THE RELATIONS BETWEEN PEDAGOGICAL AND SCIENTIFIC EXPLANATIONS OF ALGORITHMS: CASE STUDIES FROM THE FRENCH ADMINISTRATION
DRAFT: PLEASE DO NOT QUOTE OR DISTRIBUTE

A PREPRINT

Mael Pégny *
Archives Poincaré
Université de Lorraine (Nancy)
OLKi Project
maelpegny@gmail.com

April 7, 2020

Keywords AI · Explainability · Interpretability · XAI · Algorithm · ML · France · Bureaucracy · Administration · Philosophy · Social Sciences

ABSTRACT

The opacity of some recent Machine Learning (ML) techniques have raised fundamental questions on their explainability, and created a whole domain dedicated to Explainable Artificial Intelligence (XAI). However, most of the literature has been dedicated to explainability as a scientific problem dealt with typical methods of computer science, from statistics to UX. In this paper, we focus on explainability as a pedagogical problem emerging from the interaction between lay users and complex technological systems. We defend an empirical methodology based on field work, which should go beyond the in-vitro analysis of UX to examine in-vivo problems emerging in the field. Our methodology is also comparative, as it chooses to steer away from the almost exclusive focus on ML to compare its challenges with those faced by more vintage algorithms. Finally, it is also philosophical, as we defend the relevance of the philosophical literature to define the epistemic desiderata of a good explanation. This study was conducted in collaboration with Etalab, a Task Force of the French Prime Minister in charge of Open Data & Open Government Policies, dealing in particular with the enforcement of the right to an explanation. In order to illustrate and refine our methodology before going up to scale, we conduct a preliminary work of case studies on the main different types of algorithms used by the French administration: computation, matching algorithms and ML. We study the merits and drawbacks of a recent approach to explanation, which we baptize *input-output black box reasoning* or *BBR* for short. We begin by presenting a conceptual framework including the distinctions necessary to a study of pedagogical explainability. We proceed to algorithmic case studies, and draw model-specific and model-agnostic lessons and conjectures.

INTRODUCTION

There is a fundamental ambiguity in the current use of the term *explainability* in the AI community. On the one hand, *explainability* or *(human) interpretability* denotes a fundamental scientific problem, the problem of understanding the behavior of complex ML systems, which can lead to the development of sophisticated techniques. On the other hand, the term *explainability* is also used to denote a pedagogical problem, the problem of explaining to a lay audience, be they policy-makers or ordinary citizens, the behavior and outcomes of those systems. Those two challenges are not

*Webpage: https://univ-lorraine.academia.edu/MaelPegny

completely independent: of course, a computer scientist needs to have a firm scientific grasp on a given issue before she tries to give a pedagogical explanation to a lay audience. As a working hypothesis, I do not consider pedagogical and scientific explanation as dichotomic classes, rather as two ends of a spectrum[2]. They nevertheless need to be distinguished if we are to understand the considerable pedagogical challenges raised by ML models and other programs. In this paper, the terms *explanation*, *explainable* and *explainability* will have the pedagogical meaning by default. We will talk about *decision explainability* when the explanans will be an output that can be described as a decision.

By its very origin and design, the XAI literature is subject to an AI-centric bias. Glancing over recent papers, it would seem that opaque ML technics are either the only type of algorithms in need of explainability or that they need a special form of explanation. The first hypothesis is obviously false, and the second is in dire need of discussion. If there is indeed a singularity to AI explainability, especially for its pedagogical aspect, its best characterization should emerge from a comparative approach of the various explainability challenges raised by different kinds of algorithms. This is our first methodological hypothesis.

Our second methodological hypothesis is that such a comparative study can only be fully accomplished through empirical methods. Algorithm analysis in the pure CS tradition might be fine if we only consider the scientific part of explainability, but its pedagogical part cannot be dealt with by pure a priori methods. It is necessary to know the concrete challenges raised by various users in real-world examples to fully grasp the challenges of explanation, and no amount of mathematical work will be able to predict a priori what those challenges are. Psychological experiments in UX are a great step towards such an empirical approach. However, just as biologists complement in vitro experiments with in vivo experiments, it seems necessary to complement lab work with field work, and to see the problems raised by explanations in the wild.

As a consequence, the author of this paper went looking for a field study which would give him contact with a wealth of real-world examples of explainability issues. The possibility of a collaboration with Etalab, the French Prime Minister Task Force in charge of Open Data, Open Algorithms Policies, was discovered by pure luck, and turned out to be very attractive for several reasons. The first is due to the very nature of the public sector. The public sector is, at least to some extent, a centralized organization subjected to stringent legal obligations of transparency. By contrast, the private sector is just a name for a huge number of loosely related organizations, whose algorithms are often protected by IP rights and trade secrets. A collaboration with a government task force such as Etalab gives access to a wealth of data and algorithms that would be extremely hard to obtain with any private organization. What is more, those data and algorithms impact dozens of millions of French citizens and foreigners living on the French territory. If this does not compare with the billions of Facebook users, this gives access to a humongous number of interactions with various social groups. The Etalab task force has also among its missions the implementation of the GDPR right to an explanation, as well as other demands for explainability coming from the French *Loi sur la République numérique*, throughout the French administration. This leads to a genuine operational need of research in the long run on explainability issues, which warrants the quality and duration of the research collaboration. Finally, while it is true that most of the top-knotch research in ML is carried in the private sector, the French administration has not been passive on this front in the last years. Etalab supervises several research projects in AI for the public sector, especially through development projects called *Entrepreneurs d'Intérêt Général* (" Entrepreneurs for the Common Good"). If this research effort might not compare with American or Chinese private sector powerhouses, it is nevertheless a real opportunity to access up-to-date ML research in the making, with an ease and transparency that would be hard to get in the private sector. However, our conclusions will not be limited to "public sector algorithms" in any way. We will examine types of algorithms which are used both in private and public sectors, and our conclusions will bear on scientific and pedagogical explanation in general, not on the specific legal and political issues of public sector algorithms.

This entails an embarrassment of riches: how to proceed when one tries to study a gigantic organization such as the French administration with one full-time researcher, and a couple collaborators with limited availability? This paper presents a preliminary study whose aim is to articulate a conceptual framework, a set of problems and working hypotheses before going up to scale. It is based on a general a priori reflection on our methods and objectives, combined with a small set of explainability case studies of various ML and non-ML algorithms in the French public sector. In this paper, we discuss general issues of explainability, but also try to study the merits and drawbacks of a particular approach to explainability, which we baptize *input-output black box reasoning*, or *Black Box Reasoning* (BBR) for short. In a nutshell, this approach defends the optimistic stance that the outcomes of many AI algorithms can be explained in simple terms without getting into the sophisticated details of the algorithm, via causal reasoning on the relations

[2]The philosopher should not project the analytic tradition attached to the phrase "scientific explanation" onto our current use of this same phrase: "scientific explanation" is just a term to denote the rough definition we provided above, leaving its rigorous philosophical understanding to further discussion. In the same fashion, our working hypothesis on the relations between scientific and pedagogical explanations is not a hard philosophical position, just a call to remain cautious and open-minded before we examine sophisticated philosophical quarrels relevant to our object, such as the opposition between pragmatic and non-pragmatic accounts of scientific explanation in the analytic tradition (woodward, 2019, (fraassen, 1980) (achinstein).

between inputs and outputs. This approach is chosen simply because it looks like a plausible candidate for pedagogical explanation, as it is extremely sober in its technical prerequisites. This choice is not so much an endorsement of BBR as a the golden path to explainability as an acknowledgement that it is a natural candidate to begin an in-depth study of explainability in the real world[3].

In our first section, we try to articulate a conceptual framework for the general issues of pedagogical explainability. We will defend several definitions and distinctions on explanation, present the input-output black box reasoning approach, and discuss the respective roles of empirical methodology and philosophy in our overall methodology. In our second section, we will apply our conceptual framework to four case studies of algorithms used by the French public sector, and draw model-specific and model-agnostic lessons for pedagogical and scientific explanation.

Before we proceed, it is necessary to justify a particular feature of this work. Our case studies have raised issues which would be treated in separate papers in the XAI literature, which calls for two remarks. The first remark is that, since we want to establish a list of issues that we will carry through a large scale study of explainability challenges, our aim is precisely to centralize problems that might be dispersed in the literature. The second remark is that, in our limited experience, explainability issues show up very early, for very simple examples, they are much more mdel-agnostic than I have expected, and they are probably hard. This point could only be made convincingly by putting all our case studies in one place, and telling through anecdotes how and when those issues appeared in this research, which was always sooner than expected.

Contents

[3]This should not be read as a hint that this type of approach is not relevant for scientific explanations. Counterfactual reasoning, which can roughly be seen as a branch of BBR, has recently been presented as the emerging standard in XAI [Poyiadzi et al.(2020)Poyiadzi, Sokol, Santos-Rodriguez, De Bie, and Flach].

1 The Challenges of Pedagogical Explanation: A Conceptual Framework

1.1 The Role of Philosophy

The reader might be surprised to find a philosopher of science as a lead investigator in a work at the crossroads of XAI, UX and sociology. This presence is not an institutional anecdote, as I believe that philosophy has an important role to play in this investigation, even if its full extent might not be apparent at the first stages of this research. Let me now explain my vision of this role.

There is a huge philosophical literature on the problem of explanation, its nature, its normative value and cognitive effects. However, as rightly noted Tim Miller in [Miller(2019)] and Wachter et al. in [Mittelstadt, Brent et al.(2019)Mittelstadt, Brent, Russell, Chris, and Wachter, Sandra] , this literature has been largely ignored in the XAI literature on explainability. The first reason is cultural and contingent: computer scientists will more spontaneously think of empirical psychology and law when they look for interdisciplinary collaborations on those issues [Doshi-Velez(2018)]. The second is more structural. The philosophical literature on explanation has been largely dominated by issues related to explanations of phenomena in natural science and everyday experience[4]. A quick search for the word "algorithm" shows that it does not appear in the entire article. There also exists a subliterature on explanation in mathematics, which covers both the explanatory power of mathematics in empirical sciences and explanation within mathematics. Even more surprisingly, a search for the word "algorithm" in the "Explanation in Mathematics article also gives no result. Consequently, phenomena, patterns, experiments, laws, theories and models have been considered typical objects of explanation, not algorithms. A considerable and delicate effort has to be made in order to adapt this body of literature to the specific challenges of algorithms and decision-making processes in general. The road between philosophy and computer science is a two-way street: if computer scientists will have to work to discover philosophy of explanation, philosophers will also have to provide an effort to adapt traditional definitions and arguments to the new objects created by computer science, or possibly provide new ones. However, it would be surprising if some nuggets could not be found in such a sophisticated intellectual tradition, and the recent success of counterfactuals, a classic topic for philosophers, in the XAI literature is ample demonstration that this expectation is reasonable [Poyiadzi et al.(2020)Poyiadzi, Sokol, Santos-Rodriguez, De Bie, and Flach]. Furthermore, the philosophical literature relevant to XAI should not be reduced to the analytic tradition on scientific explanation. Cognate topics, such as justification in the philosophy of action or the philosophy of law, causal reasoning [noa()] understanding (see [Páez(2019)] and references there) and many more might also be relevant. This is the first reason why philosophy seems important for our present purposes, and even if the adaptation of the explanation literature to the challenges of modern computer science is a long-term work which shall not be conducted in this paper, we hope to show in the next subsections a couple examples of the interest of philosophical analysis.

The interest of philosophy only grows when one takes into consideration the more normative aspect of our work. For XAI to be successful, especially for explanations to the general public, it is not only a description of how human subjects do produce explanations that it is needed, but a normative reflection on what kind of explanation is, just to mention a couple epistemic and moral virtues, relevant, honest, accurate and accessible. Such a reflection is particularly needed for the formulation of industrial standards and legal requirements. At first sight, it might be tempting to think that the description of explanatory practices belong to empirical sciences such as psychology and sociology, while the normative reflection belongs to philosophy. Such a distinction, if not completely inaccurate, would be too schematic, since normative assessments are part of practice in explanation. However, it is true that philosophy has a rich tradition on norms, be they epistemic or moral, and this tradition should be used to its full extent.

There are other reasons why philosophy might come into play. The author of this paper is a self-defined member of an intellectual tradition, very well developed in France, of interactions between history and philosophy of science. This tradition might be called *historical epistemology*, or *Integrated History and Philosophy of Science* (IHPS)[5]. This integration of history and philosophy should not be read as a rejection of a priori methods in philosophy, or even a defense of some *empirical turn* in philosophy : it is just grounded in the belief that philosophical reflection, in its most traditional form, can be informed by the study of actual practices, without ever being reduced to or directly determined

[4]The reader unfamiliar with the philosophical literature might just take a look at the "Scientific Explanation" article in the Stanford Encyclopedia of Philosophy, the international encyclopedic reference for philosophy in the English language, which states the same point [Woodward(2019)].

