TYPE Original Research
PUBLISHED 24 April 2023
DOI 10.3389/frai.2023.1128212

OPEN ACCESS

EDITED BY
Dursun Delen,
Oklahoma State University, United States

REVIEWED BY
Francesca Pratesi,
Pisa Research Area (CNR), Italy
Laurence Rozé,
Institut national des sciences appliquées de Rennes, France

*CORRESPONDENCE
Joao Marques-Silva
✉ joao.marques-silva@irit.fr

RECEIVED 20 December 2022
ACCEPTED 29 March 2023
PUBLISHED 24 April 2023

CITATION
Marques-Silva J and Ignatiev A (2023) No silver bullet: interpretable ML models must be explained. *Front. Artif. Intell.* 6:1128212.
doi: 10.3389/frai.2023.1128212

No silver bullet: interpretable ML models must be explained

Joao Marques-Silva[1]* and Alexey Ignatiev[2]

[1]IRIT, CNRS, Toulouse, France, [2]Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Monash University, Melbourne, VIC, Australia

Recent years witnessed a number of proposals for the use of the so-called interpretable models in specific application domains. These include high-risk, but also safety-critical domains. In contrast, other works reported some pitfalls of machine learning model interpretability, in part justified by the lack of a rigorous definition of what an interpretable model should represent. This study proposes to relate interpretability with the ability of a model to offer explanations of why a prediction is made given some point in feature space. Under this general goal of offering explanations to predictions, this study reveals additional limitations of interpretable models. Concretely, this study considers application domains where the purpose is to help human decision makers to understand why some prediction was made or why was not some other prediction made, and where irreducible (and so minimal) information is sought. In such domains, this study argues that answers to such why (or why not) questions can exhibit arbitrary redundancy, i.e., the answers can be simplified, as long as these answers are obtained by human inspection of the interpretable ML model representation.

KEYWORDS

explainable AI (XAI), model interpretability, logic-based explainability, decision trees, decision lists, decision sets

1. Introduction

Recent years witnessed many successes of machine learning (ML) (LeCun et al., 2015; Goodfellow et al., 2016, 2020; Krizhevsky et al., 2017; Bengio et al., 2021). Despite these successes, there are shortcomings to the deployment of ML models (Szegedy et al., 2014; Goodfellow et al., 2015, 2016). Indeed, complex ML models can exhibit lack of robustness, can display bias, and their operation is invariably inscrutable for human decision makers (Gunning and Aha, 2019). As a result, there have been efforts to devising logically rigorous (and so formal) approaches to reasoning about ML models (Marques-Silva and Ignatiev, 2022).

In some application domains, e.g., in high-risk and safety-critical settings, a number of researchers have proposed the use of so-called interpretable models (Rudin, 2019; Molnar, 2020), which include, for example, decision trees, decision lists, and decision sets, among others (Molnar, 2020). Despite the term "*interpretable model*" being extremely popular (Rudin, 2019; Molnar, 2020), it is also the case that there is no rigorous definition for what an interpretable model should be. The subjectivity of what interpretability should mean indicates that a rigorous widely accepted definition is at least fairly unlikely. Accordingly, some other researchers have raised important concerns about what *interpretability* of ML models might represent (Lipton, 2018).

In the case of decision trees, we have recently shown (Izza et al., 2020, 2022a; Huang et al., 2021b) that, when compared with logically rigorous explanations, decision trees can yield explanations which are arbitrarily redundant on the number of features.[1]

[1] Extraction of rules from decision trees has been studied before (Quinlan, 1987), but not for addressing explaination redundancy.

Concretely, given some point in feature space and a predicted class, the question "why does the ML model predict the class?" is referred to as a WHY question. For decision trees, it has been shown (Izza et al., 2020, 2022a; Huang et al., 2021b) that the answer to this WHY question can be arbitrarily redundant when the explanation corresponds to the path in the decision tree that is consistent with the values assigned to the features. A corollary of these results is that, if succinct explanations can be viewed as a measure of model interpretability, then decision trees can hardly be deemed interpretable. Furthermore, a human decision maker will in most cases be unable to propose explanations less redundant than the tree path consistent with the input, and so automated computation of explanations is required.

This study extends further these earlier results on the redundancy of decision trees. We consider additional families of so-called interpretable ML models, and investigate what would be the answer to WHY questions. Since the internal details of the model are in general of no interest to a human decision maker, the answer to such a WHY question is to be expressed as an irreducible subset of the features, such that such set is *sufficient* for the prediction. A set of features is sufficient for the prediction if those features are fixed to their given values, then the value of the prediction must be the given one. Such definition enables *interpreting* the answers to WHY questions as logically correct universally valid rules, which can be conveyed to a human decision maker.

As with other related work, we seek answers to WHY questions which can be trusted. As a result, we need first to formalize what the answers to WHY questions mean. Afterwards, we argue that it is not intuitive (quite the contrary) to obtain such rigorous answers to WHY questions from (manual) inspection of the model. Thus, this further supports the argument against declaring ML model to be interpretable, even when these are claimed to be interpretable. The experimental results included in the study support extensively our conclusion. Concretely, the results show that explanations obtained by inspection of an ML model often include a significant degree of redundancy, and this represents information that is unnecessary to understand the reasons for why a prediction is being made.

Despite their shortcomings, there are still important reasons to advocate the use of these so-called interpretable models. One of these reasons is that such models can be efficiently explained in practice by using the rigorous definitions of explanations proposed in recent years. Given such definitions, we have provided empirical evidence that logically correct rules can be efficiently computed for several families of ML classifiers widely regarded as interpretable (Izza et al., 2020, 2022a; Huang et al., 2021b; Ignatiev and Marques-Silva, 2021). The assessment of the so-called interpretable models included in this study hinges on the fact that rigorous explanations are efficient to compute, even when in theory computing some of these explanations is computationally hard.

This article is organized as follows: Section 2 introduces the notations and definitions used throughout the article. Section 3 introduces logic-based explanations, and briefly overviews recent work on this topic. Section 4 proposes a measure of understanding ML models, namely model comprehensibility, and discusses examples that suggest that even interpretable models are not simple to comprehend. Section 5 summarizes a number of results which offer additional evidence to the difficulty in comprehending interpretable models. Section 6 analyzes experimental results on comprehending interpretable models, concretely decision trees and decision lists. The results of the article are briefly put into perspective in Section 7. Finally, the article concludes in Section 8.

2. Preliminaries

2.1. Classification problems

Classification problems in ML are defined on a set of features (or attributes) $\mathcal{F} = \{1, \ldots, m\}$ and a set of classes $\mathcal{K} = \{c_1, c_2, \ldots, c_K\}$. Each feature $i \in \mathcal{F}$ takes values from a domain $\mathbb{D}_i$. In general, domains can be categorical or ordinal, with values that can be boolean, integer, or real-valued. Feature space is defined as $\mathbb{F} = \mathbb{D}_1 \times \mathbb{D}_2 \times \ldots \times \mathbb{D}_m$. For boolean domains, $\mathbb{D}_i = \{0, 1\} = \mathbb{B}$, $i = 1, \ldots, m$, and $\mathbb{F} = \mathbb{B}^m$. The notation $\mathbf{x} = (x_1, \ldots, x_m)$ denotes an arbitrary point in feature space, where each x_i is a variable taking values from $\mathbb{D}_i$. The set of variables associated with features is $X = \{x_1, \ldots, x_m\}$. Moreover, the notation $\mathbf{v} = (v_1, \ldots, v_m)$ represents a specific point in feature space, where each v_i is a constant representing one concrete value from $\mathbb{D}_i$. When referring to the domains of one of more features, we use $\mathbb{D} = \langle \mathbb{D}_1, \ldots, \mathbb{D}_m \rangle$, which serves solely to aggregate all the features' domains in a single dedicated structure.

With respect to the set of classes $\mathcal{K}$, the size of $\mathcal{K}$ is assumed to be finite; no additional restrictions are imposed on $\mathcal{K}$. An ML classifier $\mathcal{M}$ is characterized by a (non-constant) *classification function* κ that maps feature space $\mathbb{F}$ into the set of classes $\mathcal{K}$, i.e., $\kappa : \mathbb{F} \to \mathcal{K}$. An *instance* (or observation) denotes a pair $(\mathbf{v}, c)$, where $\mathbf{v} \in \mathbb{F}$ and $c \in \mathcal{K}$, with $c = \kappa(\mathbf{v})$. In should be plain to conclude that the formalization of ML classifiers imposes few (if any) restrictions on the families of classifiers that can be studied by using logic-based representations of those classifiers.

Given the definitions above, a classification problem is a tuple $\mathcal{M} = (\mathcal{F}, \mathbb{D}, \mathbb{F}, \mathcal{K}, \kappa)$, and $\mathbb{M}$ denotes the set of all classification problems.

2.2. ML models regarded as interpretable

Although a wide range of ML models are often deemed interpretable (Molnar, 2020), we will consider tree and rule models (Flach, 2012), namely decision trees, decision lists, and decision sets, in their simplest forms. These are widely regarded as interpretable (Lakkaraju et al., 2016; Rudin, 2019, 2022; Molnar, 2020).

2.2.1. Decision trees

A decision tree is a directed acyclic graph, with one root node that has no incoming edges, and the remaining nodes having exactly one incoming edge. Terminal nodes have no outgoing edges, and non-terminal nodes have two or more outgoing edges. Each terminal node is associated with a class, i.e., the predicted class for the node. Each non-terminal node is associated with exactly one feature (i.e., unless otherwise stated, we consider *univariate* DTs).

Each outgoing edge is associated with a literal defined using the values of the feature, and such that any value of the feature domain is consistent with exactly one of the literals of the outgoing edges. In general, we allow literals to use the $\in$ relational operator, as in earlier work (Izza et al., 2022a). Thus, a literal $x_i \in \{S_i\}$ is consistent if x_i takes one of the values in S_i. For simplicity, when $S_i = \{v_i\}$, then we will also allow for a literal to be of the form $x_i = v_i$. Common (implicit) assumptions of DTs are that: i) all paths in a DT are consistent; and ii) the branches at each node capture all values in the domain of the tested feature. An example of a DT is shown in Figure 1. (This example will be analyzed in greater detail below).

