Predictive models & reasoning with explanations

Ryma Boumazouza

UNIVERSITÉ D'ARTOIS

ÉCOLE DOCTORALE SCIENCES TECHNOLOGIES ET SANTÉ N°585

Predictive models & reasoning with explanations

DOCTORAL THESIS

presented and publicly defended on 08 December 2022

in partial fulfillment of requirements for the degree of

Doctor of Philosophy of Artois University

in Computer Science

by

Ryma BOUMAZOUZA

Doctoral Committee

Advisors:	Bertrand MAZURE	Professor, Artois University
	Karim TABIA	Assoc. Professor, Artois University
Supervisor:	Fahima CHEIKH-ALILI	Assoc. Professor, Artois University
Chair:	Sylvain LAGRUE	Professor, Compiègne University
Reviewer:	Marie-Jeanne LESOT	Assoc. Professor, Sorbonne University
Examiner:	Christine SOLNON	Professor, INSA Lyon

CENTRE DE RECHERCHE EN INFORMATIQUE DE LENS – CNRS UMR 8188
Université d'Artois, rue Jean Souvraz, S.P. 18 F-62307, Lens Cedex France
Secrétariat : +33 (0)3 21 79 17 23
http://www.cril.univ-artois.fr

Mise en page avec memcril (B. Mazure, CRIL) et thloria (D. Roegel, LORIA).

Acknowledgements

This doctoral journey has been quite incredible despite the several ups and downs including the enormous pressure during the pandemic. Although it is often said that to pursue a thesis is a lonesome journey, that was not the case with this one. I am grateful to all of those with whom I have had the pleasure to spend this journey.

This work would not have been possible without the financiel support of the Région Hauts-de-France and the Artois University. I would like to thank them along with the CRIL laboratory for supporting this thesis.

I take this moment to express my gratitude to my supervisors: starting with my co-thesis director Mr. Karim TABIA, who have been supportive and pushed me to learn by doing. My supervisor Mme. Fahima CHEIKH-ALILI has been quite encouraging and supportive throughout this journey, and I must thank her for the pleasing conversations on both scientific research and life in general. My thesis director Mr. Bertrand MAZURE has provided me with his thoughtful advice and pitched in at important times.

I am grateful to each of the members of my Dissertation Committee for evaluating my thesis work. I would like to begin by thanking Mme. Marie-Jeanne LESOT and Sylvain LAGRUE for reviewing my thesis. I would also like to thank Mme. Christine SOLNON for examining my thesis. I am very thankful to all of you for your insightful comments and thought-provoking questions during the defense session.

During this thesis, as I moved to a foreign country, the experience has been enlightening. I am thankful to each one of my friends and colleagues at CRIL laboratory for the in-depth discussion on various topics and the light moments during the coffee and tea breaks. I cherish to have known Sara, Marie, Alexis, Thanh, Hugues, Anasse, Ikram, Yazid, Chouaib, Thibault, Karen, Antony and Romain.

Lastly, I would be remiss in not mentioning or not thanking the members of my family and loved ones : Hiba, Sidahmed, Wafa, Ines, Azwaw and Maïa. I could not have undertaken the pursuit of this project without them. I am especially thankful to my parents, whose love and guidance are with me whatever I pursue and whose support and proud were always a motivation for me.

I would like to dedicate this thesis in memory of my loved grandmother who always prayed for me and to my loved grandfather who always encouraged me to work hard and live with dignity and pride.

I dedicate this thesis
to my family.

Contents

List of Figures

General introduction

Context and motivations

As our daily environment increasingly incorporates artificial intelligence (AI), it has become important to understand how models make decisions, particularly in a context where user confidence in the AI technology has become a societal issue. The explainability field became a hot topic and has grown along with the rapid rise of « black-box » machine learning techniques such as deep learning and the start of DARPA's eXplainable Artificial Intelligence (XAI) program[GA19] in 2016. Especially with the recent regulations of the European General Data Protection Regulation (GDPR) that came into effect in 2018, which requires providing explanations to users both for legal and ethical reasons. For instance, the Article 22 imposing a right to "information about the logic involved" forces the explanations to represent the inner logic of a system and need to be faithful to it. This is particularly important in high stakes decision making settings such as medical decision support system, military and security applications. To comply with all those regulations and needs, several approaches were proposed to explain the decision function of a classifier or explain predictions individually. The aim of such XAI methods is to provide in addition to a prediction, interpretable and useful information that justifies and explains a prediction.

Despite the rapid growth in attention on eXplainable AI, most of the current explainability methods for black-box models are ad-hoc. Namely, the feature relevance techniques that are among the most popular approaches and that have received much attention in the machine learning literature face critical issues that prevent their deployment. Indeed, much of these techniques have focused on estimating importance score, for how much a given input feature contributes to a model's output by the means of non-deterministic components. For instance, LIME[RSG16] (Local Interpretable Model-agnostic Explanations) uses random perturbation during the sampling process which results in shifts in data and instability in the generated explanations. Yet, in a context where XAI methods are expected to generate robust and stable explanations (i.e same explanations given the same instance and model with the same configuration), there is no guarantee that the explanations generated using ad-hoc methods are not accurate nor sufficient and they may even suffer from instability (i.e different explanations can be generated for the same prediction) due to random perturbations often used in this type of methods. These are critical issues that can prevent deployment of such methods in sensitive domains and limit their use. On the other hand, there exist formal methods to generate rigorous explanations for different machine learning models. However, the main bottleneck of such methods is their efficiency in computing explanations when it comes to large input feature sets. In addition, most of the current formal methods use information about the model and deal with its inner working to interpret the results, reducing the type of predictor to use to a set of interpretable models, that relatively enjoy a less good predictive performance compared to some black-box models.

The challenge that this thesis attempts to address is to understand what is happening beneath the surface and explain how black-boxes make decisions, and thus by integrating concepts and formalism of symbolic AI. We try to leverage formal methods and logical reasoning to develop a novel local model-agnostic interpretability approach for explaining the prediction of black-box classifiers, for both multi-

class and multi-label settings. Our contributions include the proposal of an explainability method that combines both the "agnostic" nature of numerical methods and offers more "rigorous" explanations that characterize symbolic explanations. We use propositional logic as a formalization framework to associate some logical representation (encoding) to our machine learning model. Thus, we reduce the given problem of explaining individual predictions to a variant of the propositional Boolean satisfiability problem (SAT). We use SAT solvers as the problem solving engine without implementing dedicated programs. This makes our approach efficient since SAT solvers are extremely well studied. Note that SAT framework is just one possibility and that other constraint solvers (e.g Satisfiability Modulo theories (SMT), Mixed-integer linear programming (MILP), etc) can be used for a similar process. Therefore, our approach is centered around Minimal Unsatisfiable Subsets (MUS) and Minimal Correction Subsets (MCS) and provides a comprehensive solution for feature importance on both explanations and variables level.

We present an approach based on modeling the problem in the form of variants of the propositional satisfiability problem (SAT) in order to take advantage of the strengths of already existing and proven SAT technologies, and of the powerful practical tools for the generation of MCSes ans MUSes. Then, we propose to overcome the complexity of encoding a machine learning classifier into an equivalent logical representation by the means of a surrogate model. This latter will be used to symbolically approximate the original model in the vicinity of the sample of interest, allowing it to be locally faithful [RSG16]. We propose two complementary types of symbolic explanations which are *sufficient reasons* and *counterfactuals*. Given an input data item and its prediction, a *sufficient reason* would answer the question : "*What are the feature values in the input which are sufficient in order to trigger the prediction whatever the values of the other variables?*" while a *counterfactual* explanation would answer the question : "*Which values are sufficient to change in the input to have a different prediction?*". This type of questions is fundamental for the end-user's understanding and for the explanations to be usable. The adequate type of explanation is directly linked with the end user. For instance, what may be important for the doctors in a XAI method is to know if they can act (need an actionable explanation) and not only understand how the algorithm works. From another perspective, domain experts may require to observe the effects and the results which will provide an intuition about the data that can directly help them to understand the decision process and how the model works.

As for the multi-label setting, we propose semantics and reasoning approaches to infer from first level explanations provided by the base classifier explanations the multi-label explanations. We adapt our agnostic and declarative approach to provide different types of symbolic explanations for multi-label classifiers. The explanations proposed are distinguished according to the associated semantics (sufficient reasons or counterfactuals), the elements composing an explanation (features, labels or a combination of the two) and the level of granularity of the explanations (the whole prediction or parts of the prediction). We will also be particularly interested in so-called feature-attribution explanations. We propose three schemes to achieve feature attribution for multi-label tasks using existing attribution methods as oracles, namely i) an aggregation-based scheme, ii) a problem transformation-based scheme and iii) a symbolic explanation-based one. A property we call label-explanation correlation, specific to multi-label classification is used in addition to the extension of the basic properties of sensitivity and stability to the multi-label setting.

Organization of the manuscript

This manuscript is composed of three parts. The first one is devoted to the state-of-the-art and the second and third parts bring together the contributions of this thesis.

The **Background and notations** section is a reminder of the basic notions of classification, propositional logic and Boolean satisfiability problem that constitute the technical preliminaries necessary to the understanding of the rest of the manuscript. In the first part, **Chapter 1** is devoted to explainability in AI. We present the eXplainable Artificial Intelligence field, the need to explanation and the different purposes and audiences of an XAI method. We present a review of the different explainability methods proposed by the community. We rely on a taxonomy that depends on what the methods explain (intrinsic/post-hoc) and how they are applied (local/global) in order to distinguish the methods of explainability. We present within the **Chapter 2** of the first part two main methodologies of the XAI methods : Ad-hoc methods and formal based ones.

The contributions are presented within the second and third part of this thesis. **Chapter 3** is dedicated to the symbolic explanations for single-label classification. We define the different types of symbolic explanations we propose to enumerate and we present a general framework of our **declarative** and **model-agnostic** approach. This latter allows to provide *sufficient reason* and *counterfactual* explanations based on SAT technologies.

Chapter 4 describes our contributions to provide different types of symbolic explanations for multi-label classifiers, where a prediction is a subset of labels. More precisely, in addition to global *sufficient reason* and *counterfactual* explanations, our approach makes it possible to generate explanations at different levels of granularity in addition to structural relationships between labels (provides both feature and label-based explanations).

Chapter 5 and **6** are dedicated to the feature-attribution explanations. We describe in **Chapter 5** how we associate scores reflecting the relevance of the explanations and the features w.r.t some properties, in order to explain individual predictions of single-label classifiers. We present a set of fine-grained properties allowing to analyze, rank and select explanations and a set of scores allowing to assess the relevance of explanations and features w.r.t the suggested properties. In **Chapter 6**, we propose three schemes to achieve feature attribution for multi-label tasks using existing attribution methods as oracles. The first scheme is based on aggregating feature attribution scores obtained for each label while the second one is based on problem transformation. The third scheme combines problem transformation with symbolic explanations. In order to evaluate the quality of feature attributions, we extend the basic properties of sensitivity and stability to the multi-label setting and propose a third property, called label-explanation correlation, specific to multi-label classification. Moreover, we propose to exploit the correlations between labels in order to infer feature attributions from already computed explanations.

Publications

→ R. Boumazouza, F. Cheikh-Alili, B. Mazure, K. Tabia, A Symbolic Approach for counterfactual explanations in the 14th International Conference on Scalable Uncertainty Management (SUM20), September 2020.

→ R. Boumazouza, F. Cheikh-Alili, B. Mazure, K. Tabia, A Model-Agnostic SAT-based approach for Symbolic Explanation Enumeration in the 23rd International Conference on Artificial Intelligence (ICAI'21), July 2021.

→ R. Boumazouza, F. Cheikh-Alili, B. Mazure, K. Tabia, ASTERYX: a model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations in the 30th ACM International Conference on Information and Knowledge Management (CIKM), November 2021. (Core rank A)

Background and notations

Machine Learning (ML) is a sub field of AI that is specifically focused on learning a model from data without being explicitly programmed. Machine learning can be supervised (relies on labelled data) or unsupervised (processes unlabelled data). The first one fits the data while the second one separates and explores the data. Supervised learning deals with two main tasks that are regression and classification. In this thesis, we deal with classification where we propose an approach to explain individual predictions of classifiers.

1 Classification problems

In this thesis, we want to explain supervised Machine Learning (ML) models trained for classification problems. We define in the following what is a classifier and present the different types of classification. We provide some details about single-label classification problems in section 1.1 and multi-label classification problems in section 1.2.

In supervised learning, a structured dataset for classification is defined as set of instances of the form (x, y) where $x = (x_1, ..., x_n)$, $n \in N$, is an input vector called features (or variables, attributes) and y is an outcome variable often called the label (or class variable or target). We denote by X the feature space and Y the outcome space. We use the notation $\{,\}$ to denote the domain of a discrete variable and $|\ .\ |$ to denote its size. Classification is a supervised machine learning task whose aim is to predict the class (output) variable based on the values of the input variables. In [dCF09], it is defined as the process of approximating the mapping function that associates input samples to corresponding target classes (labels).

Definition 1. *(Decision function) A decision function of a classifier is a function $f : X \rightarrow Y$ mapping each instantiation x of X to y=$f(x)$.*

A decision function describes the classifier's behavior to perform classification.
We distinguish three main types of classification problems :

- Binary classification: This task involves predicting one of two classes ($|Y|= 2$). A binary classifier is defined as $f : X \rightarrow \{0, 1\}$.
- Multi-class classification: This task deals with predicting one class among a set of classes with $|Y|> 2$. A multi-class classifier is defined as $f : X \rightarrow Y$ where $|Y|> 2$.
- Multi-label classification: This task involves predicting a subset of labels for each instance. Contrary to binary and multi-class classification, classes are not mutually exclusive in multi-label tasks. A multi-label classifier is defined as follows : $f : X \rightarrow Y = (Y_1, \ldots, Y_k)$ where Y is a vector of binary variables (Y_i=1 denotes the fact that the label l_i is predicted positively).

1.1 Single-label classification

Single-label classification is concerned with learning from a set of examples that are associated with a single class y from a set of disjoint labels (classes are mutually exclusive). It covers binary and multi-class classification ($|Y| > 2$). Among the main applications where binary classification is used, there is disease diagnosis [GLB+10, PGA08], spam and malware detection [SKG09]. Multi-class classification is also used in different applications such as character recognition [AJM12, CVKRBA12], biometric identification [TYRW14], computer security [RSE16], etc.

Several multi-class classification techniques exist such as multi-layer perceptron (MLP)[Hay94], decision trees (DT)[WKQ+08], k-nearest neighbors (k-NN)[Alt92], support vector machines (SVMs)[CV95], naïve bayes classifier (BNC)[HY01]. The choice of the learning algorithm to use should take into account the specific application requirements and many other factors such as the type and number of features, interpretability, accuracy, etc.

Evaluation measures There are several ways to measure a classifier accuracy and generalization quality. The most known for multi-class classification are accuracy, precision, and recall. The accuracy is a metric used to assess the performance of a model and it is used to measure how well the classifier predicts.

Confusion matrix A confusion matrix is a summary of the prediction results in a classification problem. Table 1 illustrates an example of a confusion matrix for the case of binary classification.

- True Positive (TP) : For correctly predicted positive values.
- True Negative (TN) : For correctly predicted negative values.
- False positive (FP) : For values incorrectly predicted as positive.
- False Negative (FN) : For values incorrectly predicted as negative.

		Actual class	
		Positive (P)	Negative (N)
Predicted class	Positive (P)	TP	FP
	Negative (N)	FN	TN

Table 1: Confusion matrix

Accuracy It is defined as the number of predictions a model correctly makes divided by the total number of predictions made.

$$Accuracy(acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision As the name suggests, precision determines how precise and correct the model is with respect to positive predictions (how many of them are actually positive).

$$Precision = \frac{TP}{TP + FN} \tag{2}$$

Recall Is defined as the proportion of positive predictions made among the total number of positive samples. It measures the model's ability to detect positive samples.

$$Recall = \frac{TP}{TP + FP} \tag{3}$$

1.2 Multi-label classification

In contrast to single-label classification in which instances can only belong to a single class, in multi-label classification instances are associated with a vector of (binary) variables $Y = (Y_1, \ldots, Y_k)$. Multi-label classification is very common in many real-world applications [TKV09]. It is a well-known predictive task in many domains such as text categorization (where each document can belong at the same time to several predefined topics. For example, a conference paper may at the same time be labeled as *Artificial intelligence* and *Philosophy*), object recognition in images, sentiment analysis, audio, text categorization, video categorization [BLSB04], bioinformatics, information retrieval, multimedia content annotation, web mining, protein function classification [EW01], music categorization [LO03] and so on.

A dataset in multi-label classification is a collection of couples (x,y) where x is an instance of X and y a vector of binary variables in Y encoding the true labels associated with x.

Definition 2 (Multi-label classifier)**.** *A multi-label classifier is a function mapping each input data instance x to a multi-label prediction y. Each input x is a vector of n values assigned to X. Each corresponding output is a vector y of k binary values assigned to Y. Given the prediction $y=f(x)$, the instance x is classified by f in the label Y_j if Y_j=1 in the prediction y.*

Example 1. *Table 2 represents an example of a multi-label dataset. Assume a multi-label text classification problem of studying toxic comments online. For the sake of simplicity, assume that each comment is described by a set of keywords. Using a binary bag-of-words representation, each comment will have a set of binary features where feature $X_i = 1$ (resp. $X_i = 0$) denotes that keyword X_i is present (resp. absent) in the comment. In this example, the feature space X is composed of five binary variables associated to five keywords. The set of labels Y is composed of seven classes : toxic, severe_toxic, obscene, threat, insult, identity_hate and none. A toxic comment might at the same time be about any of or none of them. The variable $Y_j = 1$ (resp. $Y_j = 0$) denotes the fact that the current comment is positively labelled i.e. $Y_j = 1$(resp. not labelled, $Y_j = 0$) in label Y_j.*

$X = (X_1, X_2, X_3, X_4, X_5)$	$Y = (Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7)$
0 1 0 1 1	1 1 0 1 0 0 0
1 0 1 1 0	0 1 0 0 0 1 1
1 0 0 1 1	0 0 0 1 1 0 1
0 0 0 1 1	1 0 0 0 1 0 1
...	...
1 0 0 0 1	1 0 0 1 1 0 1

Table 2: Example of a multi-label dataset.

Remark 1. *The label space $Y = (Y_1, .., Y_k)$ (interchangeably denoted as $L = \{l_1, .., l_k\}$) is consisting of k binary variables encoding the presence/absence of the k labels.*

As for categories of multi-label classifiers, there exist mainly three :

1. **Problem transformation approaches** where the multi-label classification problem is transformed into a set of multi-class classification or mono-label regression problems. Examples of methods in this category are Binary Relevance (BR) [LDB+12], Classifier Chains (CC) [RPHF09] and Label Powerset (LP) [TV07, RPH08, TK07].

2. **Method adaptation approaches** based on extending multi-class techniques to predict instead of one single class a set of relevant labels. Examples of this category are ML-kNN [ZZ05] and ML-C4.5 [CK01].

3. **Ensemble approaches** that combine ideas from the two first categories. RAndom k labEL sets (RAkEL) [TV07], Hierarchy Of multi-label classifiERs (HOMER) [TKV08], Ensemble of Classifier Chains (ECC) [RPHF09] and Ensemble of Binary Relevance (EBR) [RPHF11].

One of the characteristics of multi-label data that must be taken into consideration compared to the multi-class datasets is the density of labels. This characteristic defined as the average number of labels per dataset entry divided by the number of labels is very low in most multi-label datasets [BdSRM14] and can impact the multi-label learning. Another difference worth mentioning compared with the multi-class case is related to evaluation metrics used to assess the accuracy of multi-label techniques. Indeed, standard multi-class classification metrics (e.g. precision, recall) are no more enough and appropriate measures are specifically designed for this purpose (e.g. Hamming-Loss).

