, e.g. those reported in regulation (see [7]) and practical problems. Inside the chapter, we include the literature review on the different topics we will be dealing with in the following.

Besides the practical matters discussed in the first chapter, in Chapter 2 a more theoretical study is reported, where we describe two novel methodologies developed during the Ph.D. in the explainability field. Both methodologies exploit, in an innovative way, the well-known concept of feature importance, which is intended to assign a score to input features based on how useful they are at predicting the target. A summary of the two methodologies is here reported:

- in the first methodology, we basically recognize that in a multi-layer neural network with Rectified Linear Units (ReLU), the non-linear behavior of the ReLU function gives rise to a natural clustering when the pattern of active neurons is considered. This observation helps to deepen the learning mechanism of the network; in fact, we demonstrate that, within each cluster, the network can be fully represented as an affine map. The consequence is that we are able to recover an explanation, in the form of *feature importance*, for the predictions done by the network to the instances belonging to the cluster. Although the methodology needs to be tested in many more cases, it seems to be able to increase the level of interpretability of a fully connected feedforward ReLU neural network, downstream from the fitting phase of the model, without altering the structure of the network.

- The second novel methodology was born as the attempt to apply the well-developed framework of logic constraints in machine learning (for a detailed review, see [8]) to the explainability topic. The motivating question was the following: is it possible to drive the AI model with the human apriori knowledge[2] of the extent of features' importance? The methodology we are proposing tries to answer the question with a regularization term, aiming at encouraging the importance of the features to be a predetermined range. Hidden behind the regularization term there is a local method for the feature importance computation, that in our experiments is the Layer-wise Relevance Propagation (LRP), which links the model weights to be optimized to the user-defined constraints on feature importance.

  Many possible applications of this model-agnostic theoretical framework are described in the present thesis, together with some promising experimental results in the fairness area.

[2] in a sort of human-weighted AI.

Unfortunately, at the end of 2019, an event utterly changed the world in many respects. The outbreak of Coronavirus disease 2019 (COVID-19), caused by SARS-CoV-2, has spread worldwide within a few weeks, reaching the status of a global pandemic. In such a challenging context, a strong contribution has been provided by the scientific research community, according to the individual capacity and sense of responsibility. In particular, during the initial phases of the pandemic, a considerable effort was made to monitor the evolution of the infection in terms of new cases, recoveries and deaths, by exploiting mathematical modeling in the field of epidemiology. For example, my co-authors and I contributed by introducing an SEIR compartmental model with a mobility-dependent parametrization [1].

Later in the pandemic, after almost two years, COVID-19 has demonstrated itself to be a disease having a broad spectrum of clinical effects: from asymptomatic patients to those with severe symptoms leading to death or persistent disease (the so-called "long COVID") [9, 10]. While vaccination programs and other preventive measures are an important tool to significantly dampen infection transmission and reduce disease expression, a much deeper understanding of the interplay between SARS-CoV-2 and host genetics is required, in order to support the development of treatments for new virus variants as soon as they arise.

In Chapter 3 we report the results of the analyses carried out to discover the genetic variability explaining the different degrees of severity of patients affected by COVID-19. The challenge was that, differently from Mendelian disease, where a single variant can be responsible for the disease, complex genetic diseases like COVID-19 are characterized by a potentially high number of both rare and common variants contributing together in a cooperative way to the severity. Moreover, within the chapter, further characterization of the multi-organ phenotype of the disease is provided.

Finally, in Chapter 4 we aim at merging the notions learned in the previous chapters by introducing a new interpretable machine learning model able to predict the severity of COVID-19 from host genetic data. It is worth stressing that interpretability has been a guiding principle in the definition of the machine learning model, through the chapter. In fact, we deem that only a readily interpretable model can provide useful and reliable information for clinical practice, while also contributing significantly to diagnostics, and therapeutic targeting. Unfortunately, the high dimensionality of host genetic data poses a serious challenge to evident and reliable interpretability. We tackled this complexity, as revealed by the studies reported in Chapter 3, by separately considering both rare and common variants that were expected to contribute to the likelihood of developing a severe form of the disease. This fact, along with the enriched gene-level representation of host genetic data, was the basis for the model development, by an ensemble of interpretable machine learning algorithms.

In Chapter 5 we present the conclusions, in addition to highlighting the outlook of the work. In particular, many possible applications of the methodologies introduced in Chapter 2 are suggested. Instead, research on the genetic bases of COVID-19 severity has already made great progress but needs to be further consolidated, for instance by modeling the heterogeneous biological processes for different groups of patients.

Chapter 1

A model risk framework for Machine Learning applications

The views, thoughts, and opinions expressed in this chapter are those of the authors in their individual capacity and should not be attributed to Banco BPM S.p.A. or to the authors as representatives or employees of Banco BPM S.p.A

Machine Learning (ML) applications are becoming increasingly widespread in many different sectors: medicine, the advertising industry, recommender systems, financial applications, etc. In the banking sector, ML applications range from risk management [11, 12] to pricing [13, 14], and from customer segmentation [15] to econometric and time series forecasting [16, 17]. It is easy to foresee that the volume of business decisions driven by ML will increase in future years. Due to this, the assessment of the ML model's adequacy has become one of the major challenges to overcome, and this process becomes particularly crucial for high-stakes applications such as diagnostic techniques, autonomous guide, risk management, security, and software.

Toy examples and case studies, usually applied by the ML scientific community to a public dataset, could not reveal potential issues in ML application. Indeed, proof of an algorithm's effectiveness does not guarantee that an application operates correctly, especially in the long term in a production environment. Therefore, while developing and deploying an ML model can be relatively straightforward, maintaining it in a production system, whilst guaranteeing a high level of performance over time, is a complex task. A clear explanation is given by the so-called "technical debt" known in programming, introduced by Ward Cunningham in 1992, with a parallelism from monetary debt [18]. The technical debt indicates the additional work needed to compensate for the quick and dirty solution adopted before. Since the potential technical issues can become, over time, increasingly difficult to correct, it is said that the technical debt accumulates "interest". ML can be seen as a "quick and dirty" solution totally relying on data, which does not need a hypothesis or prior assumptions. For this reason, a deterioration of the model's performance over time may occur without the possibility for the user to check the correctness of their own hypotheses.

There is no immediate answer to the problem of identifying and mitigating the risk of ML, since regulation, industry standard, and scientific knowledge are not currently widespread or exhaustive. Some organizations have issued principles and guidelines for ethical AI [19], such as "Ethics guidelines for trustworthy AI" [7] in the European Union (EU), which outlines the principles for correct implementation and usage of ML applications. Nevertheless, a link is currently missing between the high-level regulation and practical matters in the model designing phase. Despite numerous papers discussing AI policy [20], there is a lack of works with practical suggestions for ML applications.

In the financial framework, depending on the particular jurisdiction, models are traditionally regulated by specific monitoring bodies. For instance, risk management models are regulated by the European Banking Authority (EBA), through specific regulatory technical standards and guidelines, which focus on standard and internal models involved in the computation of regulatory and economic capital requirements. In this context, the institution's Internal Validation Unit (IVU) monitors the soundness and correctness of the internal models developed and employed by the bank. The basic guiding principles for the validation of standard statistical models can be found in Federal Reserve document SR 11-7 [21]. However, with the usage of ML, new requirements emerge, since the traditional strategies based on the model's hypotheses and domain knowledge injection are substituted by the automatic research of patterns behind data. ML models usually deal with high-dimensional input data, nonlinear models, and other technical aspects. These features change the way the validation should be carried out in the direction of a more complex and frequent assessment process. Recently, EBA issued a report on the identification of the key challenges in using big data and advanced analytics [22], focusing on the increasing challenge for both financial institutions and regulators due to the widespread use of these applications. In particular, EBA recognizes that "elements of trust", such as explainability and interpretability, are key elements that need to be assured, "by design".

1.1 Related works

Many works have detected potential issues in the usage of Machine Learning (ML) models for production. In [18], a lucid analysis of the so-called "technical debt" reveals the possible problems related to the ML model. In [23], a series of rules were collected in the form of best practices to deploy an effective ML model, whereas [24] reviews the main practices to build ML-powered applications. A complete review of general requirements in ML models is provided in [25] and in [26] with a bibliographic review. In [27, 28, 29], applicative reports in the risk management area are presented. Other surveys are focused on certain sub-fields of ML or specific applications. For example, in the context of healthcare, a clinical checklist is presented in [30] to assess the suitability of ML applications. According to the authors, the current literature lacks a comprehensive framework and a checklist of concrete requirements valid in many contexts.

Concerning the methodologies for testing a specific area of ML models, some of the literature adapts notions coming from the software testing domain [31] showing that concepts such as code coverage, mutation testing, or property-based testing can

be translated to identify problems in ML. Another attempt can be found in [32] where the authors test specific implementations of two different ML ranking algorithms. Other analyses are reported in [33], which provides a list of 28 specific tests learned from deploying ML systems at Google. In [34], a technique of standard model testing based on "metamorphic testing" is applied to ML models. The aim of this is to verify necessary properties of the intended functionality of the software, such as the invariance of the predictions when an affine transformation, a permutation of the input, is performed. The framework is applied to k-nearest neighbours and the Naïve Bayes classifier.

1.2 Requirements for Machine Learning model assessment

The validation process of a Machine Learning (ML) model includes all activities carried out in order to evaluate three fundamental aspects: (1) the correctness of the model architecture, (2) adequacy of functional requirements, and (3) fulfilment of the policy guidelines. This attempts to answer the following three fundamental questions: How is the model built? How well does it work? What are the consequences of its activities?

The requirement of the correctness of the *model architecture* is related to the conceptual soundness of the model, its design and the assumption made. This includes an assessment of both the representativeness of the chosen *dataset* and the main features of the *algorithm* (scalability, stability, etc.). In particular, more than in the standard statistical techniques, in ML, the dataset representativeness has a fundamental role, since the model parameters are highly adaptive to the amount of information included in the data.

The *functional requirements* are related to the behavior of the model, i.e., to the predictions performed by the model in different contexts. First, a mandatory step is the standard analysis of the *model performances* in a testing dataset through the usual evaluation metrics. However, a large part of the functional requirements refers to the *reliability* aspects of the model, which provide a more thorough check of the model performances under many possible scenarios. For instance, the behavior of the model could be assessed whenever a feature changes (*sensitivity analysis*) or under ad-hoc adversarial instances (*adversarial examples*). Another fundamental indicator of the reliability of the model is its *explainability*, i.e., the capability of the algorithm to make its decision process human-comprehensible to some extent. This functionality is essential to verify whether the model is actually solving the task assigned to it, rather than another one, which is probably simpler and not generalizable to the original problem[1].

Finally, the requirement in the *policy area* includes the *fairness* of the predictions, the *accountability*, the *safety* of the model and the *privacy* of the sensitive dataset. Other policy requirements reported in [7] are societal and environmental *well-being*, including sustainability, environmental friendliness, social impact, and the requirement of *human agency*, referring to fundamental rights, human agency, and human oversight. Finally, EBA reports [22] stress the importance of using an "ethical by design" approach in building ML applications. In this context, the linkage with Environmental, Social, and Governance (ESG) policies cannot be overlooked.

[1] e.g., the snow detector instead of the desired wolf/husky classifier, described in [4].

In the following list, the topics for the requirements that should be evaluated during the assessment process are summarized.

A) **Architectural requirements**:

1. **dataset representativeness** through the phases of both data acquisition and preprocessing;
2. **algorithm adequacy**, such as scalability, stability, and efficiency.

B) **Functional requirements**;

3. **model performance** evaluating the testing phase, also with respect to benchmark models. Model relevance, i.e., potential underfitting/overfitting;
4. **model reliability**, including sensitivity analysis, explainability and model maintaining.

C) **Policy requirements**:

5. **privacy** and data governance, including respect for privacy, data quality and integrity, and access to data;
6. **safety**, including resilience to cyber-attacks and security, a fall-back plan, and general safety;
7. **fairness**, diversity, nondiscrimination, avoidance of unfair bias, client's accessibility to personal information and universal design, and stakeholder participation;
8. **accountability**, auditability, minimization and reporting of negative impacts, trade-offs, redress, chain of responsibility, transparency, traceability, and communication.

For each section treating the above-mentioned points, we provide a table with specific requirements in the form of technical questions. The questions are identified with an ID number and the letter: A for the requirements in the architectural area, F for the functional area, and P for the policy area.

For the sake of clarity, in Figure 1.1, a tree chart representing the phases of the validation process is shown.

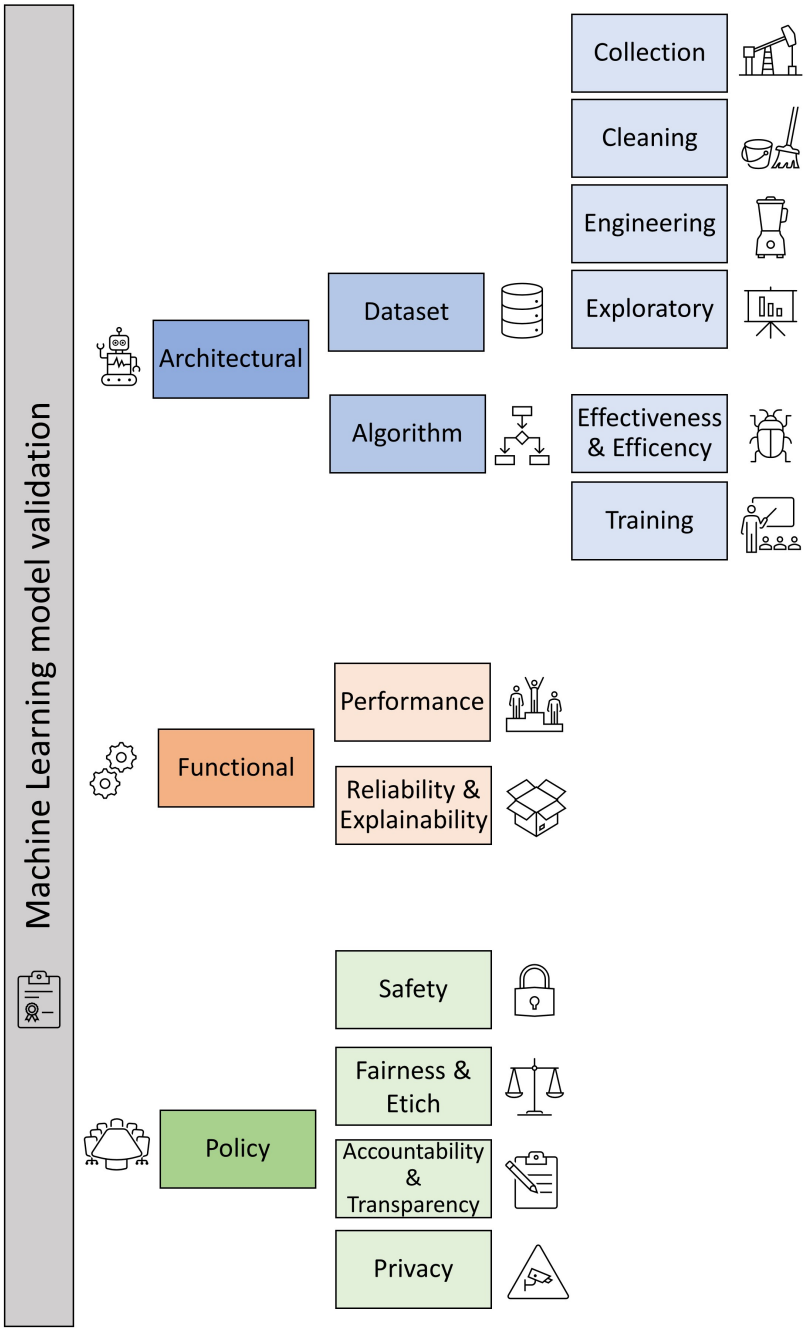


Figure 1.1: Representation of the requirements for a Machine Learning model validation process.

1.3 Dataset representativeness

In the context of the validation of the Machine Learning (ML) model, the correctness and representativeness of the dataset assume a primary role. The reason for this is that, since the model's hypotheses are not explicit, the calibration of the parameters is driven almost entirely by the information included in the data. Therefore, the validation process needs to primarily assess if data have been correctly collected and represent what they should. As a consequence, a large amount of work should be conducted to ensure that the quality of data is good. In [35], [36], [37], the standard techniques for testing the data pre-processing phase are reviewed.

1.3.1 Data collection

The data collection phase aims at gathering, merging, and organizing the input dataset of a model. The dataset can be either provided by an external source or derived from an internal data acquisition process, or a combination of both. The fact that the process is externalized does not eradicate the need for an assessment [7]. Therefore, in any case, an assessment of the dataset is needed, with the complication, in the external case, that the acquisition process could not be directly assessed.

A wide range of possibilities for dataset acquisition exist, usually relying on data scraping techniques to extract data from web content. Other methods rely on leveraging the public dataset, e.g., Kaggle and the UCI Machine Learning Repository; exploiting past datasets harvested in another context; directly observing the necessary data, etc.

Reliability of data collection

When the dataset is internally collected by the company, an adequate process for evaluating the reliability of the data acquisition process needs to be defined and followed.

A possible source of issues is the incorrect labeling of the supervised dataset. A typical ex-post warning of this problem emerges when the model has lower performances than the random guess, e.g., is better predicting the wrong class. It is, therefore, best to measure the target class multiple times (or with different sources) whenever possible. Moreover, when measurements are carried out, the estimate of the error should be annotated along with the whole numeric data value (any rounding, discretization procedures will come later) in order to evaluate the statistical significance.

Especially when the dataset is internally gathered, the cooperation between the person who takes the measurement and the data analyst needs to be stressed, in order to grant whether the interpretation of the latter, who may not be an expert in the field, is coherent and correct. In the case of an external dataset, the assessment process is more difficult due to the possible lack of transparency. One possibility is to rely on verification bodies that certify data providers as golden/not golden sources.

An important question is whether the data acquisition process will be continuous in time. When the dataset is continuously updated, the stationarity of the acquired dataset needs to be carefully evaluated. On the contrary, when the dataset is frozen, the assessment should be focused on the adequacy of past observation to reflect recent phenomena.

Possible Biases

The general requirement of the dataset exploited in the model's calibration is to be representative of the real-world data distribution, i.e., free from possible biases or outliers. This should be primary conducted by avoiding the favoring of some instances, features, or target categories during sample selection (the so-called *sample/selection bias*) or by checking possible *measurement bias* in the act of collection. Furthermore, the so-called response or *activity bias* regards content generated by humans, where few sources and a small proportion of users may not be representative of the real data distribution. Moreover, *confirmation bias* is the effect of having an expectation of target data in human labelling. Finally, *feedback loops* happen whenever a feature of the dataset is determined from a previous model, possibly correlated with the target variables.

The first step to mitigate the effect of biases is the awareness of them. Possible solutions can be adopted by exploiting multiple data sources or ad-hoc strategies in the labeling and data scientists' work organization, re-sampling, etc. [38].

Dataset size

Once the overall dataset has been collected, the data scientist needs to select a domain adapted for the scope, by excluding samples. This reduces the dimension of the dataset, but contributes to reducing the difficulty of the problem at hand, at least at the beginning.

As regards to the dataset size, its minimum depth depends upon many factors, such as the number of features, the algorithm, and the specification of the problem. There are different rules of thumb, with no full agreement in the scientific community, e.g., the number of instances of the training dataset should be higher than ten times the number of features. This should be considered as a minimum requirement considering the growing availability of big data.

Finally, a good feature of the dataset for classification problems is the balance between the different classes. Whenever the actual data distribution has an unbalanced nature and our scope is beyond the mere accuracy, the problem can be balanced through different techniques, such as class weight and random undersampling/oversampling.

The list of requirements for the data collection phase is reported in Table 1.1.

Table 1.1: Checklist for data collection requirements.

Code	Check
A1	Is the data acquisition process internal or external? If internal, how are the data sampled (scraping, other datasets, direct observation, etc.)? Is this phase properly documented?
A2	In the case of an external process, is the data source certified as a golden source? Is there any documentation on the acquisition phase?
A3	How will the data be stored and managed? Have you put in place strategies for a long-term archive and planned backups?
A4	Is the dataset properly described, through metadata/data dictionary? Is there any available documentation describing the meaning of the dataset information? Do you understand the meaning of the features?
A5	Is it intended to make the data, a part of the data, or aggregated values publicly available?
A6	Is there any selection/sample bias, e.g., where only a part of the population is tested? Did you perform traditional representativeness analysis (Z-test, Kolmogorov–Smirnov, Chi-squared, Kruskal–Wallis, ANOVA, etc.) of the target population?
A7	Is there any measurement bias?
A8	Is there any possible optimistic bias for the labeling phase?
A9	Are there any feedback loops?
A10	Is the process of label assignment reliable? Is there an error estimate (interval of confidence/multiple measures)?
A11	Are there unavailable desirable data and uncaptured relevant features, e.g., possible confounders of the task?
A12	How are the multiple data sources joined? Are the data correctly formatted before data integration?
A13	Do you put in place data quality and a data filtering process, with checks and inspections?
A14	Is there an ongoing dataset updating process, or does the procedure of data collection remain the same over time? Do you apply any incremental (online) learning models?
A15	Do you filter the samples to your specific domain?
A16	Is the dataset size adequate? Are the data sufficient, or will other observations need to be collected?

After the acquisition of data, the phase of pre-processing includes any transformation of raw data into a dataset with meaningful information for subsequent analyses. The process can be very different from dataset to dataset and involves the *data cleaning* phase for the detection and correction of errors, as well as *feature engineering* and *exploratory data analysis* for the study of possible preliminary patterns.

1.3.2 Data cleaning

The data cleaning phase is the process of detection and correction of the information included in a raw dataset.

Structural errors

Usually, it is first necessary to correct structural errors in the input dataset. Common issues are typographical errors (naming conventions, incorrect capitalization, special characters, etc.), domain value violations, values not satisfying specific constraints, or the absence of mandatory data; in these cases, remedial action is needed. Moreover, especially when multiple data sources are combined, data can be duplicated. A particular focus should be placed on data type conversions.

Missing values

Another common issue is the management of the absence of information, the so-called missing data. The source of unknownness can be either forgetfulness/an omission in the data acquisition process or an actual absence of information for the specific feature–instance couple (censored data). The simplest way to cope with this issue is to remove the instances with missing values, with the disadvantage of dataset reduction. Other approaches assign average feature values, the most common, or a random value based on distribution. Other methods rely on regression or classification for the prediction of the missing values. Especially when missing data are the minority due to the actual absence of information, the coding of the information's absence into specific values is the most appropriate choice. For example, for categorical variables, a typical solution is the null vector in one-hot encoding.

Outliers

An important step in ML data cleaning is the identification and treatment of outliers. Many possibilities exist for the detection of anomalous values, such as box plot visualization, quantile or skewness of distribution (the distribution is skewed in the direction of the outliers). Possible replacement methodologies are based on log-transformation, quantile-based flooring, capping, and replacing outliers with median values; see, for instance, [39].

In order to check the correctness of the input dataset, a set of tests can be performed. The first step is to assess that summary statistical indicators are consistent with expectations. Common aggregated values are central tendencies (mean, median, or mode), standard deviations, percentiles, sum, count, and min-max. Another possibility is to test if a feature distribution follows a schema, e.g., the number of calls in an hour is similar to the Poisson distribution.

In [31], a survey of testing methods for the quality of data is reported, and the main goal is to try to identify errors by applying analytical queries. Data linting introduced in [40] automatically identifies potential issues or inefficiencies in the dataset by exploiting best practices, such as inspecting the training data's summary statistics or individual examples. Another automatic pipeline for improving model accuracy through error detection and repair is described in [41]. The list of requirements for data cleaning is reported in Table 1.2.

Table 1.2: Checklist for data cleaning requirements.

A17	Are mandatory constraints, uniqueness constraints (unique field for a single instance) or data type constraints satisfied?
A18	Are there any typographical errors in values (incorrect capitalization, extra spaces, different naming conventions, etc.)?
A19	Are there any values outside the domain or physically impossible data combinations (e.g., sex: male; pregnant: yes)?
A20	Are aggregated values consistent? Did you test if feature distributions follow a schema, e.g., number of calls in an hour similar to the Poisson distribution?
A21	Any issues in data type conversion?
A22	Are there duplicated observations? Is there a unique ID?
A23	How are the missing values treated?
A24	How are outliers identified and treated? Is the algorithm sensitive to the outlier (e.g., LR) or not (e.g., DT)?
A25	Did you remove non informative features, e.g., features with all constant zeros?

1.3.3 Feature Engineering

The feature engineering phase aims at converting and mapping original data into more appropriate and informative representation. The process includes *feature selection*, *feature transformation* and *feature scaling*.

Feature selection

A particularly delicate process in ML design is the selection of the features based on their potential contribution to the prediction. Usually, this study relies on correlations among features, the explained variance, and the evaluation of the model performances when removing/adding features to the fitting process.

In [42, 43], a review of the principal feature selection techniques is provided, broken down into *feature extraction*, *filtering* and *wrapper* methods. The goal of feature extraction methods is to aggregate the original features into a smaller set of synthetic (and usually non-interpretable) features (PCA, LDA, etc. [44]). The filter approach instead selects the features independently by the potential induction

algorithm [45], e.g., by performing univariate tests, with the disadvantage of not facing the complexity of the problem. Finally, wrapper methods consist of exploring the entire power set of the features set and selecting the subset of features providing the best performances for the task. Here, the problem lies in the computational cost for the high dimensionality of the feature set.

It is worth recalling that, for each feature, there is a software engineering cost for the maintenance, fitting, correlation with other features, and possible instability.

Feature transformation

Features can be represented in different ways. Common transformations include binarization, categorization, and log-normal transformation. New features can be obtained by polynomial terms or a combination of other features. A particular focus should be placed on noisy features, which show, in addition to the signal, a large amount of additional meaningless information. The signal-to-noise ratio and the information content should be evaluated.

Feature scaling

Feature scaling is the process of normalization and standardization of the features in the dataset. Many strategies can be followed, such as the min-max, percentile normalization, and standardization in Gaussian distribution, paying attention to outliers.

The list of requirements for the feature engineering phase is reported in Table 1.3.

Table 1.3: Checklist for feature engineering requirements.

Code	Check
A26	Is there a feature selection process? Are the non-informative (constant, low variance) features ruled out? Did you remove highly correlated features?
A27	How are the features encoded (continuous, binary, categorical, ordinal, etc.)?
A28	Is there a data transformation process (logarithm, binned)?
A29	Are the features aggregated or decomposed, e.g., address decomposed in city, street, etc.?
A30	How are features normalized/standardized? Did you consider outliers in the scaling process?
A31	Did you perform feature engineering among the features? Is there any PCA or synthetic variables?
A32	Did you evaluate the signal to noise ratio, or the information content of features?
A33	Is there any data leakage in the pre-processing phase? Are you sure that the engineered variables are not “contaminated” by the target variable?

1.3.4 Exploratory Data Analysis

Exploratory data analysis (EDA) is a preliminary important step in order to understand the structure of data and the relationships among features. Univariate and bivariate analyses, as well as correlation plots, are instruments exploited in the visualization analysis. Unsupervised techniques, such as PCA, TDA, and SOM, aim to identify patterns, including possible clusters of instances, related to possible confounders. EDA is usually iterative for a complex problem where outliers, trends, or other features of the data are discovered in a dynamic manner. The list of requirements is reported in Table 1.4.

Table 1.4: Checklist for exploratory data analysis requirements.

Code	**Check**
A34	How is the data exploration phase performed? Are there any anomalies or clusters of recurrent patterns?
A35	Did you carry out univariate or bivariate statistics among features or between a feature and a target?
A36	Did you represent distributions of data (PDF, CDF, QQplot) in addition to a summary of statistics (multi-modal behaviour, outliers)? Did you perform statistics for groups?

1.4 Algorithm adequacy

The Machine Learning (ML) algorithm is a coding procedure, which, once fed with the data, is able to provide the automatic programming of the statistical model. Therefore, the task of an algorithm is to learn the model, without being explicitly programmed, starting from data.

The classical distinction of ML algorithms is among supervised learning (SL), unsupervised learning (UL), and reinforcement learning (RL). In SL, the algorithm learns a function $\tilde{f} : \boldsymbol{x} \rightarrow \boldsymbol{y}$ mapping the (usually multidimensional) input $\boldsymbol{x}$ to the output $\boldsymbol{y}$ given a dataset of realized pairs of input/output: $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{y})\}$. In UL, the algorithm learns an informative representation, $h(\boldsymbol{x})$, of the input, $\boldsymbol{x}$, given exclusively a dataset of the input, $\mathcal{D} = \{\boldsymbol{x}\}$. Finally, RL provides the learning of the actions needed to maximize a cumulative reward in an interactive environment, where the dataset of the possible response or the environment is physically evaluated. In Table 1.5, we list the most commonly used ML algorithms for SL, divided into classification and regression, in the case of categorical or continuous target variables, respectively, and UL.

1.4.1 Algorithm effectiveness

The effectiveness of the ML algorithm depends almost exclusively on the peculiarities of the problem at hand, in relation to the available dataset. In SL, some of the algorithms require the target variable to be of a specific nature, e.g., continuous, Boolean, or categorical, and the input features to be represented in determined format, e.g., tabular, image, text, or time series. For this reason, the pool of available algorithms, see Table 1.5, reduces for the specific problem at hand.

Usually, in SL classification or regression problems with a tabular dataset, models such as regression analysis (LR/LogR), decision tree (DT), random forest (RF), neural network (NN), or support vector machine (SVM), are tested. It is important to mention that only certain algorithms are able to provide a natural probabilistic interpretation of the prediction, e.g., LogR contrarily to SVM for SL. Instead, when data are structured in images, texts, or time series, other more sophisticated algorithms are usually used, such as deep NN architectures, that are suitable for the extraction of increasingly abstract information through the different layers. In UL, depending on the task, some algorithms give a hierarchical representation of the cluster (hierarchical clustering) or provide a unique clusterization by specifying the number of clusters, eventually calibrated with hyperparameter tuning (e.g., k-Means and k-Mode) or automatically provided by the algorithm (e.g., DBSCAN).

The choice of algorithm can be also driven by the size of training dataset in relation to the number of features. Generally speaking, when this ratio is high, e.g., more than 10 instances for feature, low bias/high variance algorithms, such as kNN/DT/SVM, can be adopted; on the contrary, when the number of instances is low, e.g., 1:1, high bias/low variance algorithms, such as LR/linear SVM/NB, can be a better option.

One useful rule is to test all the algorithms that are simpler, in terms of parametrization, non-linearity, and optimization, with respect to the chosen one. When the problem is linearly separable, the linear algorithms both in the input space (LR, LogR) and in the transformed feature space (SVM with the linear kernel) are ef-

ficient, and the residual error can be acceptable. On the other hand, nonlinear models can be adopted. Finally, the degree of explainability of an algorithm can be a discriminant feature in choosing the simpler models.

Table 1.5: Summary of most commonly used algorithms for Unsupervised Learning (UL) and Supervised Learning (SL) divided into classification (clas.) and regression (reg.)

Algorithm	**SL clas.**	**SL reg.**	**UL**
Logistic Regression (LogR)	x		
Naive Bayes (NB)	x		
K-Nearest Neighbour (KNN)	x		
Ordinal Regression (OR)	x		
One Rule (OneR)	x		
Discriminant Analysis (DA)	x		x
Apriori (AP)	x		x
Support Vector Machine (SVM)	x	x	
Decision Tree (DT)	x	x	
Random Forest (RF)	x	x	
Extremely Randomized Tree (ExtraT)	x	x	
Neural Network (NN)	x	x	
Convolutional NN (CNN)	x	x	
Linear Regression (LR)		x	
Polynomial Regression (PR)		x	
Stepwise Regression (SR)		x	
Quantile Regression (QR)		x	
Linear Basis Function (LBF)		x	
Generalized Linear Model (GLM)		x	
Gradient Boosting (GBM, XGBoost)		x	
Adaptive Boosting (AB)		x	
Recurrent NN (RNN)		x	
Long Short-Term Memory (LSTMs)		x	
Autoencoders (AE)			x
k-Means / k-Medians (kM)			x
Hierarchical Clustering (HC)			x
Gaussian Mixture (GMix)			x
Principal Component Analysis (PCA)			x
Singular Value Decomposition (SVD)			x
Latent Dirichlet Analysis (LDiA)			x
Factor Analysis (FA)			x
Self-Organizing Map (SOM)			x
DBSCAN			x
Generative Adversarial Networks (GANs)			x
Variational Autoencoders (VA)			x
Kernel Density Estimator (KDE)			x

1.4.2 Efficiency

Another important property of an ML algorithm is the computational efficiency, i.e., the amount of time required for the calibration and prediction phases of the model. It may happen that, with a conspicuous dataset (both in the instances and in the features), the training time can be significant and prohibitive with respect to the available computational resource. Obviously, algorithms with low parameters, such as NB and LR, are more efficient with respect to others, e.g., NN or RF. In addition, the computational time for the train/test of a model depends upon the choice of its hyperparameters, e.g., the number of layers and the choice of kernels. Linked to the computational efficiency, the scalability of an algorithm measures the reduction in the efficiency as a function of the increasing dataset size. The stability of the calibration procedure should also be taken into account.

1.4.3 Correctness

In general, the fundamental ML algorithms are included in the open-source libraries of many software packages. For this reason, the correctness of the source code of the algorithms should be monitored by the scientific community. The main tools for ML are Waikato Environment for Knowledge Analysis (WEKA), R, and Python. For the Python programming language, some well-known modules are Scikit-learn, Keras, Tensor Flow, and PyTorch. Regarding the possible monitoring of the algorithm's code, the unit test aims to assess a particular unit of code, whereas the integrated tests check the overall behavior. Test-Driven Development is intended to build the test before building the functionality, and then refactor the code. The list of requirements for the algorithm adequacy is reported in Table 1.6.

1.4.4 Training of the model

The training phase of a model includes the two fundamental steps of hyperparameter choice and weight calibration.

The hyperparameters are variables that determine the algorithm's structure (e.g., the number of layers in NN, number of neurons, activation function, and loss function) and control the learning process (e.g., the learning rate, batch size, number of epochs, and optimizer). Hyperparameters are not explicitly optimized during the training phase, and their choice could be a demanding process. Rules of thumb, industry standards, choices made for related problems, or trial and error strategies are the common ways to fix the hyperparameters. Otherwise, analytical methods, such as grid search and random search, optimize a performance metric in a test dataset spanning the space of possible hyperparameter values.

Before starting the fitting phase, it should be assured that data are coherently split into training, validation, and test sets and properly randomized, to avoid a spurious effect due to a particular order in the data collection. The behavior of the loss metric in the training and validation sets should be monitored during the fitting phase to measure possible overfitting/underfitting. For the training, many optimization algorithms exist: Stochastic Gradient Descent (SGD), ADAptive Moment estimation (ADAM), Momentum, Root Mean Square Propagation (RMSprop), etc. Another useful instrument to evaluate the model performances is the *learning curve*.The learning curve is the representation of the prediction error vs. the training

Table 1.6: Checklist for algorithm adequacy requirements.

Code	Check
A37	Is the algorithm suitable for the addressed problem and the available dataset? Does the solution require a probabilistic interpretation? Is the loss function adequate?
A38	Is the algorithm adapted to the nature of the target variable (categorical, ordinal, binary)? Is it appropriate for the possible sparsity of features?
A39	Is the problem linearly separable? Did you take care of the bias/variance trade-off depending on the dataset size? Have you tested all the simpler algorithms?
A40	Is there any algorithmic bias? Are you unintentionally favouring some features, e.g., the continuous ones in logistic regression, or those having many categories in random forest?
A41	Are you balancing the dataset or the errors evaluated by the algorithm?
A42	How much is the computational cost? Remember that instance-based methods require a retrain for each prediction.
A43	Does the model keep working with an increase in the data? Is it scalable?
A44	Is the algorithm correctly coded?

set size for training and validation sets. As the dataset becomes larger, the training score should increase, and the validation score should decrease. If the two scores do not converge, this means that there is high variance, potentially solvable by collecting more data. When the two curves converge toward too high values, the model is subject to high bias and should be made more complex.