2.2.2. Decision lists and sets

Both decision lists and sets represent sets of rules. A rule is of the form: IF cond THEN class, i.e., if the condition cond is true given the values assigned to features, then class is predicted. When cond is true, we say that the rule *fires*; cond can for example represent a conjunction of literals, where a literal is defined as in the case of DTs. Moreover, the difference between decision rules and sets is that for decision lists, the rules are ordered, and for decision sets, the rules are unordered. Thus, a decision list is organized as follows:

$$\begin{array}{llcll} R_1: & \text{IF} & (\tau_1) & \text{THEN} & d_1 \\ R_2: & \text{ELSE IF} & (\tau_2) & \text{THEN} & d_2 \\ \ldots & & & & \\ R_r: & \text{ELSE IF} & (\tau_r) & \text{THEN} & d_r \\ [R_{\text{DEF}}: & \text{ELSE} & & & d_{r+1}] \end{array} \tag{1}$$

In contrast, a decision set is organized as follows:

$$\begin{array}{llcll} R_1: & \text{IF} & (\tau_1) & \text{THEN} & d_1 \\ R_2: & \text{IF} & (\tau_2) & \text{THEN} & d_2 \\ \ldots & & & & \\ R_r: & \text{IF} & (\tau_r) & \text{THEN} & d_r \\ [R_{\text{DEF}}: & & & & d_{r+1}] \end{array} \tag{2}$$

Where the last rule is optional.

A difficulty with decision sets is rule overlap, i.e., the existence of situations when two or more rules predicting different classes fire (Observe that if overlapping rules predict different classes, then the decision set does not implement a classification function). Rule overlap was investigated in recent work (Lakkaraju et al., 2016), but with a definition of overlap that is restricted to the instances in the dataset. As a result, as first observed in Ignatiev et al. (2018), the solution proposed in Lakkaraju et al. (2016) is susceptible to overlap for points in feature space that are not in the dataset.

Another issue with decision sets (when a default rule is not used) is the fact that, for some points in feature space, it may be the case that no rule will fire. Approaches that guarantee no overlap were proposed in Ignatiev et al. (2018). To the best of our knowledge, no rigorous approach exists that guarantees that a decision sets implements a total function, i.e., guarantee of i) no overlap and ii) a prediction for every point in feature space. If these conditions are not met, interpretability is even more of a challenge (Furthermore, Ignatiev et al. (2018) conjectures that learning a DS that respects the two conditions above is Σ_2^{P}-hard).

The learning of decision sets that implement a total function without overlap is believed to be a computationally challenging task (Ignatiev et al., 2018). Moreover, no solution exists that guarantees that a decision set implements a total function without overlap. Thus, in the remainder of this article, we will focus on decision trees and decision lists.

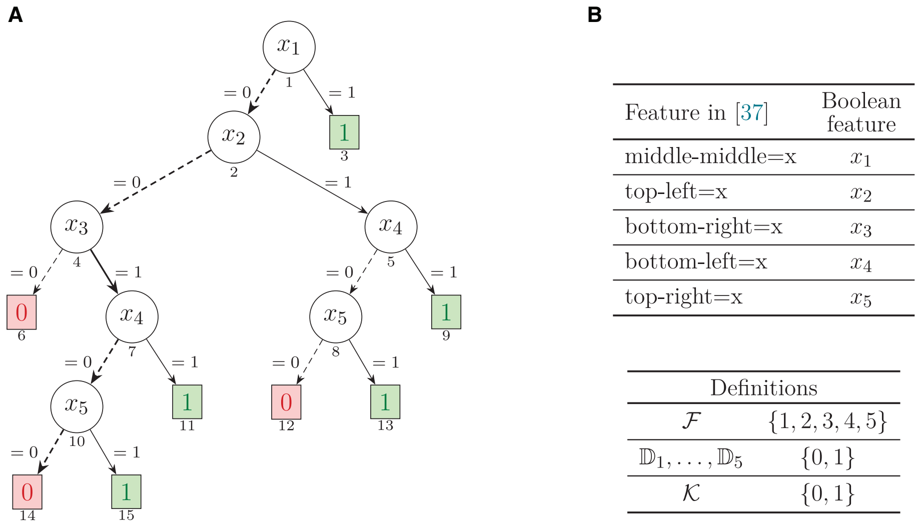
2.3. Logic foundations

Throughout this article, we will use notations and definitions that are standard when reasoning about the decision problem for propositional logic, i.e., the Boolean Satisfiability (SAT) problem (Biere et al., 2021). SAT is well-known to be an NP-complete (Cook, 1971) decision problem. A propositional formula φ is defined over a finite set of propositional atoms $X = \{x_1, x_2, \ldots, x_n\}$ (The elements of X are also referred to as boolean variables). Well-formed propositional formulas are defined inductively given a set of logic operators, $\wedge, \vee$, and $\neg$ (resp. AND, OR, and NOT). Additionally often used logic operators include $\rightarrow$ and $\leftrightarrow$ (resp. implication and equivalence). In practice, propositional formula are most often represented in *conjunctive normal form* (CNF). A CNF formula is a conjunction of clauses, a clause is a disjunction of literals, and a literal is a variable (x_i) or its negation ($\neg x_i$). A term is a conjunction of literals. Whenever convenient, a formula is viewed as a set of sets of literals. A boolean interpretation (or valuation) ν of a formula φ is a total mapping of X to $\{0, 1\}$ (0 corresponds to **false** and 1 corresponds to **true**). Interpretations can be extended to literals, clauses, and formulas with the usual semantics of propositional logic; hence, we can refer to l^ν, ω^ν, τ^ν, and φ^ν, to denote, respectively, the value assigned to a literal, clause, term, and formula, given an interpretation. Given a formula φ, ν is a *model* of φ if it makes φ **true**, i.e., $\varphi^\nu = 1$. A formula φ is *satisfiable* ($\varphi \nvDash \bot$) if it admits a model; otherwise, it is *unsatisfiable* ($\varphi \vDash \bot$). Given two formulas φ and ψ, we say that φ *entails* ψ (denoted $\varphi \vDash \psi$) if all models of φ are also models of ψ. φ and ψ are equivalent (denoted $\varphi \equiv \psi$) if $\varphi \vDash \psi$ and $\psi \vDash \varphi$.

For an unsatisfiable CNF formula φ, let $\mathcal{T}$ denote the set of clauses in φ. In this case, a *minimal unsatisfiable subset* (MUS) $\mathcal{U}$ is an irreducible subset of the clauses in $\mathcal{T}$ that is also unsatisfiable. A *minimal correction subset* (MCS) is an irreducible subset $\mathcal{C}$ of $\mathcal{T}$, such that $\mathcal{T} \setminus \mathcal{C}$ is satisfiable. In general, these definitions can assume some background knowledge $\mathcal{B}$, which is known to be consistent, and some other knowledge $\mathcal{S}$, such that $\mathcal{B} \cup \mathcal{S}$ is unsatisfiable. A fundamental result in the analysis of inconsistent formulas is the minimal hitting set (MHS) duality between MUSes and MHSes (Reiter, 1987) (Recall that a set $\mathcal{H}$ is a *hitting set* of a set of sets $\mathcal{S} = \{S_1, \ldots, S_k\}$ if $\mathcal{H} \cap S_i \neq \emptyset$ for $i = 1, \ldots, k$. $\mathcal{H}$ is a minimal hitting set of $\mathcal{S}$, if $\mathcal{H}$ is a hitting set of $\mathcal{S}$, and there is no proper subset of $\mathcal{H}$ that is also a hitting set of $\mathcal{S}$). There exist in-depth overviews of algorithms for reasoning about inconsistent (or unsatisfiable) formulas, e.g., Marques-Silva and Mencía (2020).

3. Logic-based explainable AI

Logic-based (or formal) explanation approaches have been studied in a growing body of research in recent years (Shih et al., 2018, 2019; Ignatiev et al., 2019a,b,c, 2020a, 2022; Narodytska et al., 2019; Wolf et al., 2019; Audemard et al., 2020, 2021, 2022a,b;



Feature in [37]	Boolean feature
middle-middle=x	x_1
top-left=x	x_2
bottom-right=x	x_3
bottom-left=x	x_4
top-right=x	x_5

Definitions	
$\mathcal{F}$	$\{1, 2, 3, 4, 5\}$
$\mathbb{D}_1, \ldots, \mathbb{D}_5$	$\{0, 1\}$
$\mathcal{K}$	$\{0, 1\}$

FIGURE 1
Decision tree, adapted from (Hu et al., 2019, Figure 5b), for the tic-tac-toe dataset. This DT is also studied more recently in Izza et al. (2022a) and Marques-Silva (2022). Each feature tests a possible play for the X player. The (boxed) terminal nodes display the predicted class. The (circled) non-terminal nodes display the tested feature. The edges label depicts the feature's tested literal, for the edge to be consistent. The number below each node denotes a unique number given to each node, which enables representing paths in the DT. **(A)** Decision tree. **(B)** Mapping of features.

Boumazouza et al., 2020, 2021; Darwiche, 2020; Darwiche and Hirth, 2020, 2022; Izza et al., 2020, 2021, 2022a,b; Marques-Silva et al., 2020, 2021; Rago et al., 2020, 2021; Shi et al., 2020; Amgoud, 2021; Arenas et al., 2021; Asher et al., 2021; Blanc et al., 2021, 2022a,b; Cooper and Marques-Silva, 2021; Darwiche and Marquis, 2021; Huang et al., 2021a,b, 2022; Ignatiev and Marques-Silva, 2021; Izza and Marques-Silva, 2021, 2022; Liu and Lorini, 2021, 2022a; Malfa et al., 2021; Wäldchen et al., 2021; Amgoud and Ben-Naim, 2022; Ferreira et al., 2022; Gorji and Rubin, 2022; Huang and Marques-Silva, 2022; Marques-Silva and Ignatiev, 2022; Wäldchen, 2022; Yu et al., 2022), and are characterized by formally provable guarantees of rigor, given the underlying ML models. Given such guarantees of rigor, logic-based explainability should be contrasted with well-known model-agnostic approaches to XAI (Ribeiro et al., 2016, 2018; Lundberg and Lee, 2017; Guidotti et al., 2019), which offer no guarantees of rigor. The rest of this section offers a brief overview of logic-based explainability. More detailed overviews can be found elsewhere (Marques-Silva, 2022; Marques-Silva and Ignatiev, 2022).