[5]The subtle differences and theoretical quarrels between those two denominations do not interest us for the time being. The use of the term "history" rather than sociology for the study of present-time phenomena might be puzzling for some, even if the history of present time has become a standard branch of historical studies in recent decades. But again, quarrels on the proper delimitation of history, sociology, anthropology, ethnography etc., or lack of relevance thereof, are not an urgent concern for this work.

by the findings of social sciences. In particular, the author has developed an interest for the history of cognitive technics such as writing and computation, and their impacts on the knowledge and organization of societies: this makes the interest for the use of complex CS technics in extra-scientific contexts all the more natural.

Furthermore, it is our working hypothesis that the normative function of philosophy might be inseparable from the empirical work done by the social sciences. A normative reflection on explanation should not only bear on abstract epistemic virtues of a good explanation, but it should also be a reflection on the set of explanatory questions themselves: which questions do people ask? which questions should they ask? Without considerable empirical work on this issue, a priori reflection might reproduce biases in current administrative practices, where some populations have very little interaction with administrative algorithms or very little understanding of them. A priori normative reflections and empirical investigation have to work hand in hand to provide an examination of both the limitations of empirical data and the limitations of a priori construction of our object.

Finally, there is a more particular ambition at play here. One of our goals is to actually turn bureaucratic decision-making into an object for philosophy of science. Bureaucracy has been a classical object for social sciences, political science, the anthropology of norms and critical theory. On the other hand, computer simulations and algorithms have been an object respectively for the philosophy of scientific modeling and simulation, and the foundations of computation theory. Those two traditions might discuss phenomena such as "algorithms", "algorithms" and "computerization of practices" in their own way, but, to the extent of our knowledge, their interface has been practically null. However, a cross-fertilization of those traditions is made necessary by the increasing use of opaque, complex algorithmic systems in everyday settings, in particular bureaucratic institutions with huge impacts on the lives of millions of men and women. The epistemological interrogations on the knowledge produced by those models immediately translate into ethical and political problems, and ethical and political ambitions might translate into impressive technical challenges, bringing their own lot of epistemological issues. This paper is a first step in this integration of philosophy of science and social sciences on the object of algorithmic aids to decision-making.

1.2 Explanation of the algorithm vs Explanation of the Output

At first sight, the demand for explanation of ML algorithms might face a fatalistic attitude in the lay public: how could we possibly explain algorithms even computer scientists have a hard time to understand[6]? It is crucial to see that this fatalistic attitude is not warranted. Explaining a algorithm does not necessarily mean understanding its intricate workings. It might be possible to answer certain questions with relevant elements of information at the appropriate level of abstraction. The complexity of the explanation is thus not only a function of the complexity of the algorithm, but also of the nature of the question, and the kind of knowledge and ability to decide and to act that should be enabled by a relevant explanation. We will soon see that the BBR approach is based on the assumption that many demands for explanation can be met without getting into an actual exposition of the algorithm itself.

In order to understand the rationale behind this assumption, it is necessary to distinguish between the explanation of a given algorithm, and an explanation of a given output of that algorithm. This distinction is sometimes made in the recent literature as the distinction between local and global explanation £ref finale doshi-velez 2018, £review for health-care, molnar?. From a purely theoretical point of view, those two issues are not completely independent: if one were to put someone in charge of giving explanations on the outputs of an algorithm, it would of course be required that this person understands it in its entirety. However, a practical demand for the explanation of a given output can be met without explaining the entire algorithm, since some parts of it might be irrelevant for a given output. This can greatly simplify the explanation needed.

This simplification is all the more relevant if we accept the following conjecture on the needs for explainability. This conjecture can be phrased as follows: in most practical cases, what the public needs is an explanation of the output, not an explanation of the algorithm. This conjecture should be tested by a systematic empirical study. However, it seems fairly reasonable a priori. With this distinction and conjecture in mind, we can re-phrase our initial problem in the following fashion: how can we satisfy a majority of demands for explanations of the output without having to explain the entire algorithm? A positive answer to this question would allow algorithms to grow in size and sophistication without compromising the right to an explanation.

[6] In ML, it is more common to speak of a "model" or "system" than it is to name '"algorithm" the end product of learning. That departure from conventional CS terminology is fully understandable, as machine learning generates programs that do not simply implement algorithms, but are the offspring of a given learning algorithm and a given dataset. However, since we have to compare ML with other branches of CS where the term "algorithm" is completely standard, and we might even have to consider cases where the algorithm is not implemented by a computer program but executed by hand, or might not even be fully defined as are some decision processes, the term "algorithm" was a natural choice for our explanandum.

The distinction between explanation of the algorithm and explanation of the output is also crucial to design proper communication channels with the public. The first reflex of a computer scientist might be to write a careful explanation of her algorithm in the documentation, and make this documentation publicly available. However, as Quentin Loridant, a data scientist working on the CibNav ML project (see section £ref below), painfully experienced, this choice of communication channel fails radically in front of a fundamental sociological fact: techies read doc, normal people don't. Even in a professional setting, where public agents were supposed to master a new tool in their daily work, the analytics showed an appallingly low rate of interaction with the documentation for an extremely reduced number of agents. It seems that non-computer scientists are not quite fond of lengthy explanations of algorithms. It is better to develop interactive tools such as search engines, simulators and interactive explainers (£ref) which allow the lay public to answer the particular questions on particular outcomes in which they are interested, and to eventually work their way up to a more global understanding of the algorithm if that is relevant for them. This inductive approach is more in phase with a fundamental hypothesis on the intellectual division of labor: users come to a technical system with a particular task in mind, and the general explanation of the system is only a costly road to that particular end, so costly that it would more often than not lead users to give up on a particular system rather than go through it. Instead of lamenting on the lack of curiosity and intellectual rigor of the general public, it is better to take their cognitive and time constraints at heart, and to provide them with the quickest available road to a solution of their specific problem. With this in mind, global explanation should only be provided as the first step of explanation when no local explanation is available[7].

1.3 Presentation of Input-Output Black Box Reasoning

Before we move on to other challenges of explanation, we would like to present briefly an approach in the XAI literature that will be of great interest to us. It does not really have an official name, so we will call it input-output black box reasoning, or BBR for short.

The BBR approach that we present was first developed in [Doshi-Velez(2018)] by the Harvard Berkman Klein Center Working Group on Explanation and the Law and by [Wachter et al.(2017)Wachter, Mittelstadt, and Russell] at the Alan Turing Institute. We will attribute them the authorhood of this approach even if they have not named it that way, and might see differences between their respective approaches that we will not discuss.

Let A be an algorithm whose outputs we have to explain. We will consider A as a black box, and only look at the list of inputs and outputs, and how they correlate. As a running toy example, we will imagine a state-run insurance system, where the algorithm A is used to attribute an insurance package to a given user on the basis of the information she provided. Let us see what information pure input-output black box reasoning enables us to get and the legal questions that information allows to answer. We quote from [Doshi-Velez(2018)], except for the toy examples of possible legal questions on the decision:

- **List of factors taken into account in a given decision and their value, ideally ordered by significance**. Ex: was a protected variable, such as gender, race, sexual orientation or confession, used in the decision?
- **Impact of a factor on a given input**. Ex: your insurance package has changed because your age is above 35.
- **Counterfactual reasoning**. In the philosophical literature, this is just a technical name for "what would happen if...?" questions and answers. Ex: "Would I get a different package if my annual income increases by 10ks?" Counterfactual reasoning is fundamental for strategic adaptations, because it explains to users what would happen if some relevant parameters change.

Input-output black box reasoning enables to see causal relations between inputs and outputs without any knowledge of the algorithm. It offers the possibility to answer a wide range of typical questions on decisions, such as the presence of a given parameter, the exact role played by a given parameter in a given decision, and possible alternative decisions if some parameter values were altered. As it was noticed in [Wachter et al.(2017)Wachter, Mittelstadt, and Russell], this is relevant and at least in some cases sufficient information to understand why a particular decision was reached, contest an adversarial decision, and understand what could be changed in order to achieve a preferable result in the future. Since this seems a good first description of the objectives behind a demand for explanation, the black box reasoning approach is thus a sound starting point to explore the challenges raised by explainability, especially local explainability. More importantly, the BBR approach is simple and intuitive, which makes it a natural candidate for pedagogical explanation, while some other approaches in explanation are so sophisticated from the get-go that they cannot even be considered for that purpose £ref LIME, SHAP, saliency mapping, self-interpretable neural networks.

[7]The reader interested in this topic should find more remarks and references at the beginning of Tim Miller's review paper £ref miller, where he discusses evidence that people naturally go for explanations of local decisions in everyday life rather than going through general theories.

However, as we have said above (£ref section), this does not mean that it does not relevant for scientific explanation, which is further evidence that the two types should not be strictly separated[8].

As we will see below (£ref section), BBR will probably not be sufficient for all practical cases, and the first authors to put this approach forward did not pretend it would. Capturing a small majority of the cases would already be a major feat: it would give us a by default model of how explanations of decision-aiding algorithms should work. In order to be confident that input-output black box reasoning can actually achieve this, it should be tested on a wide variety of case studies including actual legal cases, interviews with experts and lay subjects and most importantly, empirical tests with real subjects. In this paper, we will not conduct detailed BBR studies of each algorithm, but will focus on two issues. The first is a philosophical analysis of the scope of this approach. The second is a synthesis of various methodological challenges we have noticed during the study of our examples, which should be of interest for any future experiment with the BBR approach. Let us now turn to the philosophical analysis of the scope of BBR.

1.4 Black Box Reasoning: Explanation of the Specification or Explanation of the Implementation?

Another pivotal distinction that is not made as often as it should is the distinction between an explanation of the specification of the algorithm and the explanation of the algorithm itself. Let us clarify our terminology first. In this paper, the term "function" will only be used in the mathematical sense of a set-theoretic application, not in the CS sense denoting a particular module written in a particular programming language. The expression "implementation of a function" will be used to denote the process of selecting an algorithm to compute a given set-theoretic function, while the term "implementing an algorithm" would denote the process of selecting various programming features in order to run that algorithm on a computer. The expression "specification of an algorithm" or "algorithmic specification" will be used to denote the set-theoretic function that that algorithm is supposed to compute. In this perspective, several algorithms with very different designs and performances might have the same specification.

However, the terms "specification" and "implementation" are not only used in those narrow acceptions in CS parlance, especially if we take into account the terminology used in the industry. In this terminology, the terms "specification" and "implementation" do not denote a well-defined set of objects: being part of the specification or being part of the implementation is relative to a given level of abstraction of your analysis. At some level of abstraction, the statement "if variable x increases by 1, your total score increases by 30 points" is already part of implementation, something you might write down in one line in a high-level programming language. At another level, it might be read as specification for someone managing the implementation of counter incrementation. Any given statement might be read as specification if it can be construed as a functional description which might be realized in different fashions at a lower level of abstraction, or as an implementation of a functional description made at a higher level of abstraction[9]

Consequently, we will use the expression "explanation of the specification" to denote the explanation of the choice of a particular (set-theoretic) function to represent an informal or semi-formal task. This includes explaining why a particular list of arguments were considered both necessary and sufficient to fulfill such a task, and why the values of this function is a good model of the goal we are trying to achieve. Such an explanation of the specification already abstracts away from many concrete details of the implementation happening at a lower level of abstraction. We will use the expression "explanation of the implementation" to denote the explanation of choices of algorithmic and programming features to implement a given (set-theoretic) function.

This distinction is all the more important as obtaining a perfectly defined specification of a given informal task is by no means a menial task. Many pieces of software in industrial use only have a semi-formal specification. It might be that applying formal methods to a given task would be too difficult and costly for some applications. However, in some other cases, the mere possibility of formal specification is an open question. ML techniques were by and large conceived for tasks where the formulation of a rigorous model of the task was a mind-boggling challenge £ref doshi-velez. It is not obvious to say today whether this is a transitory state of the art or a fundamental scientific limitation. Consequently, the explanation of the specification is a fundamentally hard task in ML: for instance, we do not know why multiplying a given vector by a matrix of neuron weights gives us the result that we want in terms of image classification. However, the challenges of specification explanation are by no means limited to ML tasks, but may take other forms. For some human issues, it is not so much that formal specification is too hard to be achieved, but that looking for a rigorous specification of the task does not even makes sense and might just be a case of over-formalization. Formal and semi-formal specifications of a given task might be conceived as a translation from natural language to a language that might be amenable to computer science treatment, and the faithfulness of that translation might also be an explanandum. However, this crucial distinction between specification and implementation is not only seldom made in

[8] It is probably not a coincidence that this approach emerged in working groups that had legal scholars among them.

[9] I wish to thank Henri Sahla and Bastien Guerry for making me notice this point.

public debates, but also in the XAI literature, probably because of the ML-centric bias that immediately pushes to the side questions of specification.

One might then wonder: is the BBR approach an explanation of the specification or an explanation of the implementation? Giving a list of inputs ideally ordered by significance is undoubtedly part of specification: it is part of the description of the function the algorithm is supposed to compute. Counterfactual reasoning will provide statements of the form "if the value of variable x_0 is in the interval I_0, ..., the value of x_{n-1} in interval I_{n-1}, then the value of output y will be in interval J". Those statements provide constraints on the relation between inputs and output, which may be part of the explanation of the specification.