2 Propositional logic and Boolean satisfiability

Propositional logic (PL) is a framework for representing knowledge in a logical form. It is the simplest kind of logic and is also called Boolean logic. The term "proposition" refers to a statement which can be either true or false. The propositions are linked together with logical connectives such as **and**, **or** and **not**.

2.1 Syntax of propositional logic

The syntax of propositional logic defines the allowable propositions to be used to represent the knowledge. In what follows, we define the basic elements of the language (propositions and connectives) and the structuring rules.

A propositional language $\mathcal{L}$ is composed of :

- a set of propositional variables (atoms) noted $\mathcal{V}$;
- logical connectives (also called logical operators) $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$ respectively corresponding to the Negation, Disjunction, Conjunction, Implication and Logical equivalence;
- Boolean constants $\top$ for True (equivalent to 1) and $\bot$ for False (equivalent to 0);
- parenthesis;

Definition 3. *(Propositional variable) a propositional variable (also called atom) is a Boolean variable that takes the value true or false.*

An atomic proposition consists of a single proposition symbol (it contains no logical connectives) and it is the fundamental block from which more complex statements can be built.

Definition 4. *(Literal) A literal is either a propositional variable ℓ or its logical negation $\neg\ell$ (also called complement or negation of ℓ).*

Example 2. *Let a be a Boolean variable, a and $\neg a$ are respectively positive and negative literals.*

Definition 5. *(Formula) A formula is said to be a Well-Formed Formula (WFF) if it belongs to one of this set of formulae :*

- $\top$ *and* $\bot$ *are formulae;*
- *an atom a is a formula;*
- *if α is a formula then $\neg\alpha$ is a formula;*
- $\alpha \vee \beta$ *is a formula if α and β are formulae;*
- $\alpha \wedge \beta$ *is a formula if α and β are formulae;*
- $\alpha \Rightarrow \beta$ *is a formula if α and β are formulae;*
- $\alpha \Leftrightarrow \beta$ *is a formula if α and β are formulae;*

Example 3. *Let $\mathcal{V} = \{a, b, c, d\}$ be a set of propositional variables. An example of a well-formed formula can be $((a \vee b \vee \neg d) \wedge (a \vee c))$.*

We define in the next section how semantics are associated to formulae.

2.2 Semantics of propositional logic

Definition 6. *An interpretation μ of a propositional formula α is an application that assigns values from {0, 1} to every propositional variable. It is therefore a function :*

$$\mu : \mathcal{V}_\alpha \to \{0, 1\}$$

An interpretation μ satisfies a formula α iff μ satisfies all sub-formulae of α. We can determine the truth value of any formula using the usual semantics of logical operators.

Remark 2. *An interpretation can be complete (assigns a value to every propositional variable appearing in a formula α) or partial (assigns values to a subset of propositional variables appearing in a formula α).*

Remark 3. *An interpretation μ can be written in the form $\{v_1 \to b_1, \ldots, v_n \to b_n\}$ where $v_i \in \mathcal{V}$ is a propositional variable and, $b_i \in \{0, 1\}$ the Boolean value associated to it. It can also be represented in a more compact way using only the literals.*

Example 4. *Let $\mathcal{V} = \{a, b\}$ be a set of propositional variables. An example of an interpretation is $\mu = (a \to 1, b \to 0)$. It can also be written as $\mu = (a, \neg b)$.*

Definition 7. *(Model) An interpretation μ making a formula α true is called a model of α ($\mu(\alpha) = 1$).*

Similarly, an interpretation μ making a formula α false is called a counter-model of α ($\mu(\alpha) = 0$).

Example 5. *Let $P = \neg(a \wedge b) \Rightarrow \neg b$ be a formula. The interpretation $\mu = \{a, \neg b\}$ is a model of P while $\mu' = \{\neg a, b\}$ is a counter-model.*

Definition 8. *(Equivalent formulae) Two propositional formulas α and β are equivalent and we note $\alpha \equiv \beta$, when they take the same truth value for all the interpretations.*

Definition 9. *(Satisfiable formula) A formula α is satisfiable (consistent) if there is an interpretation that makes it true. In other terms, α has at least one model.*

Similarly, a formula α is unsatisfiable (inconsistent) if there is no model for it. A "Tautology" refers to a propositional formula that is always true. A "Contradiction" is a proposition formula that is always false.

2.3 Normal forms

We present within this section two normal forms used in propositional logic : Disjunctive Normal Form (DNF) and Conjunctive Normal Form (CNF).

Using the introduced definitions, we can define the following.

Definition 10. *(Disjunctive Normal Form (DNF)) A propositional formula is in disjunctive normal form (DNF) if it is a disjunction of conjunctive clauses. A conjunctive clause is a conjunction of literals.*

Definition 11. *(Conjunctive Normal Form (CNF)) A propositional formula is in conjunctive normal form (CNF) if it is a conjunction of disjunctive clauses. A disjunctive clause is a disjunction of literals.*

Example 6. *Here is an example of a CNF formula referred to as Σ. $\mathcal{T}_i$ refers to clauses at position i in Σ.*

$$\begin{array}{cccccccc} & \mathcal{T}_1 & & \mathcal{T}_2 & & \mathcal{T}_3 & & \mathcal{T}_4 \\ \Sigma = & (l_1) & \wedge & (\neg l_2) & \wedge & (l_1 \vee l_3) & \wedge & (\neg l_1 \vee l_2) \end{array}$$

Remark 4. *A clause composed of one literal is called a unit clause. A clause composed of n literals is called n-ary clause.*

It is possible to convert statements into a conjunctive normal form (CNF) that are written in another form, such as disjunctive normal form (DNF).

In our work, we used the Conjunctive Normal Form as the target representation. Namely, we take a binary machine learning model f, together with a data instance, and produce a propositional CNF formula Σ which has the same number of models as f and with respect to the number of the input variables. Such encoding is required in order to use the Boolean satisfiability solvers as the problem solving engine.

3 Boolean satisfiability problem

We present within this section different definitions and concepts of the Boolean satisfiability problem.

3.1 Boolean satisfiability problem

The Boolean satisfiability problem (also called SAT for short) is a decision problem consisting in determining whether the variables in a given Boolean formula can be assigned so that the formula evaluates to TRUE. SAT was the first problem to be shown to be NP-complete by Cook's theorem [Coo71] which is the basis of NP-completeness theory and the P = NP problem. Despite this worst-case complexity, recent SAT-solving algorithms are capable of solving problem instances involving tens of thousands of variables and formulas consisting of millions of symbols [OSC07] which is sufficient for many practical SAT problems.

Definition 12. ***(SAT : The Boolean satisfiability problem)*** *the Boolean satisfiability problem is the decision problem, which, given a CNF formula, determines whether there is an assignment of propositional variables that makes the formula true.*

Example 7. *The formula $(x_1 \wedge x_2) \vee \neg x_1$ where x_1 and x_2 are Boolean variables is satisfiable since if x_1 takes the value False, the formula evaluates to True.*

Using a SAT-solver, one can decide if a CNF formula is satisfiable and can enumerate its models. In case a formula is unstatisfiable, one may need to identify subsets of clauses that cause the inconsistency or identify parts of the clauses to relax in order to restore the consistency.

Definition 13. ***(MUS)*** *A Minimal Unsatisfiable Subset (MUS in short) is a minimal subset* Γ *of clauses of a CNF* Σ *such that* $\forall\ \alpha \in \Gamma$, $\Gamma \backslash \{\alpha\}$ *is satisfiable.*

Intuitively, it is enough to have the subset Γ so that the formula Σ becomes inconsistent. A MUS is a sufficient reason for inconsistency. An unsatisfiable CNF Σ can have many MUSes.

Example 8. *Let* Σ *be a CNF formula composed of six clauses* (α_i) *where* $\Sigma = \{\alpha_1 = (a \vee b), \alpha_2 = (\neg a \vee b), \alpha_3 = (a \vee \neg b), \alpha_4 = (\neg a \vee \neg b), \alpha_5 = (\neg b), \alpha_6 = (b)\}$. *The set of MUSes for* Σ *is the following:*

$$\text{MUSes}(\Sigma) \quad = \quad \{\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, \{\alpha_1, \alpha_2, \alpha_5\}, \{\alpha_3, \alpha_4, \alpha_6\}, \{\alpha_5, \alpha_6\}\}$$

Definition 14. ***(MSS)*** *A maximal satisfiable subset (in short, MSS)* Φ *of a CNF* Σ *is a subset of clauses* $\Phi \subseteq \Sigma$ *that is satisfiable and such that* $\forall\ \alpha \in \Sigma \setminus \Phi$, $\Phi \cup \{\alpha\}$ *is unsatisfiable.*

Definition 15. ***(MCS (Co-MSS))*** *A minimal correction subset (in short MCS, also called Co-MSS)* Ψ *of a CNF* Σ *is a set of formulas* $\Psi \subseteq \Sigma$ *whose complement in* Σ, *i.e.,* $\Sigma \setminus \Psi$, *is an MSS of* Σ.

Example 9. *Given the CNF formula from Example 8, the set of MCSes is the following:*

$$\text{MCSes}(\Sigma) \quad = \quad \{\{\alpha_1, \alpha_6\}, \{\alpha_2, \alpha_6\}, \{\alpha_3, \alpha_5\}, \{\alpha_4, \alpha_5\}\}$$

Intuitively, an MCS is subset of Σ that restores its satisfiability once removed and it is minimal (i.e. removing any smaller subset cannot restore the consistency). The enumeration of MUSes/MCSes is a well-known problem dealt with in many areas such as knowledge-base reparation. Recent years have witnessed the proposal of a large number of tools and novel algorithms for the extraction and enumeration MUSes/MCSes [GMP07, LS08, MPMS15, BK15, LPMMS16, BK16, MIPMS16, PMJMS18, NBMS18, BČB18].

A well-known minimal hitting set (MHS) relationship between MUSes and MCSes exists and is expressed as follows :

Proposition 1. *MCSes are MHSes of MUSes and vice-versa.*

This duality was originally presented in [Rei87] in the context of model-based diagnosis and was later investigated in [BL03] for propositional formulas in clausal form. Many of the proposed MUS enumeration algorithms are based on this duality between MUSes and MCSes. Given the set of MCSes, each MUS is an irreducible subset of the clauses that covers[1] all of these MCSes and vice versa.

The computational complexity of finding a MUS or an MCS are already established in the literature as they are well-studied problems. The extraction of a MUS is a problem of complexity FP^{NP} and checking whether there exists a MUS of size $\leq k$ is of complexity Σ_2^P-complete [Gup06, Lib05]. Computing the smallest MUS (SMUS) is in $FP^{\Sigma_2^P}$. Checking whether a subset of a CNF formula is an MCS is DP-complete[2] [CT95] and the computation of an MCS is a problem of complexity FP^{NP} [Chr94].

[1] A clause is said to cover an MCS (resp MUS) if it is included in the MCS (resp MUS).

[2] A problem $\mathcal{P}$ belongs to the class DP if it can be written as $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ with $\mathcal{P}_1 \in$ NP and $\mathcal{P}_2 \in$ coNP.

3.2 Partial Maximum Satisfiability problem

The maximum satisfiability problem (Max-SAT) is the problem of determining the maximum number of clauses, of a given Boolean formula in conjunctive normal form, that can be made true by an assignment of truth values to the variables of the formula. Formally, we have the following definitions :

Definition 16. *(**Max-SAT**) Given a Boolean CNF formula Σ, Max-SAT is the problem of finding a truth assignment that satisfies the maximum number of clauses in Σ.*

Example 10. *Let a set of propositional variables $\mathcal{V} = \{x_1, x_2, x_3, x_4\}$ and let a CNF formula $\Sigma = \{(\neg x_1 \vee \neg x_2), (\neg x_1 \vee x_3), (\neg x_1 \vee \neg x_3), (\neg x_2, x_4), (\neg x_2 \vee \neg x_4), (x_1), (x_2)\}$. An example of Max-SAT assignment that maximize satisfied clauses is ($x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 1$), where only 2 clauses are violated.*

The Partial Maximum Boolean Satisfiability (Partial Max-SAT or PMSAT) is an optimization variant of SAT problem defined as follows.

Definition 17. *(**Partial Max-SAT**) Given a Boolean CNF formula Σ in which some clauses are hard and some are soft, Partial Max-SAT is the problem of finding a truth assignment that satisfies all the hard constraints and the maximum number of soft ones.*

In order to solve Partial Max-SAT, we will consider the general setting where a formula is composed of two disjoint sets of clauses $\Sigma = \Sigma_H \cup \Sigma_S$ [BHvM09], where Σ_H denotes the hard clauses (which must be satisfied) and Σ_S denotes the soft ones (which may be relaxed).

Definition 18. *(**Hard and Soft clauses**) Let Σ_1 and Σ_2 be two sets of clauses where Σ_2 is satisfiable. Partial Max-SAT(Σ_1,Σ_2) computes one maximal subset of Σ_1 that is satisfiable with Σ_2. Σ_1 and Σ_2 are called the sets of soft and hard constraints (clauses), respectively.*

Example 11. *Using the same set of propositional variables $\mathcal{V}$ from Example 10, an example of a CNF formula (Σ) in Partial Max-SAT problem is : $\Sigma = \Sigma_H \cup \Sigma_S$ where $\Sigma_H = \{(x_1 \vee \neg x_2 \vee x_4), (\neg x_1 \vee \neg x_2 \vee x_3)\}$ and $\Sigma_S = \{(\neg x_2 \vee \neg x_4), (\neg x_3 \vee x_2), (x_1 \vee x_3)\}$. An example of Partial Max-SAT solution that satisfies all hard clauses and a maximum number of soft clauses is : ($x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 0$).*

Part I

State-of-the-art

Chapter 1

Explainable AI

Symbolic artificial intelligence, also known as "Good Old Fashioned Artificial Intelligence" (GOFAI) [Hau85] started as philosophers' attempts to describe human thinking as a symbolic system and was often the dominant paradigm in the AI community until the late 1980s. The author in [Wal12] claims that the "classical" or "symbolic" approach comes from the idea that mind and cognition can be understood in broadly mechanical specifically computational terms, giving it an explanatory power. For a long time, symbolic AI, which involves humans in the learning process, took precedence over machine learning AI considered as opaque and incomprehensible to humans. Indeed, symbolic AI systems built of nested if-then statements that allow conclusions to be drawn are considered human-readable. Unlike machine learning from data without being assisted by human beings (abstract and higher-order concepts).

From the 2010s', machine learning suddenly started exploding because of the availability of huge amounts of digital data and of powerful hardware resources (e.g. GPUs) which caused shifts from a knowledge-driven approach to a data-driven approach (IBM's Deep Blue beats the world champion at chess in 1997, ImageNet [KSH12], Deep learning [LBH15], Deepface [TYRW14] involving more than 120 million parameters, etc). The emergence of modern AI systems have raised dramatically and so has their ability to process, analyze and learn from the quantities of data that continue to grow exponentially. The fact remains that most of the performing approaches suffer from lack of robustness and reliability and may be vulnerable to attacks [SZS+13, NYC15, MDFF16], in addition to their inability to explain their decisions and actions to human users. It is in this context that the quest for more transparent and interpretable AI has intensified in recent years. Indeed, in symbolic AI, the models used are often simple and are considered as interpretable in comparison to the recent ML models such as Deep neural networks (DNNs) that are lacking transparency.

There is no consensus from the literature on the definition of interpretability or explainability. The author in [Mil19] defines interpretability as "the degree to which a human can understand the cause of a decision". Thus, a model is said to be interpretable if a human is able to understand alone the model decision-making process or predictions by looking at its parameters [Lip18, ADRDS+20].

In practice, interpretable AI approaches aim to build machine learning models with low complexity (inherently human-interpretable) while keeping a high level of performances [Mol22]. On the other hand, explainable AI approaches aim at providing accurate and user comprehensible explanations of the underlying complex models to users [WYAL19, Lip18, ADRDS+20].

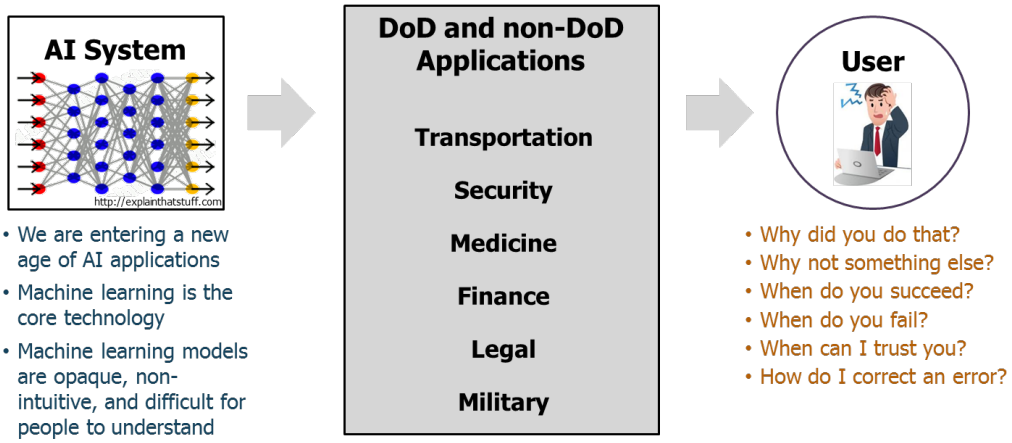


Figure 1.1: The need for explainable AI (Illustration from DARPA XAI Program [GA19])

1.1 Explainable Artificial Intelligence

The term *eXplainable Artificial Intelligence* (XAI) has been introduced by DARPA [GA19] and has been since used to refer this field that aims at helping users understand the workings of an AI system [Mil19], [GA19], [DVK17], detect data bias and discovering flows in the models. Explainability for AI systems has grown along with the success and adoption of deep learning systems that are applied in many fields where trustworthiness is critically needed such as in legal systems [Lip18, MB18, Rud19], autonomous driving [TPJR18, BTD+16], cybersecurity [Vre19, SNPS18, NZ17], phishing detection [SMA+18], transportation [ARAB+17], law enforcement [DF18], recruiting [Das18], health care and criminal justice [RU18], energy reliability [RPR+10], financial risk assessment [CLR+18], detecting heart attacks [WRK+17], diagnosing Alzheimer's disease [PHL+17], assessing recidivism risk [BB13, TvdH13], medical sciences and diagnostics [ADWF15, CNC+16, LKB+17], surveillance systems [DFL+18], biometric and handwritten characterers recognition [BLS+18, SPG+19, TVRFOG18] and so on.

Miller [Mil19] suggests the idea that explainability of AI systems can be refined if a close collaboration comes into place between the social science and the XAI researchers. He argues that interactions between human and AI systems can be improved by replicating and understanding from social science how people define, generate, select, present and evaluate explanations and that these mechanisms could be extended to the XAI field. The author in [Mil19] also reviews some of the relevant findings from social science research on human explanation, and has provided some insights into how this work can be used in explainable AI. Other papers also reviewed social science aspects of XAI systems such as [DVKB+17] that studies the role of algorithmic transparency and explanation in lawful AI (Trustworthy AI) and [LOL+18] that analyzes the fairness and accountability of algorithmic decision-making processes. Several XAI approaches have been proposed to achieve the general goals of accomplishing explainability of AI systems used for high-stakes prediction problems. There are different users or target audience of machine learning methods (e.g. users impacted by model decisions, developers, data scientists, regulatory entities, managers, etc) that require different explanations depending on their goals [ADRDS+20, VdWSNI+14]. Thereby, the form of the explanations are application and user-dependent.