3.1. Abductive explanations (AXp's)

Prime implicant (PI) explanations (Shih et al., 2018) denote a minimal set of literals (relating a feature value x_i and a constant $v_i \in \mathbb{D}_i$) that are sufficient for the prediction. PI explanations are related with abduction, and so are also referred to as abductive explanations (AXp's) (Ignatiev et al., 2019a)[1]. Formally, given $\mathbf{v} = (v_1, \ldots, v_m) \in \mathbb{F}$ with $\kappa(\mathbf{v}) = c$, a set of features $\mathcal{X} \subseteq \mathcal{F}$ is a *weak abductive explanation* (or weak AXp) if the following predicate holds true[2]:

$$\mathsf{WeakAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c) \ := \ \forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{i \in \mathcal{X}} (x_i = v_i)\right] \\ \rightarrow (\kappa(\mathbf{x}) = c) \tag{3}$$

Moreover, a set of features $\mathcal{X} \subseteq \mathcal{F}$ is an *abductive explanation* (or (plain) AXp) if the following predicate holds true:

$$\mathsf{AXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c) \ := \ \mathsf{WeakAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c) \wedge \\ \forall(\mathcal{X}' \subsetneq \mathcal{X}). \neg \mathsf{WeakAXp}(\mathcal{X}'; \mathbb{F}, \kappa, \mathbf{v}, c) \tag{4}$$

Clearly, an AXp is any weak AXp that is subset-minimal (or irreducible). It is straightforward to observe that the definition of predicate $\mathsf{WeakAXp}$ is monotone, and so an AXp can instead be defined as follows:

$$\mathsf{AXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c) \ := \ \mathsf{WeakAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c) \wedge \\ \forall(j \in \mathcal{X}). \neg \mathsf{WeakAXp}(\mathcal{X} \setminus \{j\}; \mathbb{F}, \kappa, \mathbf{v}, c) \tag{5}$$

This alternative equivalent definition of abductive explanation is at the core of most algorithms for computing one AXp (Throughout the article, we will drop the parameterization

[1] PI explanations were first proposed in the context of boolean classifiers based on restricted bayesian networks (Shih et al., 2018). Independent work (Ignatiev et al., 2019a) studied PI explanations in the case of for more general classification functions, i.e., not necessarily boolean, and related instead explanations with abduction. This article follows the formalizations used in more recent work (Marques-Silva et al., 2020, 2021; Cooper and Marques-Silva, 2021; Huang et al., 2021b, 2022; Ignatiev and Marques-Silva, 2021; Izza and Marques-Silva, 2021; Ignatiev et al., 2022; Marques-Silva and Ignatiev, 2022).

[2] Each predicate associated with a given concept will be noted in sans-serif letterform. When referring to the same concept in the text, the same acronym will be used, but in standard letterform. For example, the predicate name AXp will be used in logic statements, and the acronym AXp will be used throughout the text.

associated with each predicate, and so we will write $\mathsf{AXp}(\mathcal{X})$ instead of $\mathsf{AXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c)$, when the parameters are clear from the context).

Example 1. We consider the example decision tree from Figure 1, and the instance $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$. By inspection (or by following the discussion in Izza et al., 2022a), we can conclude that $\{3, 5\}$ is the only AXp, given the instance.

It is apparent that (3), (4), and (5) can be viewed as representing a (logic) *rule* of the form:

$$\textbf{IF} \quad \bigwedge_{i \in \mathcal{X}} (x_i = v_i) \quad \textbf{THEN} \quad \kappa(\mathbf{x}) = c \tag{6}$$

This interpretation of abductive explanations will be assumed throughout the article.

Similar to non-formal approaches to explainability (Ribeiro et al., 2018), abductive explanations can be interpreted as answering a "**WHY**" question, i.e., why is some prediction made given some point in feature space. The answer to this question is a (minimal or irreducible) set of the features, which is sufficient for (or entails) the prediction.

3.2. Contrastive explanations (CXp's)

Similarly to the case of AXp's, one can define (weak) contrastive explanations (CXp's) (Miller, 2019; Ignatiev et al., 2020a).[3] $\mathcal{Y} \subseteq \mathcal{F}$ is a weak CXp for the instance $(\mathbf{v}, c)$ if,

$$\begin{aligned}\mathsf{WeakCXp}(\mathcal{Y}; \mathbb{F}, \kappa, \mathbf{v}, c) \; := \; & \exists(\mathbf{x} \in \mathbb{F}).\left[\textstyle\bigwedge_{i \notin \mathcal{Y}} (x_i = v_i)\right] \\ & \wedge (\kappa(\mathbf{x}) \neq c)\end{aligned} \tag{7}$$

(As before, for simplicity, we will often keep the parameterization of $\mathsf{WeakCXp}$ on κ, $\mathbf{v}$, and c implicit). Thus, given an instance $(\mathbf{v}, c)$, a (weak) CXp is a set of features which if allowed to take any value from their domain, then there is an assignment to the features that changes the prediction to a class other than c, while the features not in the explanation are kept to their values (*ceteris paribus*). Furthermore, a set $\mathcal{Y} \subseteq \mathcal{F}$ is a CXp if, besides being a weak CXp, it is also subset-minimal, i.e.,

$$\begin{aligned}\mathsf{CXp}(\mathcal{Y}; \mathbb{F}, \kappa, \mathbf{v}, c) \; := \; & \mathsf{WeakCXp}(\mathcal{Y}; \mathbb{F}, \kappa, \mathbf{v}, c) \wedge \\ & \forall(\mathcal{Y}' \subsetneq \mathcal{Y}).\neg\mathsf{WeakCXp}(\mathcal{Y}'; \mathbb{F}, \kappa, \mathbf{v}, c)\end{aligned} \tag{8}$$

Similar to the case of AXp's, it is straightforward to observe that the definition of predicate $\mathsf{WeakCXp}$ is monotone, and so an CXp can instead be defined as follows:

$$\begin{aligned}\mathsf{CXp}(\mathcal{Y}; \mathbb{F}, \kappa, \mathbf{v}, c) \; := \; & \mathsf{WeakCXp}(\mathcal{Y}; \mathbb{F}, \kappa, \mathbf{v}, c) \wedge \\ & \forall(t \in \mathcal{Y}).\neg\mathsf{WeakCXp}(\mathcal{Y} \setminus \{t\}; \mathbb{F}, \kappa, \mathbf{v}, c)\end{aligned} \tag{9}$$

Moreover, and again similar to the case of AXp's, this simplified definition of CXp is at the core of algorithms for their computation.

A key observation is that *any* solution of (7), (8), or (9) (be it minimal or not) identifies not only a set non-fixed of features but also assignments to those non-fixed features that guarantee a change of the prediction. Hence, all the information required to change the prediction is readily available. Furthermore, the definition of CXp (similar to the definition of AXp) targets a subset-minimal set of features. However, other definitions could be considered, e.g., cardinality-minimal contrastive explanations, among others.

Example 2. We consider the example decision tree from Figure 1, and the instance $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$. By inspection (or by following the discussion in Izza et al., 2022a), we can conclude that $\{3\}$ and $\{5\}$ are the only CXp's, given the instance.

A CXp can be viewed as a possible answer to a "**WHYNOT**" question, i.e., "why is not the classifier's prediction a class other than c?" (Clearly, the definition can be adapted to the case when we seek a concrete change of class.) A different perspective for a contrastive explanation is the answer to a "**How?**" question, i.e., how to change the features so as to change the prediction. In recent literature, this alternative view has been investigated under the name "*actionable recourse*" (Ustun et al., 2019; Karimi et al., 2020, 2021; Venkatasubramanian and Alfano, 2020).

3.3. Duality between AXp's and CXp's

Given the definitions of AXp and CXp, and building on Reiter's seminal work (Reiter, 1987) (see Section 2.3), recent work (Ignatiev et al., 2020a,b) proved the following duality between minimal hitting sets:

Proposition 1. (Minimal hitting-set duality between AXp's and CXp's Ignatiev et al., 2020a,b) AXp's are minimal hitting sets (MHSes) of CXp's and *vice versa*.

We refer to Proposition 1 as MHS duality between AXp's and CXp's.

Example 3. We consider the DT running example from Figure 1, and the instance, $(\mathbf{v}, c) = ((0, 0, 1, 0, 0), 0)$. Once more by inspection (or by following the discussion in Izza et al., 2022a), we can conclude that the sets of AXp's is: $\{\{1, 4, 5\}\}$ and that the set of CXp's is $\{\{1\}, \{4\}, \{5\}\}$.

Proposition 1 has been used in more recent work for enabling the enumeration of explanations (Huang et al., 2021b; Ignatiev and Marques-Silva, 2021; Marques-Silva et al., 2021).

3.4. Current status of logic-based explainability

There has been steady progress in the efficient computation of explanations (Marques-Silva, 2022; Marques-Silva and Ignatiev, 2022) (and references therein). Moreover, a number of related research topics have been investigated, including enumeration of explanations (Ignatiev et al., 2020a), explainability queries (Audemard et al., 2020, 2021; Huang et al., 2021b),

[3] In this article, contrastive explanations mimic counterfactual explanations. However, in more complex explanation scenarios, they may differ (Liu and Lorini, 2022b).

probabilistic explanations (Wäldchen et al., 2021; Arenas et al., 2022; Izza et al., 2022b), or taking into account constraints in feature space (Gorji and Rubin, 2022; Yu et al., 2022). For the purposes of this article, the more important results are[4]:

1. For decision trees, there are polynomial-time algorithms for computing one AXp, all CXp's can be computed in polynomial time, and there are practically efficient algorithms for the enumeration of AXp's (Izza et al., 2020, 2022a; Huang et al., 2021b).
2. For decision lists, it is computationally hard to compute one AXp/CXp, but existing logic encodings enable the practically efficient computation of one explanation and of the enumeration of explanations (Ignatiev and Marques-Silva, 2021).
3. The approach used for decision lists can also be used in the case of decision sets (Ignatiev and Marques-Silva, 2021), but here the main limitation is requiring that the decision set computes a total function, as discussed earlier in this article.

4. How to understand interpretable ML models?

Since there is no formal definition of what interpretability means, and since such a definition is unlikely (Lipton, 2018), we ask a different question. Concretely, this section investigates how explanations can be obtained from an interpretable model. Since the model is interpretable, we require that a human decision maker be able to find such an explanation by *manual inspection*, i.e., not automated analysis is to be used. Evidently, for an interpretable model, one would expect that this should be feasible to do.

4.1. How to comprehend predictions?

A natural first question is how can a human decision maker understand predictions. Following earlier work (Miller, 2019; Molnar, 2020), we investigate how explanations can be manually identified given an interpretable model. Concretely, given some interpretable model, e.g., decision trees, lists, or sets, we pose the following question:

> "Given an instance $(\mathbf{v}, c)$, why is the prediction c?"

We refer to this question as the WHY question. Similar to recent work on non-formal interpretability and explainability (Lakkaraju et al., 2016; Ribeiro et al., 2018), we seek to answer the WHY question by finding a set of features $\mathcal{X}$, with which we associate the following rule:

$$\text{Xp}: \quad \text{IF } \bigwedge_{i\in\mathcal{X}}(x_i = v_i) \text{ THEN } \kappa(\mathbf{x}) = c \tag{10}$$

Clearly, this rule is required to be logically correct. Moreover, Occam's razor is expected of $\mathcal{X}$, i.e., we require $\mathcal{X}$ to be irreducible (i.e., subset-minimal) (Even if one ensures irreducibility, a possible criticism is that if the size of $\mathcal{X}$ is too large, then the answer to the WHY question may be beyond the cognitive grasp of a human decision maker (Miller, 1956). Methods to address this possible limitation have been studied elsewhere (Wäldchen et al., 2021; Arenas et al., 2022; Izza et al., 2022b)). Furthermore, to keep the notation as simple as possible, and similarly to the definition of AXp's and CXp's, we will talk about the answers to the WHY questions solely using only sets of features. Concretely, $\mathcal{X} \subseteq \mathcal{F}$ is a set of features, that presupposes a literal $(x_i = v_i)$ for each $i \in \mathcal{X}$. As a result, the relationship of a set of features $\mathcal{X}$ with the rule above is immediate.