The hard part is the question on the impact of a given input. This may take many different forms. For a decision problem, you might have statement of the form "if variable x has a value beyond threshold t, then the decision will be positive, no matter what the values of the other inputs are." For a weighted sum determining a given score, you might have statement of the form "variable x accounts for 80% of your points" or "if variable x increases by 1, your total score increases by 30 points." Those statements might be read as constraints on the behavior of the function on various subdomains of the dataspace, and hence as part of the specification. However, the question of the "impact" of a given input is so vast and fuzzy that it might take various forms that are hard to synthetize.

In any case, the BBR approach purposefully avoids to get down and dirty with the details of the algorithm, and seeks to provide simple explanations via the description of functional relations: in its spirit, it is about specification, not implementation. The paradoxical reason why the BBR approach might be successful in providing simple explanations of algorithm is because it does not provide explanations of algorithms at all. Focusing on the specification is a natural move for an agent trying to simplify explanations by abstracting away from details. The key problem, which cannot be answered a priori, is the relevance of that abstraction level for questions met in practice.

1.5 Descriptive and Contrastive Explanation: Explaining What it is, Explaining Why It Is The Way It Is

There is a fundamental distinction between two concepts of explanation of a given algorithm, which, to the extent of our knowledge, was not made in the XAI literature before T. Miller's review paper [Miller(2019)] that is the distinction between explaining what the algorithm is and explaining why the algorithm is the way it is.

The first type of explanation aims at describing the functionalities of the algorithm and the understanding of its inner workings (at a given level of abstraction). It will answer questions such as "what does that algorithm do?", "why did it give such output on such inputs?" and "what would happen if we were to do that?".

The second type of explanation will provide answers to questions of the type "why are we dealing with algorithm A_0, rather than one among algorithms $A_1, ..., A_{n-1}$?" Such questions probes the relevance of algorithm A_0 rather than another algorithm with identical specification, or they might challenge the relevance of the specification itself, and wonder whether we could have formalized the task in a different way. In any circumstance, the questions bear on the design choices which lead us to use an algorithm with a given specification and a given set of properties for a given task.

This distinction actually cross-cuts the other distinction we have made previously between explanation of the specification and explanation of the algorithm itself. You can explain what the specification is and explain why it is the way it is, and the same would hold for the algorithm.

Sadly enough, natural language terminology does not always preserve such an important distinction, and why-questions might easily bear on both types of explanation. For instance, the question "why should the algorithm act this way on this input?" could easily belong to both levels. It is just a trap created by natural language that answers to those two distinct questions will both be called "explanations of the algorithm."

It might be tempting to call those two forms of explanation "description" and "justification" of the algorithm, but as we will see below that terminology would be misleading. There is already a well-established name for the second form of explanation, *contrastive explanation*. To the extent of our knowledge, there is no such established terminology for the first form: we shall call it *descriptive explanation*.

Contrastive explanation presuppose the knowledge of the specification and also sometimes of the inner workings of the algorithm, which are the topic of descriptive explanation. For us to discuss a contrastive explanation of an algorithm, it is necessary that alternative algorithms of similar specification exist, and that we are able to conceive of them and compare them along relevant dimensions. At least some descriptive issues on the functioning of the algorithm need to be solved before a contrastive explanation is given. I need to know what the factors are before I can wonder whether that list is relevant and complete for the task at hand; I need to know the rules and criteria of a decision algorithm before I can wonder if those rules and criteria are relevant and fair; I need to know how a given modification of a factor affects the outcome before I question the legitimacy of its presence. In other words, if explanation is fundamentally

about contrasting P with Q, I need to have an understanding of P before I contrast it with another option. If P is an algorithm, that means answering questions similar to those presented in the BBR approach. The descriptive conception of explanation is distinct, and logically precedent to contrastive conception. Those two definitions of explanation are thus not concurrent, but complementary[10].

Another remark is necessary to understand how we adopt and adapt Miller's approach to our present purposes. Miller's approach is very general and centered on causal explanations. However, we are interested in the explanation of decisions in a bureaucratic context. Typically, an explanation of a decision invokes reasons, not causes. Natural language does not always preserve the distinction between causes and reasons. One might ask to a physicist the reason why the sky is blue, even though the expected explanation will invoke causes, not reasons. One might also ask an historian what were the causes of WW1, even if part of the answer will invoke decisions made by historical actors which were based on some reasons. In some theoretical contexts, it might also be perfectly fine to neglect the distinction between cause and reason. However, there are some contexts where this distinction is crucial, and we believe that it is the case of explanations of algorithms used as an aid to decision-making. This can easily be seen by comparing contrastive explanations of algorithms with other contrastive explanations. Coming back to our previous example, the question "why is the sky blue?" calls for a contrastive explanation: the explainee wants to know why the sky is blue rather than any other color. The physicist will then answer by giving a physical cause which selected this particular wavelength for the light entering our atmosphere. The "why did you choose this algorithm?" also calls for a contrastive explanation: the explainee wants to know why this algorithm was chosen over other possible algorithms of similar specification. The explanation will then try to show why the choice of this particular algorithm was optimal considering the goal and the set of constraints at hand. This notion of optimality is typical of actions taken by an intentional agent. In the case of the explanation of the color of the sky, the cause for the blue color is not a reason why blue is the best color a sky should have. By contrast, the question of optimality will be at the core of contrastive explanations of the design and choice of algorithms, as it would be for any means chosen for a particular end. Therefore, it is crucial that we keep in mind the distinction between causes and reasons, because only reasons can be invoked in a discussion of optimality[11].

This is not only true for explanations of algorithms, but also for explanations of specifications. Explaining why a given algorithm was chosen can be a confusing task without a clear statement of the specification the algorithm was trying to satisfy. The definition of the specification might even be more controversial than the choice of the algorithm. Let us consider for example the design of an algorithm pre-screening applicants for a position. An answer to the question "what is the specification of this algorithm?" might be "to select the best applicants for the job for an interview." If such an answer is correct, it is vacuously so. The design of an algorithm will need a more operational answer, which implies a definition of the qualities sought in a candidate and a reflection on how they can be inferred from available data. All those decisions have a momentous impact on lives, and are bound to be a topic of heated legal and political debate. To quote an anonymous French civil servant present at an oral presentation of this work, "in public bureaucratic algorithms, specification is already politics."

In the case of the introduction of an algorithm as an aid to decision making, a particular form of contrastive explanation might also be important in political debates. It will contrast the algorithm in use not with other possible algorithms, but with man-made decisions. In its most radical form, this demand for contrastive explanation might even pit all decision-aiding algorithms against man-made decisions, and wonder why we choose to use an algorithmic aid in the first place. This contrastive explanation of decision-making automation is bound to be extremely important in debates over particularly sensitive decisions. Hostility against full automation of key decisions is very much present in the public, and contrastive debate over automation are bound to happen in the years to come, as AI applications continue to grow. We will see elements of that debate in the study of our medical algorithm (see £ref section). However, the XAI literature hugely takes automation as a given instead as an explanandum. Cases of comparison between human performances and ML models abound in the literature, such as the famous examples of character recognition or medical diagnosis based on image recognition £ref. To the extent of our knowledge, there is nevertheless no widespread debate on the methodological framework necessary to conduct meaningful comparisons between human and computer system performances on a given task. That is a pity, as it shows a huge gap between the work taking place in the XAI community and the actual political expectations it will have to face.

If we examine the case of BBR explanations, we quickly realize that they are explanations of what the algorithm is, not explanation of why the algorithm is the way it is. The algorithm itself is black-boxed, and hence fixed, as we look at structural features of the input-output relations. None of the questions asked on those input-output relations lead to a comparison of the properties of our black-boxed algorithm to other possible algorithms, or to question the specification of our task.

[10]Consequently, our distinction between those two acceptions of explanation is perfectly compatible with the theoretical stance taken by Miller and others that explanations are fundamentally contrastive.

[11]This distinction is implicitly made in Tim Miller's paper under the name of how-come questions and what-for questions.

To quote a famous leitmotiv of developers, this is not a bug, this is a feature. The BBR approach aims at abstracting away from the details of the algorithm in order to provide simple explanation of some key properties of the black-boxed algorithm, for a non-expert (potential) user or an expert user deprived a full access to the algorithm, e.g. for the sake of IP rights preservation. It is not in the business of comparing the algorithm to other possibilities, which would be likely to considerably increase the complexity of the task. However, the terminology might be misleading for the general public, and sometimes for experts themselves: some of the real-world demands for explanation will naturally be contrastive.

If the BBR approach only provides descriptive explanations, it does not mean that it will not have any relation with contrastive explanations. On the contrary, we have seen that descriptive explanations are presupposed by contrastive explanations. As a consequence, one of the constraints weighing on the BBR approach is the relevance of the information it provides on the algorithm for contrastive discussions. Among the informations we need for contrastive explanations, which ones can be obtained from a BBR approach and which ones cannot be obtained from that approach? That constraint weighs on any approach to descriptive explanation, but it is particularly relevant for an approach that deliberately abstracts away from the details of the algorithm.

1.6 Reasoning on a Fixed Algorithm, Reasoning on the Evolution of an Algorithm

The explanations of algorithms face another specific challenge that is particularly relevant for ML systems. Not only does the development of a ML model necessarily goes through a training phase, but this training phase can be repeated over and over, if the system is supposed to learn from a continuous feed of new data. That characteristic is far more than anecdotical, as ML systems are present on platforms in interaction with a humongous amount of users, who continuously feed them with throngs of unpredictable data. This might lead to many scenarios where the explanation of the training phase, and hence contrastive explanation of its results, will be of a great relevance.

This continuous learning on historical data leads also to an interesting feature of contrastive explanations for ML systems: they needs not be explanations of design decisions made by developers. Since the training phase is automated, many reasons why the algorithm is the way it is have nothing to do with conscious human decisions. This is the main reason why the contrastive explanation of an algorithm is very different from justifying the algorithm, since there is no warrant that automated learning will systematically end up on an optimal result. For instance, Google was sued in France under suspicion of Anti-Semitism after it was seen that its search request completion software would suggest "Jew" as a completion for the name of some public figures. £ref Google's answer was naturally to claim that its search engine continuously and automatically learns from the requests made by its users, and that the company was not responsible for the abundance of dubious requests made around a given name. An answer to the question "why was the request completion software making dubious suggestions?" cannot be given without a reference to the continuous training of the system, and hence to a process that is not controlled in its details by any given human group.

As a side remark, it would be interesting to investigate how this property, which is of course relevant for ML systems, might be also relevant for other forms of automated software modification. Playing dumb for the sake of the argument, we may say that the idea of automated software modification is as old as the Turing Machine: the data-software duality allows to treat code just as another form of data, which can be modified by the execution of another piece of code. From a more applied viewpoint, automated software updates, especially when they affect complex software systems, may have consequences unforeseen by any developer. In those cases too, explaining why the software ended up in its current condition might be very different from any justification of that condition. Comparing the differences between continuous learning on a ML-driven platform and more "vintage" forms of automated software modifications is an interesting topic in itself, but we will not dwell on this in this article.

Regarding the BBR approach, the absence of comparison of the algorithm with other possibilities might have important effects when it comes to the explanation of ML algorithms or systems. If the black-boxed algorithm is a ML system, the BBR approach does not allow for any reference whatsoever to the training phase of the algorithm. This might seem paradoxical, as many answers to real-world questions on ML systems involve an explanation of the properties of that training phase. If we come back to the Google Anti-Semitism case, an explanation of the completion suggestions could not have been made without a reference to the training of the search engine system, and the BBR is thus ineffectual for such cases.

Again, this is a not a bug, but a feature. The BBR approach can not answer questions contrasting the algorithm under scrutiny with other, possible algorithms, including previous versions of the same system. Its purpose is to describe the essential features of a given system, not to compare it to other possibilities, and it cannot be blamed for failing at a task it does not try to perform. However, our example shows that some pressing legal cases on ML systems might actually need an answer to those questions. Explaining what the algorithm is is not enough to answer all practical questions, and that's a limitation of the BBR approach to explanation.

1.7 Explanation and Justification: Agent-Relative vs Agent-Neutral

At this point, it seems necessary to add some more comments on the distinction between justification and explanation.

Justification could easily be defined as the contrastive explanation showing that a given decision was optimal or right [**?**], [**?**]. For instance, justifying the use of an algorithm is showing that it use is our best option for a given task. However, this definition of justification ignores an ongoing philosophical debate. To put a long story short, this debate hinges on the following issue: can we say that a decision is justified even if it is objectively wrong?

The main reason for that debate is that optimality (according to a well-defined metric) is an intrinsic property of a decision, while it is debatable whether the predicate "is justified" should also be conceived in this fashion or should actually be applied to the decision making process. For instance, a subject might be able to pick the best choice by pure luck: the decision will still be good, but it will not be justified. On the other hand, a decision might perfectly be grounded in solid reasons even if it turns out to be wrong. If the concrete constraints weighing on the decision are taken into consideration, we might reach the conclusion that a given decision-maker has done the best work she could possibly do considering the circumstances, even if the final result is wrong, or even catastrophic. The decision-maker might have taken her decisions under strict time constraints, she might have been given unreliable information she could not cross-check, and she might not have the best decision-aiding tools at her disposal. In this sense, a justification might play the role of an excuse for an objectively wrong decision. This is of course extremely important for the assessment of legal, bureaucratic or political decisions, which are often extremely sensitive and taken in suboptimal conditions, or even conditions of duress. Without considering the possibility that a decision might be reasonable but false, any decision-maker in the wrong would be blamed without any consideration for the circumstances in which the decision has been taken[12]

This debate is known in the philosophical literature as the debate between agent-relative and agent-neutral conceptions of justification. A similar distinction can also be known as the difference between objectivist and perspectivist accounts of justification. Leaving many subtleties aside, the core question is whether a conception of justification should take into account the concrete conditions of decision-making, or if a justification should abstract away from those conditions to focus on the objective reasons to make a given decision, and be strictly distinguished from an excuse[13].