The list of requirements for the training phase is reported in Table 1.7.

Table 1.7: Checklist for training requirements.

Code	Check
A45	Did you randomize data, to avoid a spurious effect due to a particular order in the data collection process?
A46	How did you fix the model hyperparameters (topology and size of NN, number of layers, neurons, etc.)?
A47	How did you treat regularization terms (L1, L2, elastic net, activity regularize, batch normalization, dropout, etc.)?
A48	How did you fix algorithm hyperparameters (learning rate, batch size, number of epochs, loss function optimizer, etc.)?
A49	Did you visualize the grid search results? Are they the actual maximum or flat?
A50	Is the result of the fitting procedure stable by changing the seed?
A51	Did you visualize the loss as a function of the optimizing step and epochs? Did you evaluate the learning curve?
A52	Did you perform periodic validation of training, periodic checkpoints and analysis of key input value trends?

1.5 Model Performance requirements

The performances of a Machine Learning (ML) model represent the extent of correctness of its forecasts on a held-out clean testing dataset. It is important to highlight that the testing dataset must not be previously exploited for training activities, feature engineering, or hyperparameter optimization. For data with temporal components, the split can be challenging [46]. In any case, the performances on the test should be compared to a pre-determined minimum acceptable threshold or to other benchmark models' performances.

1.5.1 Performance metrics

For regression problems, the common metrics for evaluating the goodness of predictions are based on aggregating the errors between predictions $\tilde{y}$ and expected targets y for a sample of N instances into a single value (sum or average). Since the errors $\tilde{y}_n - y_n$ need to be made positive for a meaningful aggregation, a proper function has to be applied. The *Mean Square Error* (MSE) is defined as the average of the squared differences between predicted and expected target values:

$$\frac{1}{N}\sum_{n=1}^{N}(\tilde{y}_n - y_n)^2.$$

The square root of MSE is called the *Root Mean Square Error* (RMSE) and provides results with the same unit of measure of the target. Instead of the square, by averaging the absolute value of errors $|\tilde{y}_n - y_n|$, we have the *Mean Absolute Error*

(MAE) or the *Mean Absolute Percentage Error* (MAPE), when the absolute value is applied to the fraction of the error over the expected target value $\left|\frac{\tilde{y}_n - y_n}{y_n}\right|$. This latter measure is useful when the scale of the different observations significantly changes.

In the context of the classification problem, the fundamental evaluation metric is the *confusion matrix*[2], which collects the number of right and wrong previsions computed by comparing the predictions of the model with the true labels. In particular, for binary classification, the matrix reports in the diagonal the number of true positives (TP) and true negatives (TN), whereas the extra-diagonal elements collect the number of false positives (FP), i.e., type I error, and the number of false negatives (FN), i.e., type II error.

The *accuracy* score is an aggregated value of the confusion matrix given by the sum of diagonal terms (TP and TN) divided by the sum of all the matrix values:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

This specifies how many instances are correctly classified with respect to the total. Nevertheless, the mere accuracy indicator can be misleading, especially with an unbalanced dataset when the predicted class is the predominant one. The *precision* score measures the effectiveness of our estimator in finding the positive cases avoiding false positives:

$$Precision = \frac{TP}{TP + FP}.$$

Instead, the *recall* score (also called sensitivity) measures the goodness of the estimator in discovering the positive cases by taking into account possible errors in false negatives[3]:

$$Recall = \frac{TP}{TP + FN}.$$

The *F-beta* scores are defined as the weighted harmonic mean between precision and recall:

$$F_\beta = (1 + \beta^2)\, \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall},$$

where if $\beta > 1$, the indicator becomes more recall-oriented, and when $\beta < 1$, it becomes similar to precision. When $\beta = 1$, the so-called *F1 scores* provide a balance between precision and recall by considering their harmonic mean.

The aforementioned measures are particularly useful when TP matters more than TN. On the other hand, when TN is more relevant, the *specificity*, i.e., the equivalent of precision with TN, can be evaluated, as well as *fallout* (false positive rate), which is computed as a fraction of FP (false alarm) over the total negative ($FP + TN$).

The *receiver operating characteristic curve (ROC)* is a graphical plot representing the set of points given by the two coordinates ($fallout$, $recall$) as the discriminator threshold, usually fixed at 0.5, varies from 0 to 1. In this plot, the random guess model is represented as the bisector. Similarly to the ROC, the *precision–recall curve (PRC)* plots the points ($recall$, $precision$). For both the plots, the area under

[2]The name came from the fact that it makes easy to see if there is a mislabelling problem.

[3]Usually important for examples in diagnosis. The name is derived from the recall of FN patients that were outside the hospital.

the curve (denoted as ROC-AUC, and PRC-AUC) measures the effectiveness of the indicators, where random guess is $AUC = 0.5$.

Once the performance metrics have been defined, the stability of the results should be evaluated. Multiple tests or cross-validation strategies (see [47]) ensure the performances' reliability. Eventually, a cost matrix could be taken into account for differently weighting the wrong predictions (see, for example, cost-sensitive learning [48]).

1.5.2 Model Relevance

The model capacity is a measure of the richness of the functional space where the predictive function is drawn by the algorithm. In general, to avoid both overfitting and instability issues, the algorithm should be no more complex than necessary for the task entrusted to it. Model relevance (R) is defined as the difference between the capacity (dented with C') of a specific model (algorithm A' on the dataset D) and the simplest possible capacity C of an algorithm A with the dataset D:

$$R = |C'(A', D) - C(A, D)|.$$

When the relevance is too high, the model is likely to be subject to overfitting.

There are different approaches for the evaluation of the model's relevance. The simplest way to measure the complexity is through the number of model parameters to be optimized. Another proxy is the training time. A theoretical measure of model capacity is the VC-dimension [49], i.e., the cardinality of the largest set of points that the algorithm can shatter. It is computed by counting how many points, in every possible configuration, a linear boundary can correctly classify. For instance, in a one-dimensional boundary, only two points are always separable by a threshold point. Another way to measure capacity [50] is to train the model with randomized labels: the idea is that if the model is able to learn the random input/output pairs, the capacity is high.

The most evident effect, which is potentially dangerous, of an overcomplex model is overfitting. The correctness of both the training and validation datasets does not give clean information about the forecasting ability of the model, but can be useful to evaluate the possible overfitting/underfitting by evaluating the gap between train and validation performances. Cross-validation is again a good way to measure overfitting.

In order to avoid overfitting, various regularization techniques can be tested during the training phase. For instance, the well-known Least Absolute Shrinkage and Selection Operator (LASSO) regularization (see [51]) introduces the sum of the absolute values of the model parameters as a penalization term. This regularization has the effect of shrinking the estimated coefficients toward zero, providing a feature selection method for sparse solutions within the classification tasks. Other techniques are random dropout [52], l2 penalty and elastic penalty [53], batch normalization [54], etc.

The list of requirements for the performance assessment is reported in Table 1.8.

Table 1.8: Checklist for performance requirements.

Code	Check
F53	Are the metrics properly chosen, sufficient and effective? Are they adequate for imbalanced problems?
F54	Is there any data leak? Is the test really unseen or is feature selection/oversampling performed on the overall dataset?
F55	Are the testing samples enough and the performance stable? Did you try multiple tests or multiple cross-validation?
F56	Did you compare the results with the H0 distribution (shuffling of the target)?
F57	Are the performances adequate with respect to the goal, the competitor's algorithm, and the baseline algorithm (random guess, naive Bayes, dummy models)?
F58	Does the model quantify the error of the prediction?
F59	Is any cost matrix considered?
F60	Did you quantify the model relevance (e.g., training time, number of parameters, VC dimension, etc.)?
F61	Did you evaluate overfitting?

1.6 Reliability requirements

The trustworthiness of a Machine Learning (ML) model, i.e., the stability of its performances under many possible scenarios, is a fundamental requirement for day to day applications. In fact, in a medium–long-term view, a reliable model does not significantly lower its performance. Model trustworthiness is usually linked to other factors, including the interpretability of the algorithm, the stationary of data, and the lack of bias in data ([5], [4]). Especially in the field of interpretability, many studies have been conducted in order to explain and interpret the ML models in a human-comprehensible manner. The main reason behind these efforts is that the human experience and capacity for abstraction allow the monitoring of the process of the model decisions in a sound way, attempting to mitigate the risk of data-driven models.[4]

Visualizing methods for sensitivity analysis

In the context of ML, a set of sensitivity analyses can be carried out by representing in a plot, a projection or an aggregated value of the learning function chosen by the model in the hypothesis space[5].

Let us suppose we are interested in the feature i and we want to represent the learning function as a function of it. The *Individual Conditional Expectation*

[4]The decision-making process is also strictly related to the concept of responsibility that cannot be delegated to the algorithms.

[5]Functional space from feature space to the target space.

(ICE) plot, described in [55], considers an instance at one time, manually varies the feature i we want to visualize by keeping the other feature values fixed, and obtains a single line plot of all the predictions (red line in Figure 1.2 for the cost house task[6]. The chosen model for the example is a two hidden layer Neural Network.). Mathematically, let us consider the n-th instance, $\boldsymbol{x}^{(n)} = [x_i^{(n)}, \boldsymbol{x}_{D/i}^{(n)}]$, where D represents the overall set of features. We aim to plot the function depending on the i-th feature defined as follows:

$$ICE^{(n)}(x_i) = \tilde{f}([x_i, \boldsymbol{x}_{D/i}^{(n)}]),$$

where the other $|D/i|$ features are fixed and taken from the base instance $\boldsymbol{x}^{(n)}$. Similar paths can be found for other instances, and with N samples, we have N different plots (see Figure 1.2). In this way, we take into account possible heterogeneous dependencies of the same feature for different instances.

The average of ICE plots is called the *Partial Dependence Plot* (PDP), represented as a black line in Figure 1.2. Basically, for a single value of feature i, we aggregated the effect of all the other features by an expectation of the target over the marginal distribution of the other feature.

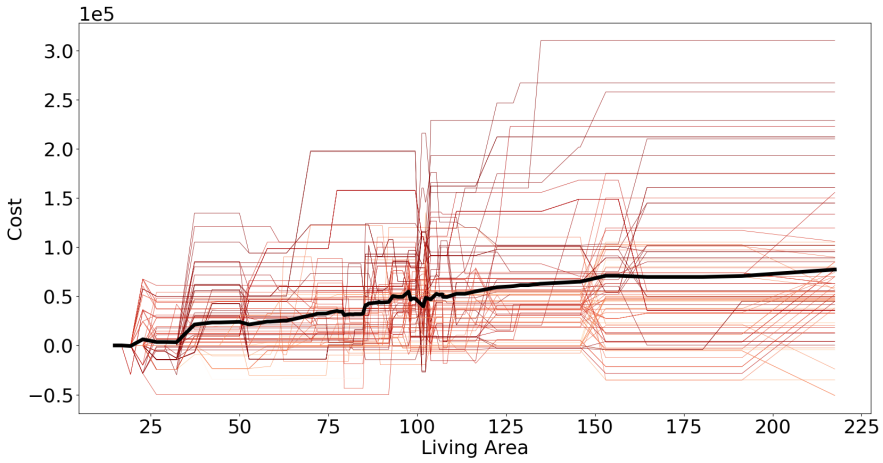


Figure 1.2: ICE plots: each red line is obtained for a single instance (house) by changing the living area feature value, and plotting the changing predicted cost of the house. With a black line, we represent the PDP. Basically, for the point with fifty square meters of living area, we average the cost of all houses by substituting 50 square meters as their living area.

The PDP problem of creating artificial, potentially unrealistic instances can be solved by *M plots* (see [57]) using only the instances with feature values close to the one we are considering. Mathematically, this means using the conditional distribution instead of the marginal one in order to compute the expectation. Anyway, the risk is to be too closely linked to the empirical joint distribution, in the sense that, if there are correlated features, with the conditional distribution we move in the data

[6]The cost house task is a regression problem aims at predicting the cost of the house with 79 explanatory variables describing the aspect of residential homes in Ames, see Kaggle competition [56]

taking the dependence hidden in the joint distribution, instead of that of the actual learned function. *ALE plot* [57] tries to solve the problem by computing variations in the predictions instead of using the prediction level itself.

Finally, other visualization techniques are designed to represent the images in the hidden layer of a deep NN; see [58].

Analysis of Stability

Most of the nonperforming ML models in the production environment are due to unreliable or unstable results of their performances metrics. Therefore, it is very important to monitor the stability of the testing results, the hyperparameters choice, and the fitting procedure with various datasets. If possible, the quantification of the error of each measure should be carried out. For example, *Conformal Prediction* (CP) is a framework to introduce a measure of confidence for the predictions produced by any traditional Machine Learning algorithm [59].

The more an ML model is able to maintain performance in the presence of noise, the more it is robust. Analyses should be carried out in order to evaluate the reliability of the models under different possible scenarios, for instance, by the techniques of adversarial example topics, thoroughly analysed in section 1.7.3. In [60], the idea is to define key risk indicators (KRIs) by generating scenarios, such as random input corrupted data or adversarial perturbations, and estimate the possible damage, by considering the likelihood of each scenario.

Error analysis test on sub-population

After leading tests on unseen samples, it is important to ask ourselves why the samples are either misclassified or correctly classified. This phase can be tedious and time-consuming but could actually be crucial to find out possible errors and eventually understand the mechanism on the basis of the ML model. In classification, you should mostly focus on false negative/positive, depending on which error is the most harmful for the task.

The test performances are usually measured as aggregated values, and this can generate misleading results. For instance, the performances of a specific slice of data can be very different with respect to the total. In general, a positive trait of an ML model is the ability to maintain its performance level in different sub-populations (e.g., defined by demographics features) or different time ranges. A possible test is to slice the data to see if the model works differently across subgroups, or to check the consistency of a prediction over time.

Metamorphic testing [34] verifies the properties that the predictions should have. For instance, if two instances are the same but the second is older, the dead probability needs to be higher.

Benchmark tests

Depending on the business area, regulators can be required to perform benchmark tests to verify the models' robustness across companies, e.g., the European Central Bank (ECB) targeted review of internal models (TRIM) in the banking area [61]. In the ML context, benchmark tests could focus on how the model works with

standardized artificial instances, e.g., for loan approval models, to see if the credit is granted to a person with particular features.

Explainability methods

Data-driven models with a large number of parameters provide non-unique and nonlinear solutions whose mathematical description is difficult to understand for humans. A fundamental challenge of ML is to explain, in a human-comprehensible manner, the working of a black-box model. This topic is actively debated [62], as it deeply involves the concept of knowledge itself. There are currently many different approaches to tackle the explainability problem, e.g., LIME and SHAP; in [63, 64, 65], a review of the existing methodologies with a particular focus on deep learning model interpretability is provided. In [63], the authors classify the methodologies in the following categories:

1. gain insight into the model's functions (*model inspection*);

2. understand how the code is exploiting the features to make a decision in a particular example (*local interpretability* based on feature importance);

3. understand how a black box machine learning model works (*global interpretability*);

4. construct self-explanatory models, which incorporate interpretability by -design into their structures (*transparent design* or intrinsic explainability).

Due to the advantage of being model-agnostic, the most used methods are those addressing local interpretability through feature importance. For a tabular dataset, the feature importance is usually represented as a rank in a histogram reporting how important each feature is for the specific prediction. As an example, in the case of LogR and linear SVC models, the importance of the features is automatically given by the coefficients of the fitted model. For images or texts, the subset of the input that is mostly in charge of the predictions gives rise to saliency maps (e.g., parts of the image or sentences of a text).

Among the different model-agnostic local interpretable methods, *permutation feature importance* methods quantify the feature importance through the variation of a loss metric, by permuting the values of a selected feature on a set of instances in the training or validation set. The approach was firstly introduced in [66] for random forest and in [67] for neural networks. Other methods, such as *class model visualization* [68], compute the partial derivative of the score function with respect to the input, and [69] introduce expert distribution for the input giving *activation maximization*. In [70], the authors introduce *deep lift*, which computes the discrete gradients with respect to a baseline instance, by backpropagating the scoring difference through each unit. *Integrated gradients* [71] cumulate the gradients with respect to inputs along the path from a given baseline to the instance. Finally, a set of well-known methods called *additive feature attribution methods* (AFAM) defined in [72] rely on the redistribution of the predicted value $\tilde{f}(\boldsymbol{x})$ over the d input features. They are designed to mimic the behaviour of a predictive function $\tilde{f}$ with a

surrogate Boolean linear function g. This surrogate function takes values in a space of the transformed vector of the input features: $\boldsymbol{x}' = h(\boldsymbol{x}) \in [0,1]^d$:

$$\tilde{f}(\boldsymbol{x}) \approx g(\boldsymbol{x}') = \phi_0 + \sum_{i=1}^{d} \phi_i \, x_i'.$$

Among the additive feature attribution methods, the popular Local Interpretable Model-agnostic Explanations (LIME) [4] builds the linear approximated model with a sampling procedure in the neighbourhood of the specific point. By considering proper weights to the linear coefficients of LIME, the author in [72] demonstrated that SHapley Additive exPlanation (SHAP)) is the unique solution of additive feature attribution methods with a set of desirable properties (local accuracy, missingness, and consistency). This last method relies on the *Shapley* method introduced in [73] and [74] for solving the problem of redistributing a reward (prediction) to a set of player features in the coalitional game theory framework. It is worth mentioning that the notion of Shapley value (whose theoretical meaning is treated in the Appendix 5) has been exploited by the author for the allocation problem of an overall risk value in the risk management framework, for the publication of [75]. Finally, *Layer-wise Relevance Propagation* (LRP) in [3] backpropagates the prediction along the network, by fixing a redistribution rule based on the weights among the neurons. Since LRP will be useful in Chapter 2, in Appendix 5 we detail the methodology starting from the original paper [3].

The set of feature importance methods, along with the data type (TAB: Tabular, IMG: Image) and the model to which the method refers (AGN: Agnostic, NN: Neural Network), are reported in Table 1.9.

Table 1.9: Feature importance methods.

Method	**Data Type**	**Model**	**Reference**	**AFAM**
SHAP	ANY	AGN	[72]	v
LIME	ANY	AGN	[4]	v
Shapley value	TAB	AGN	[73] [74]	v
Permutation feature importance	ANY	NN	[66] [67]	-
class model visualization	IMG	NN	[68]	-
activation maximization	IMG	NN	[69]	-
LRP	ANY	NN	[3]	v
Taylor Decomposition	ANY	NN	[3]	-
DeepLift	ANY	NN	[70]	v
Integrated Gradients	ANY	NN	[71]	-
GAM	ANY	AGN	[76] [77]	-

It is important to mention that most of the aforementioned feature importance methods rely on the hypothesis that there is no correlation among features. A good practice is to verify to what extent this assumption is satisfied and how much the computation of feature importance is affected; see [58, 78].

Concept drift and systemic risk

In addition to the weakness of the model's architecture, a general issue of ML modeling is the so-called *concept drift*. The concept shift is the uncertain change in the statistical properties of the target variable, which the model tries to predict. Concept shift causes a degradation of the performances over time, and continuous evaluation of the model's performances is, therefore, needed. A possible proxy is the continuous evaluation of key input values (KIDs), e.g., macroeconomic factors or external events, to consider the need for recalibration.

Another possible risk is systemic risk, due to a common calibration of the same ML model for different actors.

The list of requirements for the reliability and explainability requirements is reported in Table 1.10.

Table 1.10: Checklist for reliability requirements.

Code	Check
F62	Did you perform sensitivity analyses (ICE, PDP, M-plots, ALE plot, etc.)?
F63	Did you evaluate the limit case (e.g., extreme observations)?
F64	Did you perform "error analysis" on an individual case where the model was wrong?
F65	Did you perform metamorphic testing?
F66	Is there any measurement of confidence level of the model's predictions?
F67	Did you perform tests on a critical sub-population?
F68	Did you consider a set of benchmark models?
F69	How does the model work with the regulatory benchmark tests?
F70	Is the model explainable in a local/global way?
F71	Did you apply any explainable technique? To what extent is the assumption of feature independence satisfied?
F72	Is a maintaining process required?
F73	How are potential/future missing data treated?
F74	Did you consider concept drift?
F75	Are the limitations of the model discussed?

1.7 Policy requirements

In this section, we report the considerations related to the policy requirements of a Machine Learning (ML) model. The regulation concepts defined in [7] and [22] are relatively abstract: ensure a high-quality dataset; complete documentation; a high degree of information for the user; human oversight; and architecture that is robust, accurate and resilient to cyber-attacks. The regulation [7] applies to any model, but only a few have a high risk of violating fundamental rights. For instance, regulation is relatively restrictive with respect to activities related to video surveillance, social scoring, and any kind of business applied to fragile individuals, e.g., children. It is not the purpose of this work to provide an exhaustive view of the ethics guidelines, but we focus more on the practical issues, considering that difficulties arise when an attempt is made to concretely implement the policy. In [79], a self-assessment list is reported. It is worth noting that, in addition to the external regulation, private companies should follow their internal policy.

1.7.1 Accountability and transparency

A basic potential problem might be to identify if a specific model belongs to the category of ML. The classic definition by Mitchell in [80], "*Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience*", is quite general, does not have boundaries that are easy to identify, and states what is inside and what is outside the ML framework. Comprehensive model inventories should include all ML-based models in use by the company and can help the phase of model categorization.

Once it has been established that the model falls inside the ML framework, we need to identify and properly describe the problem that the project will tackle.[7] After the phase of model definition, it is important to identify the perimeter of application of the model's predictions and identify expected outcomes and possible alternative results.

Chain of responsibility

The business units and the chain of control having the responsibility in the ML process should be clearly identifiable, starting from the top management approving the decisions on model usage, the compliance/legal department that deals with ethical issues and the operative units that design and implement the models along with the information technology teams. During the life cycle, when an issue arises, the chain of responsibility should be assessed.

Documentation

The documentation is fundamental for the transparency of the model: it should be exhaustive and self-explanatory to allow the model to be replicated by a third party. The documentation should be clear and accessible to a general user. The code, in particular, should be documented; the usage of software for automatic documentation is an effective solution for avoiding operational errors. Additionally, previous

[7] The project starts with a question, not applying a favored technique. In fact, the regulation [7] concerns AI usage and activity, not AI research.

versions of models should be stored and accessible. If the model is transparent, the developers can more easily maintain and debug the code, whereas end users can better understand the results.

The list of requirements related to the accountability and transparency of an ML model is reported in Table 1.11.

Table 1.11: Checklist for accountability requirements.

P76	Did you track the model in the institution's model inventory?
P77	What is the problem to solve and why does the problem need to be solved? Are you able to describe it informally? Are the goal and context clearly stated?
P78	Do you have domain knowledge on the problem? In which manner would a human (expert or not) solve this problem?
P79	Are there any similar or related problems? Are they already solved? Did you use a pre-trained model?
P80	Did you pinpoint the spatial (geographical area) and temporal (deadline) scale of the project?
P81	Who is the target population?
P82	Is there an exit strategy when prediction fails?
P83	Did the model require human interaction (humans in the loop)?
P84	Have you considered financial availability for model maintenance?
P85	What is the chain of command? Who has responsibility for errors or wrong decisions?
P86	Did you periodically report the problem, solution, findings, and limitations?
P87	Did you identify the restraints of the project?
P88	Is the validation team independent and adequately skilled?
P89	Is the documentation complete, accessible, and transparent?

1.7.2 Fairness and Ethic

The requirement of model fairness is quite different with respect to those seen previously, as it involves external information we cannot find in data. While an algorithm might appear to be fair by default, since it tries to represent the dataset as smoothly as possible, the overall model can be actually unfair due to possible partiality in input data. Basically, a dataset can be unfair due to (1) a bias in the sampling procedure (specific classification of fairness biases is presented in [81]). For example, if it is easier to harvest data from a male rather than from female in a specific problem, there will be an unbalanced representation of the target variable. Furthermore, even a dataset harvested with fair methods could represent a (2) unfair world. It

is evident that the amount of discrimination, racism, and xenophobia, especially on the basis of protective attributes, can be inherently represented in data. Since the danger is that the increase in decisions carried out by automated ML algorithms amplifies the unfairness, the regulation [79] states that the models should work to reduce discrimination. Therefore, if the dataset is actually unfair, data scientists have to introduce some sort of bias to dilute the unfairness.

Fairness metrics

The unfairness is usually associated with the so-called protected feature: age, disability, gender reassignment, marriage and civil partnership, pregnancy and maternity, race, religion or belief, sex, sexual orientation, political belief, etc. In general, these features cannot be used to drive the decision of the model; even if, as shown in cross-cultural studies [82], the ways in which people approach ethical issues are quite different from country to country, and the list cannot be exhaustive.

Reviews of fairness metrics can be found in [25] and [83]. The *demographic parity* (DP)[8] fairness metric, is satisfied when, given the random variable $\tilde{Y}$ representing the model predictor $\tilde{f}(\boldsymbol{x})$ and a protected feature X_s, we have:

$$P(\tilde{Y}|X_s = 0) = P(\tilde{Y}|X_s = 1).$$

Basically, $\tilde{Y}$ needs to be independent of X_s. DP is a very strong requirement; groups based on sensitive features, e.g., black and white, should have the same rate of positive prediction (equal acceptance rate), even if differences are present. By denoting $P_0 = P(\tilde{Y}|X_s = 0)$ and $P_1 = P(\tilde{Y}|X_s = 1)$, the measures of discrepancy from DP are *disparate impact* $\frac{P_0}{P_1}$ (close to 1 when DP is satisfied) and *statistical parity difference* $|P_0 - P_1|$ (close to 0 when DP is satisfied), with the following two straightforward relaxations when the favourite class is 1: $|P_0 - P_1| \leq \epsilon$ and $\frac{P_0}{P_1} \geq 1 - \epsilon$, where ϵ is the chosen tolerance.

A possible relaxation of DP in assistive binary classification models[9] is where the protective attribute X_s can be used to discriminate among groups that are actually different in the ground truth label (so between $Y = 0$ and $Y = 1$), but not within each one. This is called *equalized odd* and is described in [84]. A predictor satisfies the equalized odd if $\tilde{Y}$ and X_s are independent conditional on Y:

$$P(\tilde{Y} = 1|X_s = 0; Y = 1) = P(\tilde{Y} = 1|X_s = 1; Y = 1),$$

$$P(\tilde{Y} = 1|X_s = 0; Y = 0) = P(\tilde{Y} = 1|X_s = 1; Y = 0).$$

Basically, the equalized odd allows dependence on X_s in an aggregate manner, but requires the independence in each single group built with ground-truth labels.

In [84], when the application is assistive, the notion of *equal opportunity* (EOP) is proposed, which consists of relaxing the equalized odd to only deal with the true positive rate parity. A predictor $\tilde{Y}$ satisfies EOP if

$$P(\tilde{Y} = 1|X_s = 0; Y = 1) = P(\tilde{Y} = 1|X_s = 1; Y = 1),$$

[8] also known as independence or statistical parity.

[9] Assistive models are those where $\tilde{Y} = 1$, i.e., the class of positive, is more relevant with respect to $\tilde{Y} = 0$.

namely, if $\tilde{Y}$ is independent from X_s, depending on Y true. Basically, the prediction needs to be independent from X_s in the group of true positives that are usually advantaged.

Another metric is discussed in [85], where they introduce SHAP as a measure of fairness based on the Shapley value.

Finally, the concept of *individual fairness* (IF) is that the algorithm is fair if it gives similar predictions to similar individuals [86]. Basically, given a metric $d(\cdot, \cdot)$, if individuals n and j are similar under this metric, then their predictions should be similar. We can compare the prediction of the instance $\boldsymbol{x}_n$ with the average predictions of the K instances in the neighbourhood, and sum this difference for many n instances:

$$\sum_n \left| \tilde{f}(\boldsymbol{x}_n) - \frac{1}{K} \sum_{j \in kNN(\boldsymbol{x}_n)} \tilde{f}(\boldsymbol{x}_j) \right|.$$

Strategies for fairness

Fairness Through Unawareness, introduced in [87], provides that any protected attributes x_s should not be explicitly used in the decision-making process. Experts should state which features are protected. Furthermore, [88] explains why unawareness can be a palliative as correlated features can share the same information of protected features. Additionally, [89] showed that unawareness is not sufficient for the fairness scope. Other methods, such as *massaging* and *reweighting* aim at modifying either the dataset or the algorithm functionalities, and are discussed in [90]. The Python package AI Fairness 360 includes the most important fairness metrics.

In the context of fairness, it is important to mention the concept of spurious correlation between two variables that are associated by correlation, e.g., political belief and low credit rating, but where there are no causalities. This association could occur due to the presence of a third factor, e.g., annual income, (the so-called confounder) which causes both the first and the second variables. If we add the confounders as a feature, we do not need the protected feature (political belief). Sometimes, the solution for a fair model is to use the confounding factor as a new feature.

The list of requirements in the fairness and ethic framework is reported in Table 1.12.

1.7.3 Safety

An ML model is considered to be safe if it is able to adequately defend itself against improper use, manipulation, and other potential damage to its components. There is extensive literature on measures of model robustness, mainly based on injecting noises into data with the scope to alter the performances (see [91, 92, 93, 94]), and the present work does not attempt to fully cover it.

The main two dangers to deal with are *adversarial example* and *data poisoning*. Adversarial examples are slightly perturbed inputs that are misclassified with high confidence. A common definition of adversarial robustness is based on the smallest perturbation necessary to produce an incorrect classification [95]. It is well known that NNs are sensitive to adversarial perturbations [96]. Methodologies designed to

Table 1.12: Checklist for fairness and ethics requirements.

Code	Check
P90	Did you evaluate the potential impact of the model on fundamental rights?
P91	Does the model interact with humans? Is the human informed and aware? Are there any possible negative influences of the interaction?
P92	Did you put in place measures to grant human oversight of the model activities?
P93	Are there any protected features?
P94	Have you defined adequate fairness measures according to whether the model is assistive or punitive?
P95	Does the model satisfy the fairness measures?

mitigate and detect adversarial attacks are usually bypassed by new, more recent techniques. Therefore, the process is dynamic, and there are no general defensive strategies, but rather special methods for different types of attack [97]. The second kind of potential attack, data poisoning, is a modification of the training data in order to change the predictive behavior [98]. It is worth noting that a reliable model, as shown in section 1.6, is more resilient to adversarial attacks or data poisoning.

The list of requirements in the safety area is reported in Table 1.13.

Table 1.13: Checklist for safety requirements.

Code	Check
P96	In the physical infrastructure prepared to face potential cyber attack?
P97	Is the model safe against adversarial attack?
P98	Is the model resilient to data poisoning?
P99	Did you put in place fall-back strategies?

1.7.4 Privacy

In general, an ML model satisfies the privacy requirement when any information related to an identifiable natural person included in the data is preserved. The most relevant regulation in the European Union (EU) of data privacy is the General Data Protection Regulation (GDPR) [7], approved by the European Parliament in May 2018. In the United States, the main statute is the California Consumer Privacy Act (CCPA) [99] for residents in California. The main drivers for GDPR regulation are the legal constraints, the individual rights of persons, the protection of how their personal data are used by organizations, the transparency of the automated

decisions, and the definition of economic penalties for noncompliance. Particular caution is given to sensitive information.

Whenever possible, the personal data of data subjects (those people about whom you hold personal data) should be made anonymous. Nevertheless, even if data are anonymized by removing personally identifiable information, privacy could not be granted, since, by linking the different datasets, individual information can still be leaked. The field of differential privacy [100] aims to ensure that no significant risk is incurred by joining statistical databases. The concept of differential privacy is that one instance (e.g., $\mathbf{x}_i$) more in the dataset B, with respect to the dataset A, should not imply that the learner could obtain much more information from B with respect to A. If this is granted, the individual data of $\mathbf{x}_i$ is demonstrated to not have a significant effect on the outcome. Differential privacy violations can be assessed via statistical tests (see [101]).

In Table 1.14, we report a list of requirements for the assessment of privacy requisites.

Table 1.14: Checklist for privacy requirements.

Code	**Check**
P100	What kind of data are being collected and stored, where and why?
P101	Which data are identifiable to a person (phone number, home address. . . .)? How many individuals are there in the database?
P102	Which data are special/sensitive personal data (genetic, health condition, etc.)? Do they require explicit consent?
P103	What do you intend to do with data (both internally and externally) and why do you need the data?
P104	How long will you store the personal data?
P105	Did you give to the person the right to access (see which data are being stored about them), rectify (make corrections), or erase (ask for their data to be deleted) data?
P106	Is it possible to anonymize, encrypt or aggregate personal data?
P107	Who has access to data (either inside or outside the business)? Did you put in place strategies for data access control? Do you have an intrusion detective policy?
P108	Is there an oversight mechanism for data collection, storage, processing and use?
P109	Which procedures and controls are in place to keep data safe?
P110	Is GDPR respected? Who is the Data Privacy Officer (DPO)?

1.8 A glance into changing the validation process

From a theoretical point of view, as shown in [102] and [103], the ML approach to solve problems is quite different and, in some ways, more complex with respect to the classical statistical one. In fact, the ML algorithm is strongly dependent on the information included in the input dataset, whose effect on the model parametrization is difficult to assess and interpret. The consequence is that testing ML reliability is generally a more challenging process.

In the ML ecosystem, more than in the standard validation process, the focus should be on the representativeness of the data, as they fully determine the model design (see Section 1.3). The increased effort in data assessment compensates, in a certain way, for the impossibility of carrying statistical tests, usually perform to verify a model's hypotheses. As shown in Section 1.3, a step-by-step analysis, especially for the data cleaning and the feature engineering phases, should be carried out, as well as a punctual check of the feature meaning. Additionally, the verification of feature selection and the feature engineering process assumes a higher relevance.

Traditionally, statistical models were calibrated and periodically re-calibrated and/or revised. In ML, if the dataset is not updated, the risk is that the model can be affected by short-term patterns that deteriorate the performances in a medium/long-term view. Therefore, a dynamical (more frequent) validation is essential. In the case that the dataset is updated with new information, a new fitting or a fine-tuning of the model may be required to reflect emerging patterns in the data; even in this case, the validation phase must be re-performed, at least partially.

The standard back-testing analyses usually performed to verify the performance of statistical model becomes, in the ML framework, a part of the testing phase; see Section 1.5. Back-testing assumes more relevance, since the conceptual soundness of the model is no more an element to prove its validity: the testing is the only way to verify the model's correctness from the performance side. Stress test methodologies in the context of ML models can be performed in different ways, for instance, by the adversarial example topic. In addition, validation shall consider possible unfairness, usually hidden in data (see Section 1.7).

Finally, an effort by the validation team needs to be made to evaluate the volume of input/output data and the architecture (CPD/GPU and RAM) required for the model.

1.8.1 Validation process and case study

In general, the model risk should be evaluated on the basis of the model complexity, the need for re-calibration, the amount of data, or compliance risk. It should be considered that often, the company policies for adopting a model are based on the model interpretability and its Risk Appetite Framework (RAF). Obviously, more risky models should be carefully evaluated, also with a continuous process of risk identification. A careful monitoring of the model should also take into account non-modeling factors, which are not included in the model but can affect the adequacy of the decisions.

All the internal models should be subject to an initial, and subsequently to a periodic, validation. To ensure the effective independence of the team from the model development process (i.e., model design, development, implementation, and moni-

toring), institutions should have appropriate organizational arrangements in place. The validation team should be qualified with both theoretical and practical experience. Nevertheless, it may happen that validation teams in the banking industry have poor application knowledge of ML techniques.

As a case study, we evaluated the possible model risk of the most important ML models (linear regression, generalized linear model, support vector machine, decision tree, random forest, and neural network) by quantifying how difficult the requirements were to validate in a hypothetical financial application, e.g., the loan approval task. The results are reported in the radar charts of Figure 1.3. The score for each of the components, i.e., dataset, algorithm, performance, reliability, and policy, range from 0 to 5, where 5 indicates the highest difficulties. We qualitatively attributed the scores based on the estimated difficulties in the fulfillment of the requirements reported in the previous sections.

Roughly speaking, the higher the model complexity, the higher the difficulties in managing the reliability and policy requirements. More generally, a possible way to measure the potential percentage of model risk in an ML model is by answering the 110 questions reported in the present work and assigning a fulfilment score to each one.