Another extensively studied type of explanation is contrastive (often referred to as counterfactual) explanations. Similarly to the case of WHY questions, given some interpretable model, e.g., decision trees, lists, or sets, we pose the following question:

> "Given an instance $(\mathbf{v}, c)$ why is the prediction not a class other than c?"

Put another way, what should be changed to change the prediction? We refer to this question as the WHYNOT question.

4.2. Defining model comprehensibility

As a measure of the actual interpretability of an ML model, we propose instead the concept of model comprehensibility. Concretely, we say that an (interpretable) ML model is comprehensible if:

> The ML model enables a human decision maker, *via* non-automated analysis (i.e., by manual inspection of the model), to rigorously answer a WHY question, thereby finding a set of features that is both sufficient for the prediction and irreducible.

Clearly, for interpretable ML models [i.e., those where the explanation is the model itself (Rudin, 2019; Molnar, 2020)], one would expect the model to be comprehensible, thus enabling a human decision maker to grasp answers to the WHY question, and express such answers as general rules, as proposed above. As shown in the rest of this section, although one can devise solutions for finding correct answers to the WHY question, those answers are hardly irreducible. More importantly, as shown in later sections, arbitrary redundancy is inherent to models that are generally deemed interpretable.

It should be observed that an answer to the WHY question corresponds ideally to an AXp[5]. Hence, if we can find AXp's, then we can provide answers to the WHY questions. Similarly, an answer to the WHYNOT questions corresponds ideally to a CXp. Given the above, we can thus conclude that what we are interested, essentially, to assess whether manual analysis of an interpretable model by a human decision maker will serve to find AXp's and/or CXp's.

[4] Following notation that is standard in theoretical computer science, computational problems that are NP-hard are deemed theoretically intractable; however, there might exist practical algorithms that are efficient in practice, albeit not in the worst case. In contrast, polynomial-time algorithms are always assumed to be efficient in practice.

[5] As clear from the outset, we are interested in rigorous explanations to predictions, i.e., explanations that are both correct, given some agreed definition of correct, and irreducible.

TABLE 1 Considering all possible values of x_1, x_2, x_4 when $x_3 = x_5 = 1$.

$x_3 = x_5$	x_1	x_2	x_4	$\kappa(\mathrm{x})$
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

As will be clarified later in the article (see Section 5.4), manually identifying answers to WHYNOT questions can be substantially more difficult that identifying answers to WHY questions.

4.3. Are interpretable models indeed comprehensible?

Before delving into theoretical results on ML model comprehensibility, let us motivate such results with a number of examples. We will analyze decision trees and decision lists, seeking to propose correct approaches for computing answers to the WHY question. Furthermore, we will also inquire how realistic it is to find answers that are both correct and irreducible.

4.3.1. Decision trees

For a decision tree, an intuitive manual approach to propose an answer to the WHY question is as follows (Izza et al., 2022a):

1. Pick the features with literals in the path that is consistent with the prediction.

Clearly, such picked set of features implies a rule that is correct. However, it is unclear whether the set of features is irreducible.

Example 4. For the example in Figure 1, let the instance be $\mathbf{v} = (0, 0, 1, 0, 1)$. Thus, an answer to the WHY question would be $\mathcal{X} = \{1, 2, 3, 4, 5\}$, thus capturing the rule,

$$\text{Xp}: \quad \text{IF } [(\neg x_1) \wedge (\neg x_2) \wedge (x_3) \wedge (\neg x_4) \wedge (x_5)] \text{ THEN } \kappa(\mathbf{x}) = 1$$

It is not too difficult to understand what $\mathcal{X}$ is not irreducible. For example, if we allow x_1 to change value, then the prediction will remain unchanged; hence $\mathcal{X}' = \{2, 3, 4, 5\}$ is also an answer to the WHY question. However, one can provide a much shorter answer. Let us allow features 1, 2, and 4 to take any value, with $x_3 = x_5 = 1$, and let us check the predicted values. The result of this exercise is shown in Table 1. As can be observed, since the prediction remains unchanged for any value assigned to features 1, 2, and 4, we can conclude that an answer to the WHY question in this case is $\mathcal{X}'' = \{3, 5\}$. It is also fairly simple to conclude that $\mathcal{X}''$ is indeed irreducible. However, it seems apparent that most human decision makers would be unable to fathom $\mathcal{X}''$ by inspection of the decision tree.

One might also wonder whether one should be interested in irreducible answers to the WHY question. As illustrated by this example, one would expect that the average human decision maker will be able to relate far better with the following (irreducible) rule,

$$\text{Xp}: \quad \text{IF } [(x_3) \wedge (x_5)] \text{ THEN } \kappa(\mathbf{x}) = 1$$

than with the rule that includes all five features (shown above).

4.3.2. Decision lists

For decision lists, we can raise similar questions. As argued below, answering WHY questions may not be immediate.

Example 5. Consider a DL classifier, with $\kappa(x_1, x_2, x_3, x_4, x_5)$ defined by,

$$\begin{array}{lllll}
\text{R}_1: & \text{IF} & (\neg x_1 \wedge \neg x_2) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\text{R}_2: & \text{ELSE IF} & (x_1 \wedge x_2 \wedge \neg x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_3: & \text{ELSE IF} & (x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_4: & \text{ELSE IF} & (x_1 \wedge x_2 \wedge \neg x_4) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_5: & \text{ELSE IF} & (x_1 \wedge x_2 \wedge \neg x_5) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_{\text{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 0
\end{array} \tag{11}$$

Let $\mathbf{v} = (1, 1, 1, 1, 1)$. Clearly, $\kappa(\mathbf{v}) = 1$.

Suppose we are interested in answering the question: "*Why is the prediction 1 for* $\mathbf{v} = (1, 1, 1, 1, 1)$*?*". Since the model is interpretable, we seek an answer by inspection of the DL. A possible answer is $\mathcal{X} = \{1, 2, 3, 4, 5\}$, i.e., as long as all the features take the value in $\mathbf{v}$, then it is certainly the case that the prediction is 1. However, if the user seeks shorter (logically) correct explanations, e.g., that contain no redundant information, then it may be possible to offer the user far more insightful information (Nevertheless, it should be noted that obtaining such information requires some degree of logical reasoning, which may not be immediate for the average human decision maker). For the example above, it can be shown that $\mathcal{X} = \{1, 2\}$ is a logically correct explanation, i.e., as long as $x_1 = x_2 = 1$, then the prediction will be 1, *independently* of the values taken by the other features.

Example 6. Even if some human decision maker can understand why the answer to the WHY question is $\mathcal{X} = \{1, 2\}$ for the DL above, more subtle scenarios can be be envisioned. Let us consider the following DL:

$$\begin{array}{lllll}
\text{R}_1: & \text{IF} & (x_1 \wedge x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_2: & \text{ELSE IF} & (x_2 \wedge x_4 \wedge x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\text{R}_3: & \text{ELSE IF} & (\neg x_1 \wedge x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_4: & \text{ELSE IF} & (x_4 \wedge x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\text{R}_5: & \text{ELSE IF} & (\neg x_1 \wedge \neg x_3) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_6: & \text{ELSE IF} & (x_6) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\text{R}_{\text{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 1
\end{array} \tag{12}$$

Let the point in feature space be $\mathbf{v} = (0, 1, 0, 1, 0, 1)$, with $\kappa(\mathbf{v}) = 0$, i.e., rule R_2 fires. If a human decision maker is interested in answering the question: "*Why is the prediction 0 for* $\mathbf{v} = (0, 1, 0, 1, 0, 1)$*?*", what are possible explanations from inspecting the model? One might be tempted to state that if $x_2 = x_4 = x_6 = 1$, then the prediction is 0, i.e., the explanation is the condition of R_2. However, such answer is incorrect. For example, if both x_1 and x_3 are flipped to 1, then the prediction would become 1, due to R_1 firing; this means that $x_2 = x_4 = x_6 = 1$ is not a correct explanation. A possible solution is to weaken the explanation, by including additional literals. For example, if either x_1 or x_3 are 0 then, if $x_2 = x_4 = x_6 = 1$, it is the case that the prediction is 0. So, a possible explanation is $x_3 = 0$ and $x_2 = x_4 = x_6 = 1$. Does this explanation represent an irreducible set of literals? Unsurprisingly, the answer is no, and a more careful analysis allows concluding that the answer to the WHY question is: $x_3 = 0$ and $x_4 = x_6 = 1$. It should be apparent from the previous example, that even for simple DLs, finding an answer to a WHY question, which is both correct and irreducible, is not a trivial task. With the purpose of highlighting the challenges of finding correct and irreducible answers to WHY questions, let us consider again the decision list in (12), and let the point in feature space be $\mathbf{v} = (0, 0, 0, 0, 0, 0)$, with $\kappa(\mathbf{v}) = 0$. In this case, since the default rule fires, there is no condition of the rule to start from. Building on the examples above, one might propose $x_3 = x_5 = x_6$ as an explanation. However, more careful analysis confirms that $x_6 = 0$ suffices as an (irreducible) answer to a WHY question, i.e., if $x_6 = 0$, then the prediction will be 1 independently of the values of all the other features. Somewhat less intuitive might be that $x_1 = x_4 = 0$ which is also an irreducible answer to a WHY question, i.e., if $x_1 = x_4 = 0$, then the prediction will be 1 independently of the values of all the other features.

The previous example highlighted some of the requirements for manually answering a WHY question in the case of a decision list. Consider the definition of decision list in (1). Pick some instance $(\mathbf{v}, c)$. Let R_j denote the rule that fires, and let the prediction be c. Hence, we propose to find a correct answer $\mathcal{A} \subseteq \mathcal{F}$ to the WHY question as follows:

1. All the features associated with literals in τ_j are added to $\mathcal{A}$;
2. For each rule R_k preceding R_j, that predicts a class other than c, let i be the feature of the first literal inconsistent with $\mathbf{v}$. Then add feature i to $\mathcal{A}$.

We could conceivably propose optimizations to the procedure above, but these would make it far more difficult for a human decision maker to find on his/her own answers to the WHY questions. However, as illustrated by the next example, that would still not guarantee that the computed answer would be irreducible.