It is of course not the place to take position on such an important debate of moral philosophy. We just want actors in the XAI community to be aware of the existence of such a debate, and wish that the terminology used in the field does not suppress such an awareness. Defining justification as the contrastive explanation that a decision is good is an inherently objectivist definition of justification: no mention is made of the concrete conditions weighing on the decision-maker. In general, justification is a contrastive explanation of the reasons leading to a decision. The authors discussing justification in the XAI community should then specify whether they wish to discuss agent-relative justification or agent-neutral justification. It is to be expected that most of the XAI literature will be concerned with agent-neutral justification, as it will mostly deal with scientific reasons to prefer an algorithm over another. However, it is not impossible that some researchers might be interested in agent-relative justification, as the choice of a given aid to decision might be justified by local constraints weighing on decision-makers.

It is important to keep this in mind to understand contrastive explanations of algorithms. Not only contrastive explanations are not necessarily justifications, as we have seen above, but justifications can be conceived in two different fashions, agent-neutral or agent-relative, and we will see examples of the relevance of that distinction in our work (£ref section). This distinction is relevant for the discussion of all type of algorithms. In the case of ML models, it will apply not only to the learning algorithm per se, but also to the choice of metrics and hyperparameters, and to the construction of the training/test set, as they are all relevant design decisions in the development of the model.

2 Case studies of explanations of algorithms

The aim of those preliminary case studies is to help formulate working hypotheses before the investigation scales up. In order to achieve this goal, we needed to find a method to take a meaningful sample of the vast number of algorithms used in the French public sector. In a recent white paper on the political stakes of public sector algorithms £ref, two members of the Etalab task force, Simon Chignard and Soizic Penicaud, presented a typology of those algorithms. The first type of algorithms was called "computation". This term is meant to denotes the set of algorithms used to

[12] As a side note for computer scientists, such a distinction is also extremely important in social sciences. Seemingly irrational decisions taken by historical actors can become perfectly understandable once the concrete circumstances weighing on their decision-making are taken into account.

[13] For more on this, see Stanford Encyclopedia of Philosophy, "Reasons for Actions: Justification, Motivation, Explanation" "Reasons for Actions: Agent-Neutral vs Agent-Relative"

compute taxes, welfare, subsidies, salaries, retirement pensions, etc.. Those algorithms are probably the most ancient type of algorithms used in the public sector, and the first subjected to computerization. The second was matching algorithms. According to the same paper, this type of algorithms was subject to computerization from the 90s onward in France, as they allowed to match a given scarce resource with a given individual selected in a vast population. It was the object of particular scrutiny in France after the controversy surrounding the algorithms (*APB*, then *ParcourSup*) matching prospective students with a slot in a given curriculum. However, this type of algorithms is also used in cases which have not been the topic of any important public debate, but are nevertheless of great practical importance, such as the attribution of slots in public kindergartens, or the attributions of grafts to patients on a national waiting list for organ transplant. The third is ML, which has been introduced in the French administration in the early 2010s. This is an important innovation for public sector algorithms and not only for the classical properties of opacity and predictive power which are generic in the comment of ML systems. As S. Chignard and S. Penicaud puts it, public sector algorithms are supposed to implement laws and regulations, and ML systems have the ability to create their own rules and criteria from data. As those algorithms are used to guide public policies, they reverse the classical order between algorithms and rules guiding public action.

After agreeing on that explorative typology, we had to choose one algorithm per type to conduct this preliminary study. The choice was mainly based on ease of access to information and an intuitive assessment of the interest and representativity of the case. For the computation type, we choose the housing tax because it was already the object of an in-house study at Etalab, and because tax computation is of course a strategically crucial part of administrative work which is always bound to raise explainability issues. It might be argued that studying the French housing tax is of limited interest, since this tax has been suppressed by the current government in 2018. However, the housing tax was an interesting example because it was criticized for its opacity. For the matching algorithm type, we choose the ScoreCœur, an algorithm used in the attribution of heart grafts. We deliberately chose to avoid *ParcourSup*, as its extremely controversial story makes it a very unique case, impossible to study as representative of a given type. By contrast, the ScoreCœur algorithm has attracted virtually no public attention despite its dramatic importance. Finally, for our ML example, we chose to study CibNav, a recent ML system designed to attribute a risk score to commercial ships in order to target them for safety inspection. It was chosen because its designers showed great care for the explainability of their model, and spend considerable time interacting with their end users, the administrative agents of the *Direction des Affaires Maritimes* in charge of safety inspection. Furthermore, the fact that this algorithm is bound to launch a re-organization of the affected administrative service and is used for a high-stake safety-related task made its explainability issues even more relevant.

However, it is difficult to claim that our case selection is in any way optimal. The representativity of each case study is problematic. Actually, the mere idea of a representative case study in the vast and diversified world of bureaucratic algorithms is problematic: explanations might face many case-specific challenges that may be hard to generalize. Furthermore, the first typology chosen as a basis for case selection is also questionable: from an explainability perspective, there might be more differences between two ML algorithms than between a ML algorithm and a matching algorithm. Those differences might not only depend on the technical type of the algorithm as a computer scientist would see it, but also on the precise function this algorithm is supposed to play, the scale and nature of the public with which it interacts, and the ultimate target public of the explanation (decision-makers, civil servants depending on the algorithm, ordinary citizens, developers of new functionalities and designers of GUI). It is still up for debate whether explanation methodologies should be conceived according to a technical typology, or according to other dimensions. The only honest answer to all those issues is to insist on the preliminary character of this study, and the importance of keeping an open mind as the investigation goes up to scale.

The sources we used were publicly available documentation (Housing Tax, ScoreCœur), interviews with developers when possible (CibNav, ScoreCœur), public hearings at the Parliament Commission for the Assessment of Technological and Scientific Choices(*Office Parlementaire d'Evaluation des Choix Scientifiques et Techniques*, OPECST) (housing tax, ScoreCœur), feedback on UX (CibNav, housing tax), and, of course, discussions with Etalab team members (Soizic Penicaud, Simon Chignard, Bastien Guerry). The sources we have are thus not homogeneous for every case study: we could not, and would not, force a perfectly homogeneous protocol on every algorithm to favor a more opportunistic approach where information availability was key.

2.1 Computation: The Housing Tax

Computation is a huge part of public sector bureaucratic activity, be it for taxes, subsidies, fines or welfare. It was only natural to consider this type of algorithms as a subclass of interest, especially since its interest is not reduced to public sector.

The term "computation" as it is used here is not a rigorous CS notion. It denotes a loose class of algorithms which can be essentially thought of as decision trees whose nodes are conditionals and computation formulas. Decision trees are

fundamental in any discussion of bureaucratic decision explainability for at least two reasons. The first is that they are an extremely common form of bureaucratic decision process. The second is that in the XAI literature they are often presented as the paradigm of an explainable algorithm.

The housing tax was also particularly interesting because it was infamous for its opacity. In the OPECST hearing of November 16 2017 [Villani, Cédric and Longuet(2018)], member of parliament (*député*) Julien Auber said that he was never able to understand how his housing tax was computed, and to estimate what the impact of a change of main residence would be. He wondered whether the desire for transparency was even applicable to tax calculation. This shows the opacity of so-called "simple calculation rules" even for a citizen as well-informed as a representative. This is a prime example of the tension between the scientific and the pedagogical meanings of explainability. For a computer scientist, the tax computing algorithm might be deemed simple, transparent and easily explainable, as it is made of clearly-stated computation rules, not a huge matrix of weighted neurons. However, this scientific perception might lead to consider unproblematic algorithms that are utterly incomprehensible for the crushing majority of users. It is crucial to maintain a clear view of the sources of cognitive challenges when the explainability of a given algorithm is discussed.

As a side remark, it should be noted that in his answer to Julien Auber, the representative of the internal revenue service (*Direction Générale des Finances Publiques*, DGFiP), Lionel Ploquin, defended the possibility of increased simplification and pedagogy for the fiscal administration, and mentioned the existence of focus groups of users to work on those issues. Bruno Rousselet also mentioned the disappearance of complaint mail on tax computation since online simulators have been introduced. It should be noted that those complaints were of the order of one or two per year, usually traceable to retired mathematics professors. However, this lack of reclamation on a system deemed complex and opaque by its own managers should not necessarily be construed as a sign of understanding. This lack of demand for explanation could be a sign of a general public who gave up on understanding the computation of their taxes, if only they ever tried to understand them, or does not dare challenge the results provided by the administration. The disappearance of complaint mail after the introduction of simulators is indeed a good sign, but it is not easy to assess their general impact on the public's understanding of their taxes. However, such an investigation would be beyond the scope of this work.

There was another contingent reason to pick the housing tax, which was simply that our partners at Etalab conducted two participatory workshops on the explanation of the housing tax. The first was conducted with a group of colleagues of the same team. They were asked to bring their own housing tax sheet, which was anonymized and studied publicly. The second was composed of *médiateurs numériques* ("digital mediators", in charge of facilitating the use of digital tools), who were not asked to bring their own tax sheets.

2.1.1 Presentation of the Algorithm: the Importance of Contrastive Explanation of Input List for Pedagogical Explanation

During those participatory workshops, the algorithm was compared to a recipe whose ingredients (inputs) had to be clearly identified. The participants were asked the following question at the beginning of the workshop: what do you think the ingredients of the housing tax are? Participants were asked at the end of the workshop if the housing tax computation was clearer to them, and what they had learned. Their knowledge was not systematically tested. The discussion of the inputs went beyond a simple descriptive explanation, since the absence of some inputs was also discussed. For instance, it was noticed that the abundance of public transportation around a given location was not taken into account in the computation of the housing tax, even if such public transportation system is usually funded by local authorities which are funded by the housing tax. This shows that contrastive explanation is relevant not only for the algorithm itself but also for the list of its inputs. As we have seen above with Julien Aubé's comments, the exact impact of a change of residence on the computation remained obscure to many, which could lead to question the fairness and relevance of this computation.

Contrastive explanation went a step further where participants were asked if they knew the relation between their housing tax and the funding of various local public services. This shows the subtlety that comes with the contrastive explanation of a tax system. The genuine product of a tax system is not simply an amount of collected money, but the funding of public services. Such an explanation considerably expands the scope of the discussion, since it is no longer only about the computation of a given tax, but the subsequent use of that money by tax-collecting authorities, which raise complex institutional and political issues. However, the design of a tax system is ultimately about this funding of public action by an appropriate set of tax payers. A contrastive explanation of a tax computation algorithm necessarily brings to the fore the complex issue of "who should pay how much for what" without which the design of the algorithm would be purposeless. This goes to show the huge difference between descriptive and contrastive explanations. For the explanation of tax computation, contrastive explanation of the list of inputs, which is part of the contrastive explanation of specification, is bound to be a main point of interest, possibly even more than the descriptive explanation of the

algorithm. This illustrates once again the contrast between the real-world challenges of pedagogical explainability and the in-vitro challenges of (scientific) explainability.

The computation of the housing tax is based on one primary input, the estimate of the locative value of the taxpayer's main residence and its dependences (parking, garage...) (*valeur locative cadastrale*). This applies indifferently whether you own, rent or occupy for free this main residence, and whether you occupy this residence for the whole year or only part of it, as long as it is your residence on January 1st. It can also apply to a secondary residence. This estimate is made by the government, which introduces an explainability issue from the get-go: the tax computation was based on an input which was not provided by tax payers, and whose origin was completely hidden from them. What is more, the estimate could take extremely puzzling forms, as some locative values seem grossly under- or overestimated compared to current market value, and two similar properties can have widely different estimates. This raises an issue of input opacity: when an input is not provided by the user, its definition as well as its mode of generation might remain obscure, and the value it takes can be puzzling.

The computation then typically proceeds as follows. Given the estimate of the locative value of the property, the tax payer can be entitled to several tax deductions, depending mainly on income, family situation, health and place of residence. For instance, widowers, individuals over 60 and disabled individuals can enjoy a deduction or full exoneration according to their income level. There also some special deductions for bed and breakfast residences located in certain rural areas which are at risk of depopulation (*Zone de Revitalisation Rurale*), or special tax allowance for overseas taxpayers, or individuals who have been forced to move to a new residence of higher value after a mandatory demolition of their previous residence. Finally, taxpayers with dependent family members enjoy a tax allowance, of 10% for the first two dependent individual, and 15% for each supplementary individual. The resulting amount is finally multiplied by a tax rate decided by local authorities, but whose variation is bounded by limits fixed by the State. The local authorities can also decide an increase of housing tax for secondary residences.