Figure 1.3: Radar charts representing the hypothetical effort of the validation process for different common ML models: Linear Regression, Generalized Linear Model, Support Vector Machine, Decision Tree, Random Forest, and Neural Network. The evaluated components are the dataset, the algorithm, the performance, the reliability, and the policy.

Last but not least, in the phases of both deployment and assessment of an ML application, it might be useful to consider the golden rules reported in Table 1.15. They are theoretical principles resulting from the abstraction of practical issues derived from experience in the AI field and from a literature search.

Table 1.15: Golden rules for the setting up of a trustworthy model.

1	**Do not touch the test (please)**. No information can be extracted from the test set, which needs to include only unseen samples. This rule also needs to be satisfied in the feature engineering phase. Be vigilant of surprising performances.
2	**When the model cannot speak, the model must be silent: be aware of randomness**. A common error is to take a noisy result as a good or bad result. Only by considering statistical errors will the experimenter not find patterns in noise. This problem is related to overfitting but is more general. For this reason, every estimator should have the notion of confidence.
3	**Models should be made as simple as possible, but no simpler**. Bearing in mind the task to solve, the algorithm should not be more complex than necessary. This problem is also known as "premature optimization". Note that a simple linear model is usually more interpretable.
4	Instead of tackling a complex problem with a complex model, try to consider the **simplification of the task**, e.g., by transforming the regression problem into a binary classification one. This is perhaps not the ultimate goal, but it would be preferable.
5	**Do not change the architecture to solve "local" issues**. You should bear in mind an overall and long-term advantage rather than a local and immediate benefit, e.g., a small improvement in a performance measure.
6	An ML model is a **compromise between data and hypothesis**. Consider the benefits of adhering to the data and the level of reliance on the hypothesis. Not relying on data when your hypotheses are strong is not a bad choice.
7	Implementing and coding with **best practice and gold standard** helps to avoid errors in the model deployment phase with a reduction in human risk factors. In sum, is quite often a good idea: standardizing the processes means repeatability and a gradual improvement in the model over time.

Chapter 2

Novel methodologies for Explainable Machine Learning

After seeing the practical matters of trustworthy Machine Learning (ML) discussed in the previous chapter, here we present two theoretical methodologies developed in the ML explainability field during the Ph.D.

The first one is devoted to the Neural Network algorithm with Rectified Linear Units, where we recognize that the non-linear behavior of the activation function gives rise to a natural clustering useful for the explainability purposes. Instead, the second methodology aims at exploiting the human prior knowledge of the features' importance for a specific task, in order to coherently aid the phase of the model's fitting. Both the methodologies exploit, in a novel way, the well-known concept of feature importance, which aims at assigning a score to input features based on how useful they are at predicting a target.

2.1 Clustering-Based Interpretation of Deep ReLU Network

There is no doubt that the recent developments of deep neural networks offer enormous progress in artificial intelligence in various sectors. In particular, the Rectified Linear Units (ReLU) functions have been shown to mitigate the vanishing gradient issue, encourage sparsity in the learned parameters and allow for efficient backpropagation [104]. Despite the benefits and the expressiveness of Rectifier Networks have been widely investigated, the cluster analysis and the consequent interpretation of the network via the modeling of the pattern of active neurons have not been discussed in the literature, to our knowledge. On the other hand, many post-hoc model-specific methodologies can be applied for gaining interpretability in neural network models (see [63]). Among them, the concept of features' importance is one of the most used strategies to gain local explainability from an opaque machine learning model [64]. Our main contribution is to provide an explainable method totally relying on the fitted structure of the network.

Relevant to our work are explorations of the roles of semantic concepts inside neural networks. In fact, the methodology we are here presenting is linked to other lines of research exploiting the connectionist paradigm, where the goal is to evaluate whether the network's structure is able to represent, within itself, a set of semantic

concepts. For example, in [105] the authors demonstrate that, in a network specifically designed for classifying scenes, individual units behave as object detectors without being explicitly trained with the notion of objects. In a similar way, in [106] the authors show that in a convolutional neural network the filters represent patterns that make sense to us visually, and help us to insect the input images.

Among the strategies aiming at extracting information from the network's structure, in [107] the authors quantify the interpretability of latent representations of CNNs, by exploiting a set of concepts drawn from a broad and dense segmentation dataset (Network Dissection). Similarly, [108] tries to interpret high-dimensional internal state of a neural network in terms of human-friendly concepts, by using the directional derivatives to quantify the degree to which a user-defined concept is important to a classification result. So, for instance, a typical problem is to assess how sensitive a prediction of an animal is to the presence of a particular texture, e.g. the stripes. Anyway, we underlying that the cited approaches need a predetermined set of semantic concepts; for instance giving labels across a range of objects, scenes, textures, colors, etc. Instead, the methodology we are proposing does not require to introduce apriori knowledge on the semantic concepts and the arising clusters can be seen as novel concepts.

2.1.1 Deep ReLU networks for the partition of the input space

In this section, we demonstrate that a deep ReLU neural network gives rise to a partition of the input dataset into a set of clusters, each one characterized by an affine map.

Let us denote by $W_i, \quad i \in [1, \dots, p]$ the weight matrices associated with the p layers of a given multilayer network with predictor $\tilde{f}$ (of a q-dim target variable) and collect[1] $\hat{W}_i$ in $\hat{W} = [\hat{W}_1, \dots, \hat{W}_p]$. For any input $\boldsymbol{u} \in \mathbb{R}^d$, the initial Directed Acyclic Graph (DAG) $\mathcal{G}$ of the deep network is reduced to $\mathcal{G}_{\boldsymbol{u}}$ which only keeps the units corresponding to active neurons[2] and the corresponding arcs. This DAG is clearly associated with a given set of weights $\hat{W}$. We can formally state this pruning for the given neural network, characterized by $\mathcal{G}$, paired with input $\boldsymbol{u}$ with weights $\hat{W}$ by

$$\mathcal{G}_{\boldsymbol{u}} = \gamma(\mathcal{G}, \hat{W}, \boldsymbol{u}),$$

where, since all neurons operate in "linear regime" (affine functions), as stated in the following, the output, the composition of affine functions, is in fact an affine function itself.

Theorem 1. *Let $\mathscr{X}_i \subset \mathbb{R}^{d_i}, \quad i \in [1, \dots, p]$ be, where $d_1 = d$. Let $\{h_1, \dots, h_p\}$ be a collection of affine functions, where*

$$h_i : \mathscr{X}_i \mapsto \mathscr{Y}_i : x \mapsto W_i\, x + \boldsymbol{b}_i = \hat{W}_i\, \hat{x},$$

and assume that $\mathscr{Y}_i$ is chosen in such a way that $\forall i = 1, \dots, p-1 : \quad \mathscr{X}_{i+1} \subset \mathscr{Y}_i$, whereas $\mathscr{Y}_p \subset \mathbb{R}^{d_q}$. Then we have that

$$\tilde{f}(\hat{W}, \boldsymbol{u}) = h_p \circ h_{p-1} \circ \cdots \circ h_2 \circ h_1(\boldsymbol{u})$$

[1] The $\hat{}$ in the notation means that the bias term is incorporated in the variable.

[2] A neuron is considered active for a particular pattern if the input falls in the right linear part of the domain's function.

is affine and we have $\tilde{f}(W, \boldsymbol{u}) = f(\hat{\Omega}, \hat{\boldsymbol{u}}) = \Omega\,\boldsymbol{u} + \boldsymbol{b}$, *where*[3]

$$\hat{\Omega} := [\Omega, \boldsymbol{b}] \tag{2.1}$$

$$\Omega(p) = \prod_{i=p}^{1} W_i \tag{2.2}$$

$$\boldsymbol{b}(p) = \sum_{i=1}^{p} \left(\prod_{t=p+1}^{i+1} W_t \right) \cdot \boldsymbol{b}_i \tag{2.3}$$

being $W_{p+1} := \mathbb{1}$.

Proof. The proof is given by induction on p.

- *Basis*: For $p = 1$ we have $\tilde{f} = W_1\,\boldsymbol{u} + \boldsymbol{b}_1$ and $\Omega(1) = W_1$ which confirms (2.2), and when considering $W_2 := \mathbb{1}$, we have

  $$\boldsymbol{b}(1) = W_2 \cdot \boldsymbol{b}_1 = \boldsymbol{b}_1,$$

  in according to (2.3).

- *Induction step*: By induction, a network with $p-1$ layers is defined by an affine transformation that is

  $$y(p-1) = \Omega(p-1)\,\boldsymbol{u} + \boldsymbol{b}(p-1).$$

  Hence

$$\begin{aligned}
y(p) &= W_p\, y(p-1) + \boldsymbol{b}_p \\
&= W_p(\Omega(p-1)\,\boldsymbol{u} + \boldsymbol{b}(p-1)) + \boldsymbol{b}_p \\
&= W_p\Big(\prod_{i=p-1}^{1} W_i\,\boldsymbol{u} + \sum_{i=1}^{p-1} \left(\prod_{t=p}^{i+1} W_t \right) \cdot \boldsymbol{b}_i \Big) + \boldsymbol{b}_p \\
&= W_p\Big(\prod_{i=p-1}^{1} W_i\,\boldsymbol{u} + \sum_{i=1}^{p-2} \left(\prod_{t=p-1}^{i+1} W_t \right) \cdot \boldsymbol{b}_i + \mathbb{1} \cdot \boldsymbol{b}_{p-1} \Big) + \boldsymbol{b}_p \\
&= \prod_{i=p}^{1} W_i\,\boldsymbol{u} + \sum_{i=1}^{p-2} \left(\prod_{t=p}^{i+1} W_t \right) \cdot \boldsymbol{b}_i + W_p\,\boldsymbol{b}_{p-1} + \boldsymbol{b}_p \\
&= \prod_{i=p}^{1} W_i\,\boldsymbol{u} + \sum_{i=1}^{p-1} \left(\prod_{t=p}^{i+1} W_t \right) \cdot \boldsymbol{b}_i + \mathbb{1} \cdot \boldsymbol{b}_p \\
&= \left(\prod_{i=p}^{1} W_i \right) \boldsymbol{u} + \sum_{i=1}^{p} \left(\prod_{t=p+1}^{i+1} W_t \right) \cdot \boldsymbol{b}_i
\end{aligned} \tag{2.4}$$

□

[3] We stress the dependence of Ω and $\boldsymbol{b}$ from the number of layer (p) since it will be useful for the proof.

Now let $\mathscr{U} \subset \mathbb{R}^d$ the input space. The given deep net yields a partition on $\mathscr{U}$ which is associated with the following equivalence relation:

$$\boldsymbol{u}_1 \sim \boldsymbol{u}_2 \leftrightarrow \mathcal{G}_{\boldsymbol{u}_1} = \mathcal{G}_{\boldsymbol{u}_2}$$

. We denote[4] by $[\boldsymbol{u}]_\sim = \{\boldsymbol{v} \in \mathscr{U} : \boldsymbol{v} \sim \boldsymbol{u}\}$ and by $\mathscr{U}/\sim$ the corresponding quotient set. Hence, $[\boldsymbol{u}]_\sim$ is the equivalent class associated with representer $\boldsymbol{u}$ which, in turn, corresponds with $\mathcal{G}_{\boldsymbol{u}}$. Notice that, as a consequence, $[\boldsymbol{u}]_\sim$ is fully defined by the set of neurons of the active neural network.

Feature Importance Explanation The characterization done above allows to assigning a matrix $\hat{\Omega}$ to each cluster of the network. In this way, the matrix is able to represent the network for the patterns of the specific cluster as an affine map.

For simplicity, if we consider a problem where the output of the network is scalar, the matrix Ω reduces to a d-dimensional effective vector $\boldsymbol{\omega}$ whose components can be interpreted as the *feature importance of the cluster's solution.*

It is important to note that the approach we are introducing can be applied to any feedforward networks where the activation function is the ReLU function. Anyway, the number of clusters highly depends on the number of network' weights.

2.1.2 Simulation study

In this section, we report the simulation studies carried out on the ReLU network architecture applied to a Boolean artificial dataset, in order to assess the power of the clustering-based interpretation.

We consider a set of 10 Boolean feature variables $v_{i\in[1,\dots,10]}$. The first 3 features determine the target variable through the following relation:

$$t = (v_1 \wedge v_3) \vee (v_2 \wedge \neg v_3) \tag{2.5}$$

whereas the other 7 features introduce noise. The rationale of the formula is that the feature v_3 split the data set in two groups (v_3 and $\neg v_3$), each ruled by a different term of the Boolean formula involving either v_1 or v_2 respectively. In the following, we investigate whether the network clustering is able to recognize the different terms of the formula.

We simulated $100,000$ samples, and we exploited a two-hidden-layer MLP with 4 and 2 neurons characterized by ReLU activation function and an output neuron with a sigmoid activation function. The cross-entropy loss is minimized via the Adam stochastic optimizer with a step size of 0.01 for 10 epochs, and a batch size of 100. An activity regularizer with 0.02 is added to the empirical loss. At the end of the training, the network solves the problem with an accuracy of 100%. The experiments are implemented with Keras in the Python environment on a regular CPU.

The analysis of the network, by considering the possible patterns of the active neurons, originates three clusters.

1. A trivial cluster characterized by all non-active neurons including all the patterns predicted as 0.

[4] In this work $[\cdot]$ is the Iverson's notation, whereas $[\cdot]_\sim$ is reserved to the equivalent class induced by equivalence relation $\sim$.

Instead, the other two clusters activate the two neurons of the last layer but a different neuron of the first hidden layer. In figure 2.1 we report the table resuming the 8 possibilities of the Boolean function restricted to the first 3 features, as well as the bar plot for the importance of the features for the two non-trivial clusters. As explained above, the importance of the feature is computed by the specific coefficient of the effective vector for that cluster.

2. The first relevant cluster, represented by the blue bars in Fig. 2.1, includes patterns predicted as 1 and characterized either by $v_1 = v_2 = v_3 = 1$ or by $v_1 = 1$, $v_2 = 0$, $v_3 = 1$. We can argue that this cluster takes in charge the patterns predicted as 1 due to the first term of Eq. (2.5), i.e. $v_1 \wedge v_3$. Coherently with this setting, the feature importance given by the effective vector is zero for the feature v_2 that does not appear in the first term of Eq. (2.5). On the other side, the coefficient of the effective vector is positive for the feature v_1.

3. In a similar way, the second cluster denoted with the orange color, represents the term $v_2 \wedge \neg v_3$ of Eq. (2.5) since the patterns belonging to it are predicted as 1 and are characterized either by $v_1 = v_2 = 1\, v_3 = 0$ or by $v_1 = 0\,, v_2 = 1\,, v_3 = 0$. As expected, the feature importance of v_1 is zero, whereas for v_2 the feature importance is positive.

Features			Target value	Clusters		
v_1	v_2	v_3	$(v_1 \wedge v_3) \vee (v_2 \wedge \neg v_3)$	I_1	I_2	I_3
1	1	1	1	> 0	0	> 0
1	0	1	1			
0	1	1	0			
0	0	1	0			
1	1	0	1	0	> 0	< 0
1	0	0	0			
0	1	0	1	0	> 0	< 0
0	0	0	0			

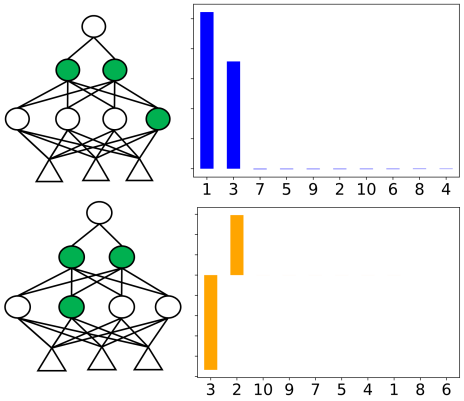


Figure 2.1: Possible results of the simulated task (8 combinations) and representation of the actual clusters provided by the cluster-based interpretation of the ReLU network.

From the simulation study, we note that the clustering-based interpretation of the ReLU network helps to achieve a more profound understanding of the solution meaning. In particular, the methodology tries to disentangle the complexity of the solutions into a set of more comprehensible linear solutions within a specific cluster.

As a further confirmation of the usefulness of cluster-based interpretation, when the activity regularizer is removed the network keeps solving the task giving rise to 8

clusters, each one specific for each combination of the first three features. Based on the cluster interpretation, we conclude that the network has chosen a less abstract way to solve the problem.

2.1.3 Titanic dataset

In this section, we report the experimental analysis performed on the well-known Titanic data set[5]. Each sample represents a passenger with specific features and the binary target variable indicates if the person survived the Titanic disaster. A standard data cleaning and feature selection procedure[6] are implemented. In Table 2.1 we report a brief description of the features of the dataset.

Table 2.1: Features for the Titanic dataset

Feature	Description	Range
Age	Age of the passenger discretized in 5 bins	$[0, 4]$
Gender	1 if the passenger is female	$\{0, 1\}$
PClass	Travel class: first, second, third	$[1, 3]$
Fare	Ticket fare discretized in 3 bins	$[0, 3]$
Embarked	Location for the embarked	$[0, 2]$
Title	Title of the passenger: Mr, Miss, Mrs, Master, Rare	$[1, 5]$
Is Alone	1 if the passenger has not relatives	$\{0, 1\}$

Similar to the previous experiment, we exploit a Multilayer Perceptron (MLP) with two hidden layers (4 and 2 neurons) characterized by ReLU activation function and an output neuron with a sigmoid activation function. The cross-entropy loss is minimized via the Adam stochastic optimizer with a step size of 0.01 for 10 epochs and a batch size of 100. An activity regularizer with 0.02 is added to the empirical loss. The experiments are implemented with Keras in the Python environment. The code is freely available at `https://github.com/nicolapicchiotti/relu_nn_clustering`.

The accuracy of the network is 77% and the study of the active neurons patterns provides a partition of the dataset into three clusters as shown in Figure 2.2.

(a) The first cluster a) includes passengers with mixed features and a percentage of survived ones equal to 38%. As expected from the univariate exploratory analyses, *gender* and *class* had the most significant relationship for survival rate.

(b) The cluster b) instead, includes only males belonging to the third class (4% of the overall): the prediction for these passengers is always 0. This cluster confirms the expectation on the male gender and third class as relevant features for not survive. We observe that the feature importance is quite similar to the one of cluster a) except for the fact that the "age" feature assumes slightly more relevance. Finally,

[5] `https://www.kaggle.com/c/titanic`

[6] `https://www.kaggle.com/startupsci/titanic-data-science-solutions`

(c) in the cluster c) the passengers are females and belonging to the first class (16% of the overall), the predicted value is always 1. The feature importance, in this case, shows that the high value of the *title*, *age*, and the other features contribute to survival, in addition to being female. This cluster helps us to understand the solution provided by the network. For instance, we note that the "age" feature has an opposite behavior with respect to the other two clusters: in this cluster the older the women, the higher the survival probability.

In this experiment, we have shown that the ReLU network can be disentangled into a set of clusters that can be analyzed individually. The clusters have a practical meaning helping the human to interpret the mechanism of prediction of the network.

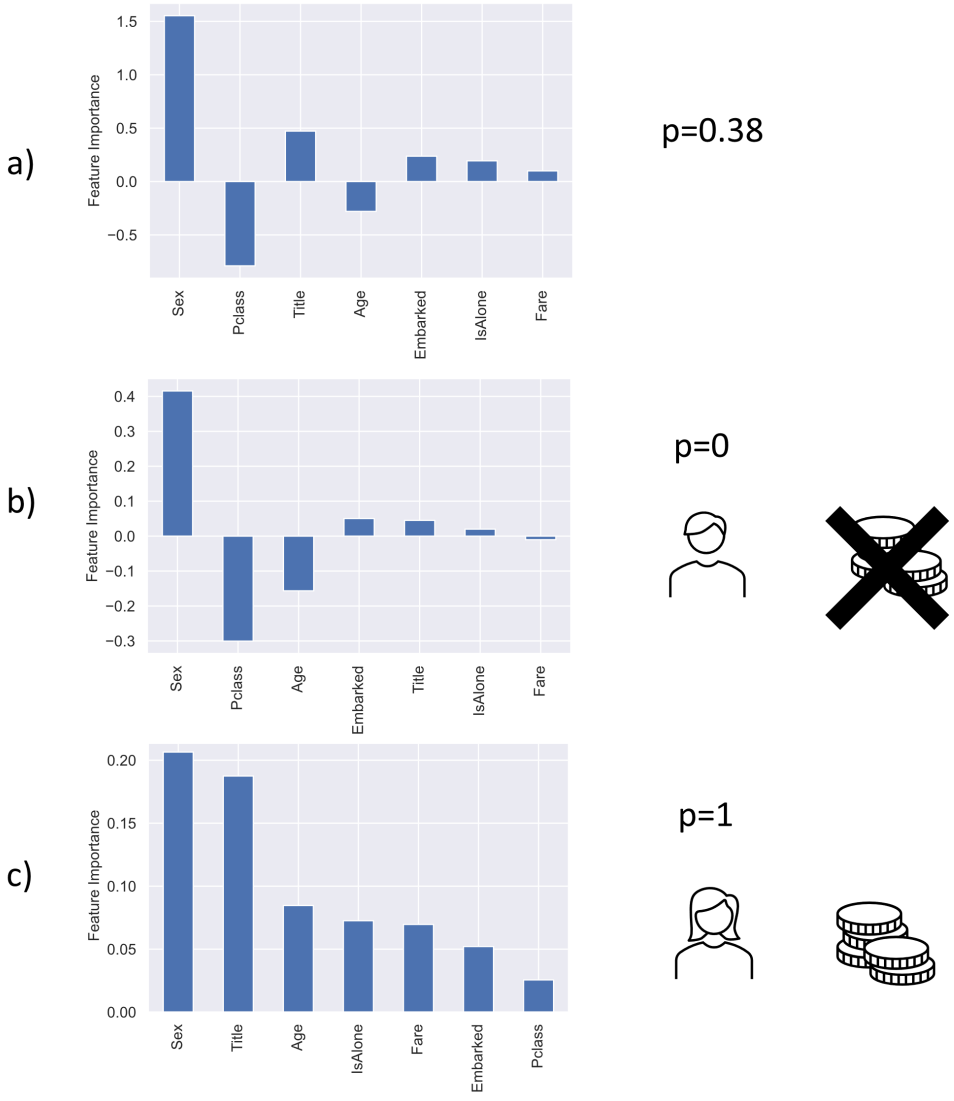


Figure 2.2: Bar plot with the feature importance for each of the three clusters originated by the ReLU neural network.

2.2 Logic Constraints to Feature Importance

As already mentioned, in recent years, Artificial Intelligence (AI) algorithms have been proven to outperform traditional statistical methods in terms of predictivity, especially when a large amount of data was available. Despite the widespread use and the extensive scientific research into Machine Learning (ML) models, effective interaction between the model and the human is still lacking and most of the current ML approaches tend to rely heavily on training/testing data. On the other hand, we deem that sources of knowledge like domain knowledge, expert opinions, understanding from the related problem, could be very important for a better definition of the model.

Here we present a novel framework trying to bridge the gap between data-driven optimization and human high-level domain knowledge. The approach provides for the inclusion of the human understanding of the relevance/importance of the input features. The basic idea is to extend the empirical loss with a regularization term depending on the constraints defined by the apriori knowledge on the importance of the features. We provide experimental results on the fairness topic.

There are few existing feature weighting approaches aimed at improving the performances of machine learning models. In [109] the author exploits weak domain knowledge in the form of feature importance to help the learning of Importance-Aided Neural Networks (IANN). The feature importance is based on the absolute weight of the first hidden layer neurons of the network. IANN is successfully applied in [110]. In [111] an ontology-based clustering algorithm is introduced along with a feature weights mechanism able to reflect the different features' importance. [112] uses both correlation and mutual information to weight the features for the algorithms SVM, KNN, and Naive Bayes. Instead, in order to accelerate the learning process, in [113] the algorithm is required to match the correlation between the features and the predictive function with the empirical correlation.

Anyway, none of the previous works define a general framework including the knowledge on the importance of the input features in the framework of explainable machine learning.

2.2.1 Mathematical setting of feature importance

In this section, we review the existing approaches aimed at assigning an importance score for each feature of a given input example in relation to the task of the model, i.e. the so-called local explainable methodologies. It is worth mentioning that the importance of a feature is one of the most used strategies to gain local explainability from an opaque machine learning model.

Let us consider a *predictor function* $\tilde{f}$ going from the d-dim feature space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 ... \times \mathcal{X}_d$ to the 1-dim target space $\mathcal{Y}$:

$$\tilde{f} : \mathcal{X} \to \mathcal{Y}.$$

Such a predictor function is the output of a learner:

$$\mathcal{L} : (\mathcal{X}^n \times \mathcal{Y}^n) \to (\mathcal{X} \to \mathcal{Y}),$$

able to process a supervised dataset $\mathcal{D} = \{X, Y\}$ where $X \in \mathcal{X}^n$ and $Y \in \mathcal{Y}^n$ with n instances. In order to fix the idea, we can think of $\mathcal{L}$ as a Deep Neural Network providing the predictor function $\tilde{f}$.

Definition 2 (Local feature importance)**.** The local feature importance is a function mapping a predictor $\tilde{f}$ and a single instance $\boldsymbol{x} \in \mathcal{X}$ to a d-dim vector of real values in the range $[0, 1]$:

$$\boldsymbol{I}(\tilde{f}, \boldsymbol{x}) : (\mathcal{X} \to \mathcal{Y}) \times \mathcal{X} \to [0, 1]^d$$

The local feature importance is a measure of how much the model relies on each feature for the prediction $\tilde{f}(\boldsymbol{x})$ made by $\tilde{f}$ on the particular pattern $\boldsymbol{x}$. Basically, the quantity $I_i(\tilde{f}, \boldsymbol{x})$ tells us how much the i-th feature contributes with respect to the others for a specific prediction. In the limit cases of $I_i(\boldsymbol{x}) = 0$ or $I_i(\boldsymbol{x}) = 1$ the feature i can be considered respectively useless or the most important one for the prediction done by the predictor on the pattern $\boldsymbol{x}$.

Example. Given a linear predictor $\tilde{f}(\boldsymbol{x}) = \sum_{i=1}^{d} w_i x_i$, the function

$$\boldsymbol{I}(\tilde{f}, \boldsymbol{x}) = \frac{|\boldsymbol{w}|}{max_{i \in [1,d]} |w_i|}$$

is a local feature importance function.

Definition 3 (Local feature importance methods)**.** Local feature importance methods are methods that given a predictor $\tilde{f}$, with its learner and the dataset, computes a local feature importance function $\boldsymbol{I}(\tilde{f}, \boldsymbol{x})$.

The set of feature importance methods, along with the data type and the model to which the method is referred are reported in Table 1.9. For a tabular dataset, the feature importances are usually represented as a rank reported in a histogram. For images or texts, the subset of the input which is mostly in charge of the predictions gives rise to saliency masks; for example, they can be parts of the image or a sentence of a text.

2.2.2 Constraints to feature importance

The overall goal of the present work is to define a framework where the local importance of the model's features can be constrained to specific intervals. We introduce a novel regularization loss term L_I, related to the not fulfillment of the feature importance's constraints:

$$L_r(\boldsymbol{x}, \boldsymbol{w}) + L_I(\boldsymbol{I}(\tilde{f}(\boldsymbol{w}, \cdot), \boldsymbol{x})), \tag{2.6}$$

where L_r is the usual empirical risk loss and $\boldsymbol{I}$ the d-dim vector of importances. It is worth observing that in Eq. (2.6) we explicated the dependence of the importance on the structure of the black box model via the weights of the model $\boldsymbol{w}$.

Let us suppose a First Order Logic (FOL) formula $E(\boldsymbol{I}(\tilde{f}, \boldsymbol{x}))$ with variable $\boldsymbol{x} = [x_1, x_2 ..., x_d]$ containing an apriori statement with inequalities on the features' importances. For example, we could require that, for every $\boldsymbol{x}$, both the feature 1 and the feature 2 should not be important for the prediction function to properly work:

$$\forall \boldsymbol{x} : \; I_1(\tilde{f}, \boldsymbol{x}) < c_1 \wedge I_2(\tilde{f}, \boldsymbol{x}) < c_2, \tag{2.7}$$

with $c_1 \in [0,1]$ and $c_2 \in [0,1]$.

In order to treat the logic formula with real value functions, each inequality of the FOL formula can be transformed into a new variable $l_{i,c_i} \in [0,1]$ through the following transformation:

$$I_i(\tilde{f}, \boldsymbol{x}) < c_i \;\longrightarrow\; l_{i,c_i}(\boldsymbol{x}) = \frac{\max(I_i(\tilde{f}, \boldsymbol{x}) - c_i,\, 0)}{1 - c_i}. \tag{2.8}$$

Although Eq. (2.8) is a quite natural choice for an increasing function from 0 to 1, other choices are possible. In Figure 2.3 we represent the variable $l_{i,c_i}(\boldsymbol{x})$ of Eq. (2.8) for the case $c_i = 0.1$ of a generic feature.

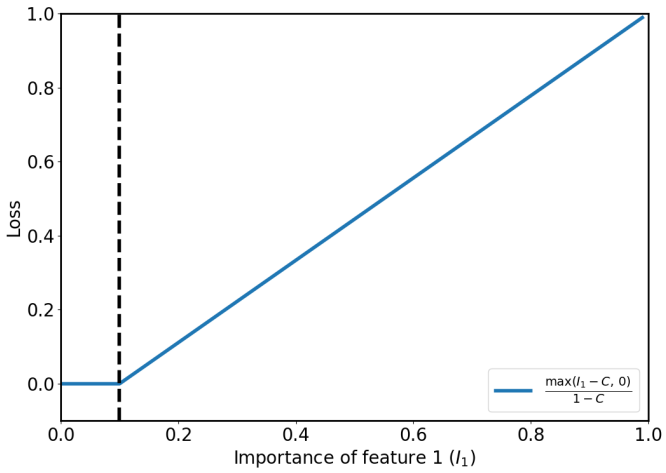


Figure 2.3: Example of loss as for the inequality on the importance of the constrained feature for $I_i < 0.1$.

So, thanks to Eq. (2.8), the aforementioned FOL formula Eq. (2.7) can be written as:

$$\forall \boldsymbol{x}: \;\; l_{1,c_1}(\tilde{f}, \boldsymbol{x}) \wedge l_{2,c_2}(\tilde{f}, \boldsymbol{x}).$$

Then, we exploit the framework of t-norm fuzzy logic that generalizes Boolean logic to variables assuming values in $[0,1]$. We can convert the formula depending on the losses $E(\boldsymbol{l}(\tilde{f}, \boldsymbol{x}))$ by exploiting a T-norm t in the following:

$$\Phi_\forall(\boldsymbol{l}(\tilde{f}, \mathcal{X})) = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} t_E \left(\boldsymbol{l}\left(\tilde{f}, \boldsymbol{x}\right)\right),$$

that is an average over the t-norm of the truth degree when grounded $\boldsymbol{x}$ over its domain. Then, a loss term can be defined by exploiting the logic constraints, e.g.

$$L_I(\boldsymbol{I}(\tilde{f}, \mathcal{X})) = \lambda(1 - \Phi_\forall(\boldsymbol{I}(\tilde{f}, \mathcal{X}))),$$

where λ is the strength of the regularization.

Finally, the partial derivative of the logic part of the loss L_I with respect to the j-th weight of the i-th importance loss function is

$$\frac{\partial L_I}{\partial w_{i,j}} = \sum_k \frac{\partial L_I}{\partial \Phi_k} \frac{\partial \Phi_k}{\partial l_i} \frac{\partial l_i}{\partial w_{i,j}},$$

and the derivative to being evaluated is: $\frac{\partial l_i}{\partial w_{i,j}}$. By resuming, the scheme is the following:

1. write the FOL formula depending on the feature importance, in turn, depending on the model weights through the chosen feature importance method;
2. convert the inequality terms $I_i < c_i$ into loss terms l_{i,c_i};
3. convert the FOL formula with the t-norm into an overall loss term;
4. the loss term is optimized in an iterative process by computing the importance at each step of the algorithm.

2.2.3 Fairness through feature importance constraints

Fairness is a natural field where the constraints to feature importance can be applied. In the following, we resume the principal fairness measures (see also Section 1.7.2 for a review) and we discuss how they can be translated by using our proposed scheme based on Constraints to Feature Importance, denoted hereafter as CTFI.

As already discussed, the *Demographic Parity* (DP) fairness metric is satisfied when, given the random variable $\tilde{Y}$ representing the binary predictor $\tilde{f}$ and a protected Boolean feature X_s we have:

$$P(\tilde{Y} = 1 | X_s = 0) = P(\tilde{Y} = 1 | X_s = 1).$$

DP is a very strong requirement: groups based on a sensitive feature, e.g. black and white, should have the same rate of positive prediction, even if differences are present.

DP can be translated into a constraint, where the importance of the protected feature s needs to be lower than a given threshold $c \in [0, 1]$:

$$\forall \boldsymbol{x} \;\; I_s(\boldsymbol{x}) < c. \tag{2.9}$$

The possible well-known issue of *unfairness due to correlated features* (see for instance [88]) can be potentially solved by setting a constraint also for the features that are correlated with the protected one. Obviously, the regularization strength (λ_i) of the i-th correlated feature should be lower with respect to λ_s, for instance given by:

$$\lambda_i = \lambda_s \cdot \rho_{s,i}. \tag{2.10}$$

The advantage of this formulation is that the constraints are smooth between 0 and 1, and can be used both with binary and continuous features.

A measure of discrepancy from DP, which will be useful for the experimental part, is the *Disparate impact* (DI):

$$\mathrm{DI} = \frac{P(\tilde{Y} = 1|X_s = 0)}{P(\tilde{Y} = 1|X_s = 1)}. \tag{2.11}$$

A possible relaxation of DP is where we grant that the protected attribute x_s can be used to discriminate among groups that are actually different in the ground truth label y, i.e., between $y = 0$ and $y = 1$, but not within each one. This is called *Equalized odd* (EOD) and is described in paper [84]. We say that a predictor $\tilde{Y}$ satisfies EOD if $\tilde{Y}$ and X_s are independent, conditional on Y:

$$P(\tilde{Y} = 1|X_s = 0; Y = 1) = P(\tilde{Y} = 1|X_s = 1; Y = 1),$$

$$P(\tilde{Y} = 1|X_s = 0; Y = 0) = P(\tilde{Y} = 1|X_s = 1; Y = 0).$$

A quite natural measure of discrepancy from EOD is the *average equality of odds difference* (EO):

$$\begin{aligned}\mathrm{EO} = &\frac{P(\tilde{Y} = 1|X_s = 0; Y = 1) - P(\tilde{Y} = 1|X_s = 1; Y = 1)}{2} \\ &+ \frac{P(\tilde{Y} = 1|X_s = 0; Y = 0) - P(\tilde{Y} = 1|X_s = 1; Y = 0)}{2}.\end{aligned} \tag{2.12}$$

Finally, another measure of fairness discrepancy defined in [86] is *counterfactual fairness difference* (CF):

$$\mathrm{CF} = P(\tilde{Y}_{x_s \leftarrow 0} = 1|X_s = 1) - P(\tilde{Y} = 1|X_s = 1), \tag{2.13}$$

where the idea is to evaluate the differences of the prediction's probabilities by changing the protected feature of the patterns from 1 to 0.

2.2.4 Toy example: constraint of the form $I_i(\boldsymbol{x}) < c$

As a toy example useful to test the effectiveness of the proposed scheme, we used the German credit risk dataset (1000 instances), available in [114], containing information about bank account holders and a binary target variable denoting the credit risk. The considered features are reported in Table 2.2.

Table 2.2: Features for the German credit risk dataset.

Feature	Description	Range
Age	Age of the costumer	numerical [19, 74]
Job	Job qualification	ordinal [0, 3]
Amount	Credit Amount (€) of the loan	numerical
Duration	Duration (year) of the loan	numerical [4, 72]
Gender	Male (1) Vs Female (0)	Boolean

We exploited a neural network with one hidden layer and 16 neurons. The learning rate of SGD is 0.01 with 10 epochs. The activation function is ReLU and the loss is given by the binary cross-entropy.