Example 7. Finally, let us consider the following DL:

$$\begin{array}{llcll}
\text{R}_1: & \text{IF} & (x_1 \land x_3) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\text{R}_2: & \text{ELSE IF} & (x_1 \land x_5) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\text{R}_3: & \text{ELSE IF} & (x_2 \land x_4) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_4: & \text{ELSE IF} & (x_1 \land x_7) & \text{THEN} & \kappa(\mathbf{x}) = 0 \\
\text{R}_5: & \text{ELSE IF} & (\neg x_4 \lor x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_6: & \text{ELSE IF} & (\neg x_4 \lor \neg x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_7: & \text{ELSE IF} & (\neg x_2 \lor x_6) & \text{THEN} & \kappa(\mathbf{x}) = 1 \\
\text{R}_{\text{DEF}}: & \text{ELSE} & & & \kappa(\mathbf{x}) = 0
\end{array} \tag{13}$$

with $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$. Clearly, the prediction is 1, due to R_3. What should be the answer to a WHY question in this case? Given what we discussed until now, we might be tempted to propose $\mathcal{X} = \{1, 2, 4\}$, since fixing feature 1 will prevent rules R_1 and R_2 from firing, and fixing features 2 and 4 will ensure that rule R_3 fires. However, since fixing feature 1 also prevents rule R_4 from firing, then the value of feature 4 is actually irrelevant, since $x_4 = 0$ would cause rule R_5 to fire, with $x_1 = 0$ and $x_2 = 1$. Thus, an irreducible answer to the WHY question should be $\mathcal{X} = \{1, 2\}$. The point here is that to find a subset minimal explanation we must not only consider the rules that precede the rule that fired, but also the rules that follow the rule that fired. As before, it appears unrealistic that the average human decision maker would grasp the answer $\{1, 2\}$ by inspection of the DL. More importantly, and similarly to earlier examples, this is a rather simple example: one should expect far more complex examples in practice. Observe that $\{1\}$ does not suffice as the justification for why the prediction is 1. Indeed, by allowing $\mathbf{u} = (0, 0, 0, 1, 0, 0, 0)$ would cause the prediction to change to 0 due to the default rule. Hence, feature 2 is necessary for preventing the prediction from changing.

As proved elsewhere (Ignatiev and Marques-Silva, 2021), it is hard to compute one AXp (resp. CXp) in the case of DLs/DSs. Hence, it would be unrealistic to expect human decision makers to be able to compute answers to the WHY (resp. WHY NOT) question by inspection, as each such answer can be mapped to an AXp (resp. CXp).

5. Non-comprehensibility of interpretable models

Given the understanding of comprehensibility proposed in the previous section, we now argue that even the simplest ML models, which are ubiquitously deemed interpretable, do not respect such understanding. The ensuing conclusion is that, for finding answers to the WHY question, even so-called interpretable models should be explained, by using a rigorous definition of explanation as proposed in Section 3.

5.1. Non-comprehensibility of decision trees

This section summarizes recent results on the non-comprehensibility of DTs. The underlying assumption is that the answer for a WHY question is the path consistent with the values of the features.

Proposition 2. (Corollary 2 of Izza et al., 2022a) There exist DTs, defined on m features, for which there exist instances exhibiting an AXp of size 1, and the path consistent with the instances has size m.

Proposition 3. (Proposition 11 of Izza et al., 2022a) A DT does not exhibit explanation redundancy if it can be represented with a disjunctive normal form (DNF) generalized decision function (GDF).

A GDF is a very restricted class of classifier, and so Proposition 3 indicates that the class of functions that can be represented with DTs without exhibiting path explanation redundancy is very restricted.

5.2. Non-comprehensibility of decision lists

The examples analyzed in Section 4 suggest that a straightforward approach for answering WHY questions in DLs is bound to yield explanations that contain redundant literals.

As experimentally validated by the results in Section 6, this is indeed the case.

Clearly, one might argue that a different approach for finding explanations would yield less or no redundancy. We conjecture that for any manual approach, producing explanations will necessarily introduce redundancy. Concretely, for any pre-specified approach for computing explanations by hand, one can construct a DL for which explanations will exhibit redundancy.

5.3. Non-comprehensibility of decision sets

Previous sections highlighted the many issues with DSs in practical settings. If a DS does not compute a function, then the core assumptions of logic-based explainability are not respected. If the DS computes a partial function, then again the core assumptions of logic-based explainability are not respected. One additional hurdle is that the learning of DSs that compute total functions (and so ensure that no overlap exists) is conjectured to be Σ_2^{P}-hard (Ignatiev et al., 2018). Finally, explanation of DSs when these compute total functions appears to raise at least the same difficulties as DLs.

5.4. Answering WHYNOT questions can be hard

In stark contrast with finding correct answers to WHY questions in DLs, this section proves that the apparently trivial problem of answering a WHYNOT question for a DL is NP-complete, i.e., it is computationally hard to decide whether the prediction can be changed to some other class. The implication of this result is that it would be rather unrealistic to expect human decision makers to decide NP-complete problems when proposing answers to WHYNOT questions. The implication of this result is that interpretability is unattainable when the goal is to answer WHYNOT questions for DLs.

Throughout this section, we consider a CNF formula φ, defined on a set of propositional atoms $\{x_1, \dots, x_m\}$, composed of clauses $\{\varsigma_1, \dots, \varsigma_n\}$, such that each clause ς_i contains three literals and it is non-tautologous. Assignments map each atom to $\{0, 1\}$. Given an assignment, the valuation of a CNF formula φ maps φ to $\{0, 1\}$ (Biere et al., 2021). The decision problem for CNF formulas (i.e., the Boolean Satisfiability (SAT) problem) is to decide whether there exists an assignment such that the formula's valuation is 1. It is well-known that SAT is NP-complete (Cook, 1971) (Technically, for CNF formulas, the decision problem is CNFSAT, and since we consider each clause to contain three literals, the decision problem is referred to as 3CNFSAT. However, we just use SAT to refer to these as well as the original decision problem on arbitrary propositional formulas).

Definition 1 (TOGGLESOME). Given a CNF formula φ, and given an assignment to the atoms of φ that falsifies at least one clause of φ, decide the satisfiability of φ.

Proposition 4. TOGGLESOME is NP-complete.

Proof. (Sketch) TOGGLESOME is in NP. We ignore the starting assignment, guess an assignment to the variables of φ, and then check in polynomial time whether φ takes value 1 given the assignment.

To prove NP-hardness, we reduce SAT to TOGGLESOME. Pick a clause ς in φ, and falsify it. For the remaining atoms, pick a random assignment. Thus, φ with its falsified clause ς (and with other clauses also possibly falsified) and the picked assignment represent an instance of TOGGLESOME. Clearly, φ is satisfiable if and only if the answer to TOGGLESOME is positive, and the reduction runs in polynomial time. □

In the case of a DL, we are interested in the following generic problem. Given an instance $(\mathbf{v}, c)$, can $\mathbf{v}$ be modified so that the prediction is $c' \in \mathcal{K} \setminus \{c\}$? As shown next, just deciding whether a prediction can be changed is in fact a computationally hard problem.

Proposition 5. Deciding whether a prediction in a DL can be changed is NP-complete.

Proof. We consider a DL, such that rule j with prediction c fired on some input. We want to decide which features to change such that the prediction changes value to a class other than c.

The problem is clearly in NP. We non-deterministically guess a certificate, i.e., an assignment of values to the features, and then check whether the resulting prediction is different than the starting one.

To prove NP-hardness, we reduce TOGGLESOME to the problem of deciding the existence of a set of features which, if changed, will allow changing the prediction.

Let φ be a CNF formula (as described above). Without loss of generality, we consider a renumbering of the clauses of φ, such that the picked assignment falsifies the first clause; let it be ς_1, with the three literals of ς_1 referenced by $l_1(\varsigma_1)$, $l_2(\varsigma_1)$, and $l_3(\varsigma_1)$. Moreover, since the clauses are non-tautologous, then $\neg\varsigma_i$ is a non-inconsistent conjunction of propositional literals for any clause ς_i of φ. Now, we construct the following decision list:

$$\begin{array}{llcl}
\text{R}_1: & \text{IF} & z \wedge \neg\varsigma_1 & \text{THEN } 0 \\
\text{R}_2: & \text{ELSE IF} & z \wedge \neg\varsigma_2 & \text{THEN } 0 \\
 & & \dots & \\
\text{R}_n: & \text{ELSE IF} & z \wedge \neg\varsigma_n & \text{THEN } 0 \\
\text{R}_{n+1}: & \text{ELSE IF} & z & \text{THEN } 1 \\
\text{R}_{n+2}: & \text{ELSE IF} & \neg z \wedge \neg\varsigma_1 & \text{THEN } 0 \\
\text{R}_{n+3}: & \text{ELSE IF} & l_1(\varsigma_1) & \text{THEN } 0 \\
\text{R}_{n+4}: & \text{ELSE IF} & l_2(\varsigma_1) & \text{THEN } 0 \\
\text{R}_{n+5}: & \text{ELSE IF} & l_3(\varsigma_1) & \text{THEN } 0 \\
\text{R}_{\text{DEF}}: & \text{ELSE} & & \text{THEN } 1
\end{array}$$

where z is a fresh propositional variable.

Furthermore, we consider some input that causes R_{n+2} to fire, resulting in prediction 0. Then, it must be the case that $z = 0$

and that ς_1 is falsified. Now, for the prediction to change, rules $\mathrm{R}_1, \ldots, \mathrm{R}_n$ must not fire, and rule R_{n+1} must fire. Since we must have $z = 1$ for R_{n+1} to fire, then each $\neg\varsigma_i$ must be falsified (and so each ς_i must be satisfied). As a result, the prediction changes to 1 if and only if φ is satisfied, i.e., that the answer to TOGGLESOME is positive. Observe that the alternative to change the prediction would be for $\mathrm{R}_{\mathrm{DEF}}$ to fire. That would require both $z = 0$ and each literal of ς_1 to be falsified; but then rule R_{n+2} would still fire before the default rule, and that would mean no change in the prediction. Hence, it is impossible for $\mathrm{R}_{\mathrm{DEF}}$ to fire, and so a change of prediction requires R_{n+1} to fire.
(It should be observed that the construction used in rules R_{n+3}, R_{n+4}, and R_{n+5}, which render $\mathrm{R}_{\mathrm{DEF}}$ unreachable, is by no means restrictive. First, the reduction is still from some CNF formula (as an instance of TOGGLESOME) to a DL. Second, a more involved reduction could have been proposed instead. A solution would be to reduce $\varphi_1 \vee \varphi_2$, where φ_1 would be encoded into rules $\mathrm{R}_1, \ldots, \mathrm{R}_{n_1}$, and φ_2 would be encoded into rules $\mathrm{R}_{n_1+3}, \ldots, \mathrm{R}_{n_1+n_2+1}$. In addition, R_{n+2} would consist of $z \wedge \neg\varsigma_{11} \wedge \neg\varsigma_{21}$, requiring at least one clause of φ_1 to be falsified and at least one clause of φ_2 to be falsified. It should be plain that the geralization of TOGGLESOME to the case of $\varphi_1 \vee \varphi_2$ is straightforward). □

Observe that answering a WHYNOT question amounts to deciding whether the prediction can be changed, and that is NP-complete as proved above. Intuitively, the complexity of finding one CXp results from the need to make consistent the condition of some rule R_k, that predicts some class other than c, and such that all the rules that precede R_k must be inconsistent.