All in all, this first presentation of the algorithm shows the relevance of the contrastive explanation of input list for pedagogical explanation. If contrastive explanation is not the topic of BBR, the presentation of the input list is. The workshops, which were led by individuals ignorant of the existence of the BBR approach, have shown the relevance of this BBR issue to their topic. This leads to the question of the relevance of this approach for the explanation of computation algorithms.

The relevance of the BBR approach seem to depend on the complexity of the calculations performed by the algorithm. Again, those might seem explainable and simple for the average computer scientist, but it is best to remember the French publishers' motto which says that every equation you print divides your readership by half. A fair amount of individuals suffer from acute mathophobia, and the simple, direct presentation of a tax sheet might be extremely alienating for them. It seems to us that a BBR approach is still relevant for an algorithm such as the computation of the housing tax. It is already extremely important to give the users with a clear list of the inputs they are supposed to provide and, when possible as it is the case here, a clear view of the hierarchy of importance of said inputs. The BBR approach is thus not only relevant for ML models which are the poster child of opacity: it is also relevant for the most common and seemingly interpretable algorithms.

2.1.2 Model-Agnostic Lesson: The Distinction Between Variable Selection and Value Setting

However, the examination of our case study also shows an important limitation of the BBR approach, at least in its current phrasing. An important problem of computation explanation is the contrastive discussion of pre-determined values, be they tax rates, income thresholds, tax allowance rates and the like. Let us call this problem "value setting" to clearly distinguish it from "variable selection". It is one thing to agree on the principle that a given input should be taken into account for a tax allowance (variable selection), it is another issue to determine the exact value of that tax allowance (value setting). The pre-determined values are conceptually different from inputs, as they do not vary according to the situation of the taxpayers: they are parameters of the algorithm itself. The determination of the exact value of those parameters is of course extremely important for the final outcome, and their contrastive explanation is an important political topic, even if it might be deemed too technical for political debates. Why is the tax allowance for overseas taxpayers 50% and not 60%? How are the various income threshold for tax allowances computed? No trace of an explanation can be found in the public documentation available online, and we have found a similar absence in the documentation of the heart graft attribution algorithm. Despite its great quality, this last documentation does not provide any explanation for the different pre-computed values of parameters which show up in the body of the algorithm. Worse still, it does not even mention the issue, as if it was absent, or so desperate it should not be mentioned. In this perspective, the public documentation is merely descriptive and not contrastive. Again, we shall not criticize the BBR approach for failing at a goal it does not try to achieve, and contrastive issues are not part of the BBR scope. However, it should be noted that this distinction between variable selection and value setting has not been made in the first presentations of BBR, and we believe it should be as those two problems are very different in nature. The list of set

values should be mentioned in the description of an algorithm and distinguished from the list of variables, even if they might be confounded under the name "list of inputs".

Furthermore, it should be noted that value setting creates a definitive feeling of arbitrariness, which is a topic on its own for explanation. This feeling might be created by extraneous factors. Let us take an example coming from our medical algorithm, as we believe the issue to be model-agnostic. The medical algorithm gives a priority to young patients who are defined as patients under the age of 18. This particular value was not decided by doctors: it is the age provided by the legal definition of minority in French law. Even though this age does not have a particular medical meaning, minors are a legally protected category, and doctors do not get to change this legal state of the art for the purpose of algorithmic design. Value setting might feel arbitrary here because it is based on considerations extraneous to the purpose of the algorithm.

However, the medical algorithm also has several parameters whose values was pre-determined for purely intrinsic, medical reasons. Those are parameters such as coupling constants between two factors which were put in the algorithm for the explicit purpose of statistical fine-tuning. In this case, the values of those parameters are not arbitrary to the doctor, who knows that they have been computed to help maximize the overall performance of the algorithm. However, for the lay user, i.e. the patient, those values fall from the sky, and the mere reason of their presence is not obvious. However, explaining those parameters might be a great explanatory challenge for pedagogical explanation, as statistical fine-tuning does not seem to have obvious equivalent in everyday reasoning. The use of statistical fine-tuning as a particular form of value setting thus create an explanatory wall, which will be tough to climb.

That wall might not only be relevant for pedagogical explanation, but also for scientific explanation. As a simple conjecture, we wish to notice that the standard example of an opaque AI method, i.e. Deep Learning (DL), is entirely based on statistical fine-tuning of parameters. In this method, everything, whether it is called a parameter or an hyperparameter in ML parlance, is a parameter fine-tuned to optimize a given metrics. In the general case, deep neural networks do not even have a list of factors understood as a list of variables relevant for the task at hand, and extracting them a posteriori is not an obvious task. The opacity of DL might come from its foundation on statistical fine-tuning. However, we shall not explore this conjecture further in this work, as DL is not present in our case studies.

Another conjecture that might be worth further investigation is an intrinsic feeling of arbitrariness created by value setting which is due neither to the pressure of extraneous factors or to technical sophistication, but from the act of value setting itself when it is applied to natural language concepts. The determination of an exact age limit for the concept of "young individual" might always feel arbitrary, because the common use of such a concept is not based on any such rigid numeric limit. As became apparent during our interview of Dr Jacquelinet, the MD in charge of the design of our heart graft attribution algorithm, there is an intellectual tradition in computer science for medicine dedicated to the mitigation of such arbitrariness, and the unwanted threshold effects it might create, e.g. by smoothing the behavior of the algorithm around the threshold value £ref. In this case, the feeling of arbitrariness created by value setting, and the demand for explanations it may prompt, might be a by-product of the translation of fuzzy natural language concepts into terms a computer might understand. In this case, the uneasiness created by value setting should not be explained away by a justification of a given value as the scientific definition of youth, but admitted and mitigated through appropriate counter-measures.

The problem of value setting is a model-agnostic problem, which should be clearly identified as an essential subproblem of explanation of the list of inputs, well-distinguished from the subproblem of variable selection. Its importance might have been underestimated not only in the BBR approach, but in the general XAI literature as well as in public documentation: the feeling of arbitrariness created by some value settings, and the difficulties tied with statistical fine-tuning, are essential pedagogical problems for many algorithms, in dire need of more attention. Inspiration might be found in intellectual traditions predating XAI, and possibly outside of computer science, as for instance threshold effects have been a topic of interest for legal scholars £ref: the identification of model-agnostic issues of explanation must be accompanied by a greater attention to various intellectual traditions.

2.2 Matching Algorithms: The Attribution of Heart Grafts by ScoreCœur

Matching algorithms are by nature a sensitive issue, as they match a scarce resource with individuals in demand. The design of an algorithm attributing heart grafts is a perfect illustration of that fact, as it manages a life-saving resource. The mere choice of automating such a dramatic decision should not be taken for obvious. However, the design and implementation of this algorithm has not been the topic of any heated public controversy or scandal, and its design cannot be seen as a political reaction to such public controversies. The situation should be contrasted with that of Germany, where German doctors who shared a common system with other countries were accused of tricking the system, especially through biological data fraud, to favor German patients £ref. However, France had an algorithm to establish a list of heart transplant receivers since 1955 (FranceTransplant). The algorithm has evolved several times

before reaching its current condition. The design of the current algorithm is partly a reaction to several perceveived defects of pre-existing algorithms, which we will explain as we present the main features of this algorithm. This presentation is hugely indebted to an interview with Dr Christian Jacquelinet, MD and computer scientist, scientific advisor to the president of the *Agence de la Biomédecine* and MD in charge of the Scorecœur algorithm.

The attribution of a graft is first based on an estimate of the compatibility of giver and receiver. This compatibility is based on simple factors such as blood type and size, as it is not desirable to transplant the heart of a light-weighted woman on a well-built man. But very complex factors such as Human Leukocyte Antigen (HLA) immuno-histocompatibility are also at play. This factor can imply to consider probabilies of the order of one in a million. This increase in granularity complexifies the decision and was a justification for the use of an aid to decision [Villani, Cédric and Longuet(2018)].

The second factor is ischemia. Once it leaves the giver's body, the graft does no longer receive blood or oxygen: its quality degrades extremely rapidly even in the best conservation conditions. The transportation time is thus a decisive factor. Consequently, the model of transplant attribution is gravitational: a patient and a transplant will attract each other on a longer distance if the compatibility between the two is higher (see [Villani, Cédric and Longuet(2018)] and public documentation). This model is a reaction to a preceding algorithm where a patient had to go through several scales (local, regional, and then national) in order to find a compatible donor. The number of compatible donors at a local scale was simply too low, and this was replaced by the current gravitational model, which gives a built-in ponderation between distance and compatibility.

Ischemia has also been used as a justification for the use of an aid to decision, as it imposes short decision time in tough conditions for human doctors: decisions to accept a transplant usually had to be taken at night , with an obligation to answer in less than 20 minutes.

The third factor are priority rules. This third factor is different from the first two, as it involves a moral decision to prioritize some patients over others, which is not a medical decision optimizing the chances of a successful transplant, and cannot be purely evidence-based. As a consequence, those factors can vary considerably between countries for cultural and political reasons. For instance, the French system first gave priority to minors (age below 18). It was then realized that young adults (between 18 and 25) were not particularly at their advantage compared to older patients in this system, and the decision was made to give them priority over senior patients. Such a prioritization does not exist in the USA where associations of retirees strongly opposed it when it was proposed [Villani, Cédric and Longuet(2018)]. Even fairly natural rules of prioritization such as vital emergencies might be a topic of moral discussion: a systematic prioritization of vital emergencies might favor desperate patients with a very small chance of long term survival even with a transplant, while potential receivers with a better shot at survival might be neglected: we will discuss this issue further below.

On top of the three basic factors of donor-receiver compatibility, ischemia and moral prioritization, the algorithmic design is deeply shaped by a quest for transparency, equity and the elimination of side effects of previous algorithms. In a precedent form, the algorithm would attribute the transplant to a medical team (*centre de greffe* [Villani, Cédric and Longuet(2018)], not to a patient. According to Dr Jacquelinet, this led to behind-the-curtain negotiations between different teams, where a given team would agree to refrain for deposing a demand at some time to warrant a successful application for another team, in exchange of reciprocity at later times. This was of course a factor of opacity, and created a considerable risk that the survival of a given patient would be more dependent on the power struggle between medical teams and the negotiating abilities of a given team member rather than medical and ethical reasons.

The third perceived defect, as we already mentioned above, was that the emergency algorithm would give priority to patients in a critical condition. If such a priority seems natural at first sight, it also increases the probability to allocate a transplant to a patient in an overall desperate condition, whose life expectancy would only be increased by a short amount. Furthermore, the systematization of emergency algorithms led to an higher mortality rates among less critical patients than among patients in critical conditions[14]. This paradoxical result was also found in the case of liver transplants [Villani, Cédric and Longuet(2018)]. By contrast, the current system eliminates the emergency algorithm altogether, and fully assumes the exclusion from the list of receivers of patients whose life expectancy would remain very low even with a transplantation (less than 50 % chances of survival within a year of the transplant for adult receivers).

The current system also made the decisive step towards practical full automation of the decision, even if the possibility of a human decision is still open [Villani, Cédric and Longuet(2018)]. According to Dr Jacquelinet, around 20% of the current cases are examined by an expert committee, especially when the data fed to the algorithm is considered

[14]It was also argued that the criteria governing the entry of the emergency algorithm did not accurately describe the severity of a given patient's condition: it was estimated that a fourth of super-emergency patients were at low risk and that a third of high-risk patients were false positives. (see public documentation)

too coarse-grained to apply to their case. The role of medical teams in the remaining ordinary decisions is reduced to inputing and updating correct measurements of critical medical values following strict guidelines, and the list is then generated in a fully automated fashion. Mixing the testimonies of the designers and normative reflection of our own, we can find at least six reasons for this step towards automation. The first is to warrant full transparency on the decision criteria, and to avoid any dependence of the patient's survival on the power and tactical skills of their medical team. The automated application of those well-defined rules then yields a second desirable property, which is the consistency of decision-making. Tranparency and consistency in decision-making are of course desirable in the quest for equity between patients. The third is to take into consideration the fragility of human decision-making abilities under conditions of duress. This is a prime example of an agent-relative justification of full automation: automation is better relative to a human agent because taking a life-and-death decision in less than 20 minutes, possibly in the middle of the night, is hard for that human agent. The fourth is the sheer complexity of the decision even when it is taken in the best conditions: as we have seen above, the success of the transplant depends on complex medical factors and extremely small probabilities. The fifth reason is that a system leaving discretionary powers to human decision-makers turn those decision-makers into a target for political pressures: desperate patients or their friends and families might be tempted to lean on decision-makers. From this perspective, the full automation of the decision is also a safety feature. The sixth and final reason takes into account the humanity of decision-makers in a different fashion. The magnitude of graft attribution decisions is psychologically crushing: denying access to a transplant to a patient is as close to a death sentence as can be in a legal system which does not know the death penalty. This would be extremely hard on the decision-makers' psyches, especially for medical teams who personally know, sometimes over a long period of time, the cardiac patients and their family[15].