After the training phase, the Layer-wise Relevance Propagation (LRP, detailed in appendix 5) method has been applied to the instances of the testing set (50%

of the overall samples) for computing the feature importances. The black line in Figure 2.4 reports the average feature importance computed with LRP. We observe that the most relevant feature is the *duration* of the loan, followed by the *amount* and the *gender*.

Let us introduce a constraint to the importance of *gender* feature that we want to be less-equal than zero (see Eq. (2.9)), with a regularization $\lambda = 0.05$. As expected, we observe (green line in Figure 2.4) that the *gender* feature has become useless for the model predictions. Basically, the model found another solution, by giving more importance to other features, e.g. the *job*.

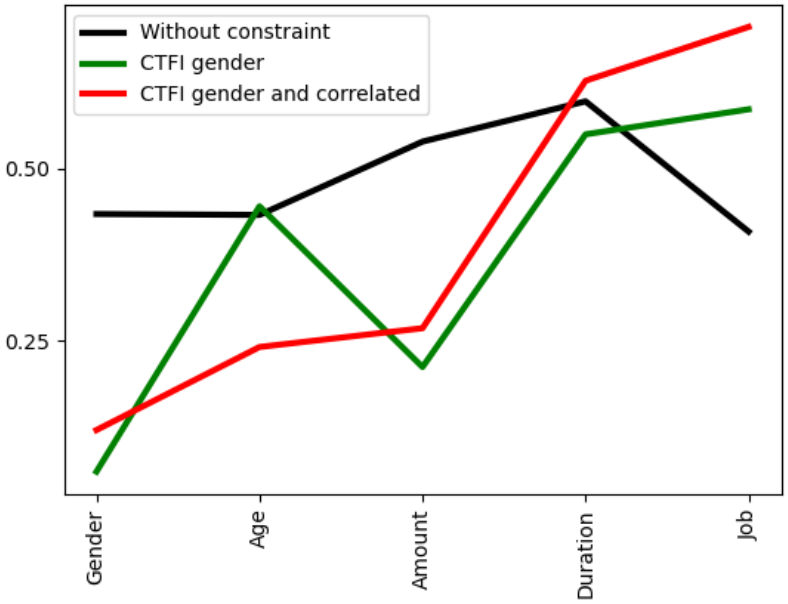


Figure 2.4: Feature importance (LRP) for the original model (black line), for the model with the constraints on the *gender* feature (green line) and that constraining also the correlated features (red line).

Furthermore, we computed the correlation matrix between the different features and we found out that, for instance, the *age* feature is correlated with the *gender* (with a Pearson correlation coefficient of $\rho_{\text{gender,age}} = 16\%$). So, in the third experiment we constrained also the other features, by using different regularization strength given by:

$$\lambda_i = \lambda \cdot \rho_{\text{gender},\, i}$$

for the i-th feature (see Eq. (2.10)). We note from the results reported in Figure 2.4 with a red line, that the correlated *age* feature decreases its importance, coherently with the expectation.

2.2.5 Fairness through constraints to feature importance

In this section, we report the results of the experimental part related to fairness. We tested the CTFI scheme proposed in the previous section to the *Adult income*

data set, also considered by [115]. It contains 48,842 instances with 12 attributes (see Table 2.3 for the description) and a binary classification task for people earning more or less than \$50,000 per year. The protected attribute we will examine is the *race*, categorized as white and non-white. In order to better evaluate the fairness metrics with a uniform test set, the dataset has been balanced and the chosen split of training/test is 50%. The model is a Neural Network with one hidden layer and 4 neurons. The learning rate is 0.1, the number of epochs is 10 and the batch size is fixed to 1 in order to compute the local feature importance for each analyzed pattern. The activation function is the ReLU function and the loss is given by the binary cross-entropy.

Table 2.3: Features for the Adult income dataset.

Feature	Range
Age	numerical [19, 74]
Race	Boolean: white Vs non-white
Sex	Boolean: female Vs male
Education	ordinal: [1, 5]
Native-country	Boolean: US Vs other
Marital-status	Boolean: single Vs couple
Relationship	ordinal: [1, 5]
Employment type	ordinal: [1, 5]
fnlwgt	continuous
Capital loss	Boolean: Yes Vs NO
Capital gain	Boolean: Yes Vs NO
hours-per-week	continuous

We used the constraint defined in Eq. (2.9) with $c = 0$ for the *race* feature; whereas as a fairness metric we consider both the disparate impact (DI) defined in Eq. (2.11), the average equality of odds difference (EO) in Eq. (2.12) and counterfactual fairness (CF) reported in Eq. (2.13). For the accuracy, we calculate the Area under the ROC curve (ROC-AUC).

Firstly, we evaluated the different fairness metrics in the testing set, with an increasing level of the regularization strength λ (10 values from 0 to 0.5). In Figure 2.5 we report the accuracy metric (ROC-AUC in the lower plot) and the three measures of fairness: disparate impact (DI), average equality of odds difference (EO), and counterfactual fairness (CF) as a function of the regularization strength.

In Figure 2.5 we observe that, while the level of the ROC-AUC score practically remains the same, both DI, EO, and CF grow as the regularization strength augments, denoting an increased level of all the fairness measures. In particular, the CF measure reaches the value of 0, meaning that the protected feature no longer affects the predictions. The other two measures, DI and EO, do not reach the maximum level (1 and 0 respectively) because of the issue of correlated features. However, when also the correlated features are constrained through Eq. (2.10), we note that the increase of fairness is more pronounced (right panel of Figure 2.5).

Then, as a further analysis we compared the fairness/accuracy levels obtained with

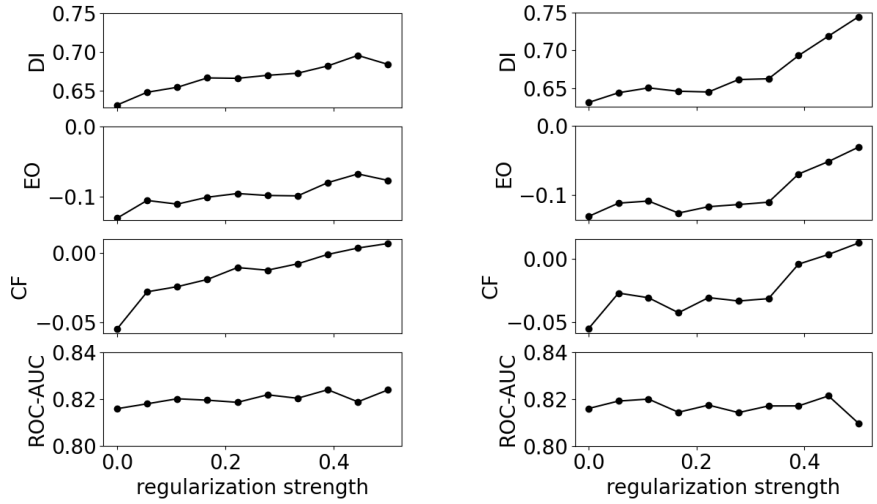


Figure 2.5: Accuracy (ROC-AUC); and fairness measured as disparate impact (DI), average equality of odds difference (EO) and counterfactual fairness (CF) (for each measure, the higher the values the higher the fairness levels), by constraining only the *race* feature (left Panel) and by constraining both *race* and the correlated ones (right Panel), as a function of the regularization strength.

the CTFI methodology[7] to the following benchmark methodologies:

1. the *unawareness* method [87], avoiding to use the *race* feature during the training phase;

2. a pre-processing method based on the *undersampling* of the samples with protected attribute;

3. the pre-processing method called *reweighing* [116] that assigns weights to the samples in the training dataset to reduce bias.

The AIF-360 library was used to apply the benchmark methodologies and the fairness metrics. All the models are coded in the Pytorch environment and available at the Github repository `https://github.com/nicolapicchiotti/ctfi`.

In Figure 2.6 we report the results of the accuracy measure given by the ROC-AUC (x-axis) and fairness metric EO (y-axis) for the different methodologies (unawareness, undersampling, reweighting) and the CFTI. The values are reported in Table 2.4.

We note that, with respect to the original model, the unawareness, the undersampling, and the reweighting methodologies grant a high level of fairness at the expense of accuracy. On the other side, the CTFI methodology provides higher fairness metrics with a similar level of accuracy.

[7]With regularization chosen to be 0.1 and constraining also the other features, by using regularization strengths given by $\lambda \cdot \rho_{\text{gender},i}$ (see Eq. (2.10))

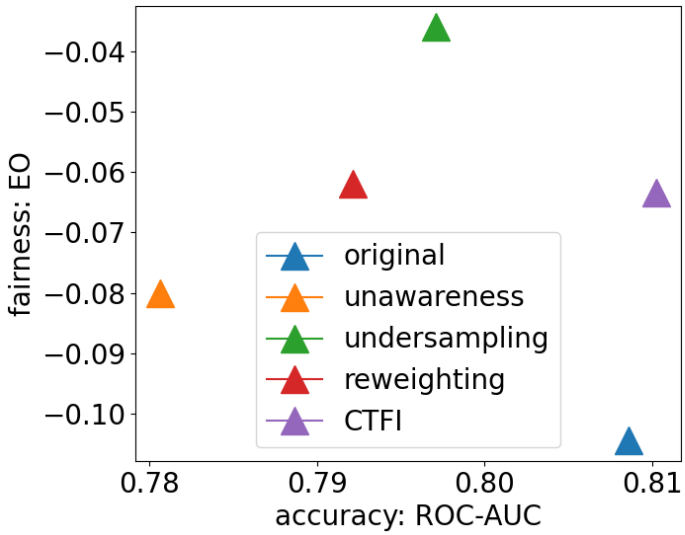


Figure 2.6: Results of accuracy (ROC-AUC) and fairness (EO) for the different methodologies: original, unawareness, undersampling, reweighting and CTFI.

method	ROC-AUC	EO
original	0.809	-0.104
unawareness	0.781	-0.080
undersampling	0.797	-0.036
reweighting	0.792	-0.062
CTFI	0.810	-0.063

Table 2.4: Results of the trade-off between accuracy (ROC-AUC) and fairness (EO) for the different methodologies: original, unawareness, undersampling, reweighting and CTFI.

Chapter 3

Machine Learning strategies for Gene discovery in COVID-19

The outbreak of the coronavirus disease 2019 (COVID-19), the Severe Acute Respiratory Syndrome caused by coronavirus SARS-CoV-2, that first appeared in December 2019 in Wuhan (China), has resulted in millions of cases worldwide within a few short months, and rapidly evolved into a real pandemic.

The COVID-19 pandemic has been representing an enormous challenge to the world's healthcare systems. Among the European countries, Italy was the first to experience the epidemic wave of SARS-CoV-2 infection, accompanied by a severe clinical picture and a high level of mortality rate. Up to August 28th, 2021, in Italy, the number of overall confirmed COVID-19 cases since the beginning of the pandemic was $4,524,292$ with $129,056$ related deaths as reported in [117].

In such a challenging context, a strong contribution has been provided by the research community, according to the individual capacity and sense of responsibility. Especially during the initial phases of the pandemic, a considerable effort of the scientific community was on monitoring the infection evolution in terms of the number of new cases, recoveries, and deaths, by focusing on the mathematical modeling in the field of epidemiology. I, too, along with others, contributed with [1]. The paper introduces a SEIR[1] compartmental model, taking into account the fraction of undetected cases, the effects of mobility restrictions, and the estimates of personal protective measures adopted, such as wearing a mask and washing hands frequently. The challenge was to find the best strategy to progressively relax the control measures, e.g. the lockdown, keeping the number of new infections below a certain threshold.

In Figure 3.1 we report the scheme of the proposed compartmental model (Panel A) and a representation of the modeled mobility effect with a decreasing logistic function (Panel B) calibrated on the basis of Google mobility data [118] (see Panels C and D).

The model has been experimentally validated for France, Germany, Italy, Spain, United Kingdom, and the United States for the period February-June, 2020. Similarly, the model has been applied to the Italian regions, as shown in Figure 3.2.

[1] Acronym for Susceptible, Exposed, Infectious, or Recovered.

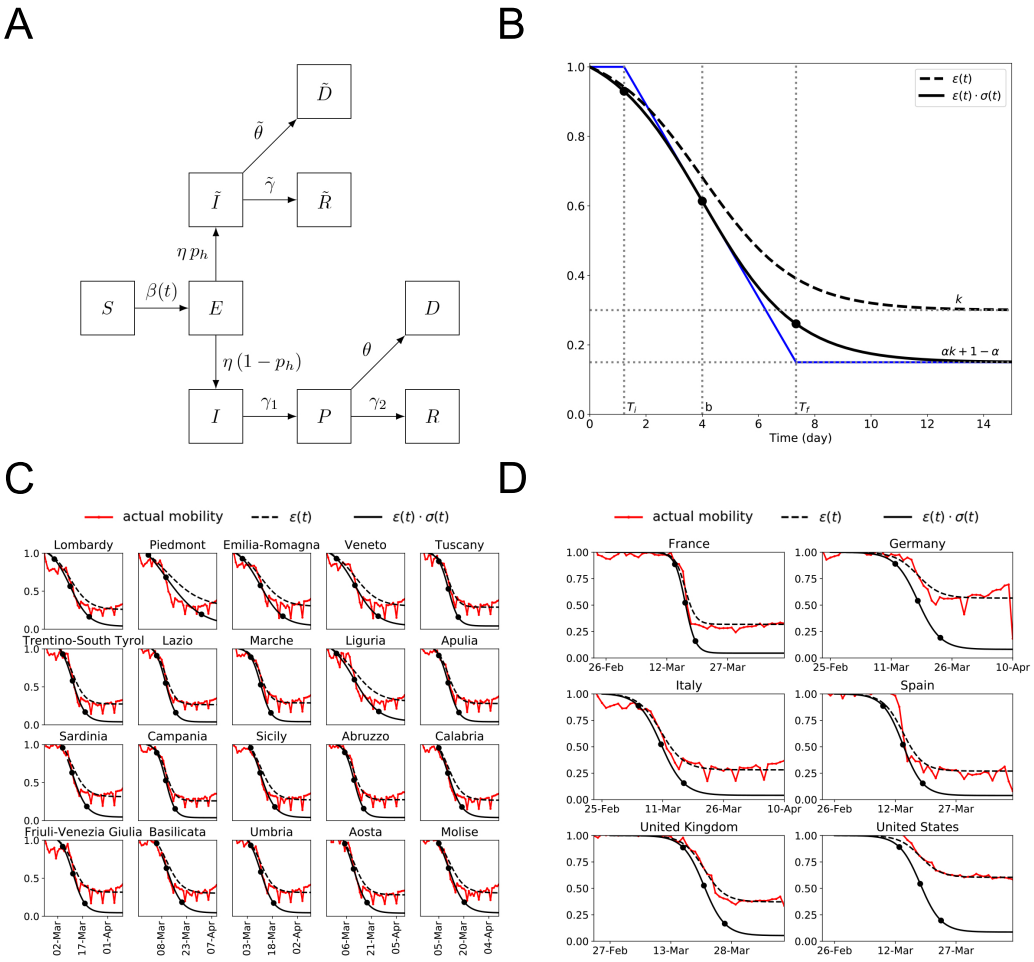


Figure 3.1: Panel A: The scheme of the compartmental model: susceptible (S); exposed (E); infected (I); positive tested (P); recovered (R); deaths (D); the symbol ˜ indicates undetected groups. Panel B: The method for the $\epsilon(t)\cdot\sigma(t)$ logistic function, extensively treated in supplementary methods of [1] considering both mobility and personal protective measures (PPM) effect: k, plateau level of mobility decreasing; $\epsilon(t)$, logistic function interpolating the mobility changes; α, coefficient representing the effect of PPM; T_i, start of public health measures; b, inflection time point of the logistic functions $\epsilon(t)\cdot\sigma(t)$ and $\epsilon(t)$. Panels C and D show the $\epsilon(t)\cdot\sigma(t)$ logistic functions (black thick lines) calibrated with Google data of mobility changes, in Italian regions and across countries, respectively.

On the other side, in the medical field, after almost two years COVID-19 has demonstrated itself to be a disease having a broad spectrum of clinical presentations: from asymptomatic patients to those with severe symptoms leading to death or persistent disease ("long COVID") [9, 10]. While developing vaccination programs and other preventive measures to significantly dampen infection transmission and reduce disease expression, a much deeper understanding of the interplay between SARS-CoV-2 and host genetics is required also to support the development of treat-

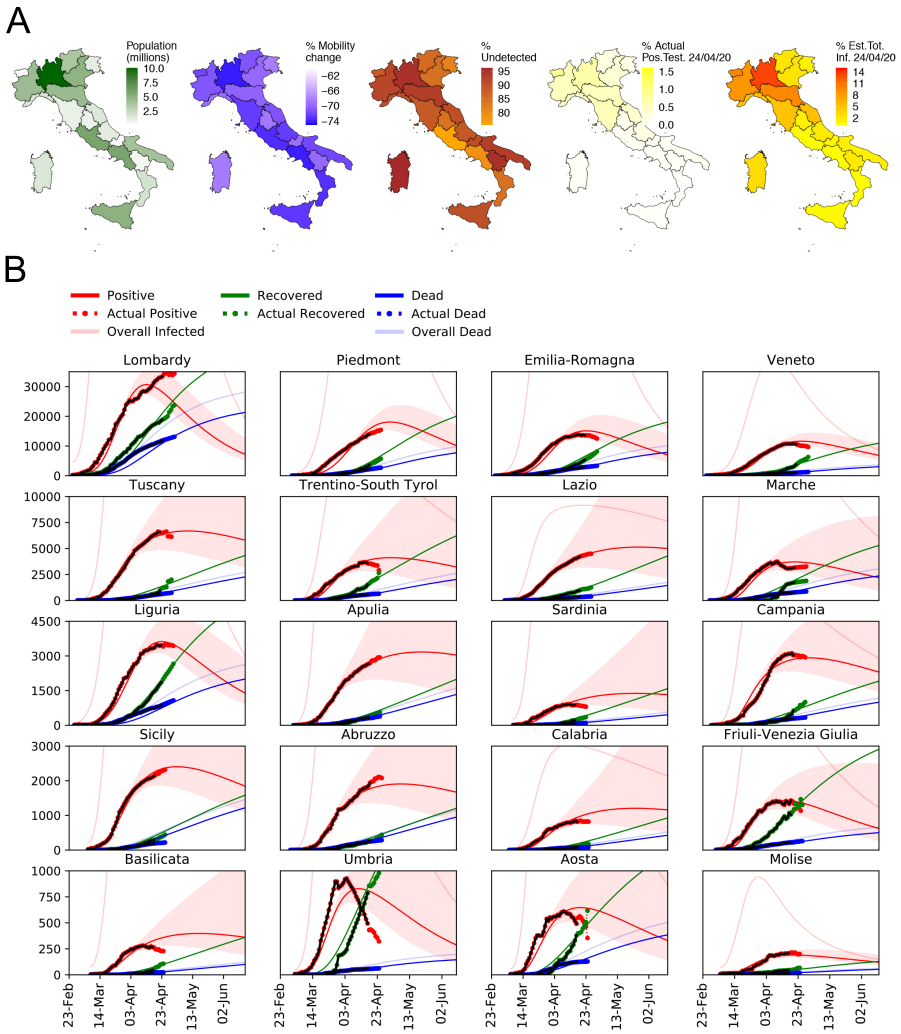


Figure 3.2: Panel A: Epidemic statistics for Italian regions; Panel B: Epidemic predicted and actual reported data in different Italian Regions; overall infected and deaths include undetected cases. The models were fitted on actual data until 17th April 2020 (black line); whereas cumulative observed values from 18th, April 2020 to 24th, April 2020 are used for judging models' accuracy. The shadow regions represent the reasonable confidence intervals.

ments for new virus variants as they arise.

Moreover, by better understanding the role of host genetics in COVID-19 susceptibility and disease severity, we are also in a stronger position to identify public health measures that will curb the impact of the disease on society as a whole. This should help us to genetically screen already affected patients as well as, eventually, individuals who may potentially be patients in order to predict those who are more or less susceptible to developing COVID-19 post-infection, especially the more severe cases. It should further help us in, not only reassigning therapeutics or developing new interventions (including vaccines) but also in decision-making regarding

therapeutics and vaccine allocations.

Furthermore, COVID-19 presents an important test case for developing new gene science models for studying complex disorders with a background of combined genetic and environmental factors. Unlike other multifactorial disorders, the main environmental factor for COVID-19, SARS-CoV-2, can easily be identified through PCR-based tests on swabs. Assuming a relatively low impact of viral genome variability [119], the remaining variability in clinical outcome may likely be associated with age, comorbidities, and host genetics, including sex. Anyway, differently for Mendelian diseases where a single variant can be responsible for the disease, complex genetic disease is characterized by a high number of variants contributing together in a cooperative way. In Figure 3.3 (representation of chromosomes from [120]) we represent the difference between a Mendelian genetic disease where a single variant is causing the disease (upper plot) and a complex genetic disease, formed by a potentially large number of genetic variants (lower plot).

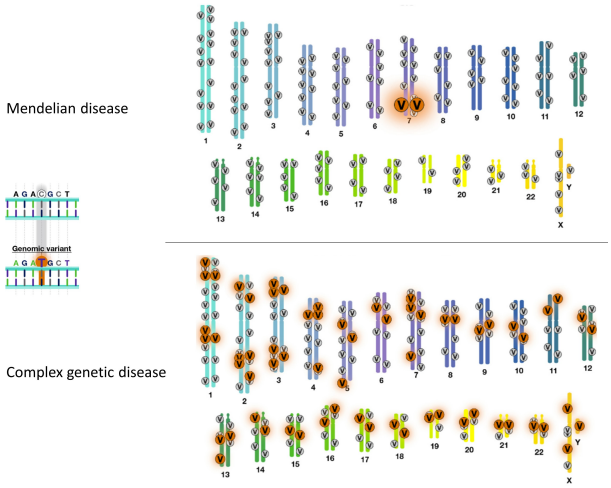


Figure 3.3: Representation of the difference between Mendelian diseases and complex genetic diseases.

It is worth noting that in recent years, Machine Learning (ML) has started to play an important role in the classification/clustering tasks related to genomic dataset [121]. The ML approach we followed in the thesis aims at bridging the gap between the complex genetic nature of the disease and the classical genetic modeling, mostly focusing on specific components of the disease.

In the present chapter, we start by exploring the genetic variability from the different points of view of rare and common variants, whereas in Chapter 4 we aim at merging the genetic information into an overall predictive model.

3.1 Review of classical statistical methods

Classical studies, such as Genome-Wide Association Studies (GWASs) have identified a certain number of common polymorphisms in relevant genes [122, 123]. The analysis is based on the comparison of around 700.000 genomic Single-Nucleotide

Polymorphism (SNPs) frequencies in cases/controls (mostly non-coding). However, these associations do not satisfactorily explain the variability of clinical outcomes.

Instead, the candidate gene approach has shown that, as with many other complex disorders, a simple Mendelian inheritance is also found in COVID-19, affecting some rare individuals with defective gene variants related to innate immunity [10, 124]. Another methodology, the burden testing, (see [125]), focuses only on rare coding variants and identified up to September 2021, 2 loci, by an aggregation on a gene level of the variants and a comparison between case and control subjects.

These two most important models in the statistical framework of genetic diseases, GWAS and Burden gene test are represented in Figures 3.4 and 3.5. We listed for both, the strengths and weaknesses, as well as the achieved results with respect to the research in the COVID-19 area.

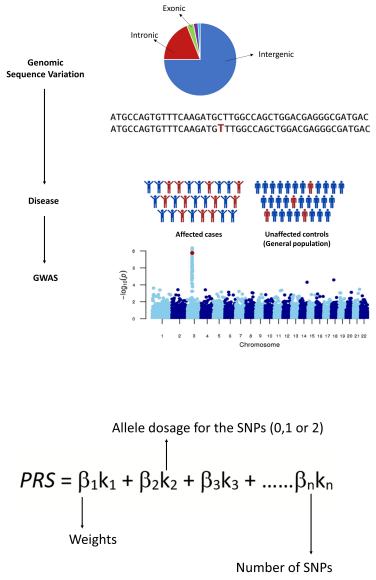
Method principle	Strengths and weakness	COVID-19 results
	• Straightforward comparison of 700.000 genomic SNPs frequencies in cases/controls (mostly non-coding) • Coverage of coding SNPs only throughout imputed data (imputing 2 M SNPs from 0.7 SNPs by LD) • Multiple independent tests and high threshold for the significativity • Need of ten/hundred of thousands subjects • GWAS focuses on common variants (MAF ≥ 5%) whose effects are small (1.2, 1.5) • Missing hereditability: rare variants	FEW GENES 25 loci identified at September 2021 • Pairo-Castineira E., et al. *Nature* December 2020 • Severe COVID-19 GWAS Group, Ellinghaus D. et al. *N Eng J Med.* October 2020 • COVID-19 Host Genetics Initiative, Niemi M.E.K., Karjalainen J., et al. *Nature* July 2021 • Kousathanas A, et al. *medRxiv* 2021

Figure 3.4: Scheme of the GWAS benchmark methodology for the study of SNPs.

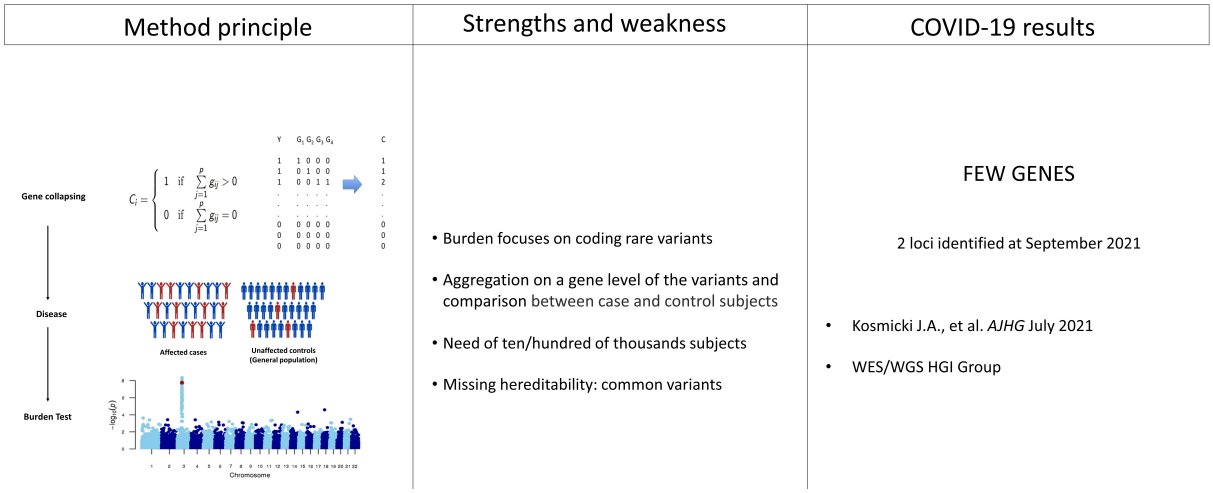


Figure 3.5: Scheme of Burden gene benchmark methodology for the study of rare coding variants.

3.2 The GEN-COVID Biobank

The whole dataset exploited for the following analyses has been collected and managed by the GEN-COVID Multicenter Study [126]. The study was designed to collect and systematize biological samples and clinical data across multiple hospitals and healthcare facilities in Italy with the purpose of deriving patient-level phenotypic and genotypic data. The specific intention is to make samples and data available to COVID-19 researchers globally. To this end, a GEN-COVID Biobank (GCB) and a GEN-COVID Patient Registry (GCPR) were established utilizing already existing biobanking and patient registry infrastructure.

The socio-demographic information of the patients includes sex, age, and ethnicity. Information about family history, (pre-existing) chronic conditions, and SARS-CoV-2 related symptoms were collected through a detailed core clinical questionnaire. The COVID-19 severity has been assessed using a slightly modified version of the WHO COVID-19 Outcome Scale [127] as coded into the following five categories:

0. not hospitalized;
1. hospitalized, not receiving supplemental oxygen;
2. hospitalized, receiving low-flow supplemental oxygen;
3. hospitalized, receiving continuous positive airway pressure (CPAP) or bilevel positive airway pressure (BiPAP) ventilation; and
4. hospitalized receiving invasive mechanical ventilation.

The collection of samples and data are utilized in the GEN-COVID Multicenter Study for generating Genotyping (GWAS) and Whole Exome Sequencing (WES) results. The data resulting from these studies are then stored and made available through the GEN-COVID Genetic Data Repository (GCGDR).

It is important to recall that all samples and data have also been systematized in accordance with the FAIR (Findability, Accessibility, Interoperability, and Reuse) Data Principles [128] to promote their international availability and use for COVID-19 research.

3.3 Unsupervised analysis for multi-organ phenotype characterization of COVID-19

COVID-19 disease is characterized by a highly heterogeneous phenotypic response to SARS-CoV-2 infection [129], with the large majority of infected individuals having only mild or even no symptoms. However, the severe cases can rapidly evolve towards a critical respiratory distress syndrome and multiple organ failure. The symptoms of COVID-19 range from fever, cough, sore throat, congestion, and fatigue to shortness of breath, hemoptysis, pneumonia followed by respiratory disorders, and septic shock.

In order to simplify the multi-organ characterization of the disease, as a first step, the laboratory values related to the different organs (see Section 3.2) have been represented as a binary clinical classification for organ/system damage[2] as reported in Table 3.1.

Then, a descriptive analysis of the phenotypes by using the hierarchically clustered heatmap was performed (see [126]). In particular, both patients and phenotypes have been clusterized with the agglomerative hierarchical clustering methodology, where the chosen metric is the hamming distance and the linkage criterion is the "average" one (Unweighted Pair Group Method with Arithmetic mean, UPGMA). Other distances, such as the Jaccard one, and other methodologies have been tested, with similar results.

The resulting dendrograms of the clusterization are reported in the upper and in the left part of the heat plot of Figure 3.6; whereas, the information on the grading of severity of the patients is added a posteriori on the left strip. The resulting plot is obtained with the Python Seaborn package.

From Figure 3.6 we observe that the clustering analysis identifies five main clinical categories and several subcategories. Specifically, the clusters represent:

(A) severe multisystemic disease, with either thromboembolic (A1) or pancreatic variant (A2);

(B) cytokine storm, either moderate (B1) or severe with liver involvement (B2);

(C) mild disease, either with (C1) or without hyposmia (C2);

(D) moderate disease, either without (D1) or with (D2) liver damage;

(E) heart-type, either with (E1) or without (E2) liver damage (Figure 4).

[2] cTnT, cardiac Troponin T; NT-proBNP, N-terminal (NT)-pro hormone BNP; ALT, Alanine transaminase; AST, Aspartate transaminase; CD4, CD4+ T cells; NK, Natural killer; IL6, Interleukin 6; LDH, Lactate dehydrogenase; CRP, c-reactive protein.

Table 3.1: Binary clinical classification of multi-organs involvements.

Organ /system	**Rule**	**Clinical Interpretation**
Lung	1 if severity grading in the range $[3, 5]$ and 0 if severity grading in $[1, 2]$	Lung disease
Heart	1 if cTnT > reference value or NT-proBNP gender specific reference value or Arrhythmia	Heart disease
Liver	1 if ALT and AST > gender specific reference value	Liver disease
Pancreas	1 if lipase and/or pancreatic amylase > or < specific reference value	Pancreas disease (either inflammation or depletion)
Kidney	1 if creatinine > gender specific reference value	Kidney disease
Lymphoid system	1 if NK cells < reference value or CD4 lymphocytes < reference value	Innate and adaptive immune deficit
Olfactory / gustatory system	1 if Hypogeusia or Hyposmia	Olfactory and Gustatory deficit
Clotting system	1 if D-dimer > 10X W/wo low Fibrinogen level (with high basal level)	Thromboembolism
Pro-inflammatory cytokines system	1 if IL6 > reference value or LDH and CRP > reference value	Hyperinflammatory response

3.3.1 Biological interpretation of the clusters

The emerging clinical categories from Hierarchical Cluster Analysis point to specific types and subtypes that are more likely to have common genetic factors.

As unmasked by the dendrogram (group A), there is indeed a growing body of evidence suggesting that, in addition to the common respiratory symptoms (fever, cough, and dyspnea), COVID-19 severely ill patients can often have symptoms of a multisystemic disorder [130]. Multiple organ failure due to diffuse microvascular damage is an important cause of death in COVID-19 severely affected patients [131]. In line with our definition of an A1 subgroup, a retrospective study on 21 deaths after SARS-Co-V2 infection recently reported that 71% of the patients who died had disseminated intravascular coagulation (DIC), while the incidence of DIC in surviving patients was 0.6% [132]. These data suggest that DIC is an important risk

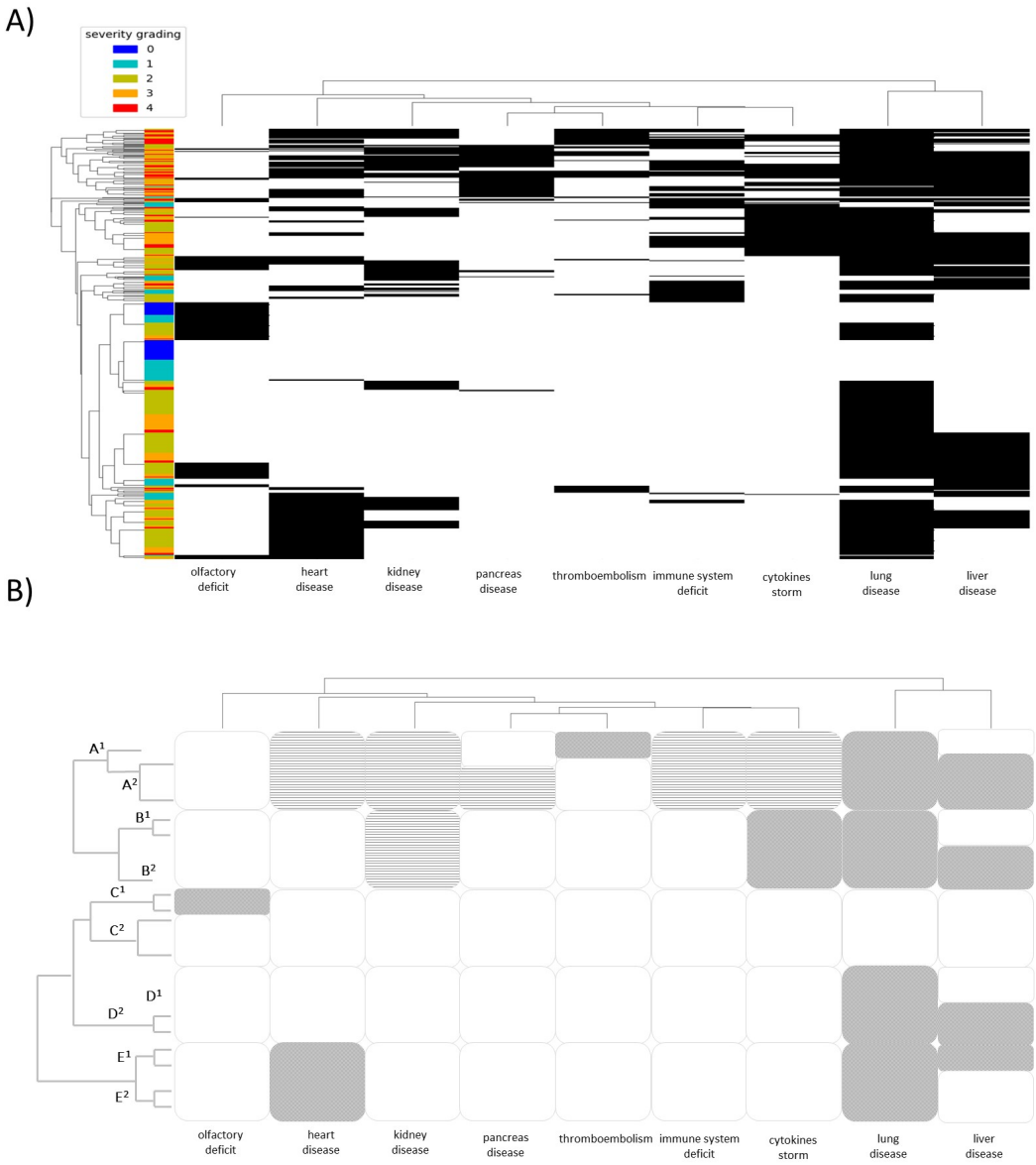


Figure 3.6: Panel A: Dendrogram of COVID-19 patients' clinical phenotypes by hierarchical clustering of organ/system involvement. Panel B: Drawing of the above-reported graph helping interpretation and simplification in the main branch of the tree. A1 severe multisystemic with either thromboembolic; A2 severe multisystemic with pancreatic variant; B1 cytokine storm with moderate liver involvement; B2 cytokine storm with severe liver involvement; C1 mild either with hyposmia; C2 mild without hyposmia; D1 moderate without liver damage; D2 moderate with liver damage; E1 heart with liver damage; E2 heart without liver damage. The image is taken from the original paper [126].

factor for increased in-hospital mortality and special attention should be paid to its early diagnosis and treatment.