1.1.1 The need to explanation

Despite the widespread use of XAI methods recently, no proper formal definition of what is an explanation is provided in the literature. We try to present it in the following through the different roles it can have from an AI perspective. Based on the research in [Mil19] and [AB18], an explanation can be used to justify the result of a system to a user in order to gain his trust, identify errors in a system to enable control, get an oversight in case of adverse or unwanted effects such as biased decision-making or social discrimination, understand a system well in order to improve it and learn knowledge by allowing humans to discover new facts that are not directly explicit from an AI system.

Explainability is not a new topic and has been dealt with in the 1970's where expert systems were already able to provide pieces of explanation [SB75]. It emerged again and became a hot topic since the adoption of deep learning models, a new generation of AI system called "black-box" models. These models would return a prediction to a given input but could not provide an explanation for its decision. "The current generation of AI systems offer tremendous benefits, but their effectiveness will be limited by the machine's inability to explain its decisions to users" [GA19]. This trade-off between the accuracy of these ML models and their ability to be understood by humans is illustrated in Figure 1.2. It is particularly the case of AI systems which are developed to be used in areas where the decision of an AI system can have serious consequences and critical situations. The domain experts and users are reluctant to adopt decisions they cannot understand, which limits the use of machine learning methods in such sensitive contexts. Recent work have highlighted the vulnerability of deep learning models in different tasks such as speech recognition [CW18], text classification [ERLD17], malware detection [GPM+17] and particularly image classification [GSS15, PMJ+16, CW17]. Thus, this kind of system decisions must be explained in order to prevent errors and gain trust of the user [AB18]. In addition to recent regulations that has grown a need to explain for both legal and ethical reasons such as the "right to explanation" introduced by the European General Data Protection Regulation (GDPR) [Vos16] in 2018. Therefor, we need explanations to decide whether an AI system is robust enough or trustworthy to be used in safety-critical environments.

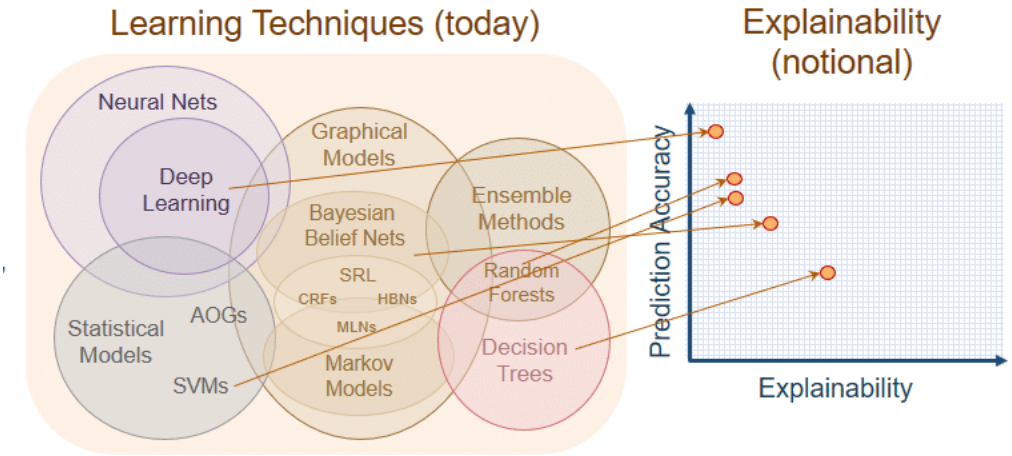


Figure 1.2: Trade-off between the accuracy of models and their interpretability. (Illustration from DARPA XAI Program [GA19])

1.1.2 Purpose of interpretability

The need for interpretable AI models is motivated by different purposes that can be technical, social or ethical [BVKV+18, DSZ16, Kas21, KK19a, LOL+18]. For instance, methods for creating white-box models aim to increase societal acceptance of machine learning models by establishing trust in decision results. On the other hand, methods for explaining complex black-box models aim to provide users with the reasons for models' decision and actionable insights to the results of algorithms. Other methods were created to promote fairness or to mitigate unwanted algorithmic bias and discrimination and lastly, methods for analysing the sensitivity of a model predictions to identify errors or biases in training data that result in adverse and unexpected behaviors (reliability).

XAI methods of the first category design systems based on learning algorithms that are easily understandable for a human being. This type of models are also referred to as intrinsic, transparent, or white-box models. Most of the methods of this category include self-explaining models such as linear models, decision trees and rule-based learners. They are discussed in Section 1.2.1 and a summary of existing white-box approaches can be found in Table 1.1.

Methods that attempt to explain black-boxes are the most common in the literature. These methods are also called post-hoc methods because they are designed to provide a posteriori explanations for predictive models (e.g random forests, deep neural networks, etc). Several desiderata can be considered to categorize these methods : the type of model to explain (specific/agnostic), the level of interpretability (local/global) or the representation of the explanation. Each of these sub-categories are presented in Section 1.2.2 and a summary of the discussed interpretability methods can be found in Table 1.2 and 1.3.

Another category for interpretability methods are the ones created to promote fairness by checking that the learned model generates decisions that are free from discrimination of specific inputs (e.g., admission to a university, and the goal is to prevent discrimination against individuals based on their membership to some group). These kinds of methods are necessary especially when working with sensitive data that can potentially affect human lives. The existing fairness methods usually evaluate the fairness of a ML system by checking the models' predictions and errors across certain demographic segments (e.g. groups of a specific ethnicity or gender). The authors in [KAAS12] presented prejudice, underestimation, and negative legacy as three major causes of unfairness in ML models and analyzed them. In [HPS16], the authors proposed a framework for quantifying and reducing discrimination in ML models. In [ZVRG17], a new metric for evaluating decision boundary fairness with respect to one or more sensitive features was introduced. Other approaches were proposed to train models to make fair predictions by removing bias from both training data and model predictions. The authors in [ZVR+17] propose to use collective preference of different demographic groups as a base to define notions of fairness. The authors in [ABD+18] considered certain definitions of fairness previously outlined as special cases and proposed to incorporate them into a systematic framework.

The last category is the sensitivity-based interpretability methods proposed to analyze the trustworthiness and reliability of the model's predictions. Those methods rely on the property of stability of the decision function and the sensitivity of its output. The idea is to check if a small perturbation of an input may lead to a significant perturbation of the outcome. A large set of adversarial examples approaches based on sensitivity analysis have been proposed within the literature [GSS15, MDFF16, MDFFF17, DLP+18, CW17, NK17, LCLS17, BRB18, DLT+18, ZAG18, LLS+17]. They aim at finding imperceptible changes in the input to fool models into producing incorrect predictions. Sensitivity analysis can be broadly partitioned into black-box and white-box methods and can also be both a global and local interpretation technique.

1.1.3 Audiences interested in explainable AI

The purpose of the explanation and its form are linked to the audience for which it is intended. The type of explanations will vary according to different groups of people and their role in the system [BXS+20, Hin19]. Thus, to provide meaningful explanations we need to determine who is receiving the explanation (see Figure 1.3). For domain experts such as medical doctors or insurance agents, the purpose of an explanation is to gain knowledge about its reasoning in order to gain trust in the model itself. For users such as regulatory entities, the explanation should certify model compliance with the legislation in place and stated regulatory requirements [Hin19]. The purpose of the explanation for the users affected by model decisions is to understand their situation and verify the outcome's fairness. As for data scientists and developers, explanations are meant to improve the model efficiency, evaluate the training data and facilitate the debugging process.

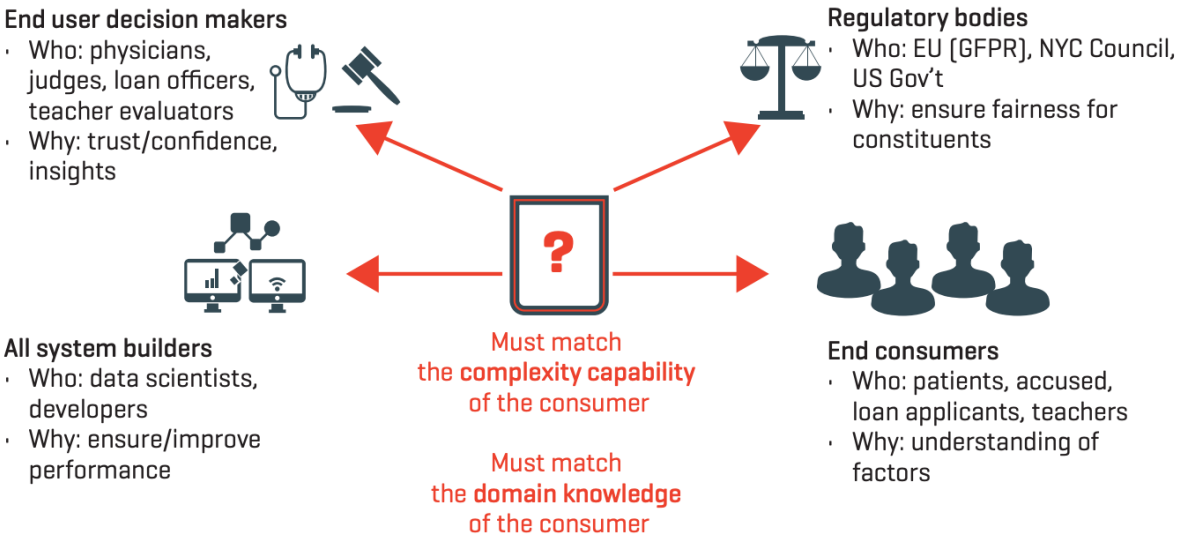


Figure 1.3: Audience interested in explainable AI (Illustration from [Hin19])

In this thesis, we consider all the cases where an explanation is provided to a user without making any assumptions about his background (can be a domain expert, a decision maker or any user who is not familiar with AI technology).

1.2 Related works

There is more than a single way to classify XAI methods in the literature, which makes it not easy to list them giving the different perspectives. Especially for supervised ML classification techniques that has been extensively studied in the field of explainable AI, where most of the works in the current literature propose methods to explain the predictions of this type of models such as feature attributions approaches[RSG16, LL17], decision rules [RSG18, JCS+20], logic-based [SCD18b, DH22, INMS19a] and counterfactual examples that has been attracting attention in recent years [WMR17, VDH20, KTKA20, AALC21, DPB+19, DCL+18, GWE+19, HHDA18, KTKA20, KBBV20, LK21, MST20, MTS19, Rus19, vdWRvD+18]. We will mention different interpretability methods according to the taxonomy presented in [Mol22] where the interpretability methods were summarized as : intrinsic or post-hoc methods, designed as model-specific or model-agnostic methods and provide local or global

Table 1.1: Summary of intrinsic interpretability methods.

Acronym	Ref	Target Model	Data Type	Year
InterpretML	[CLG+15]	Generalized Additive Model (GAM)	tab[3]	2015
iBCM	[KGJS15]	Bayesian Models	tab	2015
—	[LRMM15]	Decision Lists[4]	tab	2015
Slim	[UR16]	Linear Integer Model	tab	2016
AIX360	[DGW18]	Boolean Decision Rules	tab	2018
—	[IPNMS18]	Decision sets[5]	tab	2018
—	[WDGG19]	Generalized Linear Models (GLM)	tab	2019
—	[HRS19]	Decision Trees	tab	2019

explanations. Accordingly, we review and categorize in this section the state-of-the-art approaches for XAI.

1.2.1 Interpretable models (intrinsic methods)

Interpretable models are readily interpretable by design which makes their predictions and behaviors directly (human) understandable. Intrinsic interpretability consists in using interpretable models that have a low complexity given their simple structure and then derive straightforward explanations from it (self-explaining). The common learning algorithms that are considered interpretable by humans are linear and logistic regression, rule-based models and short decision trees [Fre14, HDM+11, RSG16]. Although these models offer better interpretability, they have limited performance on high-dimensional data and are considered directly interpretable only if the number of features and classes is limited and the size of the model is reasonable.

Linear models

Linear models are among the simplest classification models in supervised learning and demonstrate a good generalization abilities. A linear model associates an output class to an input instance x by computing a weighted sum of the features, where each weight represents the relevance of a feature. Such models make the assumption of linear dependence between the input variables and the output class which makes them highly interpretable. The influence of a feature on a prediction is actually the value of each parameter w^i, associated to the variable X_i [RSG16]. The feature X^i contributes to increase the model's output by w^i if the weight is positive, and in contrast, decreases it if the weight is negative.

Example 12. *An example of the features importance explanation is given in Figure 1.4 where we can visualize both the sign and the magnitude of the contribution of the attributes toward the model's behavior (global explanation) and the prediction of the input sample (local explanation).*

However, real-world applications does not necessarily assume a linear relationship. In order to cope with this restrictions, other extensions of linear models exist. For instance, the Generalized Linear Models (GLMs) which allow to model all types of outcomes and Generalized Additive Models (GAMs) to model nonlinear relationships. GAM is able to interpret linear and logistic regression, single trees and tree ensembles (bagged trees, boosted trees, boosted bagged trees and random forests).

[3] tab is a short of tabular data.
[4] Decision lists models are a series of IF-THEN statements (for example, if high blood pressure, then stroke).
[5] Decision sets can be viewed as unordered sets of rules, under some sort of rule non-overlap constraint.

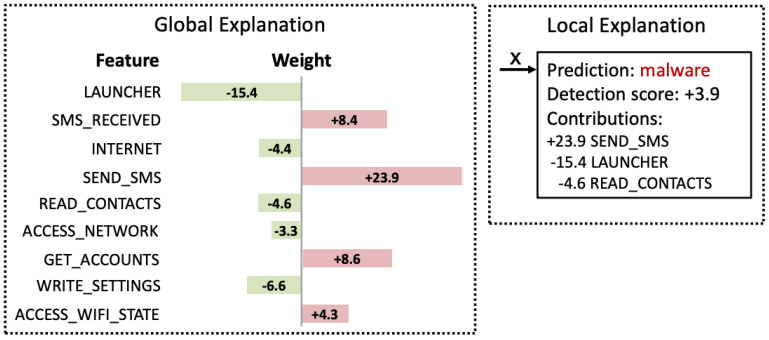


Figure 1.4: Example of a feature importance for a global and local explanation for a malware detection system based on a linear model and boolean features (Illustration from [Mel21]).

Decision trees

A Decision tree (DT) is a hierarchical structures used for both classification and regression problems [BFOS84, LR76, Qui87b]. It is a powerful technique used to fit data and it is widely adopted when the relation between the features and the outcome is non-linear or when the features are not independent from each other. A decision tree is designed with an explainable structure where a tree is grown on training set. The internal nodes corresponding to the features are split nodes where each one represents a test (for numerical features, does the feature have a value lower or greater than a threshold). In a classification setting, the leaf nodes of a tree represent the output classes. Given an input instance, the solution path in the decision tree presents the path from the root to the leaf. This path called a *decision rule* is associated to every decision made by the tree and can be understood even by non-expert users as long as the number of features remains reasonable. Thus, the interpretability of DTs relies on the *decision rule* set.

Example 13. *An example of a decision tree trained on IRIS-flower dataset.*

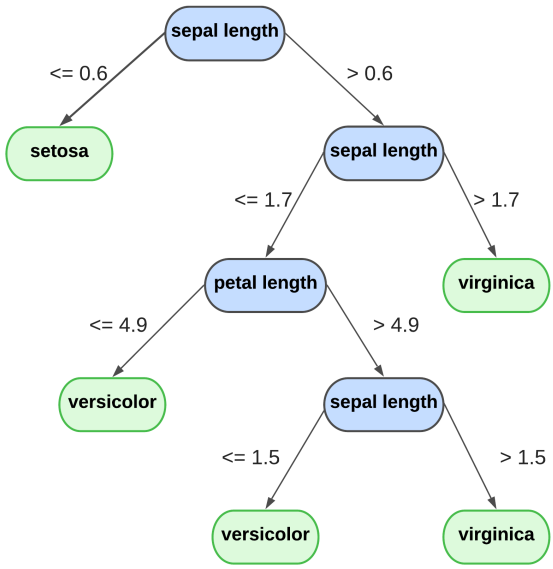


Figure 1.5: Example of a decision tree classifier with 3-classes (Illustration from [Fis36]).

Decision trees have long been considered as embodying interpretable classifiers in ML [Bre01b,

Fre14, Mol22] and have been adopted to globally explain neural networks [CS95, KSB99] and tree ensembles [HH18, TSHW20]. Nevertheless, recent works argued that interpretability of DTs can be compromised due to the large size of its explanations [BMPS20, ABB+21, HIIM21].

Rule-based learners

Together with decision trees, rule-based learners (RBML) are considered closer to human reasoning [BJL09, PGG+19]. Rule-based learners are systems that create their own models by learning rules that characterize the data to fit. Given the data and their corresponding output classes, a rule-based model learns a set of rules in different forms (IF-THEN structure, a combination of rules, etc).

Several rule-based methods have been proposed in the last few years where the learned rules are used to explain the model's prediction [NAC02, LBL16, MVED17]. Authors in [WRDV+17] proposed an algorithm ((Bayesian Rule Sets) to build Bayesian classifiers by learning short rule sets. These models are interpretable by humans since they produce a set of rules that concisely describe a specific class. The proposed method in [LBL16] learns rules by optimizing a loss function that directly depends on the length of the explanations (rules) generated.

Example 14. *This example presents a rule set learned using the method proposed in [LBL16].*

IF *Respiratory-Illness*=Yes and *Smoker*=Yes and *Age* $\geqslant$ 50 then Lung cancer
IF *Risk-LungCancer*=Yes and *Blood-Pressure* $\geqslant$ 0.3 then Lung Cancer
IF *Risk-Depression*=Yes and *Past-Depression*=Yes then Depression
IF *BMI* $\geqslant$ 0.3 and *Insurance*=None and *Blood-Pressure* $\geqslant$ 0.2 then Depression
IF *Smoker*=Yes and *BMI* $\geqslant$ 0.2 and *Age* $\geqslant$ 60 then Diabetes
IF *Risk-Diabetes*=Yes and *BMI* $\geqslant$ 0.4 and *Prob-Infections* $\geqslant$ 0.2 then Diabetes
IF *Doctor-Visits* $\geqslant$ and *Childhood-Obesity*=Yes then Diabetes

Figure 1.6: Example of interpretable decision set.

The rule generation approaches have the advantage of being transparent and explainable if the generated rules set coverage (number of rules) and specificity (number of predicates in a rule) are kept constrained since the number of conditions can significantly grow given the number of features or the output classes [HDM+11].

1.2.2 Post-hoc interpretability

Post-hoc interpretability (also referred to as post-modeling explainability) is another approach mainly used for the models with higher complexity [RPF+21, Lip18]. These methods analyze the models that are not transparent by design. A black-box predictor is an opaque system where the mapping from input to output is invisible to the user or they are known but uninterpretable by humans. Depending on the level of interpretation, post-hoc methods may explain a model locally or globally.

Model-specific methods

Model-specific methods can only be used to interpret a specific family of models and provide model-based types of explanation. They are usually used to weight the importance of the features for the model's decision [Bre01a, OMS17, YCN+15, JMD+05] and they rely on the internal states of the learning mechanism to derive an explanation. Most of them approximate the behavior of the black-box by

[6] tab/img/txt are a short for tabular/image/text.

Table 1.2: Summary of post-hoc XAI methods.