Let us open a short parenthesis which could be relevant for contrastive explanation of algorithmic automation. In our case study, automation plays a classical political role often commented upon in the critical analysis of bureaucracy, which is to alleviate human actors from personal responsibility over a given decision. This alleviation of personal responsibility has often been presented as a negative feature of procedural decision-making. It is certainly easier from a political perspective to implement a terrible decision if no given individual is to feel responsibility of the final outcome. One just needs to remember Hannah Arendt's classical analysis of the banality of evil in *Eichmann in Jerusalem*, where Eichmann is presented as a soulless bureaucrat, who could participate in horrible deeds without feeling any personal responsibility. Our particular example shows on the contrary that the depersonalization of decision-making through proceduralization and ultimately, automation, can be fully assumed for ethical and humanitarian reasons. Automation warrants the implementation of a safe, consistent, transparent and humanly bearable decision system that can be applied to complex decisions under tight time constraints. For this reason, depersonalization and deshumanization should be strictly distinguished. Depersonalization might lead to deshumanization, but that depends on the particular nature of the algorithm, its aims as well as the overall political context, not on the nature of depersonalization in itself.

It should be duly stressed that the computation of the score is not the only algorithmic aid to decision. Doctors also use simulations based on historical data to estimate the impact of each new design decisions, allowing to prevent unexpected harmful consequences and formulate predictions which inform final decisions. Pr Olivier Bastien, the adjunct director of the *Agence de la biomédecine* in 2017, declared in front of the OPECST that the agency was favorable to the use of a ML model, as it would be more adapted to the constant flow of new data than the current painstaking manual update, as some scoring systems for graft attributions have had 26 different versions as of 2017 [Villani, Cédric and Longuet(2018)]. However, such a choice would raise considerable explainability challenges. What would happen if ML was used to improve on cardiac graft attribution? Two issues should be distinguished here. The first depends on the use of ML technics to improve on the scientific survival model. As we have seen above, the legitimacy of the Scoreœur algorithm depends highly on the underlying survival model. In the current condition, this model is a black box for the vast majority of the population anyway, since they do not have the expert knowledge in cardiological science. Switching to an opaque ML system would be mainly a problem for medical scientists trying to understand and improve this model. Another scenario would be the complete substitution of the entire algorithm by an ML system. This might increase survival performances at the expense of explainability. Explainability warrants the safe, transparent and consistent nature of the algorithm: abandoning it might have a considerable political cost, and in the worst-case scenario it might completely

[15] In our interview with Clément Hénin, a CS doctoral student who spent a lot of time discussing with cardiologists interacting with the algorithm, there was barely any mention of that psychological load. Cardiologists would on the contrary insist on the extreme joy produced by a successful transplant, with its spectacular and life-saving effects: the doctors seem to be thinking more about the patients they are saving than the patients they keep on the waiting list. However, as anyone familiar with interviews knows all too well, lack of mention of a given item in a conversation is not proof of its absence in real life. More importantly though, this potential psychological load is an objective feature of the decision-making process, and it is worth being discussed as such, even if it does not show up in a particular population. Dr Jacquelinet's interview confirmed that hypothesis: not only do such decisions take their toll, but everyday interactions with patients might lead doctors to favor patients according to the relation they have with them, which would in turn put patients in an unseemly competition for the doctor's attention. This brings together our safety issue and our psychological issue.

ruin the public trust in the system. Furthermore, opaque ML systems are prone to mysterious bugs, which may be hard to diagnose and correct. Would we accept a couple of aberrant decisions in the name of overall performances? However, as Dr Jacquelinet mentioned in our interview, doctors are supposed to be passionate advocates of their patients' survival and well-being, not lawyers fearing legal liabilities. An improvement of the overall survival rate might be welcomed with great enthusiasm in this perspective, and not feared as a potential source of legal and political risks. The exact importance of explainability might not only be highly context-dependent, but value-dependent, and depend on a sharp definition of the fundamental values pursued by a task.

Another key medical issue underlying the algorithm is the robustness of the underlying model of survival. As we have seen, this model practically automates away the decision to exclude a patient from the transplant waiting list. For the contrastive explanation of such a model, it is important to know whether the model represents a worldwide scientific consensus, or if it is only representative of the scientific options of the French medical authorities. In the absence of a scientific consensus, a given patient could discover that she would not have been excluded from the list, or would have had a more favorable overall score, if she had been a citizen of another country. It is also crucial to examine the evolution of the model in time. As we have just seen, the model is regularly updated thanks to evolution of medical knowledge and inclusion of new data. This could also imply that a deceased patient's relatives might discover that she would not have excluded from the list, and she would have had a more favorable score, if she had applied in later years. This tragic situation is of course a regular part of medical history. Patients regularly die from diseases that are easily cured in the following decades, or are victims of medical mistakes that are later thoroughly eliminated. However, the legitimacy of a given complaint might depend on the following tough question: could the medical authorities have had a better model at the time of the decision? This question is often treated in philosophy of science as the contingency of scientific and technological evolution. This theoretical question could become a crucial argument in possible contestations or even lawsuits relating to outputs of the survival model. Again, all those questions can only be treated from a contrastive viewpoint, and cannot be tackled by descriptive approaches such as BBR.

Now that we have given a coarse-grained view of the factors taken into consideration and of the contrastive explanation of full automation, let us look at the score computing algorithm in more details. The current system is based on a National Score for the Attribution of Cardiac Grafts (*Score National d'Attribution des Greffons Cardiaques* or SNAGC). There are four main steps in the computation of such a score:

1. Computation of an Index of Cardiac Risk (*Index de Risque Cardiaque* or ICAR). It is based on a study made on patients between 2010 and 2013. Its role is to compute the risk of death in the waiting list for any given adult patient. The four factors taken into consideration are highly technical: existence of a short-term mechanical breathing assistance, natriuretique peptides values (neuro-hormone synthetized by cardyomyocit cells in response to a rise in blood pressure, particularly in the case of cardiac defiency, which is considered an indicator of cardiac deficiency), glomerular filtration rate (describes the flow rate of filtered fluid through the kidney, and is an indicator of kidney condition), total bilirubine value. The bilirubine is a product of hemoglobin degradation, and also other hemoproteins. If the degradation of hemoglobin cells is a normal process, an abnormally high rate of degradation is an indicator of several conditions.
2. Computation of the Brute Cardiac Composite Score (*Score Cardiaque Composite Brut*, or *Score CCB*). It divides the population into four components: standard adult, standard pediatric, expert pediatric and expert adult. Pediatric patients are given a number of extra points which increases with the duration of their time on the waiting list. The expert components of the population are given extra-points because more advanced medical exams show the emergency of their situation, which could aggravate into death or list exclusion. The decisions are based on medical factors, some of them highly technical, including haemorrhage, infection and various issues with circulatory assistance.
3. Computation of the Weighted Composite Score (*Score Cardiaque Composite Pondéré* or *Score CCP*). This computation takes the CCB score as an input and applies a set of filters and matching giver-receiver functions. This computation takes into account the age difference between giver and receiver: a difference inferior or equal to 15 years warrants that patients keeps 100% of their points. A difference between 15 and 40 decreases progressively the number of points, and a difference strictly superior to 40 cancel all points, effectively excluding any donor-receiver match. The computation also takes into account the compatibility of blood types, and a matching between morphological properties (body mass, corporeal surface). There is also a filter dedicated to transplant failure, defined as a survival rate inferior to 50% in the year after the transplant. It is only applied to adult receivers. This exclusion is not necessarily definitive: an improvement of the patient's condition can lead to a re-inclusion. This criterium is intended to avoid matching risky donors to risky receivers, and identify receivers who would not have good chances of survival no matter what the graft is.
4. Final computation. This takes the CCP score as an input and computes the final score taking into account the time of transportation between the place of collection and the place of transplantation surgery. This

transportation is made by road if the distance is inferior to 100km, and by plane and road if it is superior to that threshold.

2.2.1 Model-Agnostic Lesson for Pedagogical and Scientific Explanation: Synthetic Overview of Factors

Now that we have presented the main features of this algorithm and the motivations behind its conception, let us examine in more details some of its explanatory issues.

There are around 40 variables in this algorithm. According to a source who wishes to remain anonymous, surgeons interacting on a regular basis with the algorithm typically remember between 10 to 20 variables, the exact list depending on the surgeons and their particular experience.

The number 40 would be considered low-dimensional from a ML point of view, but it is already too large for human memory, including for professionals famous for their memorization abilities such as MDs. However, in order to have a discussion of an algorithm (specification and implementation), it is necessary to have a synthetic overview of the factors it takes in. Physicians are able to do this through a coarse-graining of the factor list. This is the type of coarse-graining we have used in our presentation when we have written that the algorithm takes the typical indicators of heart health, donor-receiver compatibility and ischemia as factors. The concepts "indicators of heart health" and "donor-receiver compatibility indicators" serve as abbreviations of a list of factors that might potentially be long and hard to remember on the spot. Even if a medical doctor might be unable to remember all the typical factors of ischemia on the spot, the concept acts as a pointer to relevant resources where the list will easily be found. Proper terminology, as it creates a supplementary abstraction layer, is thus a typical solution to the challenge of factor list length.

This extra-layer of abstraction does not only diminish the memorization load: it also facilitates reasoning. Sentences including an enumeration of 40 factors would be hard to manipulate. The introduction of a concept acts as a meta-variable facilitating the manipulation of ground variables, a common feature both in natural language terminology and formal notation.

However, the presence of the relevant concepts cannot be taken for granted: for many algorithms taking a large number of inputs, there might not be a ready-made terminology allowing a synthetic overview of the long list of factors. A definite description of a subgroup of variables might play the same role, but it requires careful phrasing. All in all, it is very relevant to explanation to wonder whether it is possible to group all variables used by an algorithm in a handful of classes which can easily be described to the explainee. ML models are particularly prone to this issue, as they are often high-dimensional, and try to examine new list of factors, which might not have been considered by domain expert terminology.

The challenge of producing a synthetic overview of factors could be seen as a purely pedagogical challenge, as it takes into consideration problems induced by human cognitive limitations. However, it is worth being remembered that scientists are not too fond of list of 300 factors too: simplicity is also a scientific value. In this perspective, producing a synthetic overview of factor list through coarse-graining is also a challenge for scientific explanation. The importance of a synthetic overview of factors is thus a model-agnostic lesson which applies to both pedagogical and scientific explanations. It is interesting to notice that this reflection on the importance of factor list length came to us by chance, during the course of conversations with a source, but comes to conclusions v similar to recent articles in XAI (see for instance [Sharma et al.(2019)Sharma, Henderson, and Ghosh]), which underline the importance for explainability of the extraction of robust, higher-order features. There is here a convergence of methodologies which demonstrate that such research are close to the concerns emerging in practice.

2.2.2 Model-agnostic Lesson for Pedagogical Explanation: the Explanation of Domain Expert Terminology and Domain Expert Knowledge

However, the troubles do not stop there for pedagogical explanation. Even if we have the chance to get a synthetic overview of factors from a well-defined terminology, we still face the challenge of explaining that terminology to the lay public if it is based on domain expert knowledge. Pedagogical explanation needs to face the issues raised by the use of domain expert knowledge in an algorithm.

As can seen in the public documentation, the code for our medical algorithm is in itself extremely simple, as it is essentially a list of if-then-elses with conditions depending on the value of a medical variable, and a couple filter functions. Our medical example is thus a perfect example of an interpretable algorithm with considerable explainability issues. The main obstacle to understanding the code is not properly algorithmic. It is the high technicality of most variables and constants, which are utterly incomprehensible for someone who is not trained in medicine in general, and in cardiology in particular. This has momentous consequences for explanatory efforts, especially in the BBR approach. Without access to the meaning of inputs, there is very little room for explanation, both contrastive and descriptive. From

a contrastive standpoint, it is of course impossible for the general public, and even sometimes for non-cardiologist doctors, to decide whether the list of technical inputs is legitimate, both in what it excludes and what it includes, or whether the hierarchy between technical inputs is sound. It is also hard to understand and discuss the logic of the processing when the determination of threshold values for the conditions is left completely in the dark for lack of knowledge of what we are talking about. From a descriptive viewpoint, this lack of understanding of input variables is also extremely problematic.

Let us illustrate this terminological issue with an example, which will again come from an anecdote of our research. The first time we met this issue was during our first reading of the public documentation, when we tried to understand what "Glomerular Filtration Rate" (GFR) meant. To cut a long story short, a quick search would explain that the GFR is an indicator of kidney health. The next obvious, contrastive issue that came to our mind was of course "what is an indicator of kidney health doing among the inputs of an algorithm for heart transplants?" The answer to this tougher question was given to us by Clément Hénin, a Computer Science doctoral student who is currently working on this medical algorithm. Kidney health is relevant to heart transplant for two reasons:

- As a direct indicator of the global condition and its emergency. Since kidney failure is fatal, it is of course crucial for MDs to monitor kidney health as an indicator of the patient's overall emergency.

- As an indirect indicator of heart health. The malfunction of the heart impacts heavily many other parts of the organism, and especially the kidneys: where the heart go, the kidneys go. Monitoring the kidneys is thus a good indirect way to monitor cardiac condition.