While a debate still exists about the significance of pancreatic enzyme elevations during COVID-19 infection and the capability of the SARS-CoV-2 virus to induce pancreatic injury due to cytotoxic effects [133, 134], it is worth noting that among patients with a multisystemic involvement we observe a subclass of individuals (group A2) with pancreatic damage, likely suggesting a secondary effect of SARS-CoV-2 infection on a subgroup of genetically predisposed individuals.

Inflammatory cytokine "storm," has been reported as playing a key role in the severe immune injury to the lungs caused by T-cell over-activation (group B) [135]. While some investigators have suggested a potential mechanism of myocardial injury due to COVID-19-induced cytokine storm that is mediated by a mixed T helper cell response in combination to hypoxia [136], our findings indicate rather a distinct class of patients, (group E), presenting with heart involvement in the absence of an inflammatory cascade. This would tend to support the hypothesis that SARS-CoV-2 may directly damage myocardial tissue and induce a major cardiovascular event. Thus, as currently recommended, our research reinforces the need to monitor plasma cTnT and NT-proBNP levels in COVID-19 patients.

In line with current evidence [137, 138], although liver injury seems to occur more frequently among critically ill patients with COVID-19 (group B), it can also be present in non-critically ill patients (groups D and E) and, as suggested, it could be mostly related to prolonged hospitalization and viral shedding duration. This allows defining, for each group, a clinical subclass according to this organ involvement.

Finally, a recent extensive review determined the prevalence of chemosensory deficits based on pooling together forty-two studies reporting on 23,353 patients [139]. No correlation with age was detected, but anosmia/hypogeusia decreased with disease severity. In accordance with evidence found in the literature, hyposmia was mostly represented among patients in group C with mild clinical symptoms [140].

3.4 Gene Discovery via LASSO Logistic Regression

In this section, we tackle the problem of extracting knowledge on the most relevant genes involved in the classification tasks of COVID-19 severity. The search for such discriminating genes can be interpreted in the classical framework of feature selection analysis. Anyway, because of the specificities of the problem, i.e. the high dimensionality compared to the sample size and the complex classification task, we devised a customized feature selection approach based on the LASSO logistic regression model.

3.4.1 Theoretical framework of feature selection methodologies

Let us consider a predictor function $\tilde{f}$ going from the d-dim space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 ... \times \mathcal{X}_d$ to the 1-dim target space $\mathcal{Y}$:

$$\tilde{f} : \mathcal{X} \to \mathcal{Y} : \boldsymbol{x} \to y.$$

In our context, the features assume the meaning of genes, whereas the samples are the patients involved in the study. The space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 ... \times \mathcal{X}_d$ is denoted *input space*, whereas the *hypothesis space* is the space of all the possible functions

$\tilde{f}$ mapping the inputs to the output. One peculiarity of our problem is that the number of features (d) is substantially higher than the number of samples (N); in such circumstances, the inference process becomes challenging since the input space and the hypothesis space are very large and hard to explore with the available samples.

A classical way to tackle this problem is by reducing the dimensions of the input space through feature selection techniques. In this context [42] provides a review of the principal techniques: *feature extracting* methods, *filtering* methods and *wrapper* methods.

Anyway, the feature extracting methods are not adapted here since their goal is to aggregate the original features into a smaller set, usually non-interpretable, of synthetic features (e.g., the principal components). The filter approach instead, selects the features independently from the potential induction algorithm, e.g., by performing univariate tests, with the disadvantage of not facing the complexity of the classification problem. Finally, wrapper methods consist of exploring the entire powerset of the d features set and in selecting the subset providing the better performances for the task; here the problem lies in the computational cost for the high dimensionality of the feature set.

Another way to approach the so-called *large d, small N* problem is through the introduction of constraints to the hypothesis space. This is classically done by adding a regularization term to the empirical loss. For instance, the well-known LASSO regularization (Least Absolute Shrinkage and Selection Operator) introduces the sum of the absolute values of the model parameters as a penalization term (see [51]). This regularization has the effect of shrinking the estimated coefficients to zero, providing a feature selection method for sparse solutions within the classification tasks. Feature selection methods based on such regularization structures are called *Embedded methods* and are the most adapted for our scope because they are computationally treatable and strictly connected with the classification task of the ML algorithm.

As a baseline algorithm for the embedded method, we adopted the logistic regression model that is a state-of-the-art ML algorithm for binary classification tasks with a probabilistic interpretation. It models a function ($\log \frac{t}{1-t}$) of the success probability of a binary target variable, denoted as $\tilde{y} = P(\tilde{Y} = 1|X = \boldsymbol{x})$, as the linear combination of the input features:

$$\log \frac{\tilde{y}}{1-\tilde{y}} = \beta_0 + \sum_{i=1}^{d} \beta_i \, x_i, \tag{3.1}$$

where $\boldsymbol{x}$ is the input vector, β_i are the coefficients of the regression and X and $\tilde{Y}$ are the random variables representing the inputs and the predicted output respectively. The chosen loss function to be minimized can be given by the binary cross-entropy loss:

$$-\sum_{n=1}^{N} y_n \log \tilde{y}_n \cdot (1-y_n) \log (1-\tilde{y}_n), \tag{3.2}$$

where y_n is the true label of the n-th instance.

As already mentioned, in order to enforce both the sparsity and the interpretability of the results, the model is trained with the additional LASSO regularization term:

$$\lambda \sum_{i=1}^{d} |\beta_i|. \tag{3.3}$$

where λ is the hyperparameter driving the strength of the LASSO regularization term.

Then, the absolute value of the surviving weights of the logistic regression algorithm can be interpreted as the feature importances of the subset of most relevant genes for the task ([58]) as explained in Section 3.4.2.

3.4.2 Interpretation of coefficients

As shown in Chapter 1, the interpretability of a model is a very important theme in ML. It distinguishes black-box models, where the user is not able to directly understand the inherent mechanism of the prediction, from white-box models where one can explain how the model behaves and which are the influencing variables.

The simplest example of white-box model[3] is linear regression, where the coefficients of the model provide a natural interpretation for the feature importance: an increase of one unit of the feature i determines an increase of β_i for the target variable. In this way, we are able to compare the importance of the different features.

Despite the logistic regression model, as opposed to linear regression, includes a non-linearity in the sigmoid function, there is still an interpretation of the coefficients similar to what happens for linear regression. The price to pay for the non-linearity is the need to introduce a new variable, the *odds*, already appearing in Eq. (3.1), given by the ratio between the success probability and the failure probability:

$$odds = \frac{\tilde{y}}{1-\tilde{y}}.$$

The odds has the same meaning of probability, but represented in the domain $[0, +\infty)$. For instance, an odds of 4 refers to the probability $\tilde{y} = 0.8$ and tells us how many times the success probability (0.8) is higher than the failure one (0.2).

In terms of odds, Eq. (3.1) can be written as:

$$odds = e^{\beta_0 + \dots \beta_i x_i + \dots + \beta_d x_d}. \tag{3.4}$$

Now, let us compute how many times the odds obtained by increasing 1 unit the feature i is higher than the base odds. Mathematically, we can evaluate the fraction:

$$\frac{odds(x_i+1)}{odds} = \frac{e^{\beta_0 + \dots + \beta_i (x_i+1) + \dots + \beta_d x_d}}{e^{\beta_0 + \dots + \beta_i x_i + \dots + \beta_d x_d}} \tag{3.5}$$

$$= e^{\beta_i}, \tag{3.6}$$

and

$$odds(x_i+1) = e^{\beta_i}\ odds. \tag{3.7}$$

[3]Actually a distinction is possible between white-box models with many features, sometimes called grey-box models.

So, by increasing the feature of one unit, we obtain a new odds given by the base one multiplied by a factor e^{β_i}. We conclude that the factor e^{β_i}, that is also called *Odds ratio*, represents the contribution of the feature i, in a multiplicative way, to the odds of the classification task.

3.4.3 Fitting of the LASSO Logistic Regression model

The fundamental hyper-parameter of the LASSO logistic regression model is the strength of the LASSO term (λ in Eq. (3.3)) that we tune with a grid search procedure on the 10-folds cross-validation average accuracy. We recall that the k-fold cross-validation provides the partition of the dataset into k batches, then exploits $k-1$ batches for the training and the remaining batch as a test, by repeating this procedure k times.

In the grid search method, a cross-validation procedure is carried out for each value of the regularization hyperparameter in the range $[10^{-3}, \ldots, 10^{2}]$. The optimal regularization parameter is then chosen by selecting the most parsimonious parameter whose cross-validation average accuracy falls in the range of the best one along with its standard deviation.

Once the regularization parameter has been calibrated, the model is fitted on the train cohort, and the class unbalancing is tackled by penalizing the misclassification of minority class with a multiplicative factor inversely proportional to the class frequencies. The data pre-processing is coded in *Python*, whereas the logistic regression model is included in the *scikit-learn* module with the *liblinear* coordinate descend optimization algorithm.

3.4.4 Benchmark methods

During the phase of model definition, other benchmark methodologies for the embedded feature selection have been analyzed. We report here a brief description of the tested methodologies.

- *Support Vector Machine* is an ML algorithm that solves the classification task by exploiting the geometrical properties of the feature space. In particular, it aims at finding the optimal hyperplane in the feature space that separates the classes by leaving the maximum margin between its support vectors. Also in this case, in order to enforce both the sparsity and the interpretability of the results, it is possible to exploit the LASSO regularization [141]. Moreover, when the chosen kernel of the feature space is the linear one, the magnitude of the model's weights are related to the importance of the features and can be used as a ranking criterion [142].

- *Extremely Randomized Trees* is a tree-based ensemble method for supervised classification [143]. The "random" splitting strategy at each node involves a set of ($m < d$) features randomly selected and one casual cutpoint for each feature, so the best split is selected based on the chosen criterion. Contrary to random forest, the method uses the whole learning sample. In this context, a well-known way to compute feature importance is the *mean decrease impurity index* [144]. For a particular feature, it is computed as the total decrease of impurity on the nodes where the feature is exploited for the split (weighted by

the proportion of samples reaching that node) averaged over all trees of the ensemble.

- Among the ML models providing rules [145], *RuleFit* [146] is an algorithm working in two stages, firstly by selecting a wide set of rules with a tree ensemble method (see Figure 3.7) and then by reducing the set with a penalized criterion applied to a classification model with a linear combination of rules. The phase of rule creation is based on the conjunction of the splits of decision trees included in a gradient boosting classifier. Boosting creates new estimators of an ensemble in an adaptive way, by focusing effort on those samples that are mislabelled by the previous estimators.

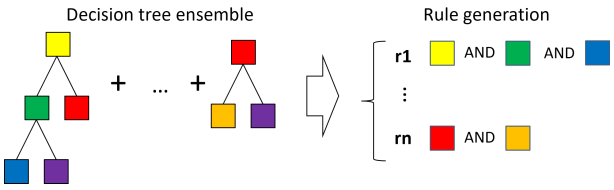


Figure 3.7: Representation of the basic idea of RuleFit algorithm.

Anyway, none of the aforementioned benchmark models have been proved to outperform LASSO logistic regression model, neither in terms of stability of the result, ease of hyperparameter tuning nor as regards to the interpretability of the extracted features.

3.5 Sex differences in COVID-19 severity: the case of Androgen Receptor

In the present section, we report the first analysis carried out by exploiting the LASSO logistic regression on the genetic dataset of COVID-19. The analyses are reported in [147].

First of all, it is important to mention that in the last two years, epidemiological studies indicated that men and women are similarly infected by COVID-19, but the outcome is less favorable in men, independently of age [148]. In addition, several studies also showed that patients with hypogonadism tend to be more severely affected. In order to deepen the sex differences in COVID-19, we firstly investigated from a genetic point of view the potential effect on COVID-19 severity of the poly-amino acids repeat polymorphisms, such as the polyQ tract of the Androgen Receptor (AR).

3.5.1 WES data representation for poly-amino acids triplet repeats

A total of 40 genes with 43 triplet repeat regions were taken from UniProtKB. For any of these genes, a feature X_{ij} was defined as equal to 1 if for the i-th patient the j-th gene presented a deletion in the region characterized by repeated triplets.

Thus, the input features of the LASSO logistic regression were the features representing poly-amino acids triplet repeats as well as gender, comorbidity[4] and the age, the latter as a continuous variable normalized between 0 and 1.

Cases were selected according to the following inclusion criteria: endotracheal intubation or CPAP/biPAP ventilation (categories 3 and 4 as defined in 3.2). As controls, participants were selected using the sole criterion of being oligo-asymptomatic not requiring hospitalization (category 0). Cases and controls represented the extreme phenotypic presentations of the GEN-COVID cohort.

3.5.2 Results of the LASSO logistic regression

The LASSO logistic regression model was fitted on the cohort as described in Section 3.4.3. As expected, the grid search curve of the cross-validation score for the LASSO logistic regression model (Figure 3.8, panel C) shows a maximum for an intermediate value of the L1 regularization parameters, and the chosen parameter, once considering the standard deviation of the performance on the 10 folds (see Section 3.4.3) is 6.31.

With this calibration setting, the 10-fold cross-validation provides good average performances in terms of accuracy (77%), precision (81%), sensitivity (77%), and specificity (78%) as shown in Figure 3.8, panel D. The confusion matrix is reported in Figure 3.8, panel B, whereas the Receiver Operating Characteristic (ROC) curve (Figure 3.8, panel E) provides an Area Under the Curve (AUC) score of 86%.

Besides the good performance of the model, the main result is given by the analysis of the coefficients of the features selected by the models. In Figure 3.8 (Panel A), the histogram of the LASSO logistic regression weights represents the importance of each feature for the classification task. The positive weights reflect a susceptible behavior of the features to the target COVID-19 disease severity, whereas the negative weights a protective action. The analysis identified AR as the only protective gene, suggesting an important contribution of this factor to the genetic of the disease. The calculated odds ratio of AR short repeats ($\leq$ 22) is 0.79 i.e. protective. Therefore, the odds ratio of long repeats ($\geq$ 23) is 1/0.79= 1.27 i.e. severity.

In order to confirm the results, we fitted a logistic regression without regularization for the features selected by the LASSO embedded method. The performances for the selected set of features (age, gender, comorbidity, and AR gene) are 79% accuracy, 81% precision, 81% sensitivity, 78% specificity, 88% roc-auc. The model shows a slight decrease of almost all the performance measures when the AR gene is removed from the set (accuracy -1.2%, precision -1.3%, sensitivity -1.4%, specificity -1.2%, ROC-AUC +0.3%).

Furthermore, the logistic regression on the male cohort with the AR gene alone provides results quite higher than the random guess with accuracy at 58%, precision 71%, sensitivity 64%, specificity 55%, and ROC-AUC 55%.

Finally, the association between long polyQ alleles ($\geq$ 23) and severe clinical outcome (p= 0.024) was also validated in an independent cohort of Spanish men $<$ 60 years of age (p=0.014).

[4]Comorbidities were defined as the presence of one or more clinical conditions (i.e. cardiac, endocrine, neurological, neoplastic diseases) at the time of infection. The variable is coded as 1 if there was at least one comorbidity.

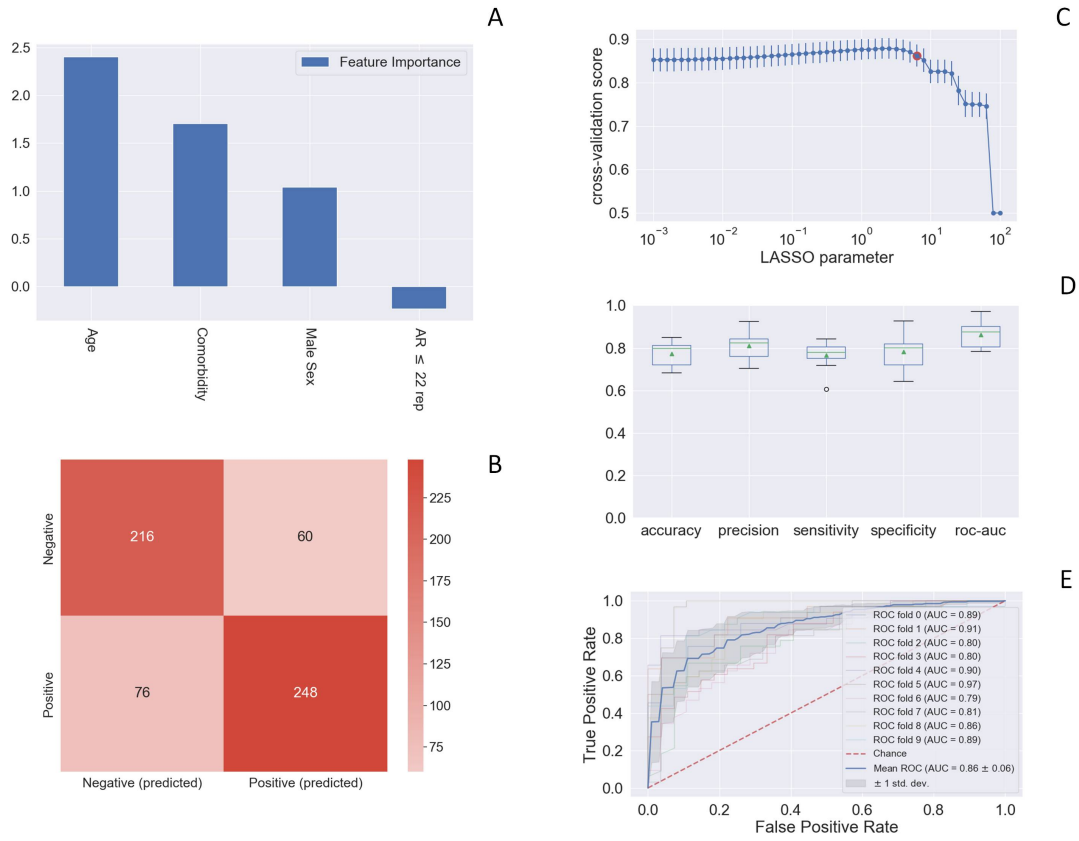


Figure 3.8: Panel A: The histogram of the LASSO logistic regression weights represents the importance of each feature for the classification task. Panel B: Confusion matrix for the aggregation of the logistic regression predictions in the 10 folds of the cross-validation. Panel C: Cross-validation ROC-AUC score for the grid of LASSO regularization parameters; the error bar is given by the standard deviation of the score within the 10 folds; the optimal regularization parameter is chosen by selecting the most parsimonious parameter whose cross-validation score falls in the range of the best one along with its standard deviation (red point). Panel D: Boxplot of accuracy, precision, sensitivity, specificity, and ROC-AUC score for the 10-fold of the cross-validation. The box extends from the Q1 to Q3 quartile, with a line at the median (Q2) and a triangle for the average. Panel E: ROC curve for the 10 folds of the cross-validation. The image is taken from the original paper [147].

3.5.3 Androgen Receptor genetic contribution to COVID-19 severity

Firstly, we recall that, from the biological point of view, AR contains a highly variable polyglutamine repeat (poly-Q) located in the N-terminal domain of the protein, spanning from 9 to 36 glutamine residues in the normal population [149]. AR polyQ length correlates with receptor functionality, with shorter polymorphic glutamine repeats typically associated with higher and longer PolyQ tracts with lower receptor activity [149]. Moreover, AR is expressed in both males and females, but the bioavailability of its ligands T and dihydroT (DHT) differs significantly,

being much higher in males.

In the present analysis, we have demonstrated that shorter polymorphic glutamine repeats (<22) confer protection against life-threatening COVID-19 in a subpopulation of individuals with age < 60 years. In addition, testosterone was proved to be higher in subjects with AR long-polyQ, possibly indicating receptor resistance ($p = 0.042$ Mann-Whitney U test). We can state that inappropriately low serum testosterone levels among carriers of the long-polyQ alleles ($p = 0.0004$ Mann-Whitney U test) predicted the need for intensive care in COVID-19 infected men.

Moreover, the relationship between the AR polyQ repeat size and 5 laboratory markers of immunity/inflammation, including CRP, Fibrinogen, IL6, CD4, and NK count has been tested. We found that older (≥ 60) males with AR polyQ tract ≥ 23 have a higher (55.92 versus 48.21 mg/dl) mean value of CRP (p-value 0.018, not accounting for multiple testing) and lower mean value of Fibrinogen and a trend of higher IL6. This is in agreement with the known anti-inflammatory action of testosterone.

In conclusion, these first results may contribute to designing reliable clinical and public health measures by studying the genetic of the patients. Basically, we suggest that sizing the AR poly-glutamine repeat has important implications in the diagnostic pipeline of patients affected by life-threatening COVID-19 infection. Most importantly, our studies open to the potential of using testosterone as adjuvant therapy for patients with severe COVID-19 having defective androgen signaling, defined by this study as ≥ 23 PolyQ repeats, and inappropriately low levels of circulating androgens.

3.6 The Mendelian face of COVID-19: rare variants of TLR7

In the previous section, we have seen how the LASSO logistic regression model was successfully used to identify a gene's common variant that is predictive for the severe or the mild COVID-19 phenotype. Here we report the analyses carried out by applying LASSO logistic regression methodology to the rare genetic variants. The results of the present section are based on [150]. In particular, rare variants of the X chromosome with frequency $\leq 1\%$ in the European Non-Finnish population were considered. In the chosen Boolean representation the gene was set to 1 if it included at least a missense, splicing, or loss of function rare variant, and 0 otherwise. In so doing we have collapsed the information of the genetic variants into a gene level.

Cases were selected according to the following inclusion criteria: i. male gender; ii. young age (< 60 years); iii endotracheal intubation or CPAP/biPAP ventilation (categories 3 and 4 as defined in Section 3.2). As controls, participants were selected using the sole criterion of being oligo-asymptomatic not requiring hospitalization (category 0). As in the previous analysis, cases and controls represented the extreme phenotypic presentations of the GEN-COVID cohort.

As usual, after the fitting of the model, the performances are evaluated through the confusion matrix of the aggregated predictions in the 10 folds of the cross-validation of Figure 3.9 (Panel C) and with the boxplot (Panel D) of accuracy (60% average value), precision (59%), sensitivity (75%), specificity (43%), and ROC-AUC score (68%).

It is important to observe that TLR7 is picked up by LASSO logistic regression

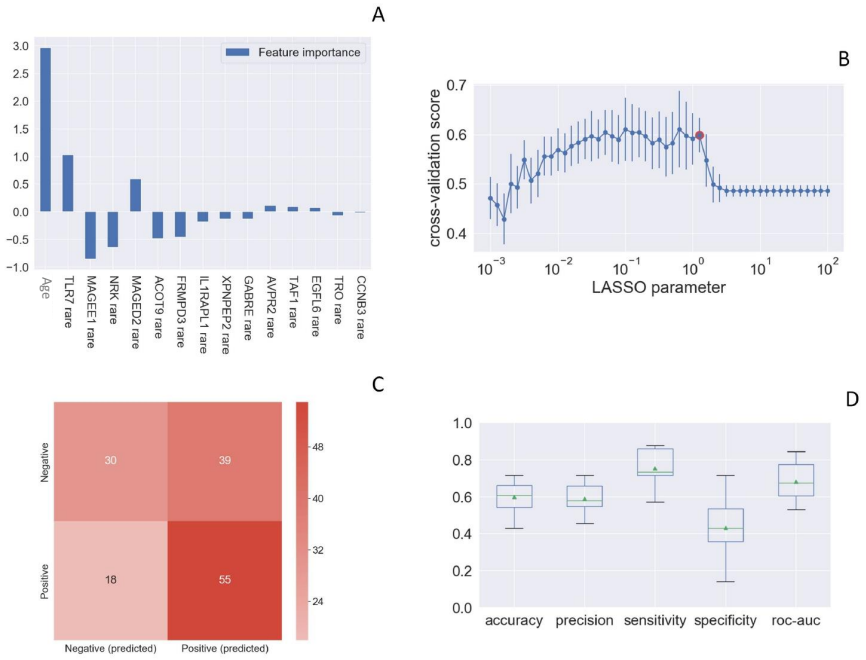


Figure 3.9: Panel A: The histogram of the LASSO logistic regression weights represents the importance of each feature for the classification task. Panel B: Cross-validation ROC-AUC score for the grid of LASSO regularization parameters. the red circle (1.26) corresponds to the parameter chosen for the fitting procedure. Panel C: Confusion matrix for the aggregation of the logistic regression predictions in the 10 folds of the cross-validation. Panel D: Boxplot of accuracy, precision, sensitivity, specificity, and ROC-AUC score for the 10 folds of the cross-validation. The box extends from the Q1 to Q3 quartile, with a line at the median (Q2) and a triangle for the average. The image is taken from the original paper [150].

as one of the most important genes on the X chromosome (see Figure 3.9 Panel A). Then, in order to confirm the significance of the association between TLR7 variants and COVID severity, the Fisher's Exact Test was used (p value 0.0037).

3.6.1 Toll-like receptors role in COVID-19

From a biological point of view, recent evidence has suggested a fundamental role of interferon genes in modulating immunity to SARS-CoV-2 [149]. In particular, rare variants have recently been identified in the interferon type I pathway that are responsible for inborn errors of immunity in a small proportion of patients and auto-antibodies against type I interferon genes in up to 10% of severe COVID-19 cases [151]. On the other side, toll-like receptors (TLRs) are crucial components in the initiation of innate immune responses to a variety of pathogens, causing the production of pro-inflammatory cytokines (TNF-α, IL-1, and IL-6) and type I and II Interferons (IFNs), that are responsible for innate antiviral responses. In particular, innate immunity is very sensitive in detecting potential pathogens, promoting synthesis and release of type I and type II IFNs in addition to a number of other proinflammatory cytokines, and leading to a severe cytokine release syndrome which

may be associated with a fatal outcome. Interestingly, among the different TLRs, TLR7 recognizes several single-stranded RNA viruses including SARS-CoV-2 [152].

Following the statistical analyses reported above, the functional gene expression profile analysis demonstrated a reduction in TLR7-related gene expression in patients compared with controls demonstrating an impairment in type I and II IFN responses. In conclusion, young males with TLR7 loss-of-function variants and severe COVID-19 represent a subset of male patients contributing to disease susceptibility in up to 2% of severe COVID-19.

3.7 Common variants and discovery of TLR3 polymorphism

As seen in the previous section (3.6), toll-Like Receptors (TLRs) are a class of proteins that play a key role in host innate immunity, causing the production of pro-inflammatory cytokines (TNF-α, IL-1, and IL-6) and type I and II Interferons (IFN), that are responsible for innate antiviral responses.

In order to evaluate the possible involvement of common polymorphism such as TLRs's ones, the LASSO logistic regression model is applied to a Boolean representation of common variants (see [153] for the full analysis). Common bi-allelic polymorphisms are defined as combinations of two polymorphisms, each with Minor Allele Frequency (MAF) above 1%, with a frequency above 5% in the cohort. So, the input features are the common bi-allelic polymorphisms from whole-exome sequencing (see 4.3 for the description of the Boolean representation) as well as gender, and age, the latter as a continuous variable normalized between 0 and 1.

We used a cohort of $1,319$ subjects from the Italian GEN-COVID Multicenter study, infected with SARS-CoV-2 diagnosed by RT-PCR on a nasopharyngeal swab, as described in 3.2. As in the previous analyses, cases were defined as patients needing endotracheal intubation or CPAP/biPAP ventilation (category 3/4). Controls were oligo-asymptomatic subjects not requiring hospitalization (category 0).

Again, the grid search curve of the cross-validation score (Figure 3.10, Panel D) shows a maximum of the regularization parameter in 10. With this calibration setting, the 10-folds cross-validation provides good performances in terms of accuracy (73%), precision (74%), sensitivity (73%), and specificity (73%) as shown in Figure 3.10 Panel C. The confusion matrix is reported in Figure 3.10 Panel D, whereas the Receiver Operating Characteristic (ROC) curve (Figure 3.10 Panel E) provides an Area Under the Curve (AUC) score of 80%.

As shown in the histograms of Figure 3.10 (Panel A) we have identified the L412F polymorphism (rs3775291; c.1234C>T) in TLR3 as a severity marker for the COVID-19 disease.

In order to confirm the role of the polymorphism, we subdivided patients into two categories, those having the polymorphism in heterozygous or homozygous state and those homozygous for the wild-type allele. We found that the prevalence of L412F polymorphism is significantly higher in cases compared to controls (p-value $2.8\,10^{-2}$). The global allele frequency of L412F in our cohort (cases and controls) is 29.38%, comparable to the allele frequency of 29.79% reported in the European (non-Finnish) population in the gnomAD database (`https://gnomad.broadinstitute.org/`). The identified frequencies were in Hardy-Weinberg equilibrium.

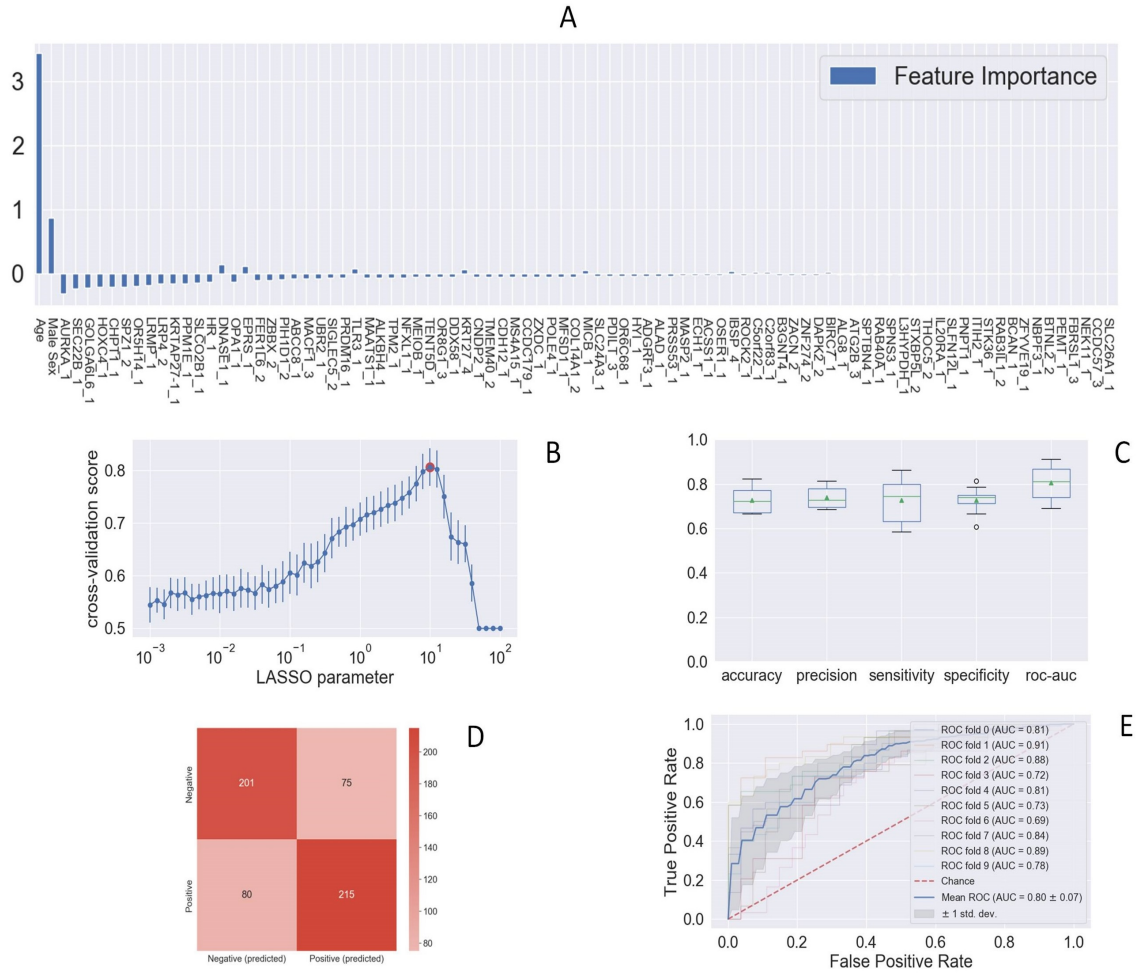


Figure 3.10: Panel A: The histogram of the LASSO logistic regression weights represents the importance of each feature for the classification task. Panel B: Cross-validation ROC-AUC score for the grid of LASSO regularization parameters; the optimal regularization parameter is chosen by selecting the one with the highest cross-validation score (red point). Panel C: Boxplot of accuracy, precision, sensitivity, specificity, and ROC-AUC score for the 10-fold of the cross-validation. Panel D: Confusion matrix for the aggregation of the logistic regression predictions in the 10 folds of the cross-validation. Panel E: ROC curve for the 10 folds of the cross-validation. The image is taken from the original paper [153].

3.7.1 Biological meaning of TLR3's biological polymorphism

The L412F polymorphism has an overall allele frequency of about 20%, ranging from 30% in European to 0.88% in African (mainly sub-Saharan) populations [154]. It is intriguing that a COVID-19-free population such as sub-Saharan has a very low frequency (0.88%) of this polymorphism and that Asian (26.97%) and European (30.01%) have a much higher frequency. The variant protein with phenylalanine is under-represented on the cell surface, it is not efficiently secreted into the culture medium when expressed as the soluble ectodomain, and it has reduced capability to

activate the expression of TLR3-dependent reporter constructs [154].

In the presented analysis, we have identified the second protein-encoding polymorphism that modulates COVID-19 outcome (see Section 3.5 for the first polymorphism). These results indicate that L412F polymorphism in the TLR3 gene makes males, in whom after puberty testosterone lowers TLR3 expression, at risk of severe COVID-19 in a context of a polygenic model. Moreover, based on the impairment of autophagy, these data provide a rationale for reinterpreting clinical trials with HCQ stratifying patients by L412F. Finally, the combination of L412F in TLR3 and specific HLA class II haplotypes may put male patients at risk of post-acute sequelae of SARS-CoOV-2 infection (PASC) pointing to the need for an appropriate follow-up.

Our experiments suggest an important role of autophagy downstream of the TLR3 receptor, possibly affecting TNFa production and susceptibility to infections, including SARS-CoV-2.

3.8 Yet another polymorphism: SELP Asp603Asn and severe thrombosis in COVID-19 males

From the analyses of LASSO logistic regression on common polymorphism (Section 3.5 and 3.7) we had confirmation that age has a powerful impact on the model effectiveness, potentially hiding the genetic effect of common variants. In order to mitigate the effect of age, the Ordered Logistic Regression (OLR) model was applied to the clinical WHO gradings, stratified by sex and adjusted by age. This procedure, thoroughly explained in the following Section 4.2, aims at defining cases and controls by getting rid of the effect of both gender and age. The analysis reported here is presented in [155].

After the phenotype definition, the usual LASSO logistic regression model has been applied to the male subset consists of 513 COVID-19 patients: 236 severe COVID-19 patients (cases as resulting from OLR) and 277 controls. The input features are given by the Boolean representation of homozygous common bi-allelic polymorphism of autosomal genes (see Section 4.3).

As shown in Figure 3.11 (Panel A), the LASSO identified SELP as a key player for severity and thromboembolism in severe COVID-19. Association was following confirmed by Chi-Square Test.