Name	**Ref**	**Model-Specific** vs**Model-Agnostic**	**Data Type**	**Year**
G-REX	[JKN04]	Agnostic	tab/img/txt[6]	2004
ICE	[KKS07]	Agnostic	tab	2007
PDP	[Fri01]	Agnostic	tab	2007
SAlib	[SRA+08]	Agnostic	tab	2008
DeLP3E	[SSF14]	Specific (Bayesian)	tab	2014
DeepExplain	[ZF14]	Specific (CNN)	img	2014
iNNvestigate	[SDBR14]	Specific (CNN)	img	2014
LIME	[RSG16]	Agnostic	tab/img/txt	2016
CAM	[ZKL+16]	Specific (CNN)	img	2016
MMD-critic	[KKK16]	Specific (Bayesian)	tab	2016
rationale	[LBJ16]	Specific (DNN)	txt	2016
SHAP	[LL17]	Agnostic	tab/img/txt	2017
fair-classification	[ZVR+17]	Agnostic	tab	2017
fairness	[DHP+12]	Agnostic	tab	2018
L2X	[CLP+18]	Agnostic	tab	2018
CAV	[KWG+18]	Specific (DNN)	tab/img	2018
ANCHOR	[RSG18]	Agnostic	tab	2018
Grad-CAM++	[CSHB18]	Specific (CNN)	img	2018
LEMNA	[GMX+18]	Agnostic	tab/img/txt	2018
—	[MWM18]	Specific (Gradient based)	tab	2018
—	[SCD18b]	Specific (Bayesian)	tab	2018
BEEF	[GPSS19]	Agnostic	tab/img/txt	2019
—	[INMS19a]	Agnostic	tab	2019
DeNNeS	[MG20]	Specific (DNN)	tab	2020
Tree explainer	[LEC+20a]	Specific (Tree-based)	tab	2020
—	[DH22]	Specific (DT/BNN)	tab	2020
Glocalx	[SGM+21]	Agnostic	tab	2021
—	[DLM+22]	Specific (Tree-based)	tab	2022
DeepGlobal	[SLZS22]	Specific (NNs)	tab/img	2022
—	[FdSRGL22]	Specific (NNs)	tab/img	2022

means of interpretable models (decision rules, decision trees and linear models) as presented in section 1.2.1.

Various methods were proposed to explain the different types of deep learning models. An example of model-specific method for explaining deep learning models is DeepLIFT [SGK17] in which the contributions of all neurons in the network are back-propagated to the input features. Several approaches used to explain different architectures of neural networks propose saliency maps (also referred to as heatmaps) as an explanation. Authors in [ZF14] present a visualization technique to produce saliency maps. The approach relies on assessing the property of sensitivity by iteratively forwarding the same image through the network occluding a different region at a time.

The TreeView method proposed in [TKSR16] extract a visual interpretation via a surrogate decision tree to explain DNNs. The approach takes as input a DNN and a number K given by the user to define the number of hierarchical partitions (clusters) of the feature space. Subsequently, the approach will create a meta-feature associated with each of the K clusters and train a random forest to predict the K

labels. Thus, the decision trees of the random forest are used to build a TreeView representation of the complex model. Other methods were proposed to interpret deep neural networks via decision trees in [WHP+18, ZYMW19].

Methods in [SVZ13a, BCC+16, ZCAW17, NDY+16] aim to explain the inner working of Convolutional Neural Network (CNN) based on activation maximization. The idea is to highlight the areas (pixels) used by the image classification black-box to make a decision. For example, the authors in [ZF14] propose a method based on sensitivity analysis of the network input/output relationship to understand the CNN using visualisation. By backtracking the network computations, the method identifies the regions in input images that are responsible of the activation of certain neurons.

The Layer-wise Relevance Propagation (LRP) [BBM+15] is another post-hoc method specific to multi-layered neural networks and Bag of Words (BoW) models built on non-linear kernels. The idea exploited by LRP is to back-propagate the effects of a decision on a given instance to the input level (layer-wise relevance propagation). For instance, building such saliency maps is based on a layer-wise conservation principle. Those heatmaps are comprehensible (to a human) and are used to visually identify which input had how much influence on the predicted output (i.e. which input contributed most to the obtained result). As shown in Figure 1.7, given an input sample x, the LRP method decomposes the classification output into sums of feature and pixel relevance scores. It is a way for obtaining the features importance then visualize it through saliency masks. The pixel-attribution representing the pixel-wise explanation is built using final relevances that correspond to the contributions of single pixels to the prediction.

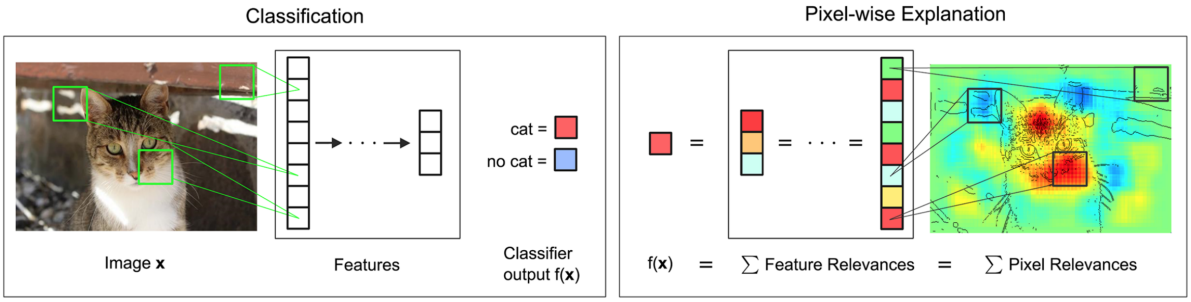


Figure 1.7: Visualization of the pixel-wise decomposition process of the LRP method. (Illustration from [BBM+15])

Another type of post-hoc methods specific to deep neural networks that are used to generate saliency maps are the gradient-based and attribution-based methods. The methods of the first category analyze the impact that small changes to the inputs have on the model's outcome (e.g. [SVZ13a]). The methods of the second one compute the contribution of input features to the model's output such that the sum of all contributions should be approximately equal to the output (e.g. [MLB+17, STY17]). One interpretation is given as a saliency map. Note that saliency maps need the input to be interpretable. The interpretation of a pixel attribution explanation generated with a gradient-based method is as follows : increasing the color values of the pixel would increase the probability of the predicted class (for positive gradient) or decrease it (for negative gradient). Besides, the effect of a change of a pixel is proportional to the absolute value of the gradient (the larger the value the more the effect) [Mol22].

Another known method for visual explanations for CNN decisions is the Gradient-weighted Class Activation Map (Grad-CAM) [SCD+17]. It is actually a generalization of the Class Activation Mapping (CAM) firstly proposed in [ZKL+16] and can be used for a wider range of CNN architectures (e.g. fully-

connected layers, structured output such as captioning in multi-task outputs and also for reinforcement learning). Grad-CAM propagates the gradient into the last convolutional layer. Thus, the relevance score assigned to each neuron for the decision of interest is the combination of the before last layer's feature maps and the output-specific weights. Figure 1.8 shows an example of visualization such as Guided Back-propagation [SDBR14] and Grad-CAM highlighting the pixels' relevance for the "cat" and "dog" classes. An extension of Grad-CAM called Grad-CAM++ was proposed in [CSHB18] to make better visual explanations of CNN models.

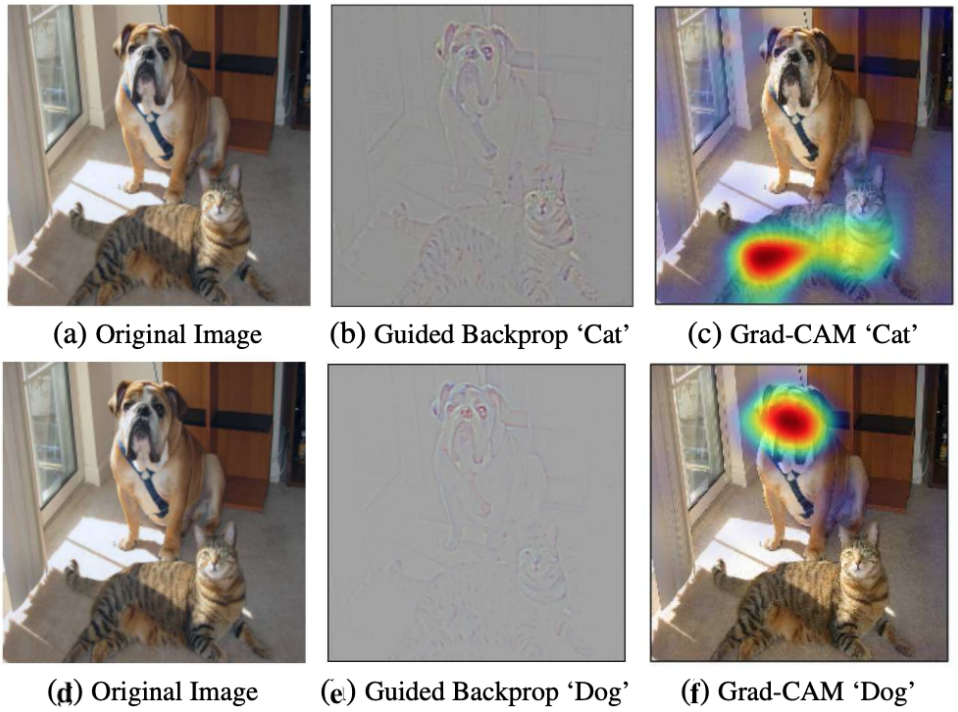


Figure 1.8: Example of visualization techniques highlighting the pixels' relevance for the "cat" and "dog" classes. Sub-figures (a,d) correspond to original image with a cat and a dog. (b,e) correspond to guided Backpropagation [SDBR14] highlighting all contributing features. (c, f) correspond to Grad-CAM output localizing class-discriminative regions. (Illustration from [SCD+17]).

Another trend for post-hoc model specific methods are the symbolic and logic-based XAI approaches that can be used for different purposes [Dar20]. For instance, sufficient reasons were introduced in [SCD18b] under the name of *PI-explanations*. Authors in [INMS19b, INMS19a] deal with some forms of symbolic explanations referred to as abductive explanations (AXp) and contrastive explanations (CXp) using SMT oracles. In [IMS21, IM21], the authors explain the predictions of decision list classifiers and decision tree classifiers using a SAT-based approach. Explaining random forests and decision trees is dealt with for instance in [AKM20b] and [INAM20, IIM20] respectively.

Model-agnostic methods

Model-agnostic methods are not tied to a particular type of function to explain. It means that post-hoc model-agnostic approaches can be used to provide an explanation for the decisions of any ML model disregarding its inner processing or internal representations (only need a query-level access). The explanations provided are "model free" and generally based on approximations of the behavior of the learning

mechanisms. Different perspectives to classify post-hoc model-agnostic methods can be considered. In our classification we take into account the explanation technique (explanation by simplification, explanation using formal representation) and the type of explanations presented to the user (feature-based, counterfactual, example-based). We will mention different XAI methods by looking at these perspectives.

Explanation by simplification Explanation by simplification refers to those techniques that rely on building an interpretable model to approximate the original decision function which can be used for explanations purposes. It can be a global surrogate model in order to approximate the behaviour of the original model to explain or a local surrogate model built in the neighborhood of the sample to explain, producing a local approximation of the original target system. The desiderata of a surrogate model is mainly accuracy (how accurate is the model performance on data) and fidelity (how much is good an interpretable model to mimic the behavior of the black-box). For instance, authors in [LRL+18] propose to generate surrogate-based explanations for individual predictions based on a sampling centered on particular place of the decision boundary and show the importance of defining the right locality in order to locally approximate accurately the black-box predictions.

The purpose of these approaches by simplification is to reduce the complexity of the model by means of a simplified approximation to gain in interpretability (for example reducing the number of architectural elements or number of parameters of a DNN). Model simplification is usually done by adopting a transparent model (cf section 1.2.1) which is easier to be implemented and explained than black-box systems.

Several methods using a simplified approximation of black-box models have been proposed withing the literature. For instance, the authors in [CS95] proposed a training algorithm TREPAN to approximate the concepts learned by black-box model using decision trees. Authors in [BCNM06, HVD+15] proposed knowledge distillation to get a smaller model that is less computationally expensive while approximating the performance of the original model, it has been used in [FH17] to get a decision tree that could explain the predictions of a black-box models. This other known and widely used model-agnostic methods proposed to explain black-box classifiers are LIME [RSG16], SHAP [LL17] and ANCHORS [RSG18]. LIME [RSG16] stands for Local Interpretable Model-agnostic Explanations (LIME), an approach to explain the predictions of any classifier by learning an interpretable model locally around the prediction. LIME uses a linear classifier to approximate local properties of the black-box models and produces coefficients of this surrogate model that is subsequently used as interpretations.
Concretely, given an input instance x and its prediction by a black-box model, a surrogate model is trained on a set of randomly generated perturbations of the sample x weighted by their distance to x (see Figure 1.9). An interpretable model g is then trained using a Lasso regression on these synthetic samples in order to optimize the objective function formally defined in [RSG16] as :

$$\xi(x) = \underset{g \in G}{argmin}(\mathcal{L}(f, g, \pi_x) + \Omega(g)) \tag{1.1}$$

The objective function $\xi(x)$ illustrates interpretability vs fidelity trade-off and represents the explanation. It is the sum of : (1) the term representing the local fidelity of g corresponding to the inverse of $\mathcal{L}(f, g, \pi_x)$, and (2) the term representing the complexity of g corresponding to $\Omega(g)$. Note that the number of features for which the explanation should be attributed is given by the user as input and it corresponds to the number of coefficients of the Lasso regression trained on the synthetic samples.

A rule-based improvement of LIME was proposed in ANCHORS [RSG18] for high-precision model-agnostic explanations. Similarly to LIME, ANCHORS generates local explanations for individual predictions of black-box ML models by sampling instances in the vicinity of the sample being explained.

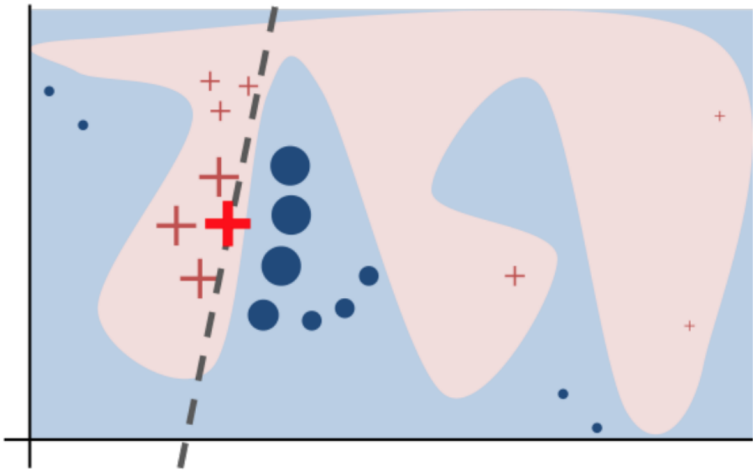


Figure 1.9: Example presenting the intuition behind LIME. The blue/pink background represents the model's decision function f. The bold red cross in the sample to explain x. The samples around x gets predictions from f and are weighted by their proximity to x (represented here by size). The dashed line is the function g learned (locally faithful to f). (Illustration from [RSG16]).

Unlike LIME, ANCHORS explanations are expressed as IF-THEN rules and are faithful by design because the coverage is adapted to the original model's behavior. A similar approach was proposed in [LKCL19] to capture the behavior of a black-box model using two-level decision sets and allowing users to input the features of interest.

SHAP [LL17] stands for Shapley additive explanations. It is a unifying approach that provides global and local attribution features by computing the Shapley values [Sha53] based on concepts from the coalitional game theory. The Shapley value of a variable is the average of its marginal contributions across all permutations w.r.t to three desirable properties: local accuracy, missingness, and consistency. Hence, SHAP explanations aim at identifying which features contribute the most to the difference in model prediction at a specific input versus a background distribution. Given a instance x, the explanation model g used to compute the importance values for each feature proposed by SHAP is defined as :

$$g(x) = \sum_{j=0}^{n} \phi_j, \phi_i \in \mathbb{R} \tag{1.2}$$

where ϕ_j is the contribution of feature X_j to $g(x)$ for all $j \in [1, n]$; and ϕ_0 is the output of the model when none of the features in x is present.

Unlike LIME that builds sparse linear models around an individual prediction in its local vicinity, SHAP computes all permutations in order to give the exact Shapley values. However, the computational cost of this evaluation is intractable. This variant of SHAP called KernelSHAP is impractical and other model-specific versions have been proposed to overcome this limitation like TreeExplainer [LEC+20b] which computes the exact Shapley values in polynomial time for tree-based models, LinearSHAP (for linear models) and DeepSHAP (for deep neural networks) to compute approximations of the Shapley values [LL17].

Several other post-hoc methods exist (see for instance [ŠK14, DSZ16, LC01, BBM+15]). They are often model-agnostic as they only need to compute the output of a model regardless of its internal working and are listed in Table 1.2.

Explanation via formal representations refers to those post-hoc explainability techniques that relies on formal methods and are mainly based on compilation knowledge or abductive reasoning (see section 2.2). Authors in [INMS19a] present a method based on encoding the machine learning model into constraints and provide cardinality and/or minimal explanations by applying abductive reasoning on the model to answer XAI queries. Then Boolean representation of the ML model will be generated and further simplified into symbolic explanations. These later are generally prime implicants, sufficient reasons referred to as abductive explanations (AXp) representing an answer to a "why?" question, or also contrastive explanations (CXp) representing an answer to "why not?" question. More details about this type of approaches are presented in the Section 2.2.

Feature-based explanations aim to explain a trained model outcome by computing a relevance score for its input features. These methods are popular in Explainable AI as they give intuitive readings on relations between features and predictions. They generally rely on model simplification explanations (cf section 1.2.2).
For instance, SHAP is a feature attribution method that computes local and global explanations. It estimates the contribution of each feature to a decision value and returns a list of feature attributions to a specific prediction (local explanation) or to the model (global explanation). LIME provides an explanation as a list of feature contributions to the prediction of the instance x and highlights the features changes that have the most influence on the prediction. The image illustrated in Figure 1.10 was assigned "tree frog" by Google's Inception neural network. The explanation generated using LIME highlights the most important features, where the classifier primarily focuses on the frog's face as an explanation for the predicted class.

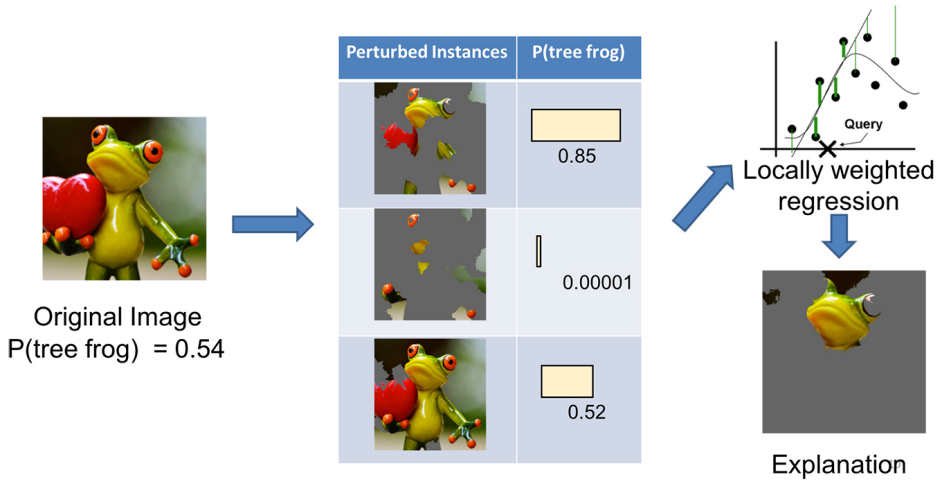


Figure 1.10: Explanation for a prediction with LIME. The top three predicted classes are "tree frog", "pool table" and "balloon". (Illustration from [MTR16]).

The advantage of feature-based attribution methods is that they are generally more scalable than their alternatives. It can associate attributions to different types of data (tabular data, text and image). However, in spite of their popularity, feature attribution methods are often accused of being inconsistent, meaning they can lower a feature's assigned importance when the true impact of that feature actually increases [LEL18, Ign20] and can be fooled by adversarial attacks [SHJ+20].