The relation of the BBR approach to this terminological problem is worth being noticed. All the basic objectives of the BBR approach could be formally achieved in the case of our medical algorithm. We could give the complete list of inputs and their hierarchy, present the impact of a given variable, check for decision inconsistency and perform counterfactual reasoning without ever explaining the meaning of technical variables. This is particularly simple here as the algorithm itself is extremely simple: there is no need to be a top-knotch programmer to understand statements of the form "if variable x has a value below threshold t, then you are given y points, else you are given z points." However, this is only an Ersatz of a genuine explanation, as most members of the public will feel utterly lost if they cannot understand the meaning of inputs. The situation might look brighter for the BBR approach when we consider the parts of the algorithm expressing moral priorities, as the variables might be much simpler in those places. However, the interest of black-boxing the algorithm is here very low, as the black-box explanation would be very close to the implementation of the algorithm in pseudo-code. There is thus a double hard lesson to be learned from our example for the BBR. The first is that there is little point to be able to apply BBR explanations if the meanings of input variables cannot be explained. The second is that black-boxing the explanation has very little interest where the algorithm is simple and close to natural language expressions. All in all, the BBR approach seems irrelevant to an algorithm with a low algorithmic sophistication, high terminological complexity due to expert domain knowledge, and dominance of constrastive over descriptive issues. This illustrates the possibility we mentioned in the introduction of this section that the types of explanatory issues raised by an algorithm might not depend on its technical classification by a computer scientist, but on cross-cutting concerns.

Leaving BBR aside, the issue of terminological opacity seems to have been neglected in the literature on ML interpretability in general. The difficult human interpretability of ML variables living in high dimensional dataspaces is of course often discussed, but problems with the semantics of inputs begin as soon as as they depend on expert knowledge. This is true of course of medical decisions, but it is also true of legal and financial vocabulary, which is involved in decisions affecting millions of citizens. The challenges of understanding a given algorithm might have little to do with the mastering of sophisticated computer science, and everything to do with the older challenge of understanding words. The algorithm might not be the problem when we say we don't understand an algorithm: it is rather that the algorithm is the place where we are confronted with our lack of control over some expert knowledge, a lack of control which can, and often will be manifested through our ignorance of the terminology. Our particular example is a case in point, as it is particularly low in algorithmic complexity but high in expert domain knowledge manifested in the terminology. As more and more of our activities are moved to the digital realm and subjected to algorithmic processing, this situation is bound to become more and more common. Algorithms become media of expert knowledge, and as such the issue of explanation of algorithms become entangled with the explanatory issues of domain expert knowledge. This was confirmed to us by an interview with Quentin Loridant, the data scientist in charge of the CibNav ML model (see section below). The CibNav model was introduced at the same time as an important reform of the regulations governing the actions of the public agents who will use CibNav. During his first presentations of the software to agents, Quentin Loridant could barely have a question on CibNav, as practically all questions were directed at the regulatory reform, a legal topic on which he naturally felt completely incompetent. However, CibNav had become the medium through which the regulatory reform was met by public agents, and as such its fate was entangled with legal expert knowledge.

From a philosophical point of view, the issues of terminological accessibility can almost be seen as off topic. Of course, one might say, one needs to understand the language and terminology used in the algorithm. That's a prerequisite of explanation, but explanatory work itself begins when the meaning of propositions is clear, and one tries to understand the functioning of the algorithm itself. However, this objection rests on a confusion. The explanation of domain expert terminology is not a linguistic issue of giving a new name to already known concepts, it is an expert issue of defining new concepts, which is already part of understanding. Furthermore, we would like to add that ignoring genuine issues of linguistic accessibility, like explaining a bureaucratic algorithm to a non-native speaker, might be legitimate in theory, but would be a dangerous abstraction to make in practice. Explainability is to deal with the real-world issue of explaining real algorithms to real people, and it should not get lost in abstraction. The first window in the GUI for an explainer should not be "Go read a book on financial law, and then I will explain to you the algorithm." Linguistic accessibility is bound to be a major problem for all attempts at developing tools for explainability, especially when it is noticed that lay users may frequently meet expert terminology coming from fields such as finance, law or medicine. It would be desirable that XAI people work hand in hand with NLP people in order to avoid an ill-founded over-specialization of subfields. The issue of terminological explanation is bound to be found in various types of algorithms, and as such it is model-agnostic lesson for pedagogical explanation.

2.3 ML Systems: Predictive Policies

For our ML case study, we are going to examine CibNav, a predictive ML system whose aim is to detect which commercial ships present a risk of accident. This system, which is in the early phase of deployment as of April 2020, is to be used by administrative agents as an aid to decision-making. The decision to target a given ship is not to be fully automated, but the agents are to use the risk score generated by the ML system to make their targeting decisions. The ultimate goal of the system is to shift from a systematic control model, where ships are subjected to a safety check-up on a regular basis, to a targeting model, where ships with a high risk score are targeted for control while ships with a low risk score are left alone or subjected to lighter control. Such an organizational shift puts considerable pressure on the development of this predictive system, as it will have considerable impact on the safety of commercial ships. The development of the system has already given ample opportunity to interact with the agents who will be the final users and who in general have virtually no knowledge of computer science. Explainability issues have thus been very present in the conception of CibNav.

2.3.1 Model-Specific Lesson: The Interaction Between Pedagogical and Scientific Explanation in ML Development

A naive vision of the explanation pipeline for a black box model might look approximately like:

- Development of Black Box Model.
- Explanation through XAI tools (LIME, SHAP, counterfactuals...).
- Pedagogical Explanation for lay users.

However, our case study at CibNav draws a radically different picture of the flow of explanatory information: there actually exists an interaction of pedagogical explanation and scientific explanation not only at the end of the development process, but during this same process.

According to our source Q. Loridant, pedagogical explanations provided to lay users, and the following feedback they enabled, played multiple roles during the development of the CibNav model:

- **Sanity-check for the developer.** The fear of falling prey to some spurious feature of your data is not null in the ML developer, and domain expert knowledge acts as a welcome sanity-check. This is the first reason why explainability was seen as a core scientific feature of his model from the beginning of development, not a legal compliance afterthought which would be taken into consideration after the model was fully trained.
- **Re-interpretation of results thanks to domain experts' causal intuitions.** In this case, lay users are also domain experts, with considerable experience on commercial ships. They were able to see some of the results of the model with a critical eye. For instance, the first model gave a great importance to ship length among possible risk factors: the longer the ship, the higher the risk of accident at sea. However, end users considered that the impact of this factor, even if it was reasonable to give it weight, was probably exaggerated by the model, as captains of small ships had a tendency to report far less incidents than captains of small ships. The dialog with end users thus prompted an investigation on a possible data bias that the data scientist could have never guessed a priori.

- **Parametrization of the model.** The first parametrization of the task produced by the developers was based on purely technical parameters, such as ship length, engine power, construction year, anomalies detected during the last check-ups and major anomalies detected during the last check-ups. However, conversations with ship inspectors showed that the organization of life aboard was also a key factor of accidents. They suggested that the turnover rate was introduced among predictive features, as a high turnover rate is usually a good indicator of tough conditions aboard the ship. The causal intuitions of end users thus lead to a re-conceptualization of the model.

Our description of the practice of development, and the role of explanation in it, drifts away from the image pushed by sometimes promotional discourses, which present ML models as all-mighty oracles replacing primitive seat-of-the pants feelings and traditions with the scientific power of data. Such an image is not only based on a caricatural view of human decision making in administrative contexts: it also gives the wrong picture of the development process. Developers do not necessarily consider their models as god-like oracles. They consider the possibility that their model might be ill-specified, ill-implemented and victim of a biased data set. Contrastive and descriptive explanations given to or by end users, both at the specification and implementation level, are thus a tool in the developer's box to re-think their model and re-interpret their results. This is nothing new for anyone familiar with development methodology. After all, agile methods were partly conceived with the ideas that intense and regular exchanges with the end user are essential to a sound development process. However, it should be noted that the opacity of ML models does not make this idea irrelevant for ML development. On the contrary, this opacity makes even more necessary to have a critical stance on key development decisions and performance assessments. This critical stance is supported by exchanges with end users and comparison with domain expert knowledge, which can only be enabled through an abundance of contrastive and descriptive explanations given to end users. The opacity of ML should not lead developers to give up on explanatory exchanges with their end users, but on the contrary it should sometimes lead to increased attention to this part of development.

2.3.2 The Relevance of BBR for ML models

The initial objectives of the software were to define a risk score for each ship, to be able to predict that risk score through regression, to offer a proper interpretation of that score and identify impacting factors.

After discussions with end users, Q. Loridant realized that the identification of the impact of a given parameter on a given risk score was a recurrent and crucial question. Users would also frequently ask counterfactual questions to evaluate the role of a factor, such as "how would the score change if the ship had a smaller length?" He decided to put this question at the heart of his efforts towards interpretability. This is of course consistent with BBR, even if the data scientist was ignorant of that approach at the time. This shows again the relevance of BBR questions to actual explanations to end users, as those questions seem to emerge spontaneously during conversations with them.

In order to better understand their original model, the CibNav designers conducted a comparative study, building other, supposedly more interpretable models on the same data. They actually started by building a decision tree with Scikit Learn. The results were consistent with the typical expectation for this type of model. The identification of important factors was easy, but the performance was weak, and the impact analysis and relativization of different parameters was difficult. At some point, as the decision tree grew to depth 49, it became difficult for developers themselves to understand how a given factor would impact a given score, making it very unlikely that they could explain this to end users. This relativizes greatly the expectations of explainability of so-called interpretable models. Decision trees are the paradigm of interpretable models in the literature, but they can become hard to interpret even for expert users. One should not expect systematic and easy explainability from the so-called interpretable models. This particular experience is consistent with remarks previously made in the XAI literature £ref Lipton.

After this disappointing experience with decision trees, the developers decided to try another family of models, GANS, (Generalized Additive Models). After some trials, They chose to use the Explainable Boosting Machine, or Boosted GA^2M. This is a fairly complex model, and the data scientist himself would admit that some parts of it are still obscure to him. However, this opacity of the model did not prevent an analysis of factor impact which was much easier than for the decision tree. Paradoxically, a white box model such as a decision tree does not necessarily lead to an easier answer to BBR questions, such as factor impact, compared to a black box model such as Boosted GA^2M. The variation of explainability according to the models is here opposite to the basic intuition that explainability issues should be easier to solve with white box, interpretable models: that's another model-agnostic lesson for pedagogical and scientific explainability.

Another experience reinforced the data scientist's interest in BBR, even though he was still unaware of the existence of this approach. Q. Loridant tried to generate by hand an approximation of his GAN model by a list of rules. This was meant to give a feeling of the inner workings of the model to lay users. The appallingly low rate of interaction with

this documentation might not only be due to the standard lay user's disdain of technical documentation. Q. Loridant made the conjecture that knowledge of the inner workings of the algorithm are actually not at the center of interest for lay users, and might only act as noise for the questions of true interest, such as BBR approach of factor impact. Black boxing the model might not always be an epistemic curse, forced upon us by the dazzling complexity of models, but a positive choice of abstraction level to facilitate the end users' understanding. In this perspective, black boxing is a positive pedagogical decision, which is another lesson for pedagogical explanation. However, this lesson is still tentative, as our ML case study, which cannot be mistaken for an exhaustive study of this issue, can not be completed as of April 2020. The CibNav model is still in the early phase of deployment, and feedback is slowly coming from end users. The relevance of BBR approach will only be fully demonstrated when that feedback is collected and analyzed.

Q. Loridant's approach was also interesting for its ambition. He wanted administrative agents to be able to contest the decision made by the model, following the motto "I understand you if I can prove you wrong." This ability to contest the decision through precise arguments was key to unable users to override the model's recommendations. Even if the agent might have this power on paper, it might be very hard for her to override the risk score out of fear of being wrong, and ending in the delicate position of justifying your failed opposition to a sophisticate scientific model. Loridant's working hypothesis is that explainability should allow agents to overcome that political fear through a new ability to contest the system with sound arguments, so that they could defend their position even if they turn out to be wrong. However, we still have to wait for post-deployment data to see if this ambitious philosophy of explainability is a practical success.

2.3.3 Model-Agnostic Lesson: The Importance of Post Deployment Decisions

Our conversation with CibNav developers also lead to an increased awareness of a key issue on the object of output explanation. In the case of a ML system such as CibNav, outputs are predictions, and their full understanding and explanation can only take place if those predictions are compared with actual events, not only in the development phase but also in the deployment phase. This entails that a large amount of data has to be collected at the very least in the early deployment phase, and preferably throughout the software life cycle, in order to assess the quality of prediction. This is not a menial task, as it sometimes implies an integration of data sources that are dispersed in several data silos under several legal regimens. The methodological obligation to collect post-deployment data thus represents a considerable demand for project management: it might be more work-intensive to collect this data than to develop the model. This problem will of course be present for data collection in private institutions, but public administrations are typically subjected to more stringent legal obligations on their data management.