In conclusion, we identified SELP rs6127 polymorphism as the elusive genetic factor predisposing COVID-19 patients to thromboembolism leading to life-threatening disease. We showed that predisposition increases if the protective effect of testosterone is lost either by age or because of additional genetic factors such as poly Q$\geq$23 in the AR gene. This knowledge provides a rationale for repurposing anti-P-selectin monoclonal antibodies as personalized adjuvant therapy in men affected by COVID-19.

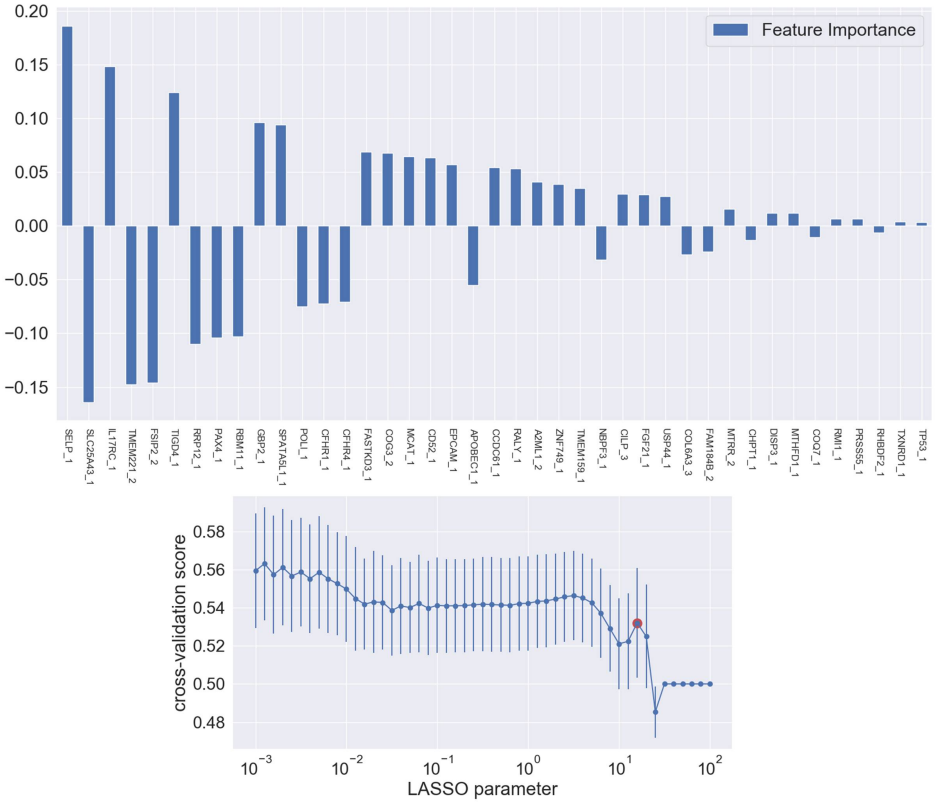


Figure 3.11: Upper Panel: selected features by the LASSO logistic regression. The upward histogram means positive weights, i.e the specific variant at the specific locus (feature) contribute to severity of COVID-19. SELP_1_homo = homozygous genotype Asn/Asn at the polymorphic locus Asp603Asn (rs6127). The downward histograms means negative weights, contributing to mildness of COVID-19. COG3_1_homo = homozygous genotype Ser/Ser at the polymorphic locus Leu825Ser (rs3014902). COG3 gene encodes for a vesicle docking protein involved in viral trafficking. TMEM221_2_homo = homozygous genotype Ala/Ala at the polymorphic locus Thr66Ala (rs4808641). TMEM221 gene encodes for a transmembrane protein. The image is taken from the original paper [155]. Lower Panel: grid search results for the fitted model.

Chapter 4

Disentangling complex genetic diseases: an explainable AI model for COVID-19

In the previous chapter, very important genetic information on COVID-19 has been extracted through Machine Learning (ML) methodologies. Several common polymorphism (see Sections 3.5, 3.7 and 3.8) and rare variants (see Section 3.6) have been selected and individually studied, given rise to potential patients' treatment. Anyway, a comprehensive framework of the COVID-19 genetic disease was still missing. In the present chapter, we describe the ML model developed in order to represent and aggregate the effects of all the genetic components into an interpretable predictive model.

The ultimate goal of the proposed model is extremely significant for public health: it aims at providing, on the basis of the genetic, a binary classification between severe and non-severe COVID-19 affected patients, where patients were considered severe when hospitalized and receiving any form of respiratory support. The focus on this target variable is motivated by the practical importance of rapidly identifying which patients are more likely to require oxygen support, in an effort to prevent further complications. As already mentioned in Chapter 3, beyond what this method can help us understand regarding the role of host genetics in COVID-19 susceptibility and the potential implications for clinical and public health responses, the model also has strong potential for understanding the role of host genetics in other complex disorders. As we move into an era of precision, patient-centric medicine - an era being propelled by the lurking ubiquitous character of COVID-19 - this method can help us tailor treatments to the specific needs of individual patients.

From the very beginning, interpretability has been a guiding principle in the definition of the ML model, as only a readily interpretable model can provide useful and reliable information for clinical practice while also contributing significantly to diagnostic, and therapeutic targeting. With this respect, in the recent literature, [156] provides a review of the different strategies for interpreting ML models, and examples of how these strategies have been applied in genetics and genomics. In the same framework [157] provides a guideline for the selection of ML methods and their practical applications. Instead, [158] discusses the recent applications

of supervised ML to population genetics that outperform competing methods, and describes promising future directions in this area.

Anyway, the high dimensionality of host genetic data poses a serious challenge to evident and reliable interpretability. So far, the development of a robust predictive model able to make a direct association between single variants and disease severity grading compared to a much smaller number of individual patients has proven to be too complex and ultimately unreliable. In order to address the complexity, an enriched gene-level representation of host genetic data was modeled by an ensemble of ML algorithms.

The analyses reported in this chapter are detailed in the original papers [159, 2].

4.1 A post Mendelian paradigm for complex genetic disease

The complexity of COVID-19, as emerged also in Chapter 3, immediately suggests that both common and rare variants contribute to the likelihood of developing a severe form of the disease. However, the contribution of common and rare variants to the severe phenotype is not expected to be the same. A single rare variant that impairs the protein function might cause a severe phenotype by itself after viral infection, while this is not so probable for a common polymorphism, which is likely to have a less marked effect on protein functionality.

These observations led to the definition of a score, named Integrated PolyGenic Score (IPGS), that includes information regarding the variants at different frequencies with the following formula:

$$
\begin{aligned}
IPGS = {} & n_C^s - n_C^m && (4.1)\\
& + F_{LF} \cdot (n_{LF}^s - n_{LF}^m) && (4.2)\\
& + F_R \cdot (n_R^s - n_R^m) && (4.3)\\
& + F_{UR} \cdot (n_{UR}^s - n_{UR}^m) && (4.4)
\end{aligned}
$$

In the above equation, the symbol n is used to indicate the count of input features that promotes the severe outcome (superscript s) or that protects from a severe outcome (i.e. mildness with superscript m) and with genetic variants having Minor Allele Frequency (MAF)$\geq$ 5% (common, subscript C), 1% <MAF$\leq$ 5% (low-frequency, subscript LF), 0.1%<MAF$\leq$ 1% (rare, subscript R), and MAF<0.1% (ultra-rare, subscript UR).

For the sake of clarity, we list the definition of each variable in the following:

- n_C^s count of features representing common genetic variants promoting severe outcome;
- n_C^m count of features representing common genetic variants protecting from severe outcome;
- n_{LF}^s count of features representing low frequency genetic variants promoting severe outcome;
- n_{LF}^m count of features representing low frequency genetic variants protecting from severe outcome;

- n_R^s count of features representing rare genetic variants promoting severe outcome;
- n_R^m count of features representing rare genetic variants protecting from severe outcome;
- n_{UR}^s count of features representing ultra-rare genetic variants promoting severe outcome;
- n_{UR}^m count of features representing ultra-rare genetic variants protecting from severe outcome.

The weighting factors F_{LF}, F_R, and F_{UR} were included to model the different penetrant effects of low-frequency, rare, and ultra-rare variants, compared to common variants. Thus, the 4 terms of Eq. (4.1), Eq. (4.2), Eq. (4.3), Eq. (4.4) can be interpreted as the contributions of common, low-frequency, rare, and ultra-rare variants to a score that represents the genetic propensity of a patient to develop a severe form of COVID-19.

The definition of the terms of the IPGS formula requires 4 separate steps, detailed in the following subsections (see also Figure 4.1):

1. the definition of a severity phenotype adjusted by age and sex, with the purpose of facilitating the extraction of features associated with the genetic basis of COVID-19 severity (Figure 4.1 Panel A). In particular, the adjusted phenotype was computed with an ordered logistic regression model as detailed in Section 4.2;

2. the conversion of genetic variants into Boolean features representing the presence of variants in different frequency ranges in each gene (described in Section 4.3). This step led to the definition of 12 separate sets of input features. Specifically, 9 sets of input features are designed to represent in a binary way the autosomal dominant hereditary model, the autosomal recessive and X-linked models of inheritance for both ultra-rare, rare, and low frequency variants. In the case of common variants, the same 3 sets of Boolean features representing the autosomal dominant, autosomal recessive, and X-linked models of inheritance were used; however, instead of simply defining the binary variables as "absence/presence of variants", the absence/presence of variant combinations was tested (Figure 4.1 Panel B).

3. The selection of those features that are associated with disease severity. In fact, despite the Boolean representation of the genetic variability described in the previous point significantly reduces the dimensionality of the problem, the number of input features is still orders of magnitude higher than the number of patients. In order to further reduce the number of input features, the already seen feature selection strategy based on logistic models with the Least Absolute Shrinkage and Selection Operator (LASSO) regularization was employed (Figure 4.1 Panel C, see Section 4.4.1 for details). Separate logistic models with LASSO regularization were trained for the 12 sets of Boolean features for predicting COVID-19 severity.

4. Given the subset of features, and once the counting variables (denoted with the letter n) in IPGS formula (Figure 4.1 Panel D) are defined, we carry out the optimization of the weighting factors appearing in Eq. (4.2), Eq. (4.3), and Eq. (4.4). This step of the model's fitting requires the maximization of a metric, i.e. the Silhouette score, measuring the goodness of the clustering between severe and non-severe patients on the basis of the IPGS scores (Figure 4.1 Panel E). This phase of the model's fitting is described in Section 4.4.2.

In order to train the model, the dataset was divided into a training set and a testing set (90/10), and the entire procedure described by these 4 steps was performed using only samples in the training set extracted from the GEN-COVID dataset.

Then, once the IPGS formula has been completely defined,

5. a final logistic regression model using as features: Age, Sex and IPGS, is defined with the purpose of predicting a binary classification of patients into mild and severe cases, where a patient is considered severe if hospitalized and receiving any form of respiratory support (Figure 4.1 Panel F). For this step of the analysis, we use a logistic regression model without regularization and the performance of the model has been extensively tested, as described in Section 4.7, with many independent test cohorts.

In Figure 4.1 we report a summary of the steps followed for the definition of the interpretable predictive model.

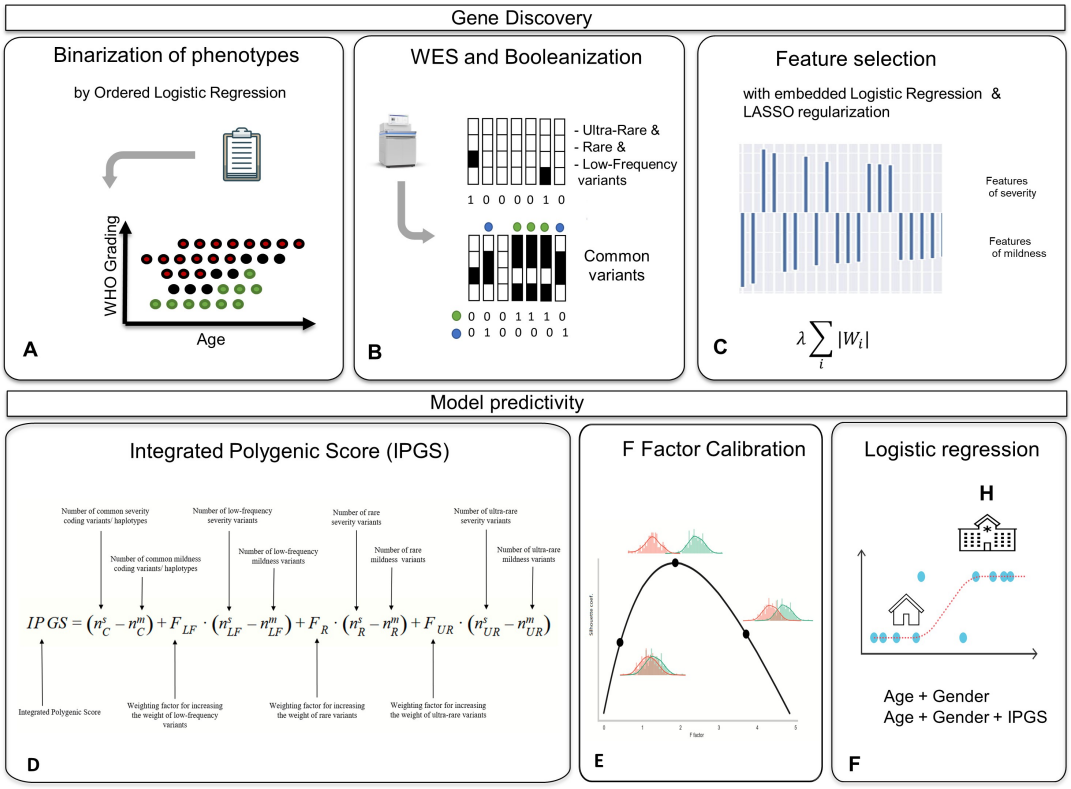


Figure 4.1: Steps for the model's fitting. Panel A: Clinical severity classification into severe and mild cases was performed by Ordered Logistic Regression (OLR) starting from the WHO grading and patient age classifications. Panel B: WES data were binarized into 0 or 1 depending on the absence (0) or the presence (1) of variants (or the combination of two or more variants only for common polymorphisms) in each gene. Panel C: LASSO logistic regression feature selection methodology on multiple train-test splits of the cohort leads to the identification of the final set of features contributing to the clinical variability of COVID-19. Then the IPGS score is defined (Panel D) by optimizing the weighting factors (Panel E). Finally, the logistic regression model with IPGS is calibrated (Panel F). The image is taken from the original paper [2].

4.2 Ordered Logistic Regression model

In order to perform the feature selection phase with a clinical classification as independent as possible from age and sex, an Ordered Logistic Regression (OLR) model was used (step 1 as defined in Section 4.1).

Firstly, since *sex* has been proved to be a significant driver of the disease severity (see Chapter 3), the patients were first divided into males and females. Then for each sex, an OLR model was fitted by using the age to predict the WHO phenotype classification into 6 grades[1]. The OLR model was chosen among the other possible

[1]For the present analysis, the additional category of deceased was added with respect to the analyses carried out in Chapter 3 and the phenotype classification of Section 3.2.

ordinal models because it imposes a simple monotone relation between input feature and target variable, and again, it provides easily interpretable thresholds between the predicted classes.

Then, the patients with an OLR's predicted grading equal to the actual grading were excluded from the feature selection analyses, since we deemed that the effect of their genetic component did not deviate the phenotype from the expected one due to age. Instead, the remaining patients were divided into two classes depending on whether their actual phenotype from OLR was milder or more severe than the one expected for a patient of that age and sex. Overall, this procedure has two main advantages:

- the first is to isolate patients whose genetic factors are most important for predicting COVID-19 severity. In particular, the dominant effect of age may be a strong bias in the feature selection, where protective features are advantaged with respect to features correlated with severity. Moreover, the modeling through OLR can be done separately from male and female cohorts, helping us to disentangle the genetic factors for both sexes.

- The results of the OLR predictions give us a way to make the phenotype of the disease binary, and the binary classification problem is much easier to solve with respect to other kinds of tasks, especially when dealing with very high dimensional space. In addition, the binary classification's performances and their reliability are easier to check, e.g. via the confusion matrix, also when datasets of other cohorts are exploited.

The model based on all-threshold variant [160] of the *mord* Python package [161] is exploited. It is based on the assumption that mistakes should be differently weighted based on the ordinal relationship, e.g., if the target label is 2, predict 3 is better than predict 4. The chosen all-threshold variant model aims at fixing a set of thresholds θ_l with $l \in [1, d-1]$ as a decision function. Then, the loss function sums the differences between the predicted score $\tilde{f}(\boldsymbol{x})$ and the thresholds θ_l:

$$\mathcal{L}(y, \tilde{f}(\boldsymbol{x})) = \sum_{l=1}^{d-1} f\left(g(l, y)(\theta_l - \tilde{f}(\boldsymbol{x}))\right), \tag{4.5}$$

where $f(\cdot)$ is some margin penalty function, i. e. the cross-entropy loss in our case, whereas the term $g(l, y)$ takes into account the y target value:

$$g(l, y) = (2 \cdot [l \geq y] - 1).$$

In Figure 4.2 we report the result of the clinical classification adjusted by age for the patients of the training set of GEN-COV cohort. On the y axis, the grading according to patients' treatment is reported (5=deceased; 4=intubated; 3=CPAP, biPAP; 2=oxygen therapy; 1=hospitalized without oxygen support; 0=not hospitalized oligo-asymptomatic patients), while on the x axis, age is reported. The red dots represent subjects falling above the expected treatment outcomes according to age (hence considered severe), whereas the green dots are subjects falling below the expected treatment outcomes according to age (hence considered mild) and black dots are subjects matching the expected treatment outcomes according to age (hence considered intermediate).

It is worth noting the steeper "diagonal" composed by the black dots for the male cohorts with respect to the female one. For females almost up to 50 years, the predicted grading is 0, whereas the grading 3 is predicted for over-80 years females.

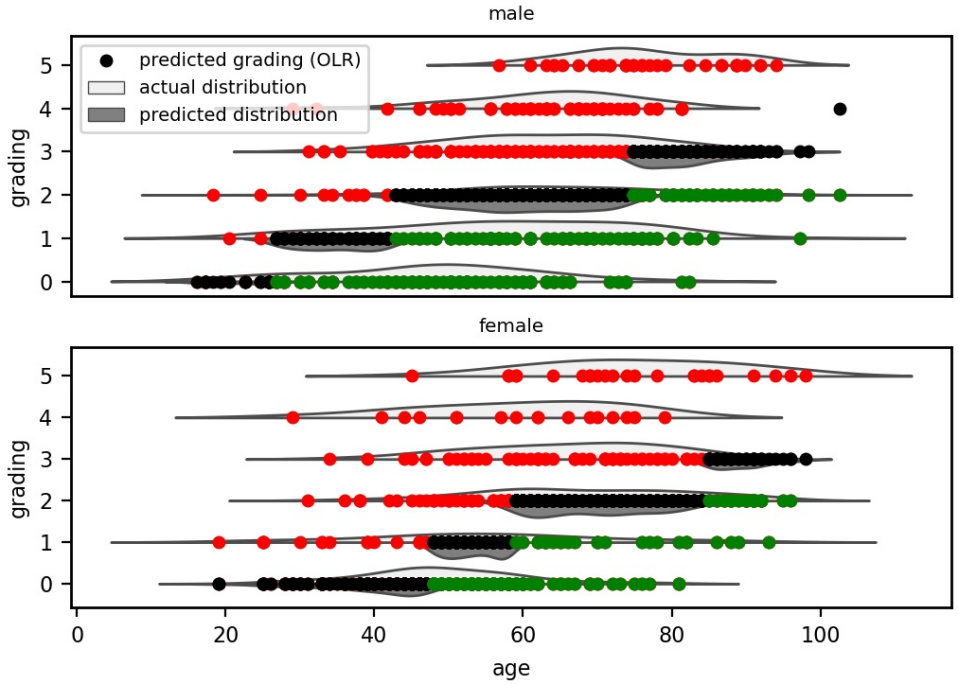


Figure 4.2: Results of the two Ordered Logistic Regression (OLR) models, stratified by sex, fitted using age to predict the ordinal grading (0, 1, 2, 3, 4, 5) dependent variable. On the y axis, the grading according to patients' treatment is reported. On the x-axis, age is reported. Red dots represent subjects falling above the expected treatment outcomes according to age (hence considered severe), green dots are subjects falling below the expected treatment outcomes according to age (hence considered mild) and black dots are subjects matching the expected treatment outcomes according to age (hence considered intermediate). The image is taken from the original paper [159].

4.3 Definition of the Boolean features

As already mentioned in Chapter 3, a big challenge of statistical models for complex genetic diseases is to find a way to map the whole genetic variability into a set of informative features (step 2 as defined in Section 4.1). In the proposed model, the genetic variants coming from WES were converted into 12 sets of Boolean features to better represent the variability at the gene-level. The decision to move from a variant representation to a Boolean gene base one is due to the necessity to reduce the feature space and to increase the interpretability of the biological meaning of the extracted features.

Firstly, any variant not impacting on protein sequence was discarded. Then the remaining variants were classified according to their Minor Allele Frequency (MAF) as reported in gnomAD for the reference population as:

- ultra-rare, MAF<0.1%;
- rare, 0.1%<=MAF<1%;
- low-frequency, 1%<=MAF<5%; and
- common, MAF>=5%.

Non-Finnish European (NFE) was used as a reference population. SNPs with MAF not available in gnomAD were treated as ultra-rare, and INDELs with frequency not available in gnomAD were treated as ultra-rare when present only once in the cohort and otherwise discarded as possible artifacts of sequencing.

4.3.1 Ultra-rare, rare and low frequency variants

Let us start with the ultra-rare variants, where 3 alternative Boolean representations were defined. They are designed to capture the autosomal dominant (AD), autosomal recessive (AR), and X-linked (XL) model of inheritance, respectively.

- The AD and AR representations includes a feature for all the genes on autosomes. These features are equal to 1 when the corresponding gene presented variants in the ultra-rare frequency range and 0 otherwise: at least 1 for the AD model, or 2 for the AR model.
- The XL representation includes only genes belonging to the X chromosome; these features are equal to 1 when the corresponding gene presents at least 1 variant in the ultra-rare frequency range and 0 otherwise.

The same approach was used to define AD, AR, and XL Boolean features for the rare (other 3 Boolean representations) and low-frequency variants, by ending up with 9 different Boolean representations for the non-common variants.

4.3.2 Common variants

A slightly more complicated approach was carried out for representing the common variants, able to better capture the presence of alternative haplotypes.

For each gene, all the possible combinations of common variants were computed. For instance, in the case of a gene belonging to an autosome with 2 common variants (named α and β), 3 combinations are possible (α, β, and $\alpha\beta$), and (consequently) 3 Boolean features were defined both for the AD and AR model. In the AR model, each of these 3 features was equal to 1 if all the variants in that particular combination were present in the homozygous state and 0 otherwise. The same rule was used for the AD model, but setting the feature to 1 even if the variants in that particular combination are in the heterozygous state.

In both models, AD and AR, a further feature was defined for each gene to represent the absence of any of the previously defined combinations. In the AD model this feature was equal to 1 if no common variant is present and 0 otherwise;

in the AR model, it is equal to 1 if no common variant is present in the homozygous state and 0 otherwise. The same approach was used to define the set of Boolean features for common variants in genes belonging to the X chromosome.

In Figure 4.3 we report a sketch for the Boolean representation of the genetic variants.

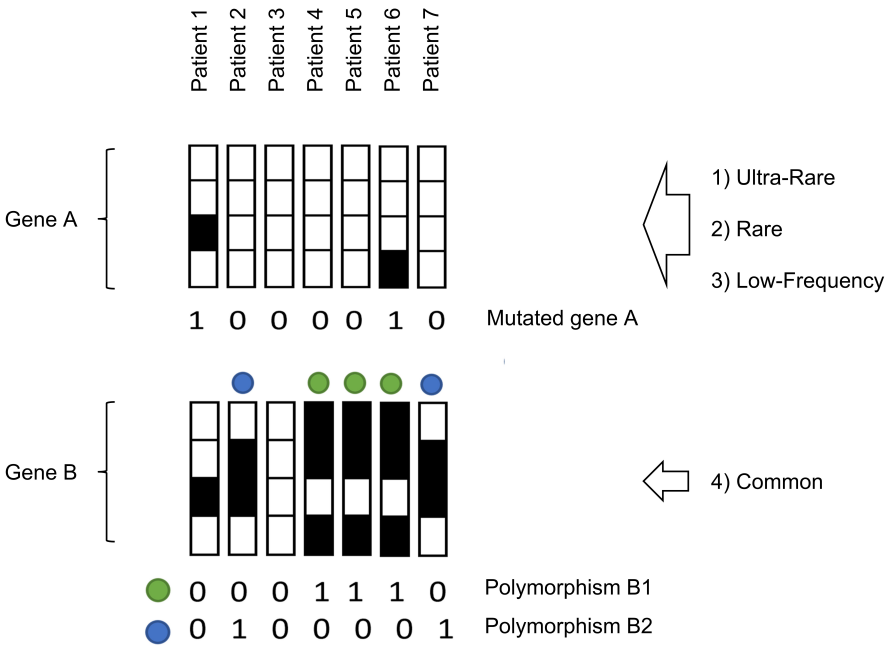


Figure 4.3: Boolean representation of genetic variants. For Ultra-Rare, Rare and Low-Frequency variants the upper chart shows that a feature "Mutated gene A" is defined by considering possible variants of the gene A. For common variants, two different polymorphisms (green and blue) are showed starting from the patterns of the variants for gene B.

In Table 4.1 we report the description of the 12 Boolean representations of genetic variability.

Table 4.1: Representations of the genetic variability for ultra rare, rare, low frequency and common variants. The Table is taken from the original paper [2].

	Representation	1	0
UR_AD	Ultra-rare variants (dominant)	At least one variant (MAF < 1/1000)	Otherwise
UR_AR	Ultra-rare variants (recessive)	At least 2 variants (MAF < 1/1000)	Otherwise
UR_X	Ultra-rare variants on the X chr genes (X-linked inheritance)	At least one variant (MAF < 1/1000)	Otherwise
R_AD	Rare variants (dominant)	At least one variant (MAF between 1/100 and 1/1000)	Otherwise
R_AR	Rare variants (recessive)	At least 2 variants (MAF between 1/100 and 1/1000)	Otherwise
R_X	Rare variants on the X chr genes (X-linked inheritance)	At least one variant (MAF between 1/100 and 1/1000)	Otherwise
LF_AD	Low-frequency variants (dominant)	At least one variant (MAF between 5/100 and 1/100) (If more than one coding LF variant, different combinations are represented)	Otherwise
LF_AR	Low-frequency variants (recessive)	Variant or variant combination as at 3a, in homozygosity (MAF between 5/100 and 1/100)	Otherwise
LF_X	Low-frequency variants on the X chr genes (X-linked inheritance)	At least one variant (MAF between 5/100 and 1/100) (If more than one coding LF variant, different combinations are represented)	Otherwise
C_AD	Common variants (dominant)	At least one variant (MAF > 5/100) (If more than one coding low-frequency variant impacts in that gene, different combinations -unique-are represented separately)	Otherwise
C_AR	Common variants (recessive)	Variant or variant combination as at 4a, in homozygosity (MAF > 5/100)	Otherwise
C_X	Common variants on the X chr genes (X-linked inheritance)	At least one variant (MAF > 5/100) (If more than one coding low-frequency variant impacts in that gene, different combinations -unique-are represented separately)	Otherwise

4.4 Definition of the Integrated Polygenic Score (IPGS)

Once both the input features (12 sets of genetic Boolean representations, see Section 4.3) and the target variable (phenotypes deviating from OLR predicted ones, see Section 4.2) have been defined, we are able to select the features that will define the IPGS's terms. In this phase the following two steps were performed:

- selection of the most relevant features for each Boolean representation with LASSO feature selection (step 3 as defined in Section 4.1). These features contribute to the definition of the count terms of the Integrated Polygenic Score (IPGS) formula. The step is described in the following Section 4.4.1.
- The definition of the weights of the IPGS (step 4, described in Section 4.4.2).

In order to augment the reliability of the model, a bootstrap approach with 100 iterations was adopted to train the model, where at each iteration the aforementioned two steps are performed. At each bootstrap iteration, 90% of the samples were selected without replication.

Principal Component Analysis Before proceeding with feature selection, a preprocessing analysis of the principal components has been carried out. Since the model is based on a combination of rare and common variants, the latter largely depending on the population, the training procedure should be performed using a dataset with homogeneous ancestry. The standard analysis of Principal Components was performed and the first principal components turned out to be connected with the patient's ethnicity collected in the medical records. Therefore, the genetic ancestry of the patients was estimated using a random forest classifier trained on samples from the 1000 genomes project and using as input features the first 20 principal components computed from the common variants by PLINK37. In order to avoid bias in the analysis due to the different ethnicity, only patients of genetic European ancestry were retained for the following analyses.

4.4.1 LASSO for Embedded Feature Selection

As already mentioned, the subsets of the most relevant features were identified using logistic regression models with Least Absolute Shrinkage and Selection Operator (LASSO) regularization, widely exploited and described in Chapter 3.

Separate logistic regression models were trained for each of the 12 sets of Boolean features described in Section 4.3, by using the two cohorts of male and female separately and the overall cohort. The target variable for each of these models was the re-classified phenotype adjusted by age and sex described in Section 4.2.

For each LASSO model, the regularization strength was optimized by 10-folds cross-validation with 50 equally spaced values in the logarithmic scale in the range $[10^{-2}, 10^{1}]$. The optimal regularization strength was selected as the one with the best trade-off between the simplicity of the model and the cross-validation score, i.e. as the highest regularization strength providing an average score closer to the highest average score than 0.5 standard deviations.

Once the regularization strength was defined, the LASSO model was re-trained using all the samples in that particular bootstrap iteration. The features with non-null coefficients are the ones selected for the next step.

In summary, for each bootstrap iteration, the LASSO logistic regression procedure returns 12 lists of features (one for each Boolean representation) that are expected to be the most important features for predicting the phenotype adjusted by age and sex in that particular bootstrap iteration.

Definition of IPGS terms The aim of the IPGS is to combine information from different representations (Eq. (4.1)-Eq. (4.4)). The list of relevant features extracted as described in the previous section (both for the specific sex cohort and for the overall one) are used to compute the number of features that are associated with mildness or severity for the different frequency ranges. For instance, n_R^m corresponds to the number of features associated with the mild phenotype coming from Boolean features computed for variants in the frequency range $[0.1\%, 1\%]$. A feature is considered associated with the mild phenotype when its coefficient in the LASSO model estimated in step 1 is negative, i.e. it contributes to the prediction of the phenotype adjusted by age and sex in the direction of a phenotype less severe than what expected at that particular age and sex. The same rule, applied to the corresponding Boolean representation, is used to define the other feature-counts appearing in Eq. (4.1) - Eq. (4.4). In this way, we have defined the count terms of the IPGS formula for males and females needed for the optimization of weights of the Integrated PolyGenic Score (IPGS).

4.4.2 Optimization of weights of the Integrated PolyGenic Score (IPGS)

The weighing factors taking into account the relative weights of the different Boolean representations in Equations (4.2), (4.3), (4.4) were estimated as those that maximize the Silhouette coefficient of the separation between the clusters of patients more/less severe than expected, on the basis of the IPGS score.

The minimization was performed with the weighting factors restricted to the integers in the following ranges: F_{LF} $[1, 4]$, F_R $[2, 8]$, and F_{UR} $[5, 100]$. This procedure returns 3 optimal values for the weighting factors associated with each Bootstrap iteration.

4.4.3 Aggregation of bootstrap results

For each of the Boolean feature, of all the 12 representations, the number of times this feature was selected in the 100 bootstrap iterations is computed. Then, the entire bootstrap procedure is repeated using random input phenotypes, and the 5th percentile of the number of times that a feature is associated with a random phenotype is estimated ($t_{5\%}$). This threshold, computed separately for each Boolean representation, was used to select which Boolean features are included in the final model, i.e. those appearing in the bootstrap more than $t_{5\%}$ times. As no significant association is expected among the Boolean features and the random phenotype, the threshold of the 5th percentile is expected to exclude with a 95% level of confidence the possible false-positive associations. The merge of all the features exceeding the threshold $t_{5\%}$ (selected in any of the bootstrap iterations) gives rise to the final version of the count terms of the IPGS formula.

As regards the weighting factors, they are computed as the median values of

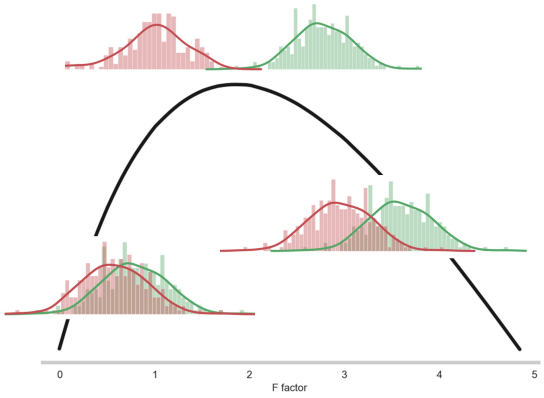


Figure 4.4: Sketch of the optimization procedure of the Silhuette coefficient for mild/severe patients on the basis of IPGS for the computation of F factors (projection of the multidimensional function on a single F).

the estimates obtained in the 100 bootstrap iterations; resulting in 2, 4, 5 for low-frequency, rare and ultra-rare respectively, for both the males and females.

4.5 Biological meaning of the extracted features

From the initial set of 163,099 features (divided into 36,540 ultra-rare, 23,470 rare, 13,056 low-frequency and 90,033 common features) in 12 Boolean representations, the selected features contributing to COVID-19 clinical variability are 7,249, which correspond to $\sim 4.4\%$ of the initial number of Boolean features. The total number of genes contributing to COVID-19 clinical variability was 4,260 in males and 4,360 in females, 75% of which were in common.

In the following points, we describe the most biologically significant extracted genes, as reported in the original paper [2].

- Among the extracted ultra-rare variants (Figure 4.5 Panel A) there was a group of genes, such as TLR3, TLR7, and TICAM1, already shown to be directly involved in the Mendelian-like forms of COVID-19. Furthermore, another group of genes are natural candidates because of their function: these include the ACE2 shedding protein, ADAM17, CFTR-related genes, genes involved in glycolipid metabolism, genes expressed by cells of the innate immune system, and genes involved in the coagulation pathway. Finally, a group of genes led by ACE2 (if affected by ultra-rare variants) confers protection from the severe disease. This group includes several genes whose mutations are responsible for auto-inflammatory disorders.

- Among the rare variants extracted (Figure 4.5 Panel B), we identified some genes as candidates for COVID-19 severity, including TLR5 and SLC26A9 as well as other genes involved in the inflammatory response.

- Among the low-frequency variants extracted, we identified some genes associated with either severity or protection from severe COVID-19 that are linked to the CFTR pathway (e.g., PSMA6) as well as specific genes involved in the immune response (e.g., NOD2) (see Figure 4.5 Panel C).

- The model was also able to identify a group of extracted common variants already shown to be linked to either severe or mild COVID-19 (4.5 Panel D). Among them are the L412F TLR3 and D603N SELP polymorphisms, already reported to be associated with the severe disease [153, 155] and several coding polymorphisms in Linkage Disequilibrium (LD) with already reported genomic SNP, such as the ABO blood group, OAS1-3 genes, PPP1R15A gene, and others [162]. In conclusion, considering their functions, genes involved in the immune and inflammatory responses, or those involved in the coagulation pathway and NK and T cell receptor, are to be considered natural candidates for severe or mild COVID-19.