It should be noted that the result above could also be established by relating with earlier results on the complexity of computing explanations for DLs (Ignatiev and Marques-Silva, 2021, Prop. 3) and the relationship between the complexity of computing AXp's and CXp's (Cooper and Marques-Silva, 2021, Th. 15) (for a more restricted family of classifiers). The proposed proof offers a more direct argument. Practical efficient algorithms for computing both AXp's and CXp's of DLs are described elsewhere (Ignatiev and Marques-Silva, 2021).

One final comment regarding DTs. There are polynomial time algorithms for computing CXp's (Huang et al., 2021b; Izza et al., 2022a), but also for enumerating all CXp's (Huang et al., 2021b; Izza et al., 2022a). Hence, for DTs, one can argue that there are efficient solutions to answering WHYNOT questions. However, the bookkeeping involved to prevent redundancy in reported CXp's (and so in answering WHYNOT questions) is arguably beyond the reach of the average human decision maker. The proposed algorithm (Huang et al., 2021b; Izza et al., 2022a) lists all possible ways to change a prediction (in polynomial time), and then removes the ways that exhibit redundancies (also in polynomial time). However, such algorithms would require a non-negligible amount of work if the solution were to be computed manually.

6. Experimental evidence

This section overviews the experimental results aiming to practically confirm the claims made earlier in the article. Namely, this section will assess the redundancy (as explained below) of the explanations offered "by default" by decision tree and decision list models trained with well-known and publicly available tools.

The experiments were built on the earlier results and data are published by Ignatiev and Marques-Silva (2021) and Izza et al. (2022a). The datasets and the induced DTs/DLs are adapted from these earlier works. In particular, for DTs, we use the SAT-based implementation of the explanation redundancy checker proposed in Izza et al. (2022a). For DLs, we use the SAT-based implementation of the formal explainer proposed in Ignatiev and Marques-Silva (2021) publicly available online[6]. The latter tool was augmented with the capability to measure redundancy of a given explanation as defined below.

The experiments were performed on a MacBook Pro laptop running macOS Ventura 13.0.1. Each individual process was run on a 6-Core Intel Core i7 2.60 GHz processor with 16 GB of memory. Despite the use of the 4 GB memory limit and 1800 s time limit, none of these limits has been reached for any of the problem instances used. In fact, the redundancy checkers were effective enough to stop and output a redundancy report *long before* the time limit.

As mentioned above, we considered the data from the earlier work. Therefore, all the datasets considered here are taken from the publicly available sources (Friedler et al., 2015; FairML, 2016; PennML, 2020; UCI, 2020) and are taken directly from Ignatiev and Marques-Silva (2021); Izza et al. (2022a). Following prior work (Izza et al., 2022a), we assessed the redundancy of explanations offered by decision trees trained by two prominent DT inference tools: ITI (Incremental Tree Induction) (Utgoff et al., 1997; ITI, 2020) and IAI (Interpretable AI) (Bertsimas and Dunn, 2017; IAI, 2020). When training IAI models, the tool was run being instructed to train high-accuracy DTs of depth either 6 and 8 (Deeper trees are harder to learn, and do not yield significant gains in accuracy). In the following, these two configurations of IAI are referred to as IAI-6 and IAI-8, respectively. As for decision list models, those were trained by the well-known CN2 algorithm (Clark and Niblett, 1989; Clark and Boswell, 1991).

6.1. Measuring explanation redundancy

Given a data instance, an explanation offered by a decision tree model "by default" is assumed to be a set of feature literals appearing in the path that fires the prediction for the instance. In the case of decision list models, a "default" explanation is constructed as the set of feature literals comprising the rule that fires the given prediction plus the first literal in each preceding rule that is determined to be inconsistent with the instance. Note that one does not have to always consider the first such literal. However, this strategy is simple enough be used by a human decision maker.

When a default explanation $\mathcal{X}$ is computed as detailed above, a redundancy checker is run to compute how many literals can be dropped from $\mathcal{X}$ resulting in an abductive explanation $\mathcal{X}' \subseteq \mathcal{X}$, i.e., the AXp condition (4) holds for $\mathcal{X}'$. Afterwards, the redundancy of $\mathcal{X}$ is said to be equal to the portion of features in $\mathcal{X}$ that the redundancy check was able to remove, i.e., it is computed as the value of $100\% \cdot \frac{|\mathcal{X}|-|\mathcal{X}'|}{|\mathcal{X}|}$. Note that our experiment targets

[6] https://github.com/alexeyignatiev/xdl-tool.

computing subset-minimal AXps $\mathcal{X}'$ rather than cardinality-minimal. In the latter case, the redundancy statistics of the default DT and DL explanations would clearly be even higher than the one reported below.

6.2. Redundancy of default explanations

Figure 2 shows four cactus plots depicting the minimum, average, and maximum redundancy of default explanations computed for the considered DT and DL models. Here is how the plots should be interpreted. Given a dataset and the corresponding ML model, *each instance* of the dataset is provided with a default explanation by the model, as described above, which is followed by an explanation redundancy check. Given the redundancy information for all the instances of the dataset, the minimum, average, and maximum explanation redundancy for this dataset is calculated. As a result and considering all the datasets studied, the full minimum/average/maximum redundancy statistics is plotted as a line sorted in ascending order. This way, a point with coordinates (X, Y) signifies that there are X datasets/models whose default explanations have the (minimum, average, and maximum) literal redundancy upper-bounded by Y%.

As can be observed, with slight variation, all the models exhibit significant explanation redundancy. Although there are a large number of datasets whose instances are provided with irredundant default explanations (e.g., see zero minimum redundancy), some instances may receive explanations with 80%–100% redundant literals (see maximum redundancy) (Note that an explanation is 100% redundant if all the features can be removed from it; this occurs when the classifier computes a constant function). The most interesting information is associated with the average redundancy, calculated for each dataset across all of its instances. While the worst average redundancy is demonstrated by ITI's decision trees, it still reaches $\approx$ 60% for IAI's decision trees and $\approx$ 50% for CN2's decision lists. This means that for the corresponding datasets, on average 60% (50%, respectively) of literals can be dropped from a default explanation offered by an IAI decision tree (CN2 decision list, respectively).

These experimental results serve as evidence confirming the lack of *practical* interpretability of what is believed to be the most interpretable ML models. Consequently, they also demonstrate the need for computing irredundant and provably correct formal abductive (but also contrastive) explanations if interpretability and transparency of the decisions made by these models is of concern. It should also be noted that despite the use of the SAT technology dealing with NP-hard problems, the runtime of the redundancy checks applied in the experiment does not exceed a small fraction of a second per instance and can be neglected, which demonstrates that the proposed formal explainability approach is ready for widespread practical deployment.

6.3. Additional classifiers

Although the experiments reported in this article consider fairly shallow DTs (i.e., with depths not exceeding six or eight), which suffice in terms of target accuracy, the methods proposed in this article can be shown to apply for much larger (and deeper) DTs. For example, recent work (Ghiasi et al., 2020) proposes the use of DTs for diagnosis of coronary artery disease. For one of the DTs proposed in Ghiasi et al. (2020) (see Ghiasi et al., 2020, Figure 2), the longest paths have 19 non-terminal nodes. Among these, for the path with prediction cad, manual inspection[7] reveals that at least 10 literals out of 19 (i.e., more than 50%) are redundant. Evidently, for a human decision maker, an explanation with nine literals (or less) is far easier to understand than an explanation with 19 literals.

7. Discussion

This article looks at so-called interpretable models from the perspective of explaining the predictions made. Explanations can serve to answer a WHY question, or alternatively a WHYNOT question. Recent work refers to the latter as abductive explanations and the latter as contrastive explanations (Ignatiev et al., 2019a, 2020a; Miller, 2019).

Because interpretable models are expected to serve themselves as the explanations (Rudin, 2019; Molnar, 2020), we focus on (manually) extracting answers to WHY questions from the models. In contrast, (manually) finding answers to WHYNOT questions is in general far less intuitive. In fact, Section 5.4 proves that it is computationally hard to answer WHYNOT questions for DLs. Clearly, all this but precludes human decision makers from attempting to answer WHYNOT questions for DLs.

Recent results (Izza et al., 2020, 2022a; Huang et al., 2021b) showed that, both in theory and in practice, decision trees exhibit explanation redundancy, i.e., if a path is used as the explanation for a WHY question, then that explanation exhibits redundancy when compared with a rigorous (logically-defined) explanation. More problematic, redundancy can grow arbitrarily large with path length.

The previous sections show that the same limitations occur with decision lists, and that decision sets exhibit other drawbacks that also serve to challenge its interpretability.

As shown by the experiments, the amount of redundancy in manually produced explanations, for DTs and DLs, can be significant. For many of the examples considered, the fraction of redundant literals exceeds 50%, i.e., more than one out of two literals in the explanation could be discarded, and that would not affect the correctness of the explanation.

Given the above, and as long as model comprehensibility is premised on succinctness, then neither decision trees, decision lists, or decision sets can be appropriate for (manually) answering WHY questions.

For WHYNOT questions, the situation is even more problematic. For DTs, CXp's (and so the answer to WHYNOT questions) can be computed in polynomial time, but such algorithms are beyond the reach of (the average) human decision makers. For DLs, it seems unrealistic to even ask a human decision maker to change a decision, since this problem is by itself computationally hard.

[7] Unfortunately, we have been unable to obtain from the authors this concrete DT in a format suitable for automated analysis.

Given the results in this article, we conclude that answers to WHY and WHYNOT questions (or alternatively the computation of (rigorous) AXp's and CXp's) should be obtained with dedicated algorithms, as proposed in recent work (see Marques-Silva and Ignatiev, 2022 and references therein).