Feature visualization explanations in this category, approaches are proposed for visualizing the relationship between the outcome class and the input variables. The aim is to increase the model's inter-

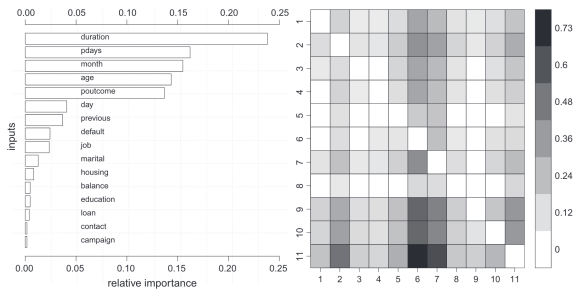
pretability by the mean of visualizations (parallel coordinate plots, scatter plots or projection methods).

Visualization techniques are generally used in data mining and statistics and lately, as visualization tools for model inspection. There are few works about the visualization techniques to explain the black-box output [KKS07] compared to the model-specific techniques. It is mainly due to the high dimension of feature space of black-box model's that makes it hard to analyze.
Authors in [GDL03] presented a survey about visualization methods based on the sensitivity analysis (SA) computation. Although most of these methods were mainly developed for neural networks (NNs), some of them can be used directly for any black-box model analysis. The authors in [CE13] presented a visualization method based on sensitivity analysis [RRK90] where they query the black-box model with sensitivity samples and analyze it to create different visualization plots for the results such as input importance bars, color matrix, variable effect characteristic curve, surface and contour (see example of Figure 1.11).

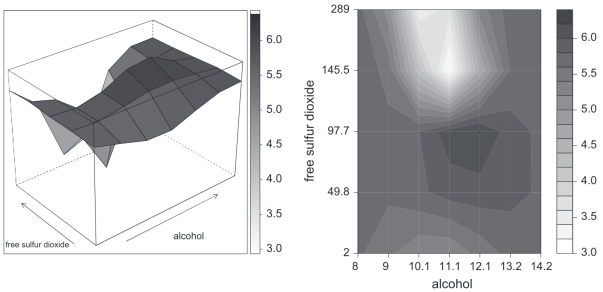
Authors in [KKS07] also propose a visualization techniques using the sensitivity analysis to deal with the visualization of the black-box model's output. They propose to study the relationships between variables for regression models and classification boundaries for classifiers in order to answer to "How is the output of the artificial model related to the measured input?". They also provide visualization to estimate the credibility of any black-box model.

A popular method for feature visualization is the Partial Dependence Plot (PDP) [Fri01]. Partial dependence plots represent the expected output of a model when the value of a specific variable (or group of variables) is fixed. In [Mol22], the authors describe it as the marginal effect of one or two features have on the predicted outcome of a machine learning model. These plots help in visualizing and understanding the relationship between the outcome of a black-box and the input in a reduced feature space. Authors in [KKS07] provide an extension of PDP named Individual Conditional Expectation (ICE). The idea is to generate for each sample, a plot to show the evolution of the prediction with respect to a grid of values of one given feature (while the others remain constant). ICE plot shows the average partial relationship between the outcome and some features. Figure 1.12 shows an example from [GKBP15] representing the ICE of a prediction with respect to a feature X_1 (dots correspond to the actual value of X_1 for each instance). We can see that there is a parabolic relationship between the model studied f and X_1. The PDP is also represented as a yellow line, which is the average of the ICE over all instances.

Counterfactual explanations: Another popular type of explanations are counterfactual explanations (CF). A counterfactual explanation tells the user how an input should be modified to make the system's decision change based on the provided explanation. This type of explanation have been widely explored



(a) Bar plot with the 1D input importances for the bank data (left) and color matrix with 2D input pair sensitivity for the wwq dataset.



(b) Surface and contour plots.

Figure 1.11: Example of visualization plots explanations from [CE13].

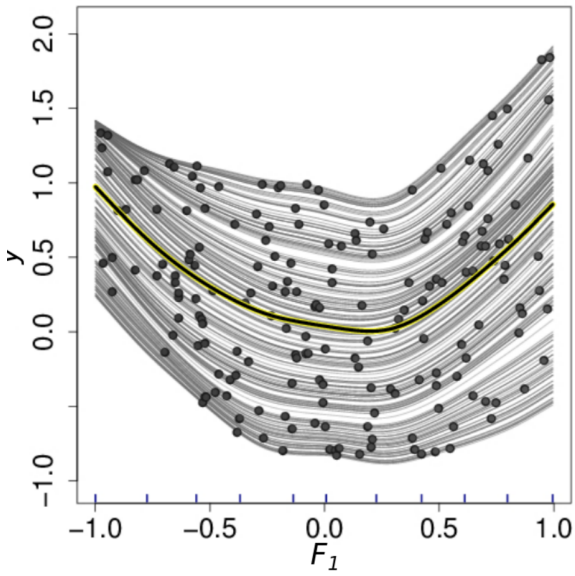


Figure 1.12: Example of the ICE and the PDP of a prediction with respect to feature X_1 (Illustration from [GKBP15]).

recently and are proved to help the user understand the decision of a models [GKC+18, LPNC+17, Byr19, AALC21, HHDA18, KTKA20, KBBV20, WMR17, DCL+18, DPB+19, GWE+19, LK21, Rus19, MTS19, MST20]. Consider the case of a bank's decisions regarding whether or not to accept a loan application as illustrated in Figure 1.13. The model decision is based on binary features like "does the requester has a stable job" and "does he have a criminal record" ? An intuitive question that a user who has obtained a refusal for his loan application would be: "what are the elements of his application that led to such a decision?" and "what he can possibly do to change his loan decision ?" The counterfactual explanation identifies the features to change in his application to get an acceptation from the model.

Most of the methods proposed within the XAI field for the generation of the nearest counterfactual explanation are optimization-based (rely on separate optimizations for each input) and there is a risk of generating unjustified counterfactual examples as shown in [LLM+19]. The methods in [Rus19, MST20] generate counterfactual explanations for the mixed datasets commonly used in the real world using an optimization based on the original features. In [JKV+19], the authors provide an optimization framework to traverse the data manifold via its latent representation and in [LKLH19] they used generative adversarial network. Authors in [SHG19] proposed CERTIFAI : "Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models". The approach uses a customized genetic algorithm to generate perturbations of an input that lead to a different outcome. The counterfactuals can be used to check the robustness and examine fairness of a ML model at individual and global level. Authors in [WMR17] propose to find a counterfactual explanation for a sample x by solving a relaxed version of the original optimization problem using gradient-based approaches. A solution to this optimization problem represents an instance x' that presents the perturbations in the original input features that can lead to a change in the prediction of the ML model. In [MTS19], the authors proposed a causal view of the feasibility of CF examples using structural causal models. They also proposed to generate counterfactual explanations using a proximity loss based on causal relationships between features instead of the standard proximity measure usually used.

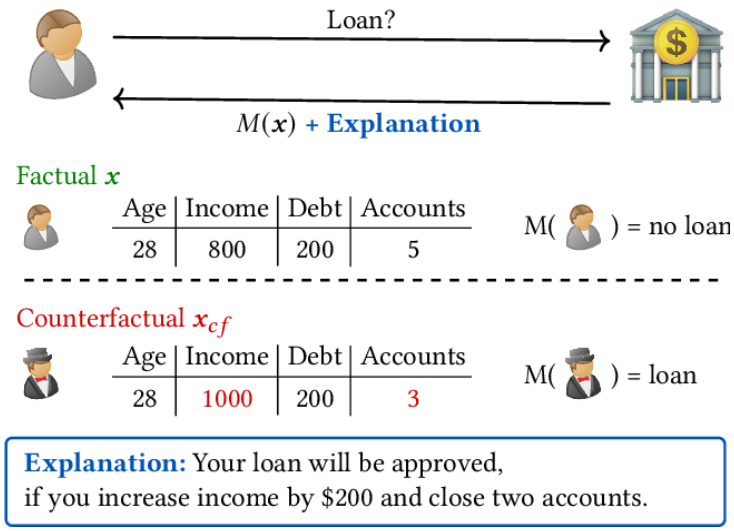


Figure 1.13: Example of a counterfactual explanation scenario (Illustration from [SGZS21]).

Other methods were used to generate counterfactual explanations without relying on an optimization problem. For instance, authors in [INMS19a] proposed a method based on abductive reasoning to generate two types of symbolic explanations. Among them, the contrastive explanations (CXp) as an answer to "why not?" question, which corresponds to CF explanation. [KBBV20] proposed MACE for "model-agnostic approach to generate nearest counterfactual explanations". MACE relies on encoding the predictive model, the distance function, the plausibility and diversity constraints into a logical formulae. Thus, the problem of finding the nearest CF becomes a sequence of satisfiability (SAT) problems where the goal is to verify if there exists a counterfactual explanation at a distance smaller than a given threshold, and can be solved using standard SMT (satisfiability modulo theories) solvers. Authors in [BL22] proposed a fuzzy framework to deal with imprecise knowledge or data and imprecise formulations of explanations. Such an approach is based on the integration of fuzzy semantics to a logical framework exploiting knowledge represented as structural causal graphs from [Mil21].

Note that adversarial examples are like counterfactual examples; however, they do not focus on explaining the model by providing an actionable explanation in the form of data instances that would have received a different outcome, but on misleading it (e.g. an adversarial example in computer vision would correspond to an imperceptible change in the image to fool models into producing incorrect predictions).

Example-based explanations: Explanations by example are mainly aiming at extracting representative examples from the dataset to explain the model being analyzed. They are actually not considered as model-agnostic methods in the taxonomy of [Mol22] since they do not generate explanations but actually select them by extracting data samples that relate to the prediction being explained. The example-based explanations are similar to how humans behave when attempting to explain a given process by thinking in examples or analogies [Mol22].

An example of such explainability method is the one of k-Nearest Neighbors (kNN). In order to explain an input instance, the idea is to return the closest instances, like the work presented in [KK19b]. The k-Nearest Neighbor is a machine learning technique based on assigning the most represented class to a sample based on the outcome of its k closest neighbors. By definition, the interpretability of this approach is local since it gives an interpretation for a particular instance by presenting the instances

from the neighborhood that were used to decide of the output class. An example suitable for that kind of interpretability is presented in [KK19b] where they applied their approach on the MNIST dataset in order to explain the prediction of an image predicted as a "0" while it represents a "6" digit. The visual explanation shows that the input image was predicted as a "0" based on its similarity with instances from its neighborhood representing a "0".

Example 15. *The example in Figure 1.14 shows how kNN can be used to interpret a prediction by looking at the outcome of the closest nearest neighbors of the query sample.*

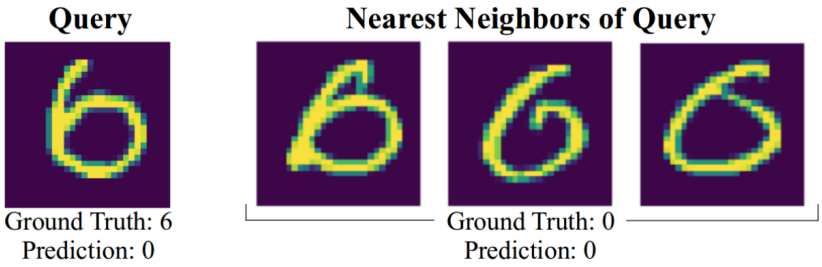


Figure 1.14: Example from [KK19b] representing the prediction of an input sample and its neighbors.

Thus, we can understand that the model was actually mislead by the outcome of the nearest neighbors of the input image.

Another example-based method is proposed to extract *prototypes* and *criticism*. Namely, the representative samples used to explain the global behaviour of a model are called *prototypes* and *criticism* are the data samples that do not fit the model well. Methods which extract those samples (such as [Zad96, LRBM08] for fuzzy prototypes, [KRS14, KK19b, AP94, LLCR18] for case-based reasoning) can be used to analyze the behaviour of a trained model by analyzing its predictions on *prototypes* and the *criticisms* samples, and, quickly extract a few instances that should be harder to predict (criticism) or easier to predict (prototypes). In [KKK16], the authors propose a Bayesian model criticism framework, called MMD-critic which efficiently learns prototypes and criticism, designed to help human explainability.

1.2.3 Local interpretability versus global interpretability

Explainable AI methods can be further distinguished into local and global methods. Local explainability methods give an interpretation for a specific decision (individual prediction) of a particular data instance. It generates the explanations with respect to a specific instance. In contrast, the global (model) explainability methods are used to understand the model's behavior through its working mechanisms and follow the entire reasoning that produces all the different possible outcomes. It generally takes a group of instances to generate one or more explanations.

Most of the post-hoc existing methods are local approximation methods. They explain a specific decision of a sample x by segmenting the data space and focusing on the samples around x to generate explanations such as the feature attribution methods (e.g. LIME [RSG16], Local Explanation Method-using Nonlinear Approximation (LEMNA) [GMX+18], SHAP [LL17]) and saliency maps ([SCD+17, ZKL+16, SDBR14, CSHB18, MLB+17, STY17, BBM+15]). Authors in [BAL+21] propose to help non-expert users understand the ML predictions by providing contextualisation elements. They propose an experimental study to assess the impact of adding contextual information on the understanding of local explanations. Explaining the global behaviour of a model is more difficult compared to interpreting

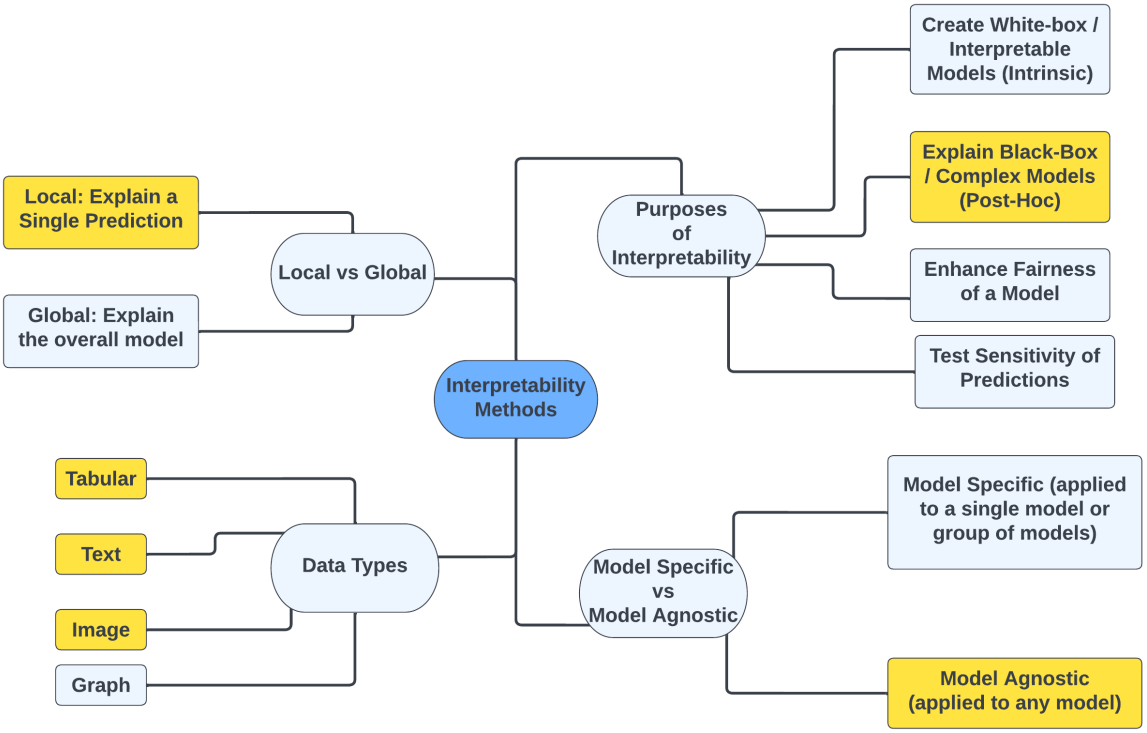


Figure 1.15: Taxonomy of machine learning interpretability techniques

the outcome of a specific instance, especially if the predictor is a complex model. Global XAI methods assess the role of features and their contribution to the model output. A summary of the included methods is shown in Table 1.3.

To contextualize and situate our work that we will be presenting in the second part of this thesis, we use Figure 1.15 of AI interpretability techniques from [LPK20] to highlight the elements describing our work. Our contributions to explain individual predictions of single and multi-label classification models consist of a post-hoc local model-agnostic approach. Indeed, our main approach for providing symbolic explanations is model-agnostic, which means it can explain the outcome of any ML model. It is declarative, which means it does not require the implementation of specific algorithms since its based on well-known concepts (more details in the upcoming sections) in SAT solving and it is local, meaning that it is used to explain individual predictions of black-box ML models.

Table 1.3: Summary of local vs global XAI methods.

Name	**Ref**	**Local** vs **Global**	**Model-Specific** vs **Model-Agnostic**	**Category**	**Year**
G-REX	[JKN04]	Global	Agnostic	Post-hoc	2004
ICE	[KKS07]	Global	Agnostic	Post-hoc	2007
PDP	[Fri01]	Global	Agnostic	Post-hoc	2007
SAlib	[SRA+08]	Global	Agnostic	Post-hoc	2008
DeLP3E	[SSF14]	Local	Specific	Post-hoc	2014
DeepExplain	[ZF14]	Local	Specific	Post-hoc	2014
iNNvestigate	[SDBR14]	Local	Specific	Post-hoc	2014
InterpretML	[CLG+15]	Global	/	Intrinsic	2015
iBCM	[KGJS15]	Global	/	Intrinsic	2015
—	[LRMM15]	Global	/	Intrinsic	2015
SLIM	[UR16]	Global	/	Intrinsic	2016
LIME	[RSG16]	Global/Local	Agnostic	Post-hoc	2016
CAM	[ZKL+16]	Local	Specific	Post-hoc	2016
MMD-critic	[KKK16]	Global	Specific	Post-hoc	2016
rationale	[LBJ16]	Local	Specific	Post-hoc	2016
SHAP	[LL17]	Global	Agnostic	Post-hoc	2017
fair-classification	[ZVR+17]	Global	Agnostic	Post-hoc	2017
fairness	[DHP+12]	Local	Agnostic	Post-hoc	2018
L2X	[CLP+18]	Local	Agnostic	Post-hoc	2018
CAV	[KWG+18]	Local	Specific	Post-hoc	2018
ANCHOR	[RSG18]	Local	Agnostic	Post-hoc	2018
Grad-CAM++	[CSHB18]	Local	Specific	Post-hoc	2018
LEMNA	[GMX+18]	Local	Agnostic	Post-hoc	2018
—	[MWM18]	Local	Specific	Post-hoc	2018
—	[SCD18b]	Local	Specific	Post-hoc	2018
AIX360	[DGW18]	Global	/	Intrinsic	2018
SENN	[AMJ18]	Global	/	Intrinsic	2018
—	[WDGG19]	Global	/	Intrinsic	2019
BEEF	[GPSS19]	Local	Agnostic	Post-hoc	2019
—	[INMS19a]	Local	Agnostic	Post-hoc	2019
DeNNeS	[MG20]	Global	Specific	Post-hoc	2020
Tree explainer	[LEC+20a]	Global/Local	Specific	Post-hoc	2020
—	[DH22]	Local	Specific	Post-hoc	2020
Glocalx	[SGM+21]	Global	Agnostic	Post-hoc	2021
—	[DLM+22]	Local	Specific	Post-hoc	2022
DeepGlobal	[SLZS22]	Global	Specific	Post-hoc	2022
—	[FdSRGL22]	Global	Specific	Post-hoc	2022

Chapter 2

XAI methodologies

The scientific communities may have different requirements and priorities when it comes to the characteristics of the XAI approach to be developed (e.g. tractability, scalability, formal aspects, level of explanation, need for access to data and model, parcimony, etc). This difference in design goals and desired properties is reflected by adopting diverse methodologies in the development of XAI approaches.