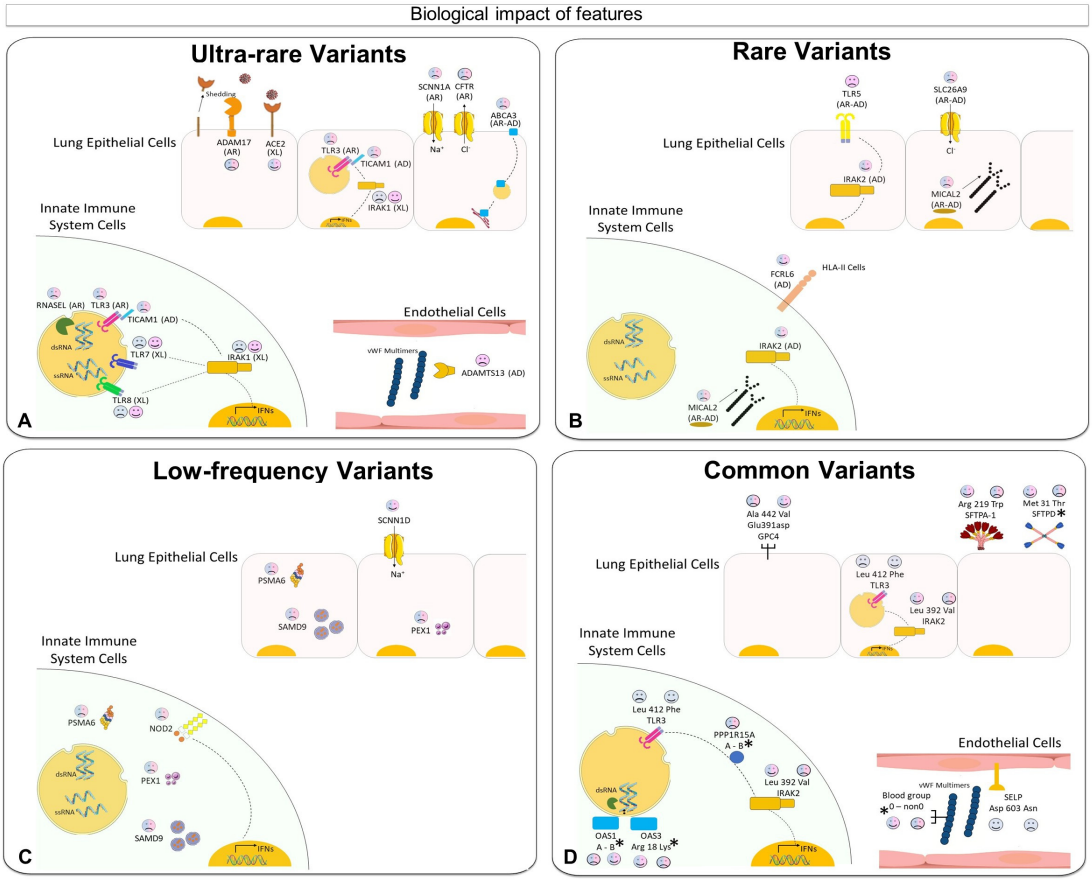


Figure 4.5: Biological interpretation of the most interesting features (for the categories of ultra-rare, rare, low-frequency and common variants) among the extracted ones. The image is taken from the original paper [2].

4.6 Training of the predictive model based on age, sex, and IPGS

The procedure described in the previous sections completely defines how to calculate the overall IPGS score. Then, the predictive model (step 5 as defined in Section 4.1) of the binary COVID-19 severity was defined as a logistic regression model without regularization, that uses as input features the IPGS, in addition to age, and sex.

It should be noted that, in the phase of feature selection/discovery described in Section 4.4.1, only patients that deviate from their expected severity based on age and sex were used (see Section 4.2 for the OLR). The procedure was in fact designed to isolate the genetic basis of COVID-19 severity. Instead, since the final model will be tested to other cohorts and could potentially be exploited as a diagnosis tool, in this final step, IPGS, age and sex are combined to predict a binary classification of the actual COVID-19 severity, i.e.

- class of severe, hospitalized patients with any form of respiratory support; represented by the following categories:

2. hospitalized, receiving low-flow supplemental oxygen;
3. hospitalized, receiving continuous positive airway pressure (CPAP) or bilevel positive airway pressure (BiPAP) ventilation;
4. hospitalized receiving invasive mechanical ventilation;
5. death;

- versus the class of mild, given by the categories of:

0. not hospitalized.
1. hospitalized, not receiving supplemental oxygen.

In order to prevent overfitting, this final model was fitted using 466 unseen samples from the GEN-COVID cohort n.2 different from the training set adopted in the feature selection phase and the Swedish cohort (having cases only, see Appendix 5 for details). During the fitting procedure, the class unbalancing is tackled by penalizing the misclassification of the minority class with a multiplicative factor inversely proportional to the class frequencies. In addition, since the aim of the analysis is to test the model to foreign cohorts, a normalization of the IPGS is needed and the percentile normalization of the IPGS scores is performed within each cohort.

An alternative logistic model using as input features only age and sex was also fitted on the same training set. The comparison between the two models is intended to evaluate if the genetic information summarized in the IPGS improves the prediction of severity compared to a model based on age and sex alone. Finally, a further logistic regression model is fitted by only considering the IPGS variable.

4.7 Model Testing

The training procedure reported in the previous section returned three logistic regression models to be compared:

1. one model using as input features only age and sex (M1);
2. a second model using IPGS in addition to age and sex (M2); and
3. the last model using only IPGS (M3).

These models were tested, without any further adjustment, using independent cohorts of European ancestry. Specifically, three different cohorts, from Germany, Canada and UK, contributed to this study. For multi ancestry cohorts (Canada and UK) the sub-population of European Ancestry was included in the study. The cohorts are described in Appendix 5.

For each cohort, the performances of the three models were evaluated and compared in terms of accuracy, precision, sensitivity, and specificity. In Figure 4.6 we report the results, where the model M1 is represented with grey, the model M2 with orange, and the model M3 with the green color.

From Figure 4.6 it is evident that the second model (M2) including IPGS, age, and sex performs better than the model considering only sex and age as inputs (M1),

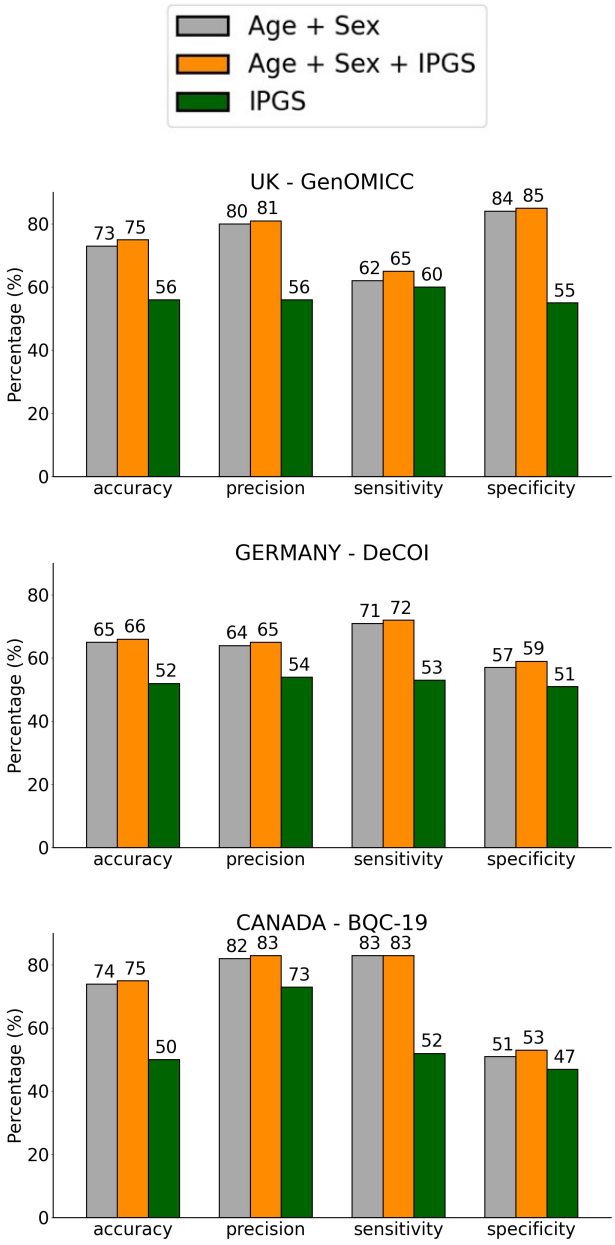


Figure 4.6: In the three tested cohorts, when the IPGS is added to age and sex as a regressor, all the performances increase: the accuracy up to +2%, the precision up to +1%, the sensitivity up to +3%, and the specificity up to +2%. We conclude that IPGS is able to improve prediction of clinical outcome in addition to the well-established powerful factors of age and sex. The image is taken from the original paper [2].

in each of the testing cohorts, separately. The increase in performance was systematically observed throughout all the cohorts: on average +1.33% for accuracy, +1% for precision, +1.33% for sensitivity, +1.67% for specificity. Considering the difference in phenotype classification inherent to a comparison among various international

cohorts, and the genetic variability among different European sub-populations, the increase in performances observed for the model with IPGS demonstrates that this score provides a robust index for predicting COVID-19 severity. We deem that the performances' increase for all three tests separately is very comforting, since one of the biggest issues of non-reliable ML model is the instability of the aggregated performances, as highlighted in Chapter 1. In addition to the two models (with and without IPGS) also the third logistic regression model fitted with IPGS alone shows performances well above the random guess.

For the overall cohort including the three independent cohorts, the results are reported in Figure 4.7.

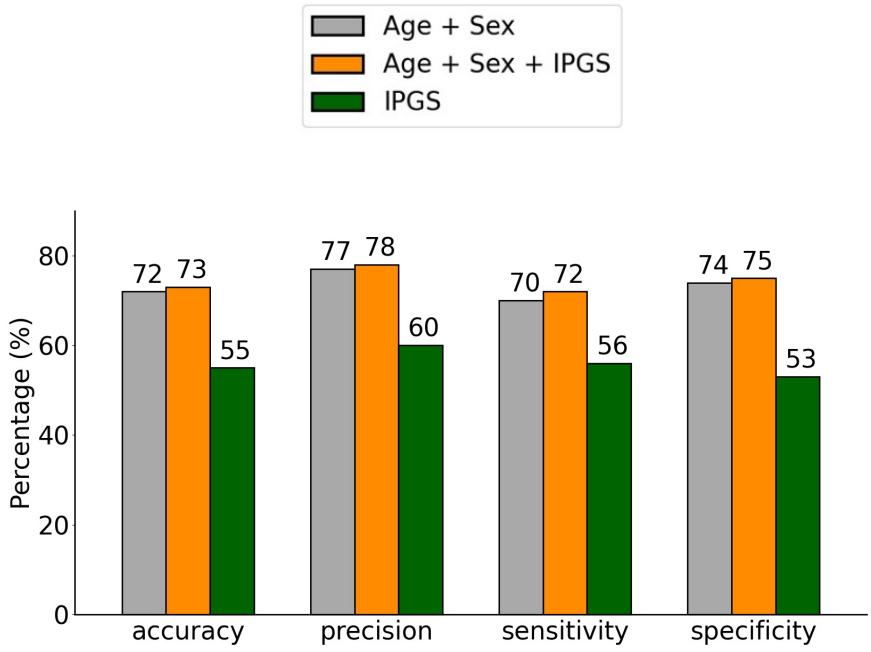


Figure 4.7: When the IPGS is added to age and gender as a regressor, the performances of the model increase: accuracy +1%, precision +1%, sensitivity +2%, specificity +1%. The image is taken from the original paper [2].

The model M2 with age, gender, and IPGS exhibited an overall accuracy of 73%, precision equal to 78%, with sensitivity, and specificity of 72% and 75%, respectively. As for the previous tests, all the aforementioned metrics are higher than the corresponding values obtained using a logistic model that adopted as input features only age and sex (M1). The increase in performances of the model with IPGS confirms that this score indeed confers additional genetic information for predicting COVID-19 severity compared to only age and sex.

In order to evaluate the statistical significance of the obtained results, the performances of the model including (M2) age, sex and IPGS are evaluated with respect to the performances of a model where the values of the IPGS feature have been shuffled. By carrying out 100 repetitions of the shuffling procedure we computed the p-value of the M2 tested performances with respect to the empirical null distribution. We concluded that the increase of the performances is statistically significant (p-value

< 0.05 for both accuracy, precision, sensitivity, and specificity) with respect to the null distribution of performances for the ensemble of models where the IPGS feature has been randomized (see Figure 4.8).

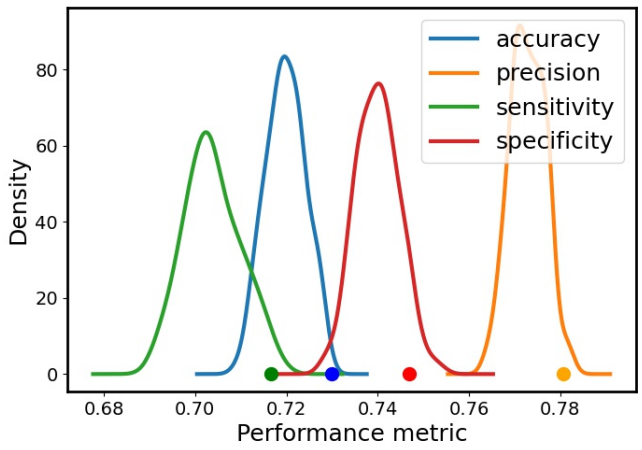


Figure 4.8: Null distribution of the performances obtained by randomizing the IPGS. The increases of the model with IPGS are statistically significant (p-value < 0.05 for accuracy, precision, sensitivity and specificity). The image is taken from the original paper [2].

Furthermore, the empirical probability density function of IPGS scores (Figure 4.9) has been estimated for the severe and non-severe patients of the cohort including the testing sets. It is worth noting the shift on the right of the IPGS distribution for the severe patients, with a significant p-value (< 0.001) for the t-test of the mean difference. This difference between severe and non-severe cases is preserved for the male and female cohorts when analyzed separately (p-values < 0.001 and 0.024, respectively).

4.7.1 Association studies

As a further test to evaluate the importance of the IPGS score for predicting COVID-19 severity, the univariate logistic models were used on the overall set including the training set of the logistic regression[2] and the testing cohorts for a total of $2,240$ patients to estimate the Odds Ratio (OR) of severe COVID-19 for IPGS, age, and sex, separately.

The test confirmed that severity was associated with IPGS, showing an OR of 2.32 ($p < 0.001$, 95% confidence interval [1.79, 3.01]) with age, measured in decades, and sex, having OR of 1.89 ($p < 0.001$, 95% confidence interval [1.79, 2.00]) and 2.99 ($p < 0.001$, 95% confidence interval [2.58, 3.46]) respectively. The results (regression coefficient, P value, Odds ratio (OR) and OR's interval of confidence at 95%) are reported in Table 4.2.

[2]Obviously, to avoid data leakage, the train set used for the feature selection and the IPGS's terms definition is not used.

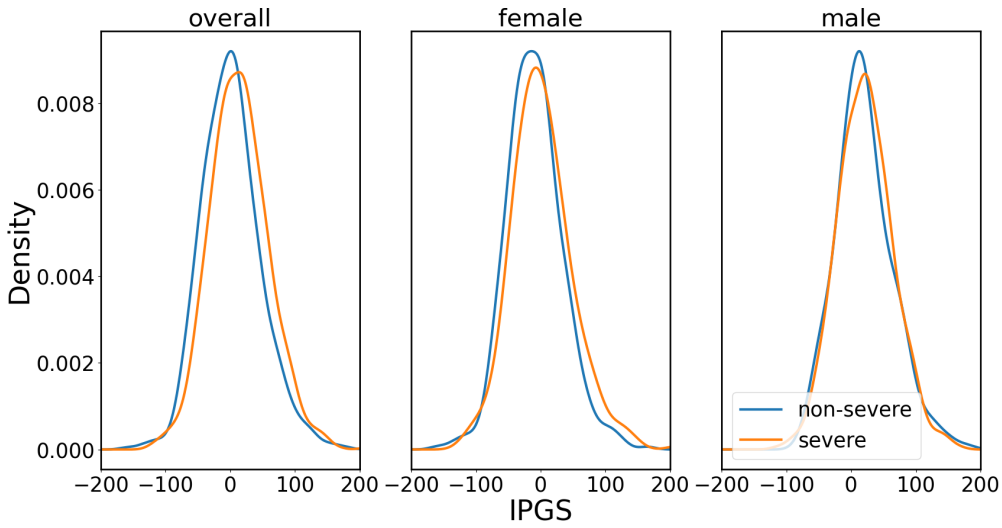


Figure 4.9: Empirical probability density function for the IPGS in the testing cohorts for severe and non-severe; separately for the overall, the female and the male cohorts. The image is taken from the original paper [2].

Table 4.2: Results of the univariate logistic regression models fitted on the cohort including the whole testing set.

Variable	**Type**	**Coef.**	**P value**	**OR**	**95 CI**
IPGS	Continuous [0,1]	0.84	<0.001	2.32	[1.79, 3.01]
Age	Continuous (Decades)	0.64	<0.001	1.89	[1.79, 2.00]
Sex	Binary (M Vs F)	1.10	<0.001	2.99	[2.58, 3.46]

Since IPGS, age and sex can be correlated, a multivariate analysis better evaluates the effect of the possible collinearity. The multivariate logistic regression using sex, age, and IPGS together, provided similar results reported in Table 4.3 confirming the goodness of the regressors' OR.

Table 4.3: Results of the multivariate logistic regression model fitted on the cohort including the whole testing set.

Variable	**Type**	**Coef.**	**P value**	**OR**	**95 CI**
IPGS	Continuous [0,1]	0.44	0.017	1.56	[1.15, 2.12]
Age	Continuous (Decades)	0.63	<0.01	1.87	[1.78, 1.98]
Sex	Binary (M Vs F)	1.01	<0.01	2.75	[2.32, 3.27]

Finally, multivariable logistic regression was performed using as predictor variables the comorbidities in addition to IPGS, age, and sex. The comorbidities are congestive/ischemic heart failure; asthma/COPD/OSAS; diabetes; hypertension; and cancer. This latter model has been fitted in the training set, where the information

on comorbidities was available. When adjusting for comorbidities, with a multivariable logistic model, OR of IPGS was 2.46 (p=0.05, 95% confidence interval [1.15, 5.25]) as shown in Table 4.4. This result further confirms that IPGS is a reliable predictor of COVID-19 clinical severity.

Table 4.4: Results of the multivariate logistic regression model fitted on the cohort where the information on comorbidities was available.

Variable	**Type**	**Coef.**	**P value**	**OR**	**95 CI**
IPGS	Continuous [0,1]	0.90	0.05	2.46	[1.15, 5.25]
Age	Continuous (Decades)	0.73	<0.01	2.08	[1.78, 2.43]
Sex	Binary (M Vs F)	0.84	<0.01	2.33	[1.49, 3.63]
Heart Failure	Binary (Y Vs N)	-0.29	0.73	0.75	[0.19, 2.97]
Asthma/COPD	Binary (Y Vs N)	0.66	0.56	1.93	[0.30, 12.49]
Diabetes	Binary (Y Vs N)	0.46	0.57	1.59	[0.42, 6.04]
Hypertension	Binary (Y Vs N)	-0.40	0.38	0.67	[0.31, 1.42]
Cancer	Binary (Y Vs N)	-0.91	0.30	0.40	[0.096, 1.70]

Code availability Data analyses were performed using Python with the Scipy ecosystem, and the scikit-learn library. The statistical association was done with the statsmodel Python library. The code is freely available at the GitHub repository: `https://github.com/gen-covid/pmm`.

4.8 An example of segregation analysis using IPGS

In order to show the potential of the explanation model's capability, in the present section we report an example of segregation analysis carried out by applying the IPGS to 9 pedigrees of the cohort of familiar cases. For this analysis, two categories of genetic variability, i.e. rare and common, are exploited, and an iteration of the extracted feature is considered (with F factors equal to 2 for males and 1.2 for females). The example is reported and detailed in [159].

In Figure 4.10 the squares represent male subjects, and the circles represent female subjects. The red is used for severely affected patients, the green for oligo-asymptomatic subjects and the grey for intermediate subjects. Under each symbol, the treatment, the age (in parenthesis), and the terms of the IPGS formula are reported. Here we describe the biological/clinical interpretation of the two most relevant cases reported in Figure 4.10.

- Panel A: Brothers of 32 and 31 years with discordant phenotypes: hospitalized CPAP treated, and oligosymptomatic, respectively. In agreement with their phenotype, they have IPGS 0 and -10 respectively, mainly due to increased common polymorphisms associated with mild disease in the asymptomatic brother (such as p.Ile57Val of AURKA, a cell cycle regulator downregulated during SARS-CoV-2 infection; p.Cys357Arg/p.Va335Met of GBP3, a strong repressor of the activity of the viral polymerase complex, which results in

decreased synthesis of viral proteins; and p.M1 of TLR8, a member of the Toll-like receptor family which plays a fundamental role in pathogen recognition and activation of innate immunity).

- Panel B: Sisters of 62 and 60 years of age with partially discordant phenotype, hospitalized with oxygen support only and hospitalized CPAP treated, respectively. In agreement with their phenotype, they have IPGS 2.8 and 7, respectively, mainly due to increased severity of rare variants in the more severely affected sister (including Amyloid Beta Precursor Protein Binding Family A Member 3 APBA3, which plays a role in immune response; and the low-density lipoprotein receptor family member LRP8, which has a role in the suppression of innate response). Treatment with immunosuppressive agents may be an option for this patient.

Through further exploration for rare variants in this individual family member, an extremely rare pathogenic mutation was detected in IFNAR1. The frequency, however, of IFNAR1 variants was too low to be identified using logistic regression in a cohort of this size.

It is worth noting that the interpretability of the proposed model is strongly favoured by the Boolean nature of the features. In fact, the genetic screening of a patient give us indications on the IPGS' genes that are mutated, and consequently, the clinicians can focus on the relevant mutated genes of the IPGS, e.g. those reported in Chapter 3.

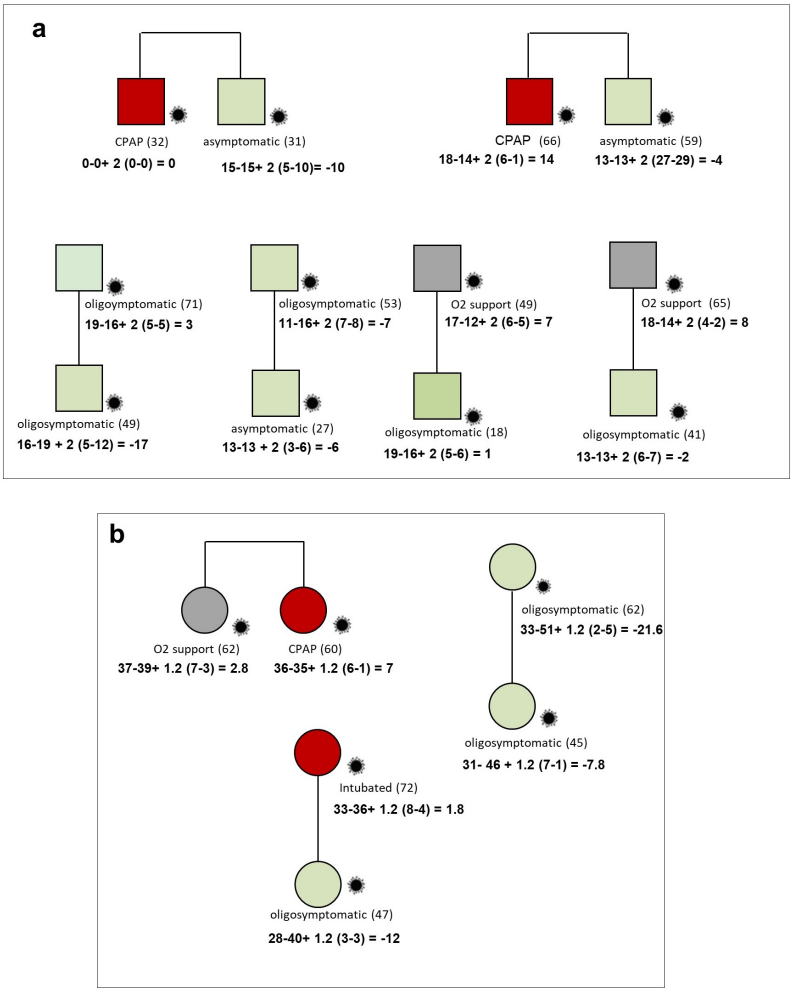


Figure 4.10: Example of segregation analysis based on the IPGS score. The squares represent male subjects, and the circles represent female subjects. The red is used for severely affected patients, the green for oligo-asymptomatic subjects, and the grey for intermediate subjects. Under each symbol, the treatment, the age (in parenthesis), and the terms of the IPGS formula ($IPGS = n^s_{common} - n^m_{common} + 2 \cdot (n^s_{rare} - n^m_{rare})$ for male, $IPGS = n^s_{common} - n^m_{common} + 1.2 \cdot (n^s_{rare} - n^m_{rare})$ for female) are reported. The image is taken from the original paper [159].

Chapter 5

Conclusions and Future works

As stated in the introduction, our main theoretical task was to deepen and systematize the reliability and the explainability topics in the context of artificial intelligence (AI) models, with a particular focus on high-stakes applications. Returning to the question posed at the beginning of this study, it is now possible to state that model trustworthiness is linked to other factors, including the interpretability of the algorithm, the stationarity of data, and the possible bias in the data (see [4], [5]).

Especially in the field of interpretability, much work has been done in order to explain and interpret the models developed by AI in a human-comprehensible manner. The main reason behind this effort is that the human experience and its capacity for abstraction allow monitoring the process of the model decisions in a sound way, trying to mitigate the risk of data-driven models. In fact, under the control of humans, the decisions taken by the models assume a stronger meaning and significantly contribute to the progress of scientific research in that particular task. This is basically the approach we tried to follow when we tackled the complex problem of the genetic variability in COVID-19 disease.

In the following, we summarize the main contributions, the limitations, and future works on the topics covered by this work.

A model risk framework for AI applications

In the first chapter, we have defined a guideline for the development and the validation of a machine learning (ML) model, with a particular focus on the trustworthiness and explainability requirements. The work aims at answering to the demands of business, especially in the FinTech industries, seldom requiring an overall assessment of the internal ML models. In particular, we have seen how the phase of design and deployment of an ML model may be driven by many choices: the algorithms that are best suited to the available dataset in respect of the problem to solve, how the performance needs to be measured, how the hyperparameters should be fixed, etc. The work tried to delineate a common set of requirements and tips to evaluate how well those different choices were made (see for example Table 1.15).

Moreover, the framework allows for the measurement of the level of "model risk" of the AI application, by assigning a score to each of the 12 boxes on the left of the chart reported in Figure 1.1. To this end, a set of questions is associated with

each box, see Tables 1.1, 1.2, 1.3, 1.4, 1.6, and 1.7 for architectural requirements; Tables 1.8, 1.10 for functional requirements; and Tables 1.11, 1.12, 1.13, and 1.14 for policy requirements.

There are two major limitations that could be addressed in future research. First, this work focuses on standard ML models and does not particularly investigate other classes of networks, e.g., the Graph Neural Network (GNN) and Generative Adversarial Network (GAN), or reinforcement learning models. Second, the work, even if generally applicable, is focused on possible models in the financial context, excluding the specificities of other different application areas. Therefore, the results of the self-assessment list must be interpreted with caution, and a number of limitations should be considered with regard to the circumstances of the problem to solve and the peculiarity of the different stakeholders. We note that, big effort of the research in this field is currently targeting application-specific quantitative metrics, e.g. Koopman's safety performance indicators in the automotive domain [163].

A more theoretical contribution to the explainability field has been achieved in Chapter 2 where two novel theoretical methodologies were developed by exploiting the concept of feature importance.

Clustering-Based Interpretation of Deep ReLU Network

The first methodology aims at increasing the level of interpretability of a fully connected feedforward neural network with ReLU activation functions, downstream from the fitting phase of the model. It is worth noting that the introduced methodology does not alter neither the structure nor the performance of the network, and can be easily applied after the training of the model since it relies on the clustering that naturally arises from the binary status of the different neurons of the network (in turn, related to the two regimes of the ReLU function).

Then, the existence of a feature importance explanation based on an affine map for each cluster has been proved, and the empirical application to the Titanic dataset showed the capability of the method to bridge the gap between the algorithm optimization and human understandability. Our results are encouraging and should be validated by a larger set of datasets, possibly with more complex networks in terms of number of layers and tasks to be solved.

Moreover, the cluster-based representation may be further exploited as a regularization to reduce the complexity of a neural network model. In fact, since the number of clusters can be seen as a measure of the model complexity, a possible strategy is to penalize solutions with numerous clusters. Nevertheless, this choice cannot be sufficient when the clusters, even few, are very similar to each other; as happening, for instance, when the activation of particular neurons changes the cluster definition without so much affecting the cluster's solution. In order to avoid both too much and similar cluster, we can enforce the orthogonality among the effective vectors representing the ℓ clusters, by minimizing for instance the following quantity:

$$R = \|\mathbb{1} - \Theta^T \Theta\|_F, \tag{5.1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and Θ is the matrix whose columns are the

ℓ, d-dimensional effective vectors:

$$\Theta = \begin{bmatrix} \tilde{\omega}_1^{(1)} & \cdot & \tilde{\omega}_1^{(\ell)} \\ \cdot & \cdot & \cdot \\ \tilde{\omega}_d^{(1)} & \cdot & \tilde{\omega}_d^{(\ell)} \end{bmatrix}$$

Therefore, once considering the gradient computation of the loss E, a second term in addition to the empirical loss V must be considered:

$$\nabla_{\hat{W}} E = \sum_{\kappa=1}^{l} \underbrace{\nabla_{\hat{W}} V(y_\kappa, \mathrm{f}(\hat{W}, \hat{u}_\kappa))}_{BP} + \lambda \underbrace{\nabla_{\hat{W}} R(\boldsymbol{\omega}(w))}_{BP^\star}$$

It is known that a very important advantage of backpropagation (BP) is the ability to being efficient in the gradient's computation[1]. The good news is that the derivative involving the term $BP^\star$, of the effective vector ω_i with respect to the network weights, can be computed by backpropagating the scalar product $\omega_i \cdot \omega_2$, where ω_i is interpreted as a "dummy" pattern and ω_2 as the equivalent network.

Another possible solution to tackle the potential issue of the high number of clusters may be followed by considering the hierarchy of the layers in the network.

Finally, a future line of research could be the generalization of the discussed approach to the Tensor Networks, e.g. the the matrix product state (MPS) [164], which retain the affine nature of ReLU networks, while improving the expressive power using entanglement properties.

Logic constraints to Feature Importance

In the second part of Chapter 2, we presented a novel model agnostic framework able to inject the apriori knowledge on the relevance of the input features into an ML model. This "weighted" approach can contribute to bridging the gap between the fully data-driven models and the human-guided ones. The advantage of the proposed method is the flexibility: the logic constraints are fully customizable and do not depend either on the nature of input features (numerical, categorical, etc.) or on the architecture of the model, or on the algorithms chosen for the computation of the feature importance, e.g. LRP.

The method has been successfully applied in the fairness topic, even if future studies are recommended in order to validate the power of the method in different datasets and tasks.

A further possible application of the proposed framework is to enforce *a priori* selective attention of the model on particular features. This can be useful for example when the user wants to focus on some relevant words in the text, or a region of an image (see Figure 5.1).

Furthermore, there could be many cases where the users want to inject prior knowledge in the form of feature importance in the model. For example, from experience, one could know that one feature should be less important than another for the business of the company, e.g. the age, the gender in a particular financial context. Another possibility is when we a priori know which feature is less reliable,

[1]The complexity for both the forward flow and the backward pass is $O(|W|)$, due to the weight matrices products.

Figure 5.1: The middle part of an image may be subject to an a priori focus.

e.g. less stationary with respect to the others.[2] It is worth noting that the constraints can be settled for just a portion of the dataset.

As future work, we are interested in providing a software solution for the integration of the proposed framework within the popular machine learning software. Another future work is to apply the logic constraints to other contexts, in terms of both datasets (images, text, etc.), and models (random forest, SVM, etc.). Finally, the usage of other measures based on information entropy can be explored in order to take into account the problem of correlation between features.

Machine Learning modeling of complex genetic diseases and COVID-19

In Chapter 3 we have reported the analyses on the discovered variants related to genes involved in the COVID-19 severity. The obtained results have further strengthened the hypothesis that COVID-19 is a complex genetic disease. Anyway, the findings contributed to disentangle the complex mechanism of the disease and leading us to make an effort for defining a comprehensive model. So, in Chapter 4 we described the interpretable model for the prediction of disease's severity, successfully tested within different independent European cohorts.

By better understanding the role of host genetics in COVID-19 susceptibility and disease severity, we are also in a stronger position to identify public health measures that will curb the impact of the disease on society as a whole. This should help us to genetically screen already affected or potential patients in order to predict those who are more or less susceptible to developing severe disease. It should further help us in, not only reassigning therapeutics or developing new interventions (including vaccines) but also in decision-making regarding therapeutics and vaccine allocations. Beyond what this ML method can help us understand regarding the role of host genetics in COVID-19 susceptibility and the potential implications for clinical and public health responses, the model also has strong potential for understanding the role of host genetics in other complex disorders.

Future works should concentrate mostly on enhancing the accuracy of the genetic model, also by considering other genetic components, e.g., the Human Leukocyte Antigen (HLA) variability. A further possibility is to focus on organ-specific IPGS

[2] In linear regression a similar problem is called attenuation bias, where errors in the input features cause the weights to go toward zero.

scores trying to better characterize the multi-organ phenotype of the disease, as described in Section 3.3. Another very promising line of research regards the study of possible clusters of patients, on the basis of the genetic, with a similar biological mechanism for the disease progression. In particular, the Topological Data Analysis (TDA) is a powerful mathematical tool for representing and visualizing complex data structures (in the form of a cloud of data points) in simplicial complexes. Preliminary attempts to apply the TDA methodology[3] to the genetic data of COVID-19 provided good results. The set of features used for TDA are those extracted from the model described in Chapter 4 and clusters of patients resulted, each one characterized by a prevalent phenotype and a set of mostly mutated genes.

It is important to mention that, the scientific contributions reported in Chapter 3 and Chapter 4 have been formalized in a pure statistical setting, without the explicit injection of human knowledge. Probably, also due to the recent advances, the time has come to explore other approaches such as the Bayesian ones, in order to exploit the acquired knowledge on the genetic components of the disease. In this context, the methodology presented in the second part of Chapter 2 provide a strategy to weight more a particular group of genes.

Finally, preliminary work has been carried out by applying a deep Neural Network model to the genetic dataset of COVID-19. The idea was to exploit the novel methodology reported in Chapter 2 "clustering-based interpretation of deep ReLU Neural Network" in order to identify the feature importance of the genes involved in the clusters' solution. For instance, a possible task was to evaluate whether the network was able to cluster the patients' features on the basis of the sex. Anyway, the major issues to solve was related to the parametrization of the network and to the overfitting in relation to the high number of features and the limited number of samples. With an increased sample size, the fitting of a deep Neural Network model is a good prospective solution.

[3]The algorithm for the dimensionality reduction is UMAP. Then, the clustering algorithm DBSCAN work within the partition on the space.

Appendix A - External cohorts contributing to the model testing

Five different cohorts (from Germany, Italy, Quebec, Sweden, and UK) contributed to this study either as a training set or as a testing set. Here we describe the main features of the cohorts.

- GEN-COVID cohort (Italy). Whole Exome Sequencing with at least 97% coverage at 20x was performed using the Illumina NovaSeq6000 System (Illumina, San Diego, CA, USA). Library preparation was performed using the Illumina Exome Panel (Illumina) according to the manufacturer's protocol. Library enrichment was tested by qPCR, and the size distribution and concentration were determined using Agilent Bioanalyzer 2100 (Agilent Technologies, Santa Clara, CA, USA). The Novaseq6000 System (Illumina) was used for DNA sequencing through 150 bp paired-end reads. Variant calling was performed according to the GATK434 best practice guidelines, using BWA35 for mapping and ANNOVAR36 for annotating.