8. Conclusion and research directions

For high-risk application domains, there has been recent interest in so-called interpretable ML models (Lakkaraju et al., 2016; Rudin, 2019, 2022; Molnar, 2020). This article proposes model comprehensibility as a measure of the understanding of ML model predictions by human decision makers. Model comprehensibility aims at finding explanations, i.e., answers to WHY and WHYNOT questions, which are both correct and irreducible. The motivation is that, for interpretable models, one would expect predictions to be comprehensible by human decision makers. As argued in this article, it is hardly the case that existing interpretable models can be deemed to enable model comprehensibility. Hence, even though there are a number of valid reasons to deploy interpretable models in high-risk domains, the

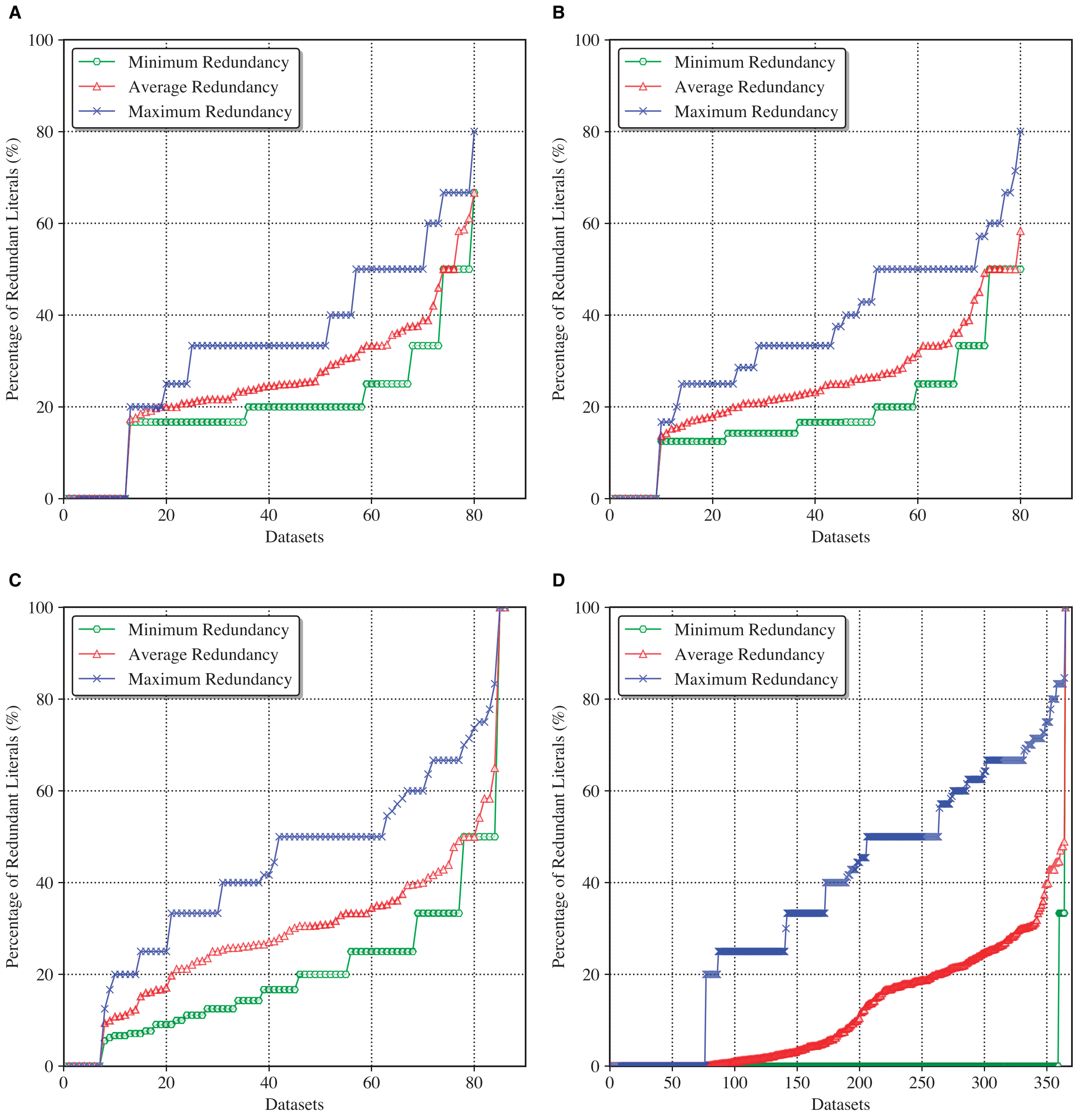


FIGURE 2
Minimum, average, and maximum redundancy of default DT and DL explanations. **(A)** Redundancy of DT/IAI6 explanations. **(B)** Redundancy of DT/IAI8 explanations. **(C)** Redundancy of DT/ITI explanations. **(D)** Redundancy of DL/CN2 explanations.

ability to find correct and irreducible explanations, by manual inspection, is not among them. Despite the fact that one can identify general rules that allow for a human decision maker to find correct explanations by inspection, it is also the case that such explanations can in general be arbitrarily redundant on the number of features. The solution for this limitation of interpretable models is, as it is also the case with non-interpretable models, to compute rigorous explanations. Moreover, it is the case that rigorous explanations can be (very) efficiently computed for both decision trees and decision lists.

Furthermore, and although decision sets can also be easily explained in practice (Ignatiev and Marques-Silva, 2021), it is also the case that most publicly available solutions for the creation of decision sets exhibit a number of crucial drawbacks. One example is prediction overlap; another is the need to use a default rule for when no other rule fires. Both drawbacks represent critical limitations to model comprehensibility.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: https://blog.fastforwardlabs.com/2017/03/09/fairml-auditing-black-box-predictive-models.html; https://github.com/EpistasisLab/pmlb; https://archive.ics.uci.edu/ml.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was supported by the AI Interdisciplinary Institute ANITI, funded by the French program Investing for the Future–PIA3 under Grant agreement no. ANR-19-PI3A-0004, and by the H2020-ICT38 project COALA COgnitive Assisted agile manufacturing for a LAbor force supported by trustworthy Artificial intelligence.

Acknowledgments

The work summarized in this document results in part from collaborations and discussions with several colleagues, including N. Asher, M. Cooper, X. Huang, and Y. Izza, among others.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Amgoud, L. (2021). "Non-monotonic explanation functions," in *ECSQARU* (Prague), 19–31.

Amgoud, L., and Ben-Naim, J. (2022). "Axiomatic foundations of explainability," in *IJCAI* (Vienna), 636–642.

Arenas, M., Baez, D., Barceló, P., Pérez, J., and Subercaseaux, B. (2021). "Foundations of symbolic languages for model interpretability," in *NeurIPS*, 11690–11701.

Arenas, M., Barceló, P., Romero, M., and Subercaseaux, B. (2022). On computing probabilistic explanations for decision trees. *CoRR*, abs/2207.12213. doi: 10.48550/arXiv.2207.12213

Asher, N., Paul, S., and Russell, C. (2021). "Fair and adequate explanations," in *CD-MAKE*, 79–97.

Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., and Marquis, P. (2021). "On the computational intelligibility of boolean classifiers," in *KR*, 74–86.

Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., and Marquis, P. (2022a). "On preferred abductive explanations for decision trees and random forests," in *IJCAI* (Vienna), 643–650.

Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., and Marquis, P. (2022b). "Trading complexity for sparsity in random forest explanations," in *AAAI*, 5461–5469.

Audemard, G., Koriche, F., and Marquis, P. (2020). "On tractable XAI queries based on compiled representations," in *KR* (Rhodes), 838–849.

Bengio, Y., LeCun, Y., and Hinton, G. E. (2021). Deep learning for AI. *Commun. ACM* 64, 58–65. doi: 10.1145/3448250

Bertsimas, D., and Dunn, J. (2017). Optimal classification trees. *Mach. Learn*. 106, 1039–1082. doi: 10.1007/s10994-017-5633-9

Biere, A., Heule, M., van Maaren, H., and Walsh, T. (Eds.). (2021). *Handbook of Satisfiability*. New York, NY: IOS Press.

Blanc, G., Koch, C., Lange, J., and Tan, L. (2022a). "The query complexity of certification," in *STOC* (Rome), 623–636.

Blanc, G., Koch, C., Lange, J., and Tan, L. (2022b). "A query-optimal algorithm for finding counterfactuals," in *ICML* (Baltimore, MD), 2075–2090.

Blanc, G., Lange, J., and Tan, L. (2021). "Provably efficient, succinct, and precise explanations," in *NeurIPS*.

Boumazouza, R., Alili, F. C., Mazure, B., and Tabia, K. (2020). "A symbolic approach for counterfactual explanations," in *SUM* (Bozen-Bolzano), 270–277.

Boumazouza, R., Alili, F. C., Mazure, B., and Tabia, K. (2021). "ASTERYX: a model-agnostic sat-based approach for symbolic and score-based explanations," in *CIKM*, 120–129.

Clark, P., and Boswell, R. (1991). "Rule induction with CN2: some recent improvements," in *EWSL* (Porto), 151–163.

Clark, P., and Niblett, T. (1989). The CN2 induction algorithm. *Mach. Learn*. 3, 261–283. doi: 10.1007/BF00116835

Cook, S. A. (1971). "The complexity of theorem-proving procedures," in *STOC*, eds M. A. Harrison, R. B. Banerji, and J. D. Ullman (Shaker Heights, OH), 151–158.

Cooper, M. C., and Marques-Silva, J. (2021). "On the tractability of explaining decisions of classifiers," in *CP*, ed L. D. Michel (Montpellier), 1–21.

Darwiche, A. (2020). "Three modern roles for logic in AI," in *PODS* (Portland, OR), 229–243.

Darwiche, A., and Hirth, A. (2020). "On the reasons behind decisions," in *ECAI* (Santiago de Compostela), 712–720.

Darwiche, A., and Hirth, A. (2022). On the (complete) reasons behind decisions. *J. Logic Lang. Inf.* 2022, 1–26. doi: 10.1007/s10849-022-09377-8

Darwiche, A., and Marquis, P. (2021). On quantifying literals in boolean logic and its applications to explainable AI. *J. Artif. Intell. Res.* 2021, 12756. doi: 10.1613/jair.1.12756

Fair,ML. (2016). *Auditing Black-Box Predictive Models*. Available online at: https://blog.fastforwardlabs.com/2017/03/09/fairml-auditing-black-box-predictive-models.html

Ferreira, J., de Sousa Ribeiro, M., Gonçalves, R., and Leite, J. (2022). "Looking inside the black-box: logic-based explanations for neural networks," in *KR* (Haifa), 432–442.

Flach, P. A. (2012). *Machine Learning - The Art and Science of Algorithms that Make Sense of Data*. CUP.

Friedler, S., Scheidegger, C., and Venkatasubramanian, S. (2015). *On Algorithmic Fairness, Discrimination and Disparate Impact*. Available online at: http://fairness.haverford.edu/

Ghiasi, M. M., Zendehboudi, S., and Mohsenipour, A. A. (2020). Decision tree-based diagnosis of coronary artery disease: CART model. *Comput. Methods Programs Biomed.* 192, 105400. doi: 10.1016/j.cmpb.2020.105400

Goodfellow, I. J., Bengio, Y., and Courville, A. C. (2016). *Deep Learning. Adaptive Computation and Machine Learning*. Cambridge, MA: MIT Press.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2020). Generative adversarial networks. *Commun. ACM* 63, 139–144. doi: 10.1145/3422622

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). "Explaining and harnessing adversarial examples," in *ICLR* (San Diego, CA).