The proposed methodologies in the literature can roughly be classified in two types: formal methods and ad-hoc or numerical methods presented in the following sections.

2.1 Ad-hoc methods

The ad-hoc methods are defined as methods designed to answer a specific question or to accomplish a goal. Although they may work well in practice, they may be considered untrustworthy and capable of producing errors. Their major issue is that they offer no guarantee of correctness w.r.t the ML model explained since they are model-agnostic, thus, the explanations may be incorrect, inaccurate or unstable and are not necessary minimal [Ign20].

Most of the current XAI approaches are based on ad-hoc methodologies and compute approximations of real explanations. For instance, SHAP [LL17] is based on the cooperative game theory Shapley value [Sha53] and determines the features that contribute the most to the difference in model prediction at a specific input versus a background distribution. LIME [RSG16] generates an explanation by approximating the underlying black-box by an interpretable model. ANCHOR [RSG18] based on LIME provide rule-based explanations on local behaviors of the models, Partial Dependence Plot (PDP) [ZH21] and Accumulated Local Effect Plot (ALEP) [AZ20] describe how features affect the decision of ML models and the nature of the relationship between them (e.g linear or monotonic).

Recent works have highlighted undesirable behavior of ad-hoc methods [INM19, Ign20, SHJ+20, FdHvE22, ADLPR22]. Among these, the work in [Ign20] where the authors showed that explanations provided by SHAP and LIME are mostly incorrect from a global perspective. The same idea is expressed in [INM19, SHJ+20], where it is shown that some ad-hoc based approaches return the same explanation for two different data point that were predicted differently (incompatible classes). This shows that ad-hoc based explanations can be misleading and incorrect.

2.2 Formal methods

Formal methods are defined as mathematically or formally verifiable. They have been widely investigated for explainability [SCD18b, SSDC20, DH22, SCD19, INMS19a, INMS19b, Ign20, IMS21, ABB+22a, AKM20a, ABB+21, APR21, WGH19, MSGC+20, IIN+21, HIIM21, IMS21, GR22]. They are mainly

based on knowledge compilation or abductive reasoning [Ign20] and provide explanations that guarantee to have the same behavior with the model (accurate), and so, trustable. The approaches based on knowledge compilation compile predictive models into Boolean circuits and propose to interpret the model by enumerating the prime implicants of the circuit [DH22, SCD18b]. The abductive based approaches represent a model into a set of constraints and provide minimal explanations by applying abductive reasoning on the model's encoding to answer XAI queries [INMS19a, Rei87, INMS19b, ABB+22a, ABB+22b, IIN+22, HM22, IIM22, GR22]. Formal methods are often too expensive to compute (worst-case exponential complexity in time and space for compilation, number of explanations is worst-case exponential), and are strongly dependent on the type of classifier to treat since we need to develop dedicated algorithms for each ML model (model-specific) when using the compilation approach.

We present in the following sections the main line works used by these methods, namely knowledge compilation and abductive reasoning.

2.2.1 Knowledge compilation

Knowledge compilation is a technique used to overcome the difficulty (intractability) of some AI problems by pre-processing the available information. It is expressed as a translation problem which is done in two phases. A first phase called off-line which allows to compile a part of the information and a second phase called on-line where the compiled form with the rest of the information is used to efficiently answer queries. Authors in [DM02] presented a map for deciding the target compilation language that is most suitable for a particular application. Note that different target compilation languages exist such as the : Conjunctive Normal Form (CNF), Disjunctive Normal Form (DNF), Negation Normal Form language (NNF) formally defined in [Dar99, Dar01], Ordered Binary Decision Diagrams (OBDD) first presented in [Bry86], Decomposable Negation Normal Form (DNNF), Deterministic Decomposable Negation Normal Form (d-DNNF), Sentential Decision Diagrams (SDD) [Dar11] and so on. Knowledge compilation has been applied in different areas : diagnosis [TT04, HD05, SH+08], configuration [AFM02, HSJ+04], planning [GT99, EMW97, JV00, CRT98] and lately, knowledge compilation had been used to address fundamental problems for explainable and robust AI.

Formal XAI methods compile ML models into Boolean circuits that can make the same predictions with the models and provide valid and complete explanations. The advantages of such approaches is the possibility of efficiently checking, verifying the symbolic representations of classifiers (e.g. [CD07, SCD18a, CSSD17, OD15, Dar11]). Authors in [SCD18b] proposed an algorithm to compile latent-tree Bayesian network classifiers into decision functions in the form of Ordered Decision Diagram (ODD). This equivalent and tractable [7] symbolic representation is used to explain Bayesian network classifiers by providing two types of explanations. The first type called the *minimum-cardinality explanations* (MC-Explanations) corresponds to the minimal subsets of inputs that is sufficient for the current decision, while the second type called *prime-implicant explanations* (PI-Explanations) corresponds to the smallest subset of features that makes the rest of the features irrelevant to the current prediction. For instance, the authors in [NKR+18] proposed a CNF encoding for Binarized Neural Networks (BNNs) for verification purposes. In [SSDC20], the authors propose a compilation algorithm of BNNs into tractable representations such as OBDDs and SDDs. In [CD03, SCD19], the authors proposed algorithms for compiling Naive and Latent-Tree Bayesian network classifiers into decision graphs (symbolic representations). Authors in [SCD18b] and [AKM20a] use knowledge compilation techniques to design tractable cases for a set of XAI queries (e.g. enumerating minimum-cardinality explanations, deriving one prime implicant explanation, enumerating counterfactual explanations). These queries were also used in [ABB+21] to evaluate the intelligibility of several families of Boolean classifiers (decision trees, DNF formulae,

[7] can be answered using a polynomial-time algorithm

decision lists, random forests, boosted trees, Boolean multilayer perceptrons, and binarized neural nets).

The major issue for knowledge compilation based methods is the tractability of the compilation phase for large feature spaces and complex models. In addition to finding a suitable representation for the compiled model, there is need to provide dedicated compilation algorithms for each new ML models since its a model-specific technique.

2.2.2 Abductive reasoning

Abduction is defined as a form of logical reasoning that allows to explain a phenomenon or an observation from certain facts. It is considered as a type of inference that is frequently employed both in everyday and in scientific reasoning, where we make a probable conclusion from what we know (e.g. a detective's identification of a criminal by piecing together evidence at a crime scene). Abductive reasoning is well-known concept [Sha89, Mar91] for computing explanations. Many works used abductive reasoning to answer queries such as verification of properties of systems [KBD+17, LNPT18, NKR+18, SDC19] and diagnosis purposes [Rei87, Rym94] were proposed. However, in the last few years it has been used specifically for XAI tasks.

Authors in [CSGD20a] tackle the problem of explaining the decisions of machine learning models with discrete/continuous input and output variables. They analyze three symbolic encodings using Boolean expressions to reason about the behavior of the system. Based on the PI-explanation[8] introduced in [SCD18b] and notions on multi-valued variables, they formally define the notion of PI-explanations in a multi-valued setting. In [INMS19a], authors propose an approach based on abductive reasoning to generate explanations using some constraint reasoning system (e.g. Satisfiability Modulo Theories (SMT), Constraint Programming (CP), or Mixed Integer Linear Programming (MILP)) to encode the ML model to be explained and answer some entailment queries. For instance, they consider neural network models and use a MILP encoding in order to compute prime implicants, which are used to find the minimal subset or cardinality minimal explanations. The paper deals with some forms of symbolic explanations referred to as abductive explanations (AXp) as an answer to a "why?" question and contrastive explanations (CXp) as an answer to "why not?" question. The authors in [INMS19b] analyze the duality relationship between explanations (defined as a prime implicants) and counterexamples (defined as negated prime implicate) and investigated how to compute adversarial examples from counterexamples. They use First Order Logic (FOL) for the representation of ML model and overview algorithms for the enumeration of explanations and adversarial examples. The authors in [Ign20] overview recent logic-based XAI approaches to explain ML model predictions. The paper proposes an empirical study to assess the correctness of the explanations of ad-hoc methods like LIME, SHAP and ANCHOR in order to validate them and argue that rigorous explanations based on abductive reasoning are trustworthy and can be independently validated. Authors in [IMS21] discuss the computational complexity of computing explanations for Decision Lists (DLs) and show the contrast compared to Decision Trees (DTs). They propose AXp and CXp explanations for Decision Lists using the SAT solvers. Authors in [ABB+22a] focus on abductive explanations of random forests. They propose (1) "direct reasons" which are an extension of the abductive explanations of decision trees, and (2) "majority reasons" which are implicants of a majority of trees in the forest and which can be computed in polynomial time. They tackle the problem of generating and minimizing (in terms of size) these two types of explanations and propose algorithms to compute and compare them empirically.

The major issue for formal explanation methods is finding a logical language suitable for describing the ML model as faithful as possible. Another limitation is their tractability in practice, although some improvement are made possible, like in recent work [IIM22] that proposes algorithms for computing

[8] defined as minimal set of instance characteristics that are sufficient to trigger the decision [SCD18b]

path explanations for DTs which run in worst-case polynomial time. The experimental results reported in [SCD18b, INMS19a] show that the approaches based on logical representations are limited and may be difficult to obtain for large datasets. Another drawback of symbolic methods is linked to the nature of data as real-world data is often numerical and not binary.

2.3 Conclusion

The evaluation of XAI methods is on early stages (e.g. [JBB+21, BWM20]). The XAI community does not have a standardized terminology yet and there is no effective evaluation methodologies to assess which methods will do well for a given use-case [DVK17]. The evaluation of an explanation often depends of the target audience (different roles, background, objectives), and thus, have a different explainability requirements [TBH+18, MZR21, ARLG20].

Evaluating explanation methods is considered as an open problem [DVK17, JG20]. Although there is no standards established for their evaluation, some desired requirements for XAI methods were summarized in [HYHI21] such as faithfulness [AGM+18, LKCL19, JG20], plausibility (explanations must be sufficiently convincing to users) [LBJ16, LCH+19, SZM19], robustness [AJ18], and readability [AET18, YRS17, ALSA+17]. For instance, the authors in [BP19] proposed a human-grounded evaluation, done by assessing 17 criteria splitted into three categories for evaluating explanations : natural language (aims at assessing the correctness of the language used in explanations), human-computer interaction (enables to evaluate what the explanation conveys when it is transmitted from the system to the user) and also the content and form (dedicated to assessing the content and the form of the explanation).

Part II

Symbolic explanations

This second part of the thesis is devoted to the contributions made to explain predictive models and reason about explanations.

Firstly, we will present in Chapter 3 the general framework of our declarative and model-agnostic approach as well as the different modules that compose it. We formally define what is a sufficient reason and what is a counterfactual and how this kind of complementary symbolic explanations are generated for single-label classification problems with the help of Boolean satisfiability concepts. The results of an experimental study is conducted on several datasets from the literature in order to evaluate the feasibility of the approach in practice. The different characteristics of the enumerated symbolic explanations are also presented.

Chapter 4 concerns the explanation of multi-label tasks. We define several symbolic explanation types and show how we can enumerate them by adapting our approach to the multi-label case. We also introduce a concept specific to multi-label problems called the label-based explanations, allowing to take advantage of the structural relationships between labels. An experimental study is also provided in order to evaluate in practice the different concepts discussed in the chapter.

In Chapter 5 of this manuscript, we go beyond symbolic explanations and address score-based explanations for classification in a single-label setting. We define some desired properties of an explanation score to assess the relevance of both explanations and features individually, in order to evaluate them in ways that are closer to how users perceive them.

The last Chapter 6 is dedicated to feature attribution for multi-label classification. We first point out some deficiencies in attribution methods based on aggregation by defining three desirable properties. As a second contribution, we present a framework based on problem transformation allowing to provide global feature attributions capturing the above properties while using existing attribution methods as an oracle. The third contribution consists in a new attribution method based on symbolic explanations such as sufficient and counterfactual reasons, from which attribution scores are generated. Finally, we propose to go further concerning the property of label-explanation correlation by exploiting it to infer features attribution on a label using the explanations already computed on another label with which it is correlated.

Chapter 3

Symbolic explanations for single-label classification

After having presented the current state of the field of XAI in the previous chapters, we introduce now a novel framework for explainable classification. In this chapter, we will provide insights of the approach to handle single-label classification problems. Chapter 4 addresses the case of multi-label classification.

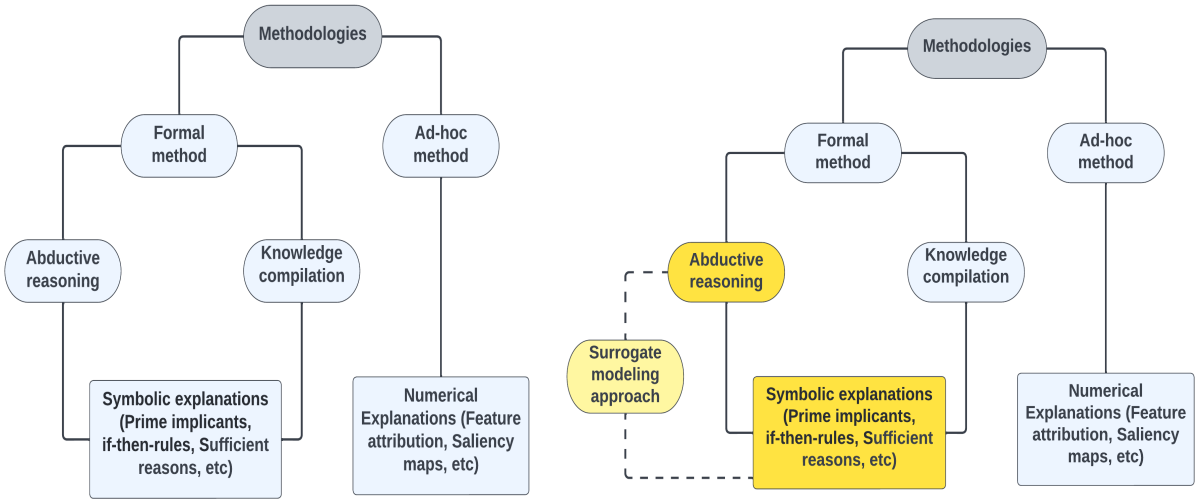
Explanations generated using symbolic reasoning approaches are called symbolic explanations and have been investigated in a number of applications such as generation of rigorous explanations, detection of decision bias and evaluation of counterfactual queries. Symbolic explanations are important as they provide the user with more detailed information beyond a simple numeric score (e.g., model inputs that contributed to the outcome). The existing symbolic explainability methods are mainly model-specific (can only be applied to specific models for which they are intended) and cannot be applied agnostically to any model, which is their main limitation. In the other hand, feature-attribution methods such as LIME [RSG16] and SHAP [LL17] provide the features' importance values for a particular prediction. These values provide an overall information on the contribution of features individually but do not really allow answering certain questions such as: "*What are the feature values which are sufficient in order to trigger the prediction whatever are the values of the other variables?* " or "*Which values are sufficient to change in the instance x to have a different prediction?*". This type of questions is fundamental for the understanding, and, above all, for the explanations to be usable. For example, if a user's application is refused, the user will naturally ask the question: "*What must be changed in my application to be accepted?*". We cannot answer this question in a straightforward manner with the features attribution explanations. Thus, the major objective of our contribution is to provide both symbolic explanations and score-based ones for a better understanding and usability of explanations.

We leverage formal methods to develop a novel model-agnostic method for explaining the prediction of single-label and multi-label classification models. We propose a model-agnostic SAT-based approach for symbolic and score-based explanations named *ASTERYX*. In particular, we focus in this chapter on the symbolic explanations and we present the general framework of *ASTERYX*. We are going to motivate the choices we made for designing a declarative and model-agnostic approach to globally or locally explain the prediction of single-label classifiers. We are interested in two complementary types of symbolic explanations: the *sufficient reasons* (SR_x) which lead to a given prediction (also known as PI-explanation [SCD18b, SCD19, Dar20] or abductive explanation (AXp) [INMS19a]) and the *counterfactuals* (CF_x) allowing to know minimal changes to apply on the data instance x to obtain a different outcome (corresponding to contrastive explanation (CXp) [INAMS20]).

In the following, we present and provide details about **Step 1** and **Step 2** from Figure 3.2 of the approach we propose. In Chapter 3, we provide insights of the approach to handle single-label classifi-

cation problems. Chapter 4 addresses the case of multi-label classification. The description of **Step 3** is given in Chapter 5 for feature-attribution explanations for the single and multi-label settings.

As depicted in Figure 3.1a, and to recall what have been presented in Chapter 2, the methodologies used in XAI are either formal methods or ad-hoc methods. The formal methods generate symbolic explanations such as prime implicants, if-then-rules, sufficient reasons, etc. On the other hand we have ad-hoc methods, developed to explain complex models by providing numerical explanations such as importance scores and saliency maps. Figure 3.1b allows to situate our work w.r.t the methodology used. Our approach relies on a formal method based on the symbolic representation of a model, and use a substitution approach to make it model agnostic. In Section 3.1, we present the general framework of the proposed approach. Section 3.2 is devoted to describe how to associate a symbolic representation to the model used. In Section 3.3, we specify how to generate the type of explanations we would like our model to return and how to reason directly from it. Finally, we complete this chapter by giving some experimental results that show the feasibility of the approach in Section 3.4.



(a) Existing methodologies

(b) Positioning of our work

Figure 3.1: Methodologies proposed within the explainability techniques

3.1 General framework

The main idea is based on associating a symbolic representation that is equivalent or almost equivalent to the decision function of the model to explain. After that, we use this symbolic encoding to generate explanations based on formal methods. We are also interested in post-processing the explanations and select the relevant ones w.r.t to some intuitive desiderata in order to select and rank the explanations according to the user expectations. We investigate a set of fine-grained properties allowing to analyze and select explanations. We also proposed some scores allowing to assess the relevance of explanations and features w.r.t the suggested properties. An illustration of the main components of our approach is given in Figure 3.2. Note that a binarization step may be performed since our approach applies on binary classification problems.

Given a decision function f representing a binary classifier $f : X \to Y$ from the input domain X to the output domain Y, we would like to explain the output $f(x)$ for some input $x \in X$ by providing ex-

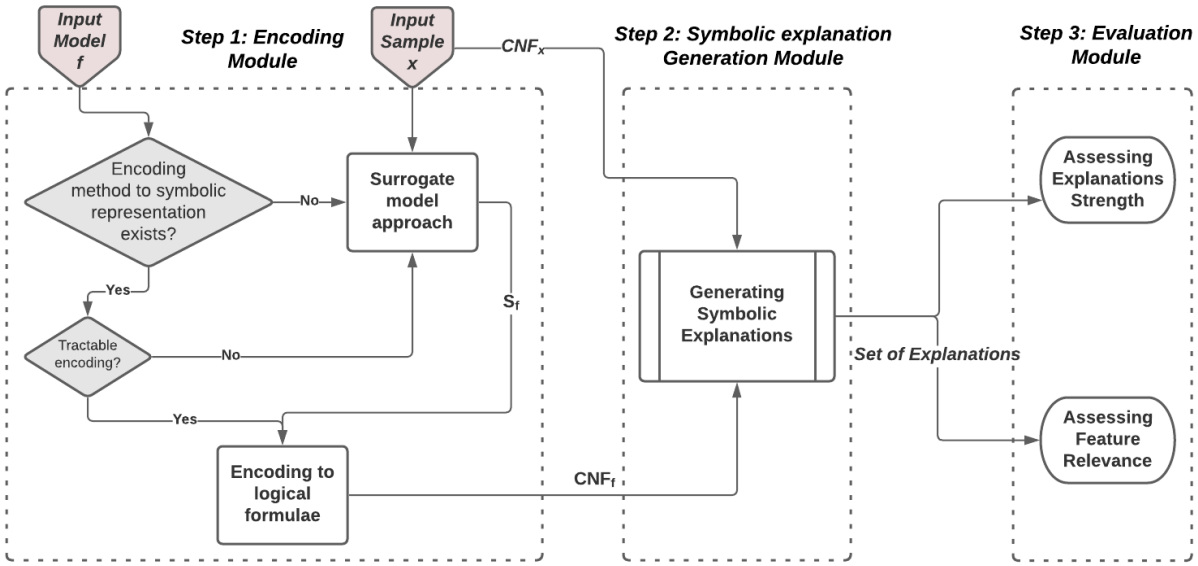


Figure 3.2: A global overview of the proposed approach

planations to justify why the output class was triggered and what modifications of the input can possibly be made to change it. Concretely, our approach proceeds in three major steps as follows:

- **Step 1 (Encoding the classifier into CNF)**: This comes down to associating an *equivalent* symbolic representation Σ_f to f as well as Σ_x encoding the input instance x whose prediction by f is to be explained. Σ_f will serve to generate symbolic explanations in the next step. The encoding is done either using model encoding algorithms if available and if the encoding is tractable as described in Section 3.2.1, or using a surrogate approach as described in Section 3.2.2.