- Swedish cohort. Whole Exome Sequencing was performed using the Twist Bioscience exome capture probe and was sequenced on the Illumina NovaSeq6000 platform. Data were then analyzed using the McGill Genome Center bioinformatics pipeline (`https://doi.org/10.1093/gigascience/giz037`) in accordance with GATK best practices.

- DeCOI (Germany). 800-1000 ng of genomic DNA of each individual was fragmented to an average length of 350 bp. Library preparation was performed using the TruSeq DNA PCR-free kit (Illumina, San Diego, CA, USA) according to the manufacturer's protocol. Whole genome sequences were obtained as 150 bp paired-end reads on S4 flow cells using the NovaSeq6000 system (Illumina). The intended average sequencing depth was 30X. The DRAGEN pipeline (Illumina, version 3.6.3 or 3.5.7) was used for alignment and joint variant calling was performed with the Glnexus software (version 1.3.2). Individuals with a 20-fold coverage in less than 96% of the protein coding sequence were removed as well as related individuals to retain only from related pairs. Variant QC was performed using hail (version 0.2.58). European individuals were selected by performing PCA analysis along with the 1000 genomes

data. Finally, annotation was performed using Variant Effect Predictor (VEP, version 101).

- BQC-19 (Quebec). Whole genome sequencing at mean coverage of 30x was performed on the Illumina NovaSeq6000 platform, then analyzed using the McGill Genome Center bioinformatics pipeline (`https://doi.org/10.1093/gigascience/giz037`), in accordance with GATK best practice guidelines.
- GenOMICC/ISARIC4C (UK). Whole genome sequencing at mean coverage of 20x was performed on the Illumina NovaSeq6000 platform and then analysed using the Dragen pipeline (software v01.011.269.3.2.22 , hardware v01.011.269). Variants were genotyped with the GATK GenotypeGVCFs tool v4.1.8.1.

Appendix B - Layer-wise Relevance Propagation (LRP)

Layer-wise Relevance Propagation (LRP) introduced in [3] and described also in [165] is an explainability method that computes an instance-specific feature importance by backpropagating the predictions of a neural network along the hidden layers.

Specifically, the method starts by redistributing the prediction $\tilde{f}(\boldsymbol{x})$ backwards to the last layer (indexed with the letter p), by assigning a relevance score R_p to each of the d_p neuron of the last layer such as their sum is equal to the prediction:

$$\sum_{p=1}^{d_p} R_p = \tilde{f}(\boldsymbol{x}).$$

This procedure is replicated through the previous layers of network until we obtain the relevance score of the input variables $R_i,\ \ i = 1 \dots d$. By denoting the layers of the network with different pedix letters, we have the following chain of equalities:

$$\sum_{i=1}^{d} R_i = \cdots = \sum_{j=1}^{d_j} R_j = \sum_{k=1}^{d_k} R_k = \cdots = \sum_{p=1}^{d_p} R_p = \tilde{f}(\boldsymbol{x})$$

How to redistribute the relevance of a neuron to the neurons of the previous layers? In order to fix the ideas, let us consider to redistribute the relevance of the third neuron of layer k: $R_{k=3}$. Suppose we have defined the parameters $\rho_{j,k}$ representing the strengths of the connections[4] between the neurons of the previous layer j and those of layer k. For each neuron of the layer j we compute the ratio of the connection strength $\rho_{j,\,k=3}$ with respect to the total connection strength of $k = 3$ (i.e., $\sum_j \rho_{j,\,k=3}$). For instance the neuron $j = 1$ will receive the fraction $\frac{\rho_{j=1,\,k=3}}{\sum_j \rho_{j,\,k=3}}$ of the relevance $R_{k=3}$.

Once we are able to compute the contribution of the relevance received from the froward neuron, the relevance score for a single neuron of layer j, e.g. $R_{j=1}$ can be computed by summing a fraction of the relevance scores of all d_k neurons of the forward layer k:

$$R_j = \sum_k \rho_{j,k} \frac{R_k}{\sum_j \rho_{j,k}}.$$

[4]The strengths of the connections, based on the weight of the network will be defined in the following.

One possible choice for the connection strength is the square of the neural network weights:

$$R_j = \sum_k w_{jk}^2 \frac{R_k}{\sum_j w_{jk}^2}.$$

Another possibility is the product of the activation a_j for the positive part of the weights, as reported in the following formula:

$$R_j = \sum_k \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} R_k.$$

In Figure 5.2 we reported a graphical representation of the LRP methods taken from the original paper [3].

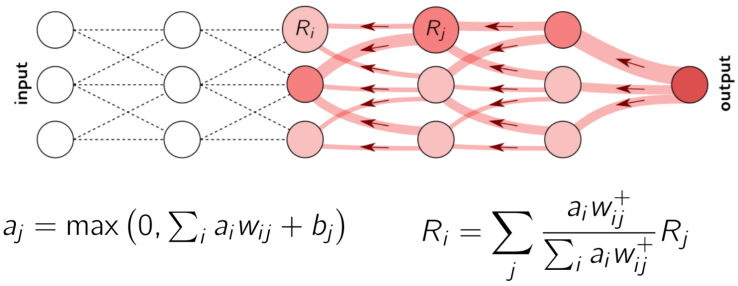


Figure 5.2: Representation of the LRP procedure taken from the original paper [3].

Appendix C - Shapley value

The Shapley value is a typical solution arising in the framework of the game theory for the problem of assigning the fair reward to each member of a coalition following an overall gain (payoff).

The formal definition of a cooperative game of a set D of d players rely on the function called *characteristic function* that maps each subset $S \subseteq D$ of players to a real number corresponding to the expected sum of payoffs the member of S can obtain by cooperation. In formulae, the characteristic function is:

$$\nu : \mathcal{P}(D) \to \mathbb{R}, \tag{5.2}$$

where $\mathcal{P}(D)$ is the power set of D, i.e the set of all the possible subsets of D, with the condition that the function returns zero on the empty subset $\nu(\{\}) = 0$.

In the following, we will uniquely identify each player D_i by the integer i in the range $[1, d]$, so that the characteristic function will be defined in the domain of the 2^d subsets of the range $[1, d]$:

$$\nu : \mathcal{P}([1, d]) \to \mathbb{R}. \tag{5.3}$$

Under quite natural assumptions, the solution of this problem is given by the Shapley value, that is a function defined for each player i that gives the fair gain of the player i for the characteristic function v. With the usual symbol notation, the Shapley value is computed as:

$$\phi_i(v) = \sum_{S \subseteq D/\{i\}} \frac{|S|!(|D| - |S| - 1)!}{|D|!} \left(v(S \cup \{i\}) - v(S)\right) \tag{5.4}$$

where the sum is done on all subsets of D not including the player i.

Let us try to understand the meaning of Eq. ((5.4)) for a chosen player i. Fixing a subset S with cardinality $|S| = k$, the term $v(S \cup \{i\}) - v(S)$ represents the extra contribution that the player i adds to the coalition S. We can compute the average of these contributes for the all subsets with a fixed cardinality k:

$$\sum_{S \subseteq D/\{i\}\, with |S|=k} \frac{v(S \cup \{i\}) - v(S)}{\frac{|D-1|!}{k!(|D|-1-k)!}}, \tag{5.5}$$

where $\frac{|D-1|!}{k!(|D|-1-k)!}$ is the number of the possible subsets with cardinality k. Sure? yes, the number of combinations of $|D-1|$ elements in groups with cardinality k.

Now we should compute the average of the introduced quantity for all the possible cardinalities of the subsets. These cardinalities are $K = \{0, 1, 2, ..., |D| - 1\}$ that is a set with cardinality $|D|$. This second average is so computed as

$$\begin{aligned} &\sum_{k \in K} \left(\sum_{S \subseteq D/\{i\} with |S|=k} \frac{(v(S \cup \{i\}) - v(S))}{\frac{|D-1|!}{k!(|D|-1-k)!}} \right) \frac{1}{|D|} = \\ &\sum_{S \subseteq D/\{i\}} \frac{|S|!(|D| - |S| - 1)!}{|D|!} (v(S \cup \{i\}) - v(S)) = \phi_i(v) \end{aligned} \tag{5.6}$$

and we retrieve the formula for the Shapley value.

One fundamental property of this solution is the Efficiency property. In fact the sum of the Shapley values of all players equals the value of the grand coalition

$$\sum_{i \in D} \phi_i(v) = v(D). \tag{5.7}$$

The super-additive property grants the individual rationality i.e. the fact that in any coalitions, a player receives more than what he could get on his own.

List of Publications

In the following, the list of the research contributions produced during the period of this Ph.D. research is provided.

Published Journal Papers.

1. **Picchiotti, N.**, Salvioli, M., Zanardini, E., & Missale, F. (2020). *"COVID-19 pandemic: a mobility-dependent SEIR model with undetected cases in Italy, Europe and US"*. Epidemiologia e prevenzione, 44(5-6). **Candidate's contributions**: Carried out theoretical analyses, algorithm design and experimental campaign.

2. Daga S., ... **Picchiotti N.**, and others. *Employing a systematic approach to biobanking and analyzing clinical and genetic data for advancing COVID-19 research*, European Journal of Human Genetics. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

3. Baldassarri M. and **Picchiotti N.**[5] and others. *Shorter androgen receptor polyQ alleles protect against life-threatening COVID-19 disease in European males*, EBioMedicine. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

4. Fallerini C.,..., **Picchiotti N.** and others. *Association of Toll-like receptor 7 variants with life-threatening COVID-19 disease in males: findings from a nested case-control study*, Elife. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

5. Croci S.,..., **Picchiotti N.** and others. *The polymorphism L412F in TLR3 inhibits autophagy and is a marker of severe COVID-19 in males*, Autophagy. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

6. Fallerini C.,..., **Picchiotti N.** and others. *SELP Asp603Asn and severe thrombosis in COVID-19 males*, Journal of hematology & oncology. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

[5] Co-first authors

7. Lugoboni A. **Picchiotti N.**, Spuntarelli A. *Risk allocation with Shapley value in the risk aggregation framework.* Risk Management Magazine. Italian Association of Financial Industry Risk Managers (AIFIRM). **Candidate's contributions**: Carried out theoretical analyses, algorithm design and part of the experimental campaign.

8. **Picchiotti N.** and others. *Post-Mendelian genetic model in COVID-19*, Cardiology and Cardiovascular Medicine. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

9. Fallerini C., **Picchiotti N.**[6] and others. *Common, low-frequency, rare, and ultra-rare coding variants contribute to COVID-19 severity*, Human Genetics. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

10. Mantovani S., ..., **Picchiotti N.** *Rare variants in Toll-like receptor 7 result in functional impairment and downregulation of cytokine-mediated signaling in COVID-19 patients.* **Candidate's contributions**: Carried out part of experimental campaign.

Peer reviewed conference papers.

11. **Picchiotti N.**, Gori M. *Logic Constraints to Feature Importance.* The 20th International Conference of the Italian Association for Artificial Intelligence and Springer LNCS/LNAI proceedings. **Candidate's contributions**: designed algorithms, carried out theoretical analyses, and experimental setup.

12. **Picchiotti N.**, Gori M. *Clustering-Based Interpretation of Deep ReLU Network.* The 20th International Conference of the Italian Association for Artificial Intelligence and Springer LNCS/LNAI proceedings. **Candidate's contributions**: designed algorithms, carried out theoretical analyses, and experimental setup.

Associated papers within COVID-19 host genetics initiative consortium.

13. *Genetic mechanisms of critical illness in COVID-19.*, Nature. **Candidate's contributions**: active participation to consortium's weekly meetings.

14. *C9orf72 Intermediate Repeats Confer Genetic Risk for Severe COVID-19 Pneumonia Independently of Age.*, Int J Mol Sci. **Candidate's contributions**: active participation to consortium's weekly meetings.

15. *Protective Role of a TMPRSS2 Variant on Severe COVID-19 Outcome in Young Males and Elderly Women.*, Genes. **Candidate's contributions**: active participation to consortium's weekly meetings.

Submitted Papers.

16. **Picchiotti N.**, Spuntarelli A. *A model risk framework for Machine Learning applications in the financial context.* Research in International Business and

[6]Co-first authors

Finance. **Candidate's contributions**: Carried out theoretical analyses, algorithm design and experimental campaign.

17. Charles J. Buchanan, ..., **Picchiotti N.** and others. *Cryptic pathogen-sugar interactions revealed by universal saturation transfer analysis*, Science. **Candidate's contributions**: Carried out algorithm design and part of experimental campaign.

18. Onoja A., **Picchiotti N.** and other. An explainable model of host genetic interactions linked to COVID-19 severity. NPJ Genomic Medicine. **Candidate's contributions**: Carried out part of the algorithm design.

Other Publications (unrefereed contributions).

18. **Picchiotti, N.**, Salvioli, M., Zanardini, E., & Missale, F. *COVID-19 Italian and Europe epidemic evolution: A SEIR model with lockdowndependent transmission rate based on Chinese data*, Available at SSRN 3562452. **Candidate's contributions**: Carried out algorithm design and experimental campaign.

Thesis supervision.

19. Assistant supervisor for the Master thesis *"Advanced statistical modelling for Credit Scoring"*, Alessia Amanti, faculty of mathematics; "Università Cattolica del Sacro Cuore", Brescia, 7th July, 2020.

20. Assistant supervisor for the Master thesis *"Advanced machine learning methods to understand the genetic mechanism of COVID-19 severity and multi-organ involvement"*, Marco Tanfoni, faculty of mathematics; "Università degli studi di Siena", Siena, 16th April, 2021.

Other works.

21. *Web sentiment model in the reputational risk framework.* Internal report. Banco BPM.

22. *Unsupervised Machine Learning model for credit spread proxy association.* Internal report. Banco BPM.

23. AIFIRM commission for the publish of the position paper: *Artificial Intelligence e Credit Risk Models* working group led by Giovanni Della Lunga (MPS)

Bibliography

[1] Nicola Picchiotti, Monica Salvioli, Elena Zanardini, and Francesco Missale. Covid-19 pandemic: a mobility-dependent seir model with undetected cases in italy, europe and us. *arXiv preprint arXiv:2005.08882*, 2020.

[2] Chiara Fallerini, Nicola Picchiotti, Margherita Baldassarri, Kristina Zguro, Sergio Daga, Francesca Fava, Elisa Benetti, Sara Amitrano, Mirella Bruttini, Maria Palmieri, et al. Common, low-frequency, rare, and ultra-rare coding variants contribute to covid-19 severity. *medRxiv*, 2021.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[5] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[6] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[7] 2018 reform of eu data protection rules, `https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf`.

[8] Michelangelo Diligenti, Marco Gori, Marco Maggini, and Leonardo Rigutini. Bridging logic and kernel machines. *Machine learning*, 86(1):57–88, 2012.

[9] Edward Livingston and Karen Bucher. Coronavirus disease 2019 (covid-19) in italy. *Jama*, 323(14):1335–1335, 2020.

[10] Xiaonan Zhang, Yun Tan, Yun Ling, Gang Lu, Feng Liu, Zhigang Yi, Xiaofang Jia, Min Wu, Bisheng Shi, Shuibao Xu, et al. Viral and host factors related to the clinical outcome of covid-19. *Nature*, 583(7816):437–440, 2020.

[11] Martin Leo, Suneel Sharma, and Koilakuntla Maddulety. Machine learning in banking risk management: A literature review. *Risks*, 7(1):29, 2019.

[12] Saqib Aziz and Michael Dowling. Machine learning and ai for risk management. In *Disrupting finance*, pages 33–50. Palgrave Pivot, Cham, 2019.

[13] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

[14] Jan De Spiegeleer, Dilip B Madan, Sofie Reyners, and Wim Schoutens. Machine learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance*, 18(10):1635–1643, 2018.

[15] Ion Smeureanu, Gheorghe Ruxanda, and Laura Maria Badea. Customer segmentation in private banking sector using machine learning techniques. *Journal of Business Economics and Management*, 14(5):923–939, 2013.

[16] Francesco Rundo, Francesca Trenta, Agatino Luigi di Stallo, and Sebastiano Battiato. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574, 2019.

[17] Saeed Nosratabadi, Amirhosein Mosavi, Puhong Duan, Pedram Ghamisi, Ferdinand Filip, Shahab S Band, Uwe Reuter, Joao Gama, and Amir H Gandomi. Data science in economics: comprehensive review of advanced machine learning and deep learning methods. *Mathematics*, 8(10):1799, 2020.

[18] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28:2503–2511, 2015.

[19] Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of ai ethics guidelines. *Nature Machine Intelligence*, 1(9):389–399, 2019.

[20] Nathalie A Smuha. The eu approach to ethics guidelines for trustworthy artificial intelligence. *Computer Law Review International*, 20(4):97–106, 2019.

[21] Sr 11-7: Guidance on model risk management, `https://www.federalreserve.gov/supervisionreg/srletters/sr1107.htm`.

[22] EBA. *EBA report on big data and advanced analytics*. 2020.

[23] Martin Zinkevich. Rules of machine learning: Best practices for ml engineering. *URL: https://developers.google.com/machine-learning/guides/rules-of-ml*, 2017.

[24] Emmanuel Ameisen. *Building Machine Learning Powered Applications: Going From Idea to Product*. " O'Reilly Media, Inc.", 2020.

[25] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 2020.

[26] Salman Sherin, Muhammad Zohaib Iqbal, et al. A systematic mapping study on testing of machine learning programs. *arXiv preprint arXiv:1907.09427*, 2019.

[27] Model behavior. nothing artificial., `https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/financial-services/deloitte-uk-ai-and-risk-management.pdf`.

[28] Ai and risk management innovating with confidence, `https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/financial-services/deloitte-uk-ai-and-risk-management.pdf`.

[29] Model risk management of ai and machine learning system., `https://www.pwc.co.uk/data-analytics/documents/model-risk-management-of-ai-machine-learning-systems.pdf`.

[30] Ian Scott, Stacy Carter, and Enrico Coiera. Clinician checklist for assessing suitability of machine learning applications in healthcare. *BMJ Health & Care Informatics*, 28(1), 2021.

[31] Houssem Ben Braiek and Foutse Khomh. On testing machine learning programs. *Journal of Systems and Software*, 164:110542, 2020.

[32] Christian Murphy, Gail E Kaiser, and Marta Arias. An approach to software testing of machine learning applications. 2007.

[33] Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, and D Sculley. The ml test score: A rubric for ml production readiness and technical debt reduction. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1123–1132. IEEE, 2017.

[34] Xiaoyuan Xie, Joshua WK Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. Testing and validating machine learning classifiers by metamorphic testing. *Journal of Systems and Software*, 84(4):544–558, 2011.

[35] Sotiris B Kotsiantis, Dimitris Kanellopoulos, and Panagiotis E Pintelas. Data preprocessing for supervised leaning. *International Journal of Computer Science*, 1(2):111–117, 2006.

[36] Salvador García, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*, volume 72. Springer, 2015.

[37] Salvador García, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, and Francisco Herrera. Big data preprocessing: methods and prospects. *Big Data Analytics*, 1(1):1–22, 2016.

[38] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9572–9581, 2019.

[39] Jonathan I Maletic and Andrian Marcus. Data cleansing: A prelude to knowledge discovery. In *Data mining and knowledge discovery handbook*, pages 19–32. Springer, 2009.

[40] Nick Hynes, D Sculley, and Michael Terry. The data linter: Lightweight, automated sanity checking for ml data sets. In *NIPS MLSys Workshop*, 2017.

[41] Sanjay Krishnan, Michael J Franklin, Ken Goldberg, and Eugene Wu. Boostclean: Automated error detection and repair for machine learning. *arXiv preprint arXiv:1711.01299*, 2017.

[42] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[43] Yvan Saeys, Inaki Inza, and Pedro Larranaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[44] Aleix M Martinez and Avinash C Kak. Pca versus lda. *IEEE transactions on pattern analysis and machine intelligence*, 23(2):228–233, 2001.

[45] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[46] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[47] Michael W Browne. Cross-validation methods. *Journal of mathematical psychology*, 44(1):108–132, 2000.

[48] Matjaz Kukar, Igor Kononenko, et al. Cost-sensitive learning with neural networks. In *ECAI*, volume 15, pages 88–94. Citeseer, 1998.

[49] Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the vc-dimension of a learning machine. *Neural computation*, 6(5):851–876, 1994.

[50] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[51] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[53] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

[54] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[55] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[56] House prices - advanced regression techniques, `https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data?select=data_description.txt`.

[57] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020.

[58] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[59] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[60] Paul Schwerdtner, Florens Greßner, Nikhil Kapoor, Felix Assion, René Sass, Wiebke Günther, Fabian Hüger, and Peter Schlicht. Risk assessment for machine learning models. *arXiv preprint arXiv:2011.04328*, 2020.

[61] ECB TRIM Guide. Guide for the targeted review of internal models (trim). *European Central Bank*, 2017.

[62] Tomaso Aste. What machines can learn about our complex world-and what can we learn from them? *Available at SSRN 3797711*, 2021.

[63] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[64] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[65] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[66] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[67] Friedrich Recknagel, Mark French, Pia Harkonen, and Ken-Ichi Yabunaka. Artificial neural network approach for modelling and prediction of algal blooms. *Ecological Modelling*, 96(1-3):11–28, 1997.

[68] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[69] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[70] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[71] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Gradients of counterfactuals. *arXiv preprint arXiv:1611.02639*, 2016.

[72] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[73] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[74] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[75] Antonio Lugoboni, Nicola Picchiotti, and Andrea Spuntarelli. Risk allocation with shapley value in the risk aggregation framework. *Risk Management Magazine. Italian Association of Financial Industry Risk Managers (AIFIRM)*, 2021.

[76] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

[77] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013.

[78] C Molnar, S Gruber, and P Kopper. Limitations of interpretable machine learning methods, molnar, c and gruber, s and kopper, p, 2020.

[79] Independent high-level expert group on artificial intelligence set up by the european commission. The assessment list for trustworthy artificial intelligence (altai) for self assessment. 2020.

[80] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[81] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

[82] Edmond Awad, Sohan Dsouza, Azim Shariff, Iyad Rahwan, and Jean-François Bonnefon. Universals and variations in moral decisions made in 42 countries by 70,000 participants. *Proceedings of the National Academy of Sciences*, 117(5):2332–2337, 2020.

[83] Pratik Gajane and Mykola Pechenizkiy. On formalizing fairness in prediction with machine learning. *arXiv preprint arXiv:1710.03184*, 2017.

[84] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[85] Juliana Cesaro and Fabio G Cozman. Measuring unfairness through game-theoretic interpretability. *arXiv preprint arXiv:1910.05591*, 2019.

[86] Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *arXiv preprint arXiv:1703.06856*, 2017.

[87] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. The case for process fairness in learning: Feature selection for fair decision making. In *NIPS Symposium on Machine Learning and the Law*, volume 1, page 2, 2016.

[88] Toon Calders and Indrė Žliobaitė. Why unbiased computational processes can lead to discriminative decision procedures. In *Discrimination and privacy in the information society*, pages 43–57. Springer, 2013.

[89] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568, 2008.

[90] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18. IEEE, 2009.

[91] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[92] Roman V Yampolskiy. *Artificial intelligence safety and security*. CRC Press, 2018.

[93] Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. Measuring neural net robustness with constraints. *Advances in neural information processing systems*, 29:2613–2621, 2016.

[94] Kush R Varshney. Engineering safety in machine learning. In *2016 Information Theory and Applications Workshop (ITA)*, pages 1–5. IEEE, 2016.

[95] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018.

[96] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[97] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[98] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3520–3532, 2017.

[99] California consumer privacy act, `https://oag.ca.gov/privacy/ccpa`.

[100] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[101] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[102] H IJ. Statistics versus machine learning. *Nature methods*, 15(4):233, 2018.

[103] Iain Carmichael and JS Marron. Data science vs. statistics: two cultures? *Japanese Journal of Statistics and Data Science*, 1(1):117–138, 2018.

[104] Xingyuan Pan and Vivek Srikumar. Expressiveness of rectifier networks. In *International Conference on Machine Learning*, pages 2427–2435. PMLR, 2016.

[105] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[106] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[107] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[108] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[109] Ridwan Al Iqbal. Empirical learning aided by weak domain knowledge in the form of feature importance. In *2011 International Conference on Multimedia and Signal Processing*, volume 1, pages 126–130. IEEE, 2011.

[110] Steve Diersen, En-Jui Lee, Diana Spears, Po Chen, and Liqiang Wang. Classification of seismic windows using artificial neural networks. *Procedia computer science*, 4:1572–1581, 2011.

[111] Lei Zhang and Zhichao Wang. Ontology-based clustering algorithm with feature weights. *Journal of Computational Information Systems*, 6(9):2959–2966, 2010.

[112] Xiaobing Peng and Yuquan Zhu. A novel feature weighted strategy on data classification. In *2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things (CCIOT)*, pages 589–594. IEEE, 2018.

[113] Ridwan Al Iqbal. Using feature weights to improve performance of neural networks. *arXiv preprint arXiv:1101.4918*, 2011.

[114] Dheeru Dua and Casey Graff. UCI machine learning repository, `http://archive.ics.uci.edu/ml`, 2017.

[115] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

[116] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

[117] Covid-19 situazione italia, dipartimento protezione civile, `https://opendatadpc.maps.arcgis.com/apps/dashboards/b0c68bce2cce478eaac82fe38d4138b1` ,

[118] Google mobility data, google, `https://www.google.com/covid19/mobility/`.

[119] M Rafiul Islam, M Nazmul Hoque, M Shaminur Rahman, ASM Rubayet Ul Alam, Masuda Akther, J Akter Puspo, Salma Akter, Munawar Sultana, Keith A Crandall, and M Anwar Hossain. Genome-wide analysis of sars-cov-2 virus strains circulating worldwide implicates heterogeneity. *Scientific reports*, 10(1):1–9, 2020.

[120] Polygenic risk scores, national human genome research institute, `https://www.genome.gov/Health/Genomics-and-Medicine/Polygenic-risk-scores`.

[121] Michael Molla, Michael Waddell, David Page, and Jude Shavlik. Using machine learning to design and interpret gene-expression microarrays. *AI Magazine*, 25(1):23–23, 2004.

[122] Severe Covid-19 GWAS Group. Genomewide association study of severe covid-19 with respiratory failure. *New England Journal of Medicine*, 383(16):1522–1534, 2020.

[123] Erola Pairo-Castineira, Sara Clohisey, Lucija Klaric, Andrew D Bretherick, Konrad Rawlik, Dorota Pasko, Susan Walker, Nick Parkinson, Max Head Fourman, Clark D Russell, et al. Genetic mechanisms of critical illness in covid-19. *Nature*, 591(7848):92–98, 2021.

[124] Caspar I Van Der Made, Annet Simons, Janneke Schuurs-Hoeijmakers, Guus Van Den Heuvel, Tuomo Mantere, Simone Kersten, Rosanne C Van Deuren, Marloes Steehouwer, Simon V Van Reijmersdal, Martin Jaeger, et al. Presence of genetic variants among young men with severe covid-19. *Jama*, 324(7):663–673, 2020.

[125] Jack A Kosmicki, Julie E Horowitz, Nilanjana Banerjee, Rouel Lanche, Anthony Marcketta, Evan Maxwell, Xiaodong Bai, Dylan Sun, Joshua D Backman, Deepika Sharma, et al. Pan-ancestry exome-wide association analyses of covid-19 outcomes in 586,157 individuals. *The American Journal of Human Genetics*, 2021.

[126] Sergio Daga, Chiara Fallerini, Margherita Baldassarri, Francesca Fava, Floriana Valentino, Gabriella Doddato, Elisa Benetti, Simone Furini, Annarita Giliberti, Rossella Tita, et al. Employing a systematic approach to biobanking and analyzing clinical and genetic data for advancing covid-19 research. *European Journal of Human Genetics*, 29(5):745–759, 2021.

[127] Covid-19 therapeutic trial synopsis. who r&d blueprint novel coronavirus. covid 19 therapeutic trial synopsis. 2020.

[128] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

[129] Zunyou Wu and Jennifer M McGoogan. Characteristics of and important lessons from the coronavirus disease 2019 (covid-19) outbreak in china: summary of a report of 72 314 cases from the chinese center for disease control and prevention. *Jama*, 323(13):1239–1242, 2020.

[130] Kenneth I Zheng, Gong Feng, Wen-Yue Liu, Giovanni Targher, Christopher D Byrne, and Ming-Hua Zheng. Extrapulmonary complications of covid-19: A multisystem disease? *Journal of Medical Virology*, 93(1):323–335, 2021.

[131] Guang Chen, DI Wu, Wei Guo, Yong Cao, Da Huang, Hongwu Wang, Tao Wang, Xiaoyun Zhang, Huilong Chen, Haijing Yu, et al. Clinical and immunological features of severe and moderate coronavirus disease 2019. *The Journal of clinical investigation*, 130(5):2620–2629, 2020.

[132] Ning Tang, Dengju Li, Xiong Wang, and Ziyong Sun. Abnormal coagulation parameters are associated with poor prognosis in patients with novel coronavirus pneumonia. *Journal of thrombosis and haemostasis*, 18(4):844–847, 2020.

[133] Aditya Ashok, Mahya Faghih, and Vikesh K Singh. Mild pancreatic enzyme elevations in covid-19 pneumonia: synonymous with injury or noise? *Gastroenterology*, 160(5):1872, 2021.

[134] Christopher Halloran. Emerging phenotype of sars-cov2 associated pancreatitis. short title: Sars-cov2 associated pancreatitis. *Gastroenterology*, 2020.

[135] Zhe Xu, Lei Shi, Yijin Wang, Jiyuan Zhang, Lei Huang, Chao Zhang, Shuhong Liu, Peng Zhao, Hongxia Liu, Li Zhu, et al. Pathological findings of covid-19 associated with acute respiratory distress syndrome. *The Lancet respiratory medicine*, 8(4):420–422, 2020.

[136] Ying-Ying Zheng, Yi-Tong Ma, Jin-Ying Zhang, and Xiang Xie. Covid-19 and the cardiovascular system. *Nature Reviews Cardiology*, 17(5):259–260, 2020.

[137] Saiping Jiang, Rongrong Wang, Lu Li, Dongsheng Hong, Renping Ru, Yuefeng Rao, Jing Miao, Na Chen, Xiuhua Wu, Ziqi Ye, et al. Liver injury in critically ill and non-critically ill covid-19 patients: a multicenter, retrospective, observational study. *Frontiers in medicine*, 7:347, 2020.

[138] Gong Feng, Kenneth I Zheng, Qin-Qin Yan, Rafael S Rios, Giovanni Targher, Christopher D Byrne, Sven Van Poucke, Wen-Yue Liu, and Ming-Hua Zheng. Covid-19 and liver dysfunction: current insights and emergent therapeutic strategies. *Journal of clinical and translational hepatology*, 8(1):18, 2020.

[139] Christopher S von Bartheld, Molly M Hagen, and Rafal Butowt. Prevalence of chemosensory dysfunction in covid-19 patients: a systematic review and meta-analysis reveals significant ethnic differences. *ACS chemical neuroscience*, 11(19):2944–2961, 2020.

[140] G-u Kim, M-J Kim, Sang Hyun Ra, Jeongsoo Lee, Seongman Bae, Jiwon Jung, and S-H Kim. Clinical characteristics of asymptomatic and symptomatic patients with mild covid-19. *Clinical microbiology and infection*, 26(7):948–e1, 2020.

[141] Daniel Lopez-Martinez. Regularization approaches for support vector machines with applications to biomedical data. *arXiv preprint arXiv:1710.10600*, 2017.

[142] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[143] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[144] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[145] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of rule learning*. Springer Science & Business Media, 2012.

[146] Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

[147] Margherita Baldassarri, Nicola Picchiotti, Francesca Fava, Chiara Fallerini, Elisa Benetti, Sergio Daga, Floriana Valentino, Gabriella Doddato, Simone Furini, Annarita Giliberti, et al. Shorter androgen receptor polyq alleles protect against life-threatening covid-19 disease in european males. *EBioMedicine*, 65:103246, 2021.

[148] Catherine Gebhard, Vera Regitz-Zagrosek, Hannelore K Neuhauser, Rosemary Morgan, and Sabra L Klein. Impact of sex and gender on covid-19 outcomes in europe. *Biology of sex differences*, 11:1–13, 2020.

[149] Qian Zhang, Paul Bastard, Zhiyong Liu, Jérémie Le Pen, Marcela Moncada-Velez, Jie Chen, Masato Ogishi, Ira KD Sabli, Stephanie Hodeib, Cecilia Korol, et al. Inborn errors of type i ifn immunity in patients with life-threatening covid-19. *Science*, 370(6515), 2020.

[150] Chiara Fallerini, Sergio Daga, Stefania Mantovani, Elisa Benetti, Nicola Picchiotti, Daniela Francisci, Francesco Paciosi, Elisabetta Schiaroli, Margherita Baldassarri, Francesca Fava, et al. Association of toll-like receptor 7 variants with life-threatening covid-19 disease in males: findings from a nested case-control study. *Elife*, 10:e67569, 2021.

[151] Paul Bastard, Lindsey B Rosen, Qian Zhang, Eleftherios Michailidis, Hans-Heinrich Hoffmann, Yu Zhang, Karim Dorgham, Quentin Philippot, Jérémie Rosain, Vivien Béziat, et al. Autoantibodies against type i ifns in patients with life-threatening covid-19. *Science*, 370(6515), 2020.

[152] Konstantinos Poulas, Konstantinos Farsalinos, and Charilaos Zanidis. Activation of tlr7 and innate immunity as an efficient method against covid-19 pandemic: imiquimod as a potential therapy. *Frontiers in Immunology*, 11:1373, 2020.

[153] Susanna Croci, Mary Anna Venneri, Stefania Mantovani, Chiara Fallerini, Elisa Benetti, Nicola Picchiotti, Federica Campolo, Francesco Imperatore, Maria Palmieri, Sergio Daga, et al. The polymorphism l412f in tlr3 inhibits autophagy and is a marker of severe covid-19 in males. *medRxiv*, 2021.

[154] CT Ranjith-Kumar, William Miller, Jingchuan Sun, Jin Xiong, Jon Santos, Ian Yarbrough, Roberta J Lamb, Juliane Mills, Karen E Duffy, Scott Hoose, et al. Effects of single nucleotide polymorphisms on toll-like receptor 3 activity and expression in cultured cells. *Journal of Biological Chemistry*, 282(24):17696–17705, 2007.

[155] Chiara Fallerini, Sergio Daga, Elisa Benetti, Nicola Picchiotti, Kristina Zguro, Francesca Catapano, Virginia Baroni, Simone Lanini, Alessandro Bucalossi, Giuseppe Marotta, et al. Selp asp603asn and severe thrombosis in covid-19 males. *Journal of hematology & oncology*, 14(1):1–4, 2021.

[156] Christina B Azodi, Jiliang Tang, and Shin-Han Shiu. Opening the black box: Interpretable machine learning for geneticists. *Trends in genetics*, 36(6):442–455, 2020.

[157] Maxwell W Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, 2015.

[158] Daniel R Schrider and Andrew D Kern. Supervised machine learning for population genetics: a new paradigm. *Trends in Genetics*, 34(4):301–312, 2018.

[159] Nicola Picchiotti, Elisa Benetti, Chiara Fallerini, Sergio Daga, Margherita Baldassarri, Francesca Fava, Kristina Zguro, Floriana Valentino, Gabriella Doddato, Annarita Giliberti, et al. Post-mendelian genetic model in covid-19. *medRxiv*, 2021.

[160] Jason DM Rennie and Nathan Srebro. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, volume 1. Citeseer, 2005.

[161] Fabian Pedregosa-Izquierdo. *Feature extraction and supervised learning on fMRI: from practice to theory*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2015.

[162] Abdelazeem Elhabyan, Saja El Yaacoub, Ehab Sanad, Abdelwahab Mohamed, Asmaa Elhabyan, and Valentin Dinu. The role of host genetics in susceptibility to severe viral infections in humans and insights into host genetics of severe covid-19: A systematic review. *Virus research*, page 198163, 2020.

[163] Philip Koopman and Beth Osyk. Safety argument considerations for public road testing of autonomous vehicles. *SAE Technical Paper*, 1(2019-01-0123), 2019.

[164] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of physics*, 326(1):96–192, 2011.

[165] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.