In the CibNav case, data collection on predictions was immediate, as agents are to use a tablet during their ship inspection which immediately centralizes all data collected on site. This is in striking contrast with some other ML applications, such as the famous COMPAS software attributing a recidivism risk score to American defendants. When the controversy on this software begun with the criticism made in the first ProPublica NGO publication [Larson et al.(2016)Larson, Mattu, Kirchner, and Angwin], it was noticed there that post-deployment data on COMPAS was simply not collected. The software was deployed in several American states with diverse legislations, and there was no systematic centralization of recidivism data for the defendants who were attributed a COMPAS risk score. This means that the quality of prediction could simply not be assessed after the software was deployed. No matter what the results of that collection would be, this is a major methodological and political problem. It is difficult to argue for a scientific, data-based shift in public policy if the quality of predictive systems is simply not assessed in the real-world: there is no such thing as an evidence-based policy without evidence. Despite the considerable technical difficulties it would entail, it could be argued that predictive ML models for high-stake decisions should not be deployed if data on the quality of their prediction cannot be collected. This is not the place to give a full argumentation for that viewpoint, but it is worthwhile to mention it for discussion, as, to the extent of our knowledge, this issue is barely discussed in the current literature.

This is the reason why we said that the key issue is the delimitation of the object of explanation, or explanandum. From a political perspective, the ultimate object of explanation for a predictive ML model are the post-deployment predictions, not the predictions made during the train-test phase. One of the most sensitive questions the developer might have to face is the following contrastive question: "why does the algorithm not behave as foreseen on the post-deployment dataset?". Explanation should not stop at the doorstep of real life, that is, stop where it should begin.

It is crucial to notice that this lesson can be generalized to other algorithms. During our interview with Dr Jacquelinet on the ScoreCœur algorithm, the importance to collect data on prediction quality was underlined many times. In Dr Jacquelinet's words, too great a focus on explainability might lead to give too much importance to our a priori expectations on the algorithm behavior, which might be defeated by confrontation with post-deployment data. This mistake could be avoided if it is clearly understood that the genuine explananda are post-deployment outputs. This seemingly trivial remark actually has significant philosophical consequences, as it means that the ultimate explanandum is not the algorithm alone, but the couple (algorithm, post-deployment dataset).

This is particularly important for two reasons. The first reason is that it generalizes issues of predictive algorithms to other types of algorithms. A tax computing algorithm is not typically considered a predictive model. However, if it is combined with a particular dataset D, and a data model stating that D should be representative of our fiscal population next year, then the pair (algorithm, dataset) yields a set of predictions on the fiscal revenue for next year. If we adopt this perspective, many issues over prediction in the ML literature might be generalized not only to non-ML, perfectly interpretable algorithms, but also to algorithms that are not even considered predictive in nature.

The second reason is that post-deployment data sets are particular datasets, with both limitations and unexpected features, and those limitations and features are also in need of explanation. Let us illustrate this by some remarks on our medical algorithm. One of the main contrastive issues that we considered during our study of this algorithm was whether its medical core could be considered an international standard. We have seen above that the decision to give priority to younger patients is a political decision, but other parts of the algorithm are based on hard medical factors, and could be discussed as a candidate for international medical standard. As we already mentioned above, this is a key contrastive issue for patients, as they might wonder whether they would have gotten a different score if they were patients in a different country.

When asked whether the algorithm could become an international standard, Dr Jacquelinet insisted that this could not be done by a pure a priori discussion of its merits, but only by an experimental confrontation with other populations. The simulations performed on the medical algorithm has shown that its performances could be sensitive to rapid changes in the population, such as an epidemic affecting heart patients for instance. This population sensitivity should be empirically tested on various real-world datasets. However, a real-world population has particular properties which should be taken impact not only the performances, but their interpretation. For instance, the algorithm might fare better in a society where access to health care is not universal. Since health issues are highly correlated to social status, the algorithm might have better performances simply because the hardest patients, who are often the poorest ones, are already dead, and never entered the waiting list. The overall performances of the algorithm are thus affected by the local social system, including, but not limited to, the health care system, as it contributes to preselect the patients and thus bias the dataset. Assessing the performances of the algorithm is thus inseparable from the study of dataset biases.

This remark has become common knowledge in ML, but it should be seen as a generic problem of the contrastive explanation of algorithms. This underlines the importance of the a posteriori, in vivo approach that we are trying to use. Explanatory issues should be explored at a least three different stages: in the mathematical, a priori study of the algorithm, in the interaction with dataset before deployment, and in the interaction with post-deployment data. This is our last model-agnostic lesson: explanations, scientific and pedagogical, descriptive and contrastive, should all be seen as parts of an empirical exploration of the interaction between algorithms and (post-deployment) data.

3 CONCLUSIONS

Some conceptual distinctions on the nature of explanation are crucial to the development of explainability as a scientific subdiscipline. If some, such as the local vs global explanation, are already well-implemented in the literature, some deserve more attention. It is the reason why we should insist on the distinction between contrastive and descriptive explanation, and the cross-cutting distinction between explanation of the implementation and explanation of the specification. If we take into consideration the philosophical debate between agent-neutral and agent-relative accounts of justification, justification is not necessarily to be understood as the contrastive explanation that a given decision is good, and this should also be known by the XAI community.

As we have said above, many of the lessons drawn out of this work turned out to be much more generic than expected. Our comparative work amply demonstrates that many issues are not model-specific, and that scientific and pedagogical explanations can share similar problems. Let us now underline our most generic lessons. From a methodological standpoint, exchanges between Computer Science and Social Sciences are important in part because the interaction between pedagogical and scientific explanation is not null. Pedagogical explanation is not something that just happens after a program is developed and scientific explanation is done. As we have seen in our ML example, explanation can participate to the design process of the algorithm, not only because it acts as a sanity check for the developers, but also as it allows communication with end users that will enrich the specification of the model and put some of the features of a given implementation in a new critical light thanks to the causal intuitions of domain experts. Explanation should not be a PR issue or a legal compliance chore for developers: it should be seen as a desirable feature, and a great opportunity for scientific progress. This is of course mainly true of ML models, but it can be read as a model-agnostic lesson.

Our medical and ML case studies have shown the importance of the couple (algorithm/model, post-deployment data) as a the ultimate explanandum for explanations of algorithms. This is "ultimate" in a political, not a scientific sense. A priori, mathematical analysis of algorithms and study of train/test data are just as scientifically important as study of post-deployment data. However, from a political perspective, the real explanandum is this couple because it represents

the effect of deployment, and faces tough contrastive questions on discrepancies between expectations taken out of the training/test phase and post-deployment results. Studying the (algorithm/model, post-deployment data) couple implies to collect and analyze post-deployment data, a huge constraint on project management which should be a topic of political and legal debate, especially when high-stake decisions are considered.

The list of factors taken into account by an algorithm is the first issue of contrastive explanation of the specification: we need to understand what the factors are and why they are taken into account for a given task. We have seen in our tax example that contrastive explanation of the list of inputs can already be a hot political topic. Furthermore, the limitations of human memory make it hard to deal with long list of factors, and this is a general explainability issue, not something specific to the gigantic dataspace dimensions found in ML. Coarse-graining through extra layers of abstraction will often be necessary in order to develop a synthetic overview of the factors at play. This synthetic overview of factors is sometimes produced by domain expert terminology, but it will not always be the case. We conjecture that the issue of coarse-graining is hard, that it will show up frequently and should be a subject of future research.

The explanation of value setting is yet another issue, which might also need more attention. The conception of an algorithm sometimes implies to fix numerical values for a given set of parameters, and those values can easily generate a feeling of arbitrariness. It is thus necessary to reflect on the constrastive explanation of value setting as opposed to value selection. It will probably be a highly technical topic, which will be hard to present in a pedagogical fashion. As a conjecture, we have proposed that opaque models such as DL are hard even from a scientific perspective because they are based on statistical fine-tuning of parameters, which is a structurally hard explanatory issue.

Some of our takeaways are model-specific, and also specific to pedagogical explanation. Algorithms are becoming media of expert knowledge: computation formulas, expert terminology, sophisticated scientific knowledge. Even if this has little to do with computer science per se, an explanation of algorithms for the general public cannot afford to ignore those issues.Terminological issues are a first major barrier to understanding, and this is bound to be a frequent occurrence in the medical, financial and legal fields, which will be major fields of pedagogical explanation considering their significant impact on large populations. We have seen that terminology encodes knowledge, and that terminological issues should not be misconstrued as linguistic issues outside of the scope of XAI. Furthermore, the issue of linguistic accessibility properly speaking is bound to be an important problem for real-world explanation, even if it is theoretically outside the scope of XAI: this should be read as an invitation for intense collaboration between XAI and NLP.

We have tried to see if the technically most simple approach to explainability, Black Box Reasoning, would perform well in front of our case studies. In order to give a fair assessment of the BBR approach, it is necessary to see that it works in a descriptive approach to the algorithm, and mostly to its specification. However, contrastive approach to specification, even for something as simple as input list, is bound to be decisive in many public debates, and it is not apart of BBR. Understanding the meaning of inputs is decisive for both contrastive and descriptive explanation, and it is not fully taken into consideration in the BBR approach. Finally, and most surprisingly, the explanation of the training phase of a ML system cannot be part of the BBR approach either, and it is also bound to play an important role in many public debates. Again, those remarks cannot be read as a criticism of the BBR approach, but as a proper understanding of its scope, which is descriptive explanation of the specification.Unsurprisingly, the BBR approach fares well with ML systems such as the one we studied here. More surprisingly, BBR may also be relevant for pedagogical explanation of programs an average computer scientist would deem simple and interpretable. Even in a simple case such as the computation of the housing tax, a BBR approach is a good way to generate accessible explanations for the large part of the population which suffers from mathophobia. Our remarks on ML have also shown that black boxing might be a positive pedagogical gesture, as knowledge of the inner workings of a white box model might only act as distracting noise for some questions of end users. By contrast, the BBR approach has limited interest in case such as the ScoreCœur algorithm, where the proper algorithmic complexity is very low, but the understanding of input factors is hard. Black boxing has limited advantage when the explanation of input-output behavior will actually be very close to the original intuitive code, and counterfactual reasoning has limited pedagogical value when one cannot understand the factors to which it applies. However, BBR might be relevant to explain some of the scientific knowledge used in the algorithm. The difficulties raised by the sheer number of factors show that it is not so easy, as our initial hypothesis might have suggested, to let the algorithm grow in size when the BBR approach is successful.

Acknowledgements

This article is hugely indebted to the many conversations and interviews we were able to conduct with developers and domain experts. First of all, the members of the Etalab task force such as Simon Chignard, Bastien Guerry and Soizic Pénicaud, but also Loup Cellard, a history doctoral student conducting another field work in the French administration. Beside this, our conversations with Clément Hénin, Quentin Loridant and Dr Christian Jacquelinet were the real drivers of this work: practically every single new idea came out of those conversations in one way or another.

This work was supported partly by the french PIA project "Lorraine Université d'Excellence", reference ANR-15-IDEX-04-LUE.

References

[noa()] The Book of Why von Judea Pearl, Dana Mackenzie. eBooks | Orell Füssli. URL https://www.orellfuessli.ch/shop/home/artikeldetails/ID85007842.html?ProvID=10917737&gclid=Cj0KCQjwmpb0BRCBARIsAG7y4zb1uol_K9bH6fhnXDGck4M24u3iQOLpQivxA5tie5X67rpM7W48nYIaAjL5EALw_wcB.

[Doshi-Velez(2018)] Alexander Kortz Mason Budish Ryan Bavitz Chris Gersham Sam O'Brien David Schieber Stuart Waldo James Weinberger David Wood Doshi-Velez, Finale. Accountability of AI Under the Law: The Role of Explanation. In *Privacy Law Scholars Conference*, 2018.

[Larson et al.(2016)Larson, Mattu, Kirchner, and Angwin] Jeff Larson, Surya Mattu, Lauren Kirchner, and Laura Angwin. How We Analyzed the COMPAS Recividism Algorithm. Technical report, Propublica, May 2016. URL https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[Miller(2019)] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, February 2019. ISSN 0004-3702. doi: 10.1016/j.artint.2018.07.007. URL http://www.sciencedirect.com/science/article/pii/S0004370218305988.

[Mittelstadt, Brent et al.(2019)Mittelstadt, Brent, Russell, Chris, and Wachter, Sandra] Mittelstadt, Brent, Russell, Chris, and Wachter, Sandra. Explaining Explanations in AI. In *ACM FAT*'19 Conference*, 2019.

[Poyiadzi et al.(2020)Poyiadzi, Sokol, Santos-Rodriguez, De Bie, and Flach] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

[Páez(2019)] Andrés Páez. The Pragmatic Turn in Explainable Artificial Intelligence. *Minds and Machines*, 29(3): 441–459, 2019. doi: 10.1007/s11023-019-09502-w. Publisher: Springer Verlag.

[Sharma et al.(2019)Sharma, Henderson, and Ghosh] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models. *arXiv preprint arXiv:1905.07857*, 2019.

[Villani, Cédric and Longuet(2018)] Villani, Cédric and Gérard Longuet. Les algorithmes au service de l'action publique : le cas du portail admission post-bac. Technical Report 305 (2017-2018), February 2018.

[Wachter et al.(2017)Wachter, Mittelstadt, and Russell] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR.(2017). *Harvard Journal of Law & Technology*, 31:841, 2017.

[Woodward(2019)] James Woodward. Scientific Explanation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2019 edition, 2019. URL https://plato.stanford.edu/archives/win2019/entries/scientific-explanation/.