- **Step 2 (SAT-based modeling of the explanation enumeration problem)**: Once we have the CNF representation Σ_f and Σ_x, we model the explanation generation task as a partial maximum satisfiability problem, also known as Partial Max-SAT [BHvM09]. The main idea is to view classifiers as a set of constraints describing a Boolean function that is inconsistent on instances that do not satisfy the constraints. This step, presented in Section 3.3, aims to provide two types of symbolic explanations: SR_x and CF_x. They respectively correspond to Minimal Unsatisfiable Subsets (MUS) and Minimal Correction Subsets (MCS) in the SAT terminology.

- **Step 3 (Explanation and feature relevance scoring)**: This step aims to assess the relevance of explanations by associating scores evaluating those explanations with regard to a set of properties presented in Section 5.2 of Chapter 5. Moreover, this step allows to assess the relevance of features using scoring functions and to evaluate their individual contributions to the outcome. We propose a set of fine-grained properties allowing to analyze and select explanations and a set of scores allowing to assess the relevance of explanations and features w.r.t the suggested properties.

Remark 5. *It is important to notice that the SAT choice is one possibility among other possible oracles such as the ones of Satisfiability Modulo Theories (SMT), Constraint satisfaction problem (CSP), Mixed Integer Linear Programming (MILP)) and so on.*

In the following we present and provide details about **Step 1** and **Step 2** of our approach.

3.2 Encoding of the model

In this section, we present the addressed ways of encoding a model into a symbolic representation. This encoding phase corresponds to **Step 1** in Figure 3.2 where the goal is to encode the decision function associated with the ML model under study into our target representation, which is the Conjunctive Normal Form (CNF) (cf Section 2.3 in **Background and notations** section) in our SAT-base modeling. Two cases are considered in our approach : Either an encoding of classifier f into an equivalent symbolic representation exists (non agnostic case) and is tractable, in which case we can use it, or we consider the classifier f as a black-box and we use a surrogate model to approximate it in the vicinity of the instance to explain x (agnostic case). Such propositional encoding allows to exploit existing algorithms for reasoning about propositional formulae and propose a declarative approach lying on well-known concepts and efficient existing algorithms for the enumeration of Minimal Unsatisfiable Subsets (MUSes) and Minimal Correction Subsets (MCSes).

3.2.1 Direct encoding into CNF

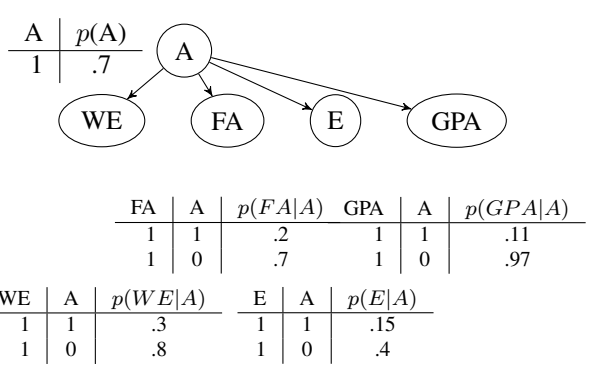
A direct encoding of the classifier f into CNF is possible for some machine learning models such as Binarized Neural Networks (BNNs) [NKR+18], Naive and Latent-Tree Bayesian networks [SCD19]. We give more insights in the following using two examples on how we would use such approaches to encode a specific ML model, for instance, Naive Bayes and Random forests classifiers into a symbolic representation.

CNF encoding of Naive Bays classifiers Authors in [SCD18b] proposed a symbolic approach to explain Naive Bayes classifiers and thus, by proposing two types of explanations called "MC-explanations" for minimal cardinality explanations and "PI-explanations" for prime implicant explanations. The MC-explanations minimize the number of positive features in an instance, while maintaining its prediction. The PI-explanations identify a smallest set of features in an instance that renders the remaining features irrelevant to a prediction. Such approach is based on compiling Naive Bayes Classifiers (NBCs) into a specific symbolic and tractable representation known as Ordered Decision Diagram (ODD).
The objective here is to equip such methods with other types of symbolic explanations, namely, counterfactuals. Given a NBC f whose predictions are to be explained, and once compiled into a decision function in the form of ODD, we show in the following how we encode it into a CNF to use our approach.

Definition 19. ***(Ordered Decision Diagram ODD)*** *An Ordered Decision Diagram is a rooted, directed acyclic graph, defined over an ordered set of discrete variables, and encoding a decision function. Each node is labeled with a variable X_i , $i = 1, \ldots, n$ and has an outgoing edge corresponding to each value x_i of the variable X_i, except for the sink nodes, which represent the terminal nodes.*

An Ordered Binary Decision Diagram (OBDD) is an ODD where all the variables are binary. As shown in Example 16, in case of an OBDD representing the decision function of a binary classifier, a node labeled with variable X_i has two outgoing edges labeled 1 and 0 respectively, and two sinks (class variable), 1-sink and 0-sink. If there is an edge from a node labeled X_i to a node labeled X_j, then $i<j$ (more on tractable representations such as ODDs can be found in [SCD19]). The proposed algorithm for NBC (and some of its variants) has many nice features in terms of tractability, explanation enumeration and formal analysis of classifiers. Authors in [SCD18b] showed how it facilitates the efficient explanation of classifiers and used it to compile such classifier into an ODD then enumerate two types of explanations : the first class is *minimum-cardinality* explanations and the second class is *prime-implicant* explanations.

Example 16. *Figure 3.3a shows a NBC for deciding whether a student will be admitted to a university (class variable: Admit (A)). The features of an applicant are: work-experience (WE), first-time-applicant (FA), entrance-exam (E) and gpa (GPA). In Figure 3.3b, we provide the OBDD representing the classifier decision function f with the variable ordering (WE, FA, E, GPA). Here, the sinks correspond to the values of the class variable (A).*



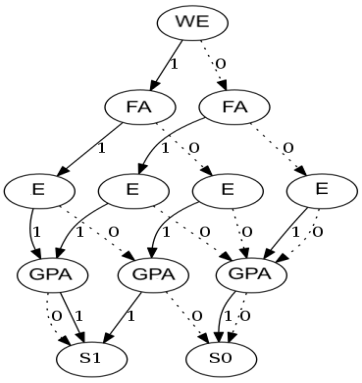
FA	A	$p(FA\|A)$
1	1	.2
1	0	.7

GPA	A	$p(GPA\|A)$
1	1	.11
1	0	.97

WE	A	$p(WE\|A)$
1	1	.3
1	0	.8

E	A	$p(E\|A)$
1	1	.15
1	0	.4

(a) Naive Bayes network classifier f



(b) Ordered Binary Decision Diagram of f

Figure 3.3: A naive Bayes network classifier and its corresponding OBDD.

To explain such Bayesian classifiers following our framework (cf Figure 3.2), we need in the first step to encode the decision diagram into our target representation, which is the Conjunctive Normal Form (CNF) (see Section 2.3).

There are several methods to encode a decision diagram as a CNF formula. For instance in [CNQ03], the authors proposed a method called "Single-Cut-Node" to store a BDD (Binary Decision Diagram) as a CNF. The BDD nodes are modeled as multiplexers. The data inputs of the multiplexer are the children nodes, the selection input is the node variable and the output is the function value which is assigned to an additional CNF variable. A second method called "The No-Cut method" creates clauses starting from f corresponding to the "off-set" and a last method called "The Auxiliary-Variable-Cut" which combines the two previous methods. For the sake of simplicity and clarity, we choose the simplest method which does not involve adding new variables during the encoding process since we want to restrict our explanations to the input variables of the classifier. We implement a simple way to encode the symbolic representation of a classifier as a CNF formula based on the "The No-Cut" method [CNQ03]. In our case, since we are dealing with binary Boolean functions (binary features and class variable), our tractable representation of the decision function f is an OBDD. Recall that we use along with this manuscript positive/true/1 and negative/false/0 interchangeably. Let us first define an "off-set" of a Boolean function and a CNF formula.

Definition 20. ***(Off-set of a Boolean function)*** *The Off-set of a Boolean function f, denoted as f^0, is f^0=$\{v \in \bigcup_{i=1}^{n}\{0,1\}^i | f(v) = 0\}$ If f^0=$\{0,1\}^n$, then f is unsatisfiable. Otherwise, f is satisfiable.*

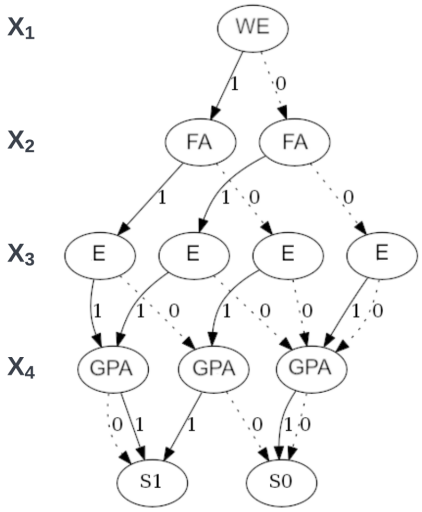
Intuitively, f^0 is the set of counter-models of f. This concept of "off-set" contains the counter-models we need to enumerate in order to construct our CNF's clauses. The OBDD is used to enumerate all the paths from the root to the 0-sink node (the off-set), where each element of f^0 corresponds to a path within it.

Definition 21. ***(CNF encoding of an OBDD)*** *Let f be the decision function encoded by an ordered binary decision diagram $OBDD_f$. Let f^0 be the off-set of $OBDD_f$. We define the obtained CNF formula from $OBDD_f$ as :*

$$\Sigma_f = \wedge \neg e_i \tag{3.1}$$

where $e_i \in f^0$ and $i \in [1, M]$ with $M = |f^0|$.

Example 17 (Example 16 continued)**.** *Given the variable ordering (WE, FA, E, GPA) associated with the OBDD of the example 16, the variable X_1, X_2, X_3 and X_4 correspond respectively to the variable WE, FA, E and GPA. To simplify, let us consider the following notation : $\neg X_2$ means that the variable numbered X_2 in a sample x is instantiated negatively (has the value 0). A decision path (X_1=1, X_2=0, X_3=1, X_4=0) corresponds to a clause $\mathcal{T} = (\neg X_1 \vee X_2 \vee \neg X_3 \vee X_4)$ and is written $(X_1, \neg X_2, X_3, \neg X_4)$. Note that the elements of f^0 are a complete assignation but can be written in a reduced form for the sake of simplicity. The corresponding "off-set" of the OBDD of Example 16 representing the counter models of the OBDD$_f$ is presented in Figure 3.4.*



$$f^0 = \{ \quad (X_1, X_2, \neg X_3, \neg X_4),$$
$$(X_1, \neg X_2, X_3, \neg X_4),$$
$$(X_1, \neg X_2, \neg X_3),$$
$$(\neg X_1, X_2, \neg X_3),$$
$$(\neg X_1, \neg X_2) \ \}$$

(a) Ordered Binary Decision Diagram of f

Figure 3.4: Off-set of a Boolean function represented by an Ordered Binary Decision Diagram.

Proposition 2. *Let f be a binary decision function and OBDD$_f$ its representation. Let also Σ_f be the CNF representation of the decision function f obtained following Definition 21. Then, an interpretation μ is model (resp. counter-model) of Σ_f iff it is mapped to 1 (resp. to 0) by f.*

Proof. The proof is straightforward. Indeed, it is easy to see that the Boolean function f encoded by an OBDD can be equivalently represented as the disjunction of its models (called disjunctive normal form, DNF for short). Let α be the associated formula of f. Similarly, $\neg\alpha$ is equivalently represented as the disjunction of the counter-models of f. Then $\neg(\neg\alpha)$ comes down to α which corresponds to f in conjunctive normal form CNF. □

Let α be the associated formula of f. The intuition is that $\neg\ \alpha \equiv \vee\ e_i$ where $e_i \in f^0$. Then f comes down to negating paths in f^0 ($\vee\ e_i$) allowing to obtain directly f in the form of a CNF. Following Definition 21, we have:

- Every variable of the feature space X= $\{X_1,...,X_n\}$ of the classifier will correspond to a Boolean variable in the CNF Σ_f.

- The class variable Y of the classifier is captured by the truth value of the CNF (Σ_f).

- Modeling a prediction made by the classifier for a given data instance x comes down to the truth value of: CNF ($\Sigma_f \wedge \Sigma_x$) where Σ_x stands for the data instance x encoded as a CNF by a set of unit[9] clauses.

The encoding of Definition 21 guarantees the logical equivalence between the OBDD$_f$ and the obtained CNF Σ_f and Proposition 2 formally states that Σ_f is *logically equivalent* to the function f, i.e, they have the same truth value for each data instance x. Thus, we can assert the following result.

Lemma 1. *Given a binary classifier, a data instance x and the predicted class $f(x)$=y, the formula ($\Sigma_f \cup \Sigma_x$) is SAT iff $f(x)$=1 .*

Example 18. *Let f be the decision function of the classifier represented by an OBDD in Figure 3.4a. The running example shows the CNF formula Σ_f corresponding to the "off-set" of f within the OBDD$_f$, and Σ_x corresponding to the data instance x=(X_1=1,X_2=0,X_3=1,X_4=0).*

$$
\begin{array}{lll}
\Sigma_f & (\neg X_1 \vee \neg X_2 \vee X_3 \vee X_4) & \wedge \\
 & (\neg X_1 \vee X_2 \vee \neg X_3 \vee X_4) & \wedge \\
 & (\neg X_1 \vee X_2 \vee X_3) & \wedge \\
 & (X_1 \vee \neg X_2 \vee X_3) & \wedge \\
 & (X_1 \vee X_2) & \wedge
\end{array}
$$

$$
\begin{array}{lll}
\Sigma_x & (X_1) & \wedge \\
 & (\neg X_2) & \wedge \\
 & (X_3) & \wedge \\
 & (\neg X_4) &
\end{array}
$$

Proposition 3. ***(CNF size)*** *Let Σ_f be the CNF obtained from OBDD$_f$ following Definition 21. The CNF size (number of clauses) is linear in the size of the OBDD$_f$ (number of nodes).*

Proof. Let N be the size of OBDD$_f$ (number of nodes). From Definition 21, we have: Σ_f=$\wedge \neg e_i$ where $e_i \in f^0$ and $i \in [1, M]$ where $M = |f^0|$. Note that for each $e_i \in f^0$ there exists a path within the OBDD from the root to the 0-sink node. The number of clauses involved in Σ_f is equal to the number of those paths. In the worst case, the number of paths is at most equal to 2*N. □

Model enumeration is a polytime operation on OBDDs since they are a subset of DNNFs [DM02]. Besides, the authors in [SCD18b] show experimentally that compiling Bayes network classifiers into ODDs can be handled efficiently and the number of PI explanations remains reasonable for small size feature spaces. Thus, such approaches may face scalability issues when applying them to real-world applications. The authors in [SCD18b] also showed that given a NBC, compiling an ODD representing its decision function is NP-hard.

[9] A unit clause involves only one Boolean variable represented by a literal

CNF encoding of Random Forest classifiers Random Forest classifiers (RF) [Ho95] are tree-based models built using a stochastic approach. The decision trees composing it are grown on randomly selected subspaces of the feature space, in order to further improve predictive performance. The output value of a RF classifier is given from a combination of the trees' predictions, often using the majority vote. RF classifiers are attractive due to their high execution speed and good predictive performance with relatively little hyper-parameter tuning.

The encoding of a random forest into a CNF amounts to encode the decision trees individually and then encode the combination rule (majority voting rule). We only consider binary variables which can either be True or False (1 or 0 respectively).

Encode in CNF every decision tree: The decision trees are seen as directed acyclic graphs whose internal nodes represent propositional variables and whose edges represent assignments to source nodes. A decision tree (DT) in our case represents a Boolean decision function. Recall that the internal nodes of a DT represent a binary test on one of the features. Each leaf of a decision tree is annotated with the predicted class (namely, 0 or 1 for binary classification). A *decision rule* is the solution path going from the root to the leaf leading to the final decision assigned to an input x. We can represent a decision tree as a CNF formula as the conjunction of k clauses $\mathcal{T}$, where a clause is a disjunction of literals of variables in the tree defined on the input features.

Let $x = (x_1, \ldots, x_n)$ be the sample of interest. The CNF encoding of a decision tree is :

$$\Sigma_{DT} = \bigwedge_{j \in [1,k]} \mathcal{T}_j$$
$$\mathcal{T}_j = \bigvee_{i \in [1,n]} l_j^i \tag{3.2}$$

where l_i^j is the value of the literal associated to the variable's value $x_i \in x$ in clause j. Example 19 shows how the Boolean function encoded by a decision tree can be captured into a CNF as the conjunction of the negation of paths leading from the root node to leaves labelled 0.

Encode in CNF the combination rule: Let y_i be a Boolean variable capturing the truth value of the CNF Σ_{DT_i} associated to a decision tree. Hence, the majority rule used in random forests to combine the predictions of m decision trees can be seen as a cardinality constraint[10] [Sin05] that can be stated as follows:

$$y \Leftrightarrow \sum_{i \in [1,m]} y_i \geq t, \tag{3.3}$$

where t is a threshold (usually t=$\frac{m}{2}$). Cardinality constraints have many CNF encodings (e.g. [Sin05, BB03, ANORC13]). To form the CNF corresponding to the entire random forest, it suffices to conjuct the m CNFs associated to the equivalences between y_i and the CNF of the decisions trees, and the CNF of the combination rule.

Definition 22. ***(Equivalence of a classifier and its CNF encoding)*** *A binary classifier f is said to be equivalently encoded as a CNF Σ_f if the following condition is fulfilled: $f(x)$=1 iff x is a model of Σ_f.*

Thus, representing the model as a CNF formula allows the capturing of all the models (resp counter-models) likely to make the formula satisfiable (resp unsatisfiable), and thus, matched with the predictions of the classifier f as reported in Definition 23.

[10] In the case of binary classification, this constraint means that at least t decision trees predicted the positive label.

Tseitin transformation Unfortunately, putting a formula in conjunctive normal form can make this formula exponentially larger than the original one. Hence, we use the Tseitin transformation [Tse83] which converts any propositional formula into a CNF formula of size linear in the size of the starting formula.

3.2.2 Surrogate model encoding into CNF

Motivation

Despite the guarantees of fidelity and completeness offered by direct encoding algorithms which are based on formal methods, a very restricted class of classifiers can be represented directly into a symbolic representation, which makes it hard to use them in practice to explain the different ML classifiers. In addition, their computational complexity makes it hard to use in real-world applications involving a large number of features and parameters.