Gorji, N., and Rubin, S. (2022). "Sufficient reasons for classifier decisions in the presence of domain constraints," in *AAAI*.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2019). A survey of methods for explaining black box models. *ACM Comput. Surv.* 93, 1–93. doi: 10.1145/3236009

Gunning, D., and Aha, D. W. (2019). DARPA's explainable artificial intelligence (XAI) program. *AI Mag.* 40, 44–58. doi: 10.1145/3301275.3308446

Hu, X., Rudin, C., and Seltzer, M. I. (2019). "Optimal sparse decision trees," in *NeurIPS* (Vancouver, BC), 7265–7273.

Huang, X., Izza, Y., Ignatiev, A., Cooper, M. C., Asher, N., and Marques-Silva, J. (2021a). Efficient explanations for knowledge compilation languages. *CoRR*, abs/2107.01654. doi: 10.48550/arXiv.2107.01654

Huang, X., Izza, Y., Ignatiev, A., Cooper, M. C., Asher, N., and Marques-Silva, J. (2022). "Tractable explanations for d-DNNF classifiers," in *AAAI*, 5719–5728.

Huang, X., Izza, Y., Ignatiev, A., and Marques-Silva, J. (2021b). "On efficiently explaining graph-based classifiers," in *KR*, 356–367.

Huang, X., and Marques-Silva, J. (2022). On deciding feature membership in explanations of SDD and related classifiers. *CoRR*, abs/2202.07553. doi: 10.48550/arXiv.2202.07553

IAI (2020). *Interpretable AI*. Availabel online at: https://www.interpretable.ai/

Ignatiev, A., Izza, Y., Stuckey, P. J., and Marques-Silva, J. (2022). "Using MaxSAT for efficient explanations of tree ensembles," in *AAAI*, 3776–3785.

Ignatiev, A., and Marques-Silva, J. (2021). "SAT-based rigorous explanations for decision lists," in *SAT* (Barcelona), 251–269.

Ignatiev, A., Narodytska, N., Asher, N., and Marques-Silva, J. (2020a). "From contrastive to abductive explanations and back again," in *AIxIA*, 335–355.

Ignatiev, A., Narodytska, N., Asher, N., and Marques-Silva, J. (2020b). On relating 'why?' and 'why not?' explanations. *CoRR*, abs/2012.11067. doi: 10.48550/arXiv.2012.11067

Ignatiev, A., Narodytska, N., and Marques-Silva, J. (2019a). "Abduction-based explanations for machine learning models," in *AAAI* (Honolulu, HI), 1511–1519.

Ignatiev, A., Narodytska, N., and Marques-Silva, J. (2019b). "On relating explanations and adversarial examples," in *NeurIPS* (Vancouver, BC), 15857–15867.

Ignatiev, A., Narodytska, N., and Marques-Silva, J. (2019c). On validating, repairing and refining heuristic ML explanations. *CoRR*, abs/1907.02509. doi: 10.48550/arXiv.1907.02509

Ignatiev, A., Pereira, F., Narodytska, N., and Marques-Silva, J. (2018). "A SAT-based approach to learn explainable decision sets," in *IJCAR* (Oxford, UK), 627–645.

ITI (2020). *Incremental Decision Tree Induction*. Available online at: https://www-lrn.cs.umass.edu/iti/

Izza, Y., Ignatiev, A., and Marques-Silva, J. (2020). On explaining decision trees. *CoRR*, abs/2010.11034. doi: 10.48550/arXiv.2010.1103

Izza, Y., Ignatiev, A., and Marques-Silva, J. (2022a). On tackling explanation redundancy in decision trees. *J. Artif. Intell. Res.* 75, 261–321. doi: 10.1613/jair.1.13575

Izza, Y., Ignatiev, A., Narodytska, N., Cooper, M. C., and Marques-Silva, J. (2021). Efficient explanations with relevant sets. *CoRR*, abs/2106.00546. doi: 10.48550/arXiv.2106.00546

Izza, Y., Ignatiev, A., Narodytska, N., Cooper, M. C., and Marques-Silva, J. (2022b). Provably precise, succinct and efficient explanations for decision trees. *CoRR*, abs/2205.09569. doi: 10.48550/arXiv.2205.09569

Izza, Y., and Marques-Silva, J. (2021). "On explaining random forests with SAT," in *IJCAI* (Montreal, QC), 2584–2591.

Izza, Y., and Marques-Silva, J. (2022). On computing relevant features for explaining NBCs. *CoRR*, abs/2207.04748. doi: 10.48550/arXiv.2207.04748

Karimi, A., Barthe, G., Schölkopf, B., and Valera, I. (2020). A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *CoRR*, abs/2010.04050. doi: 10.48550/arXiv.2010.04050

Karimi, A., Schölkopf, B., and Valera, I. (2021). "Algorithmic recourse: from counterfactual explanations to interventions," in *FAccT* (Toronto, ON), 353–362.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Lakkaraju, H., Bach, S. H., and Leskovec, J. (2016). "Interpretable decision sets: a joint framework for description and prediction," in *KDD* (San Francisco, CA), 1675–1684.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Lipton, Z. C. (2018). The mythos of model interpretability. *Commun. ACM* 61, 36–43. doi: 10.1145/3233231

Liu, X., and Lorini, E. (2021). "A logic for binary classifiers and their explanation," in *CLAR* (Hangzhou).

Liu, X., and Lorini, E. (2022a). "A logic of "black box" classifier systems," in *WoLLIC* (Iasi), 158–174.

Liu, X., and Lorini, E. (2022b). A logic of "black box" classifier systems. *CoRR*, abs/2210.07161. doi: 10.1007/978-3-031-15298-6_10

Lundberg, S. M., and Lee, S. (2017). "A unified approach to interpreting model predictions," in *NeurIPS* (Long Beach, CA), 4765–4774.

Malfa, E. L., Michelmore, R., Zbrzezny, A. M., Paoletti, N., and Kwiatkowska, M. (2021). "On guaranteed optimal robust explanations for NLP models," in *IJCAI* (Montreal, QC), 2658–2665.

Marques-Silva, J. (2022). Logic-based explainability in machine learning. *CoRR*, abs/2211.00541. doi: 10.48550/arXiv.2211.00541

Marques-Silva, J., Gerspacher, T., Cooper, M. C., Ignatiev, A., and Narodytska, N. (2020). "Explaining naive bayes and other linear classifiers with polynomial time and delay," in *NeurIPS* (Vancouver, BC).

Marques-Silva, J., Gerspacher, T., Cooper, M. C., Ignatiev, A., and Narodytska, N. (2021). "Explanations for monotonic classifiers," in *ICML*, 7469–7479.

Marques-Silva, J., and Ignatiev, A. (2022). "Delivering trustworthy AI through formal XAI," in *AAAI*, 12342–12350.

Marques-Silva, J., and Mencía, C. (2020). "Reasoning about inconsistent formulas," in *IJCAI*, 4899–4906.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.* 63, 81–97. doi: 10.1037/h0043158

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* 267, 1–38. doi: 10.1016/j.artint.2018.07.007

Molnar, C. (2020). *Interpretable Machine Learning. Leanpub*. Available online at: http://tiny.cc/6c76tz

Narodytska, N., Shrotri, A. A., Meel, K. S., Ignatiev, A., and Marques-Silva, J. (2019). "Assessing heuristic machine learning explanations with model counting," in *SAT* (Lisbon), 267–278.

Penn, M. L. (2020). *Penn Machine Learning Benchmarks*. Available online at: https://github.com/EpistasisLab/pmlb

Quinlan, J. R. (1987). "Generating production rules from decision trees," in *IJCAI* (Milan), 304–307.

Rago, A., Cocarascu, O., Bechlivanidis, C., Lagnado, D. A., and Toni, F. (2021). Argumentative explanations for interactive recommendations. *Artif. Intell.* 296, 103506. doi: 10.1016/j.artint.2021.103506

Rago, A., Cocarascu, O., Bechlivanidis, C., and Toni, F. (2020). "Argumentation as a framework for interactive explanations for recommendations," in *KR* (Rhodes), 805–815.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artif. Intell.* 32, 57–95. doi: 10.1016/0004-3702(87)90062-2

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). ""why should I trust you?": explaining the predictions of any classifier," in *KDD* (San Francisco, CA), 1135–1144.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). "Anchors: high-precision model-agnostic explanations," in *AAAI* (New Orleans, LA), 1527–1535.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* 1, 206–215. doi: 10.1038/s42256-019-0048-x

Rudin, C. (2022). Why black box machine learning should be avoided for high-stakes decisions, in brief. *Nat. Rev. Methods Primers* 2, 1–2. doi: 10.1038/s43586-022-00172-0

Shi, W., Shih, A., Darwiche, A., and Choi, A. (2020). "On tractable representations of binary neural networks," in *KR* (Rhodes), 882–892.

Shih, A., Choi, A., and Darwiche, A. (2018). "A symbolic approach to explaining bayesian network classifiers," in *IJCAI* (Stockholm), 5103–5111.

Shih, A., Choi, A., and Darwiche, A. (2019). "Compiling bayesian network classifiers into decision graphs," in *AAAI* (Honolulu, HI), 7966–7974.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., et al. (2014). "Intriguing properties of neural networks," in *ICLR* (Banff, AB).

UCI (2020). *UCI Machine Learning Repository*. Available online at: https://archive.ics.uci.edu/ml

Ustun, B., Spangher, A., and Liu, Y. (2019). "Actionable recourse in linear classification," in *FAT* (Atlanta, GA), 10–19.

Utgoff, P. E., Berkman, N. C., and Clouse, J. A. (1997). Decision tree induction based on efficient tree restructuring. *Mach. Learn.* 29, 5–44. doi: 10.1023/A:1007413323501

Venkatasubramanian, S., and Alfano, M. (2020). "The philosophical basis of algorithmic recourse," in *FAT* (Barcelona), 284–293.

Wäldchen, S. (2022). *Towards Explainable Artificial Intelligence-Interpreting Neural Network Classifiers with Probabilistic Prime Implicants* (Ph.D. thesis). Technischen Universität Berlin.

Wäldchen, S., MacDonald, J., Hauch, S., and Kutyniok, G. (2021). The computational complexity of understanding binary classifier decisions. *J. Artif. Intell. Res.* 70, 351–387. doi: 10.1613/jair.1.12359

Wolf, L., Galanti, T., and Hazan, T. (2019). "A formal approach to explainability," in *AIES* (Honolulu, HI), 255–261.

Yu, J., Ignatiev, A., Stuckey, P. J., Narodytska, N., and Marques-Silva, J. (2022). Eliminating the impossible, whatever remains must be true. *CoRR*, abs/2206.09551.