To overcome those limitations, we propose to integrate a surrogate approach in the encoding phase. The goal is to explain any classifier regardless of the used technique and implementation (model-agnostic). In our setting, no assumptions about the model are made other than that it maps from some known input feature space to a known output domain. The overall goal is to build a surrogate model to faithfully approximate the output around a sample of interest (locally faithful to the classifier). In the same manner as for other approaches such as LIME [RSG16] and Anchor [RSG18], the intuition behind using this simplification approach is that the decision boundary for the black-box can be arbitrarily complex over the whole data space but in the neighborhood of a data point there is a high probability that the decision boundary is clear and simple, and thus, can be captured by a surrogate model.

Local fidelity for a surrogate model

Generally, local surrogate models that are used to explain individual predictions of black-boxs are interpretable models (see Explanation by simplification in Section 1.2.2). For our case, the interpretability of the model is not a criterion. We require from the surrogate model f_S to be i) as faithful as possible to the initial model f (ensures same predictions) and ii) to allow obtaining a tractable symbolic encoding. Thus, the surrogate model must meet a **fidelity-tractability** trade-off. Given a classifier f and a sample data x whose prediction $f(x)$ is to be explained, we want to maximize the local fidelity of a surrogate model f_S, used to approximate f in the neighborhood of x noted $V(x, r)$ (also written V_x for clarity's sake). In the same time, the encoding of f_S must remain tractable in terms of the size of the logical representation and the time to generate it. Since our target encoding is a CNF formula, its size is expressed in terms of the number of clauses composing it and the total number of variables used.

We use the surrogate model f_S to approximate the classifier f in the neighborhood of the instance to be explained. The neighborhood of x within the radius r noted $V(x, r)$ is constructed by sampling data instances within a radius r from x. It is formally defined as $V(x, r)$: $\{v \in X \mid \text{dist}^{11}(x, v) \leqslant r\}$. Note that the value of the radius r is an input information given by the user. We consider mainly the case where a dataset is available in the following but both options can be used.

The intuition behind the explanations generated in the locality of an instance x is to detect what are the reasons used by the model to differentiate two classes that are assigned to similar samples. In case the dataset is not available, we can draw new perturbed samples around x. A similar approach for sampling is presented in [RSG16] where they propose to sample instances around x by drawing elements of x uniformly at random.

Concretely, f_S is built according to the following steps :

[11] dist(x,v) denotes a distance measure between x and v.

1. Select/sample a balanced set of neighbor instances of the given instance x;

2. Once the initial training samples determined, use the original model f as a predictor to determine their predictions;

3. Approximate the ML model chosen by a surrogate model on the pairs of the selected training samples and their corresponding predicted classes;

Given a input sample x whose prediction by a model f is to be explained, the original model f and the surrogate model f_S are supposed to have almost the same input-output behaviour in the vicinity of x. A surrogate model f_S is a logical consequence of f if any model (positive prediction) of f is a model (positive prediction) of f_S and vice-versa. In other words, for any pair (x, y) of the sampled training dataset, we have $f(x) = f_S(x)$.

Definition 23. *For a binary setting, a classifier f_S is locally equivalent to the initial classifier f in the neighborhood V_x if $f(X) = f_S(X)$.*

Two models are said to be locally equivalent if they both predict the same output value for the same samples. Thus, the explanation generated using the surrogate model f_S will be consistent (faithful) to the model f since f_S is locally faithful to the model f.

The evaluation of the faithfulness of the surrogate model f_S to f is based on the difference between the predictions of f and f_S. A low consistency between these predictions means that the outputs of f_S are not consistent with those from the original model f for $x \in V_x$.

We mainly focus in the remaining of this manuscript on the agnostic option used when no direct CNF encoding exists for f or if the encoding is intractable. We make the realistic assumption that the surrogate model is faithful to f in the neighborhood of the sample x to explain.

Random Forest as surrogate model

A machine learning model that can guarantee a good trade-off between faithfulness and tractability (tractable CNF encoding) is the one of random forests (RF).

Given a sample x and its prediction $f(x)$ we follow the steps described previously in order to build a surrogate model f_S that can be converted into logical formulae (interchangeably noted Σ_f or Σ_{f_S}). We determine the neighborhood of x by sampling instances within a radius r of x. We train a random forest on the new dataset composed of pairs (x', y') where $x' \in V(x, r)$ and $y' = f(x')$. Once we have trained a RF f_S to agnostically approximate a classifier f in the locality of x, we need to encode it into a CNF formula following the same steps explained in 3.2.1. Namely, data instances x predicted positively ($f(x)$=1) by the classifier are models of the CNF encoding the classifier. Similarly, data instances x predicted negatively ($f(x)$=0) are counter-models of the CNF encoding the classifier. Such equivalence is tighthly related to the fidelity of the surrogate model. Indeed, a high accuracy that tends towards 100% ensures that the original model and the surrogate one have the same input-output behavior. An illustration of this encoding phase is shown in Example 19

Example 19. *As a running example to illustrate the different steps, we trained a Neural Network model f on the United Stated Congressional Voting Records Dataset*[12]*. In this example, the label Republican corresponds to a positive prediction, noted* 1 *while the label Democrat corresponds to a negative prediction, noted* 0*. An input instance x is defined over the following set of binary features :*

[12] Available at https://archive.ics.uci.edu/ml/datasets/congressional+voting+records.

X_1	*handicapped-infants*	X_9	*mx-missile*
X_2	*water-project-cost-sharing*	X_{10}	*immigration*
X_3	*adoption-of-the-budget-resolution*	X_{11}	*synfuels-corporation-cutback*
X_4	*physician-fee-freeze*	X_{12}	*education-spending*
X_5	*el-salvador-aid*	X_{13}	*superfund-right-to-sue*
X_6	*religious-groups-in-schools*	X_{14}	*crime*
X_7	*anti-satellite-test-ban*	X_{15}	*duty-free-exports*
X_8	*aid-to-nicaraguan-contras*	X_{16}	*export-administration-act-south-africa*

Assume an input instance x=(1,1,1,0,0,0,1,1,1,0,0,0,0,1,0,1) *whose prediction is to be explained. As a surrogate model, we trained a random forest classifier* RF_f *composed of 3 decision trees (decision tree 1 to 3 from left to right in Figure 3.5) on the vicinity of the input sample* x.

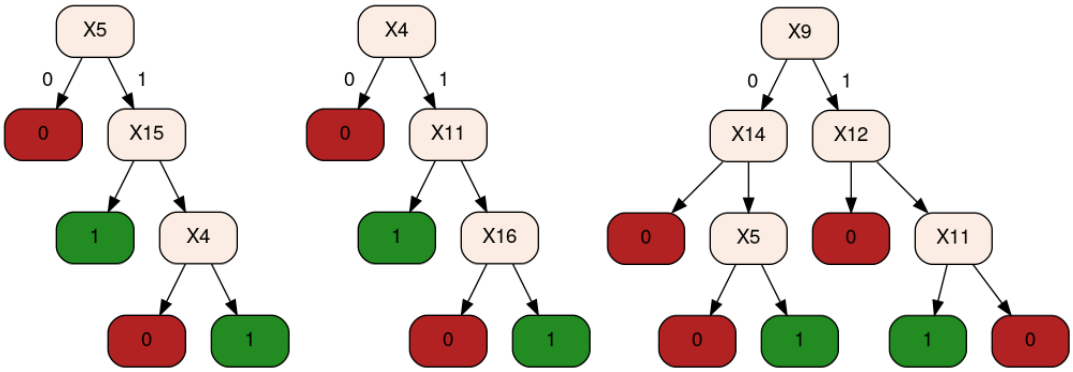


Figure 3.5: A random forest classifier trained on the neighborhood of x

In this example, RF_f *achieved an accuracy of 91.66%. This* RF_f *classifier will later be encoded into CNF to serve for the enumeration of symbolic explanations. Each decision tree (* DT_i *)* $_{i=1..3}$ *represents a Boolean function whose truth value is captured by Boolean variable* y_i. *The Boolean function of* RF_f *is captured by the variable* y. *Note that the encoding of* RF_f *is provided in this example in propositional logic in order to avoid heavy notations. The following formulae illustrate the encoding steps applied to* RF_f*:*

$$DT_1 \quad y_1 \Leftrightarrow (X_5) \wedge (\neg X_5 \vee \neg X_{15} \vee X_4)$$

$$DT_2 \quad y_2 \Leftrightarrow (X_4) \wedge (\neg X_4 \vee \neg X_{11} \vee X_{16})$$

$$DT_3 \quad y_3 \Leftrightarrow (X_9 \vee X_{14}) \wedge (X_9 \vee \neg X_{14} \vee X_5) \wedge (\neg X_9 \vee X_{12}) \wedge (\neg X_9 \vee \neg X_{12} \vee \neg X_{11})$$

$$\text{Majority vote} \quad y \Leftrightarrow (y_1 \wedge y_2) \vee (y_1 \wedge y_3) \vee (y_2 \wedge y_3) \vee (y_1 \wedge y_2 \wedge y_3)$$

3.3 Enumeration of symbolic explanations

In this section we give more insights on how we model the explanation-generation problem and present the setting used to enumerate the desired types of explanations. We provide the formal definitions of the symbolic explanations which are described as a set of selected features but with guarantees of sufficiency and minimality. This enumeration phase corresponds to **Step 2** in Figure 3.2. To generate symbolic explanations, we rely on analyzing inconsistent formulae and focus on finding and enumerating explanations of inconsistency and corrections.

3.3.1 Satisfiability solving for explanation generation

In order to enumerate the symbolic explanations, we propose to model the problem using different concepts from the propositional satisfiability problem (SAT). This choice is justified by our desire to propose a declarative approach, i.e. does not require the implementation of specific algorithms nor dedicated programs with the aim of using SAT solvers as the problem solving engine. Given a test instance, a SAT solver can be viewed as an oracle which answers either positively (for satisfiable instances) or negatively (for unsatisfiable instances). Recall that we are interested in two complementary types of symbolic explanations: *sufficient reasons* (SR_x) which lead to a given prediction and *counterfactuals* (CF_x) allowing to know the minimal changes to apply on the data instance x to obtain a different outcome.
Our approach is based on the reduction of the problem of explaining a prediction to a variant of the SAT problem called Partial-Max SAT [BHvM09] and relies on two very common concepts in SAT which are Minimal Unsatisfiable Subsets (MUSes) and Minimal Correction Subsets (MCSes) to enumerate these explanations. Such a declarative approach allows to exploit the strengths of modern SAT solvers and already existing and proven solutions and algorithms for the extraction of MUSes and MCSes such as [GMP07, LS08, MPMS15, BK15, LPMMS16, BK16, MIPMS16, PMJMS18, NBMS18, BČB18]. In addition, it makes it possible to restrict the explanations only to that which concern the input data x and do not include clauses that concern the encoding of the classifier.

Given an unsatisfiable formula, a maximum satisfiability problem (Max-SAT) returns the maximum number of OR clauses that can be satisfied. Namely, for a CNF with m clauses and n variables, a Max-SAT solution would be an assignment of the n variables to satisfy the maximum number of clauses. A Partial Max-SAT problem is composed of two disjoint sets of clauses where Σ_H denotes the hard clauses (those that could not be relaxed) and Σ_S denotes the soft ones (those that could be relaxed). The aim is finding an assignment to the variables such that no hard clause Σ_H is falsified and the minimum number of soft clauses Σ_S are falsified. In our modeling, the set of hard clauses Σ_H corresponds to Σ_f, the CNF formula encoding the classifier f while the set of soft clauses Σ_S corresponds to Σ_x, the CNF encoding of the data instance x whose prediction $f(x)$ is to be explained. Let Σ_x be the set of *soft clauses*, defined as follows :

- Each clause $\alpha \in \Sigma_x$ is composed of exactly one literal ($\forall \alpha \in \Sigma_x, |\alpha| = 1$).
- Each literal corresponds to a Boolean variable $\{X_i \in X\}$ from the input feature space of f.

Example 20. *Let us consider the input instance* x=(1,1,1,0,0,0,1,1,1,0,0,0,0,1,0,1) *from Example 19. The CNF* Σ_x *associated to it is :*

$$\Sigma_x \quad (X_1) \wedge (X_2) \wedge (X_3) \wedge \neg(X_4) \wedge \neg(X_5) \wedge \neg(X_6) \wedge (X_7) \wedge (X_8) \wedge (X_9) \wedge \neg(X_{10}) \wedge \neg(X_{11}) \wedge \neg(X_{12}) \wedge \neg(X_{13}) \wedge (X_{14}) \wedge \neg(X_{15}) \wedge (X_{16})$$

Recall that since the classifier f is "equivalently" encoded by Σ_f, then a negative prediction $f(x)$=0 corresponds to an unsatisfiable CNF $\Sigma_f \cup \Sigma_x$. Now, given an unsatisfiable CNF $\Sigma_f \cup \Sigma_x$, it is possible to identify the subsets of Σ_x responsible for the unsatisfiability (corresponding to reasons of the prediction $f(x)$=0), or the ones allowing to restore the consistency of $\Sigma_f \cup \Sigma_x$ (corresponding to *counterfactuals* allowing to flip the prediction and get $f(x)$=1). For positively predicted instances, we can simply work on the negation of the decision function of the classifier.

3.3.2 Enumerating sufficient reason explanations (SR_x)

We are interested here in identifying minimal reasons of why the prediction is $f(x)$=0. This is done by identifying subsets of clauses causing the inconsistency of the CNF $\Sigma_f \cup \Sigma_x$ (recall that the prediction $f(x)$ is captured by the truth value of $\Sigma_f \cup \Sigma_x$). A sufficient reason explanation of a given instance x (SR_x) is the minimal subset of feature values that explains why the decision was made for this specific instance. This means that it is enough to fix this group of features so that the model will make the same prediction whatever the values of the remaining features in x are. We formally define the SR_x explanations as follow:

Definition 24. ***(SR_x explanations)*** *Let x be a data instance and $f(x)$ its prediction by the classifier f. A sufficient reason explanation $\tilde{x}$ of x is such that:*

1. $\tilde{x} \subseteq x$ *($\tilde{x}$ is a part of x)*
2. $\forall \acute{x} \in X, \tilde{x} \subset \acute{x} : f(\acute{x})=f(x)$ *($\tilde{x}$ suffices to trigger the prediction)*
3. *There is no partial instance $\hat{x} \subset \tilde{x}$ satisfying 1 and 2 (minimality)*

Intuitively, a *sufficient reason* $\tilde{x}$ is defined as the part of the data instance x such that $\tilde{x}$ is minimal and triggers the prediction $f(x)$.

Example 21. *For example, let us consider a binary classification task for predicting if credit applications should be granted or denied. A model f is trained on attributes such as age, education, working hours, income, debt and accounts. Given an input instance x = (`age=20, education=high school, working hours=30, income=800, debt=200, accounts=5`) and the prediction f(x)= `"loan denied"`. Assume a* ***sufficient reason*** *to trigger the reject decision is as follows SR_x=`{Income=800, Debt=200, accounts=5}`. This means that the reject decision sticks for any instance $\acute{x}$ having the features `income, debt` and `accounts` respectively set to `800, 200` and `5`, and thus whatever the values of the other features are.*

Given Definition 13 of minimal unsatisfiable subsets, it is clear that a MUS for $\Sigma_f \cup \Sigma_x$ comes down to a subset of soft clauses, namely a part of x that is causing the inconsistency (the prediction $f(x)$=0).

Proposition 4. *Let f be a classifier, let Σ_f be its CNF representation. Let x be a data instance predicted negatively ($f(x)$=0) and let $\Sigma_f \cup \Sigma_x$ be the corresponding Partial Max-SAT encoding. Let $SR(x, f)$ be the set of sufficient reasons of x w.r.t f. Let MUS($\Sigma_{f,x}$) be the set of MUSes of $\Sigma_f \cup \Sigma_x$. Then:*

$$\forall \tilde{x} \subseteq x, \tilde{x} \in SR(x, f) \iff \tilde{x} \in MUS(\Sigma_{f,x}) \tag{3.4}$$

Proposition 4 states that each MUS of the CNF $\Sigma_f \cup \Sigma_x$ is a SR_x for the prediction $f(x)$=0 and vice versa.

Proof. The proof is straightforward. It suffices to remember that the decision function of f is equivalently encoded by Σ_f and that the definition of a MUS on $\Sigma_f \cup \Sigma_x$ corresponds exactly to the definition of an SR_x for $f(x)$. Recall that MUSes can only include soft clauses in our modeling. For the first implication (each MUS $\tilde{x}$ is a sufficient reason explanation), it is easy to verify that if $\tilde{x}$ is a MUS then it satisfies the 3 properties of Definition 24 (sufficient reason explanation). Namely, (1) $\tilde{x}$ is part of the data instance x (since the MUS are limited to soft clauses in our modeling), (2) Since $f(x)$=0 iff $(\Sigma_f \cup \Sigma_x)$ is unsatisfiable and $\tilde{x} \in MUS$ then $\forall \acute{x} \in X, \tilde{x} \subset \acute{x} : f(\acute{x})=f(x)$, (3) $\tilde{x}$ is minimal.
For the implication in the opposite direction, it holds since for each sufficient reason $\tilde{x}$, there exists a MUS containing only the elements of $\tilde{x}$ as unit clauses. From Definition 24 we have $\tilde{x} \subseteq x$, $f(x) = 0$ and $\forall \acute{x} \in X, \tilde{x} \subset \acute{x} : f(\acute{x})=f(x)$. We know that $f(x)$=0 iff $(\Sigma_f \cup \Sigma_x)$ is unsatisfiable. Hence, since f(x[$\tilde{x}$])= $f(x)$ then $(\Sigma_f \cup \Sigma_{x[\tilde{x}]})$ is unsatisfiable. Consequently, $\tilde{x}$ is a MUS ($\tilde{x}$ is minimal and causes the inconsistency). □

Example 22 (Example 19 continued)**.** *Given the CNF $\Sigma_f \cup \Sigma_x$ associated to RF_f from Example 19 and the input x=$(1,1,1,0,0,0,1,1,1,0,0,0,0,1,0,1)$, we enumerate the SR_x for $f(x)$=0 (x is predicted as Democrat). There are three SR_x:*

- *SR_x^1="X_4=0 AND X_5=0" (meaning that if the features physician-fee-freeze (X_4) and el-salvador-aid (X_5) are set to 0, then the prediction is Democrat) ;*
- *SR_x^2="X_{12}=0 AND X_5=0" (meaning that if the features education-spending (X_{12}) and el-salvador-aid (X_5) are set to 0, then the prediction is Democrat) ;*
- *SR_x^3="X_4=0 AND X_{12}=0 AND X_9=1" (meaning that if the features physician-fee-freeze (X_4) and education-spending (X_{12}) are set to 0, then the prediction is Democrat) ;*

It is easy to check for instance that if X_4=0 and X_5=0 then decision trees DT_1 and DT_2 of Figure 3.5 predict 0 leading the random forest to predict 0 thanks to the majority vote.

3.3.3 Enumerating counterfactual explanations (CF_x)

A counterfactual explanation of a given instance x (CF_x) is the minimal subset of features to modify to alter the outcome of the black-box model. It is an actionable explanation provided to the user in the form of data instances that would have received a different outcome. A CF_x distinguishes between variables that need to be modified (activated for binary classifiers) and the ones that need to remain unchanged (deactivated for binary classifiers).

Let us formally define the concept of counterfactual explanation.

Definition 25. *(CF_x Explanations) Let x be a complete data instance and let $f(x)$ be its prediction by the decision function of f. A counterfactual explanation $\tilde{x}$ of x is such that:*

1. *$\tilde{x} \subseteq x$ ($\tilde{x}$ is a part of x)*
2. *$f(x[\tilde{x}])$= 1-$f(x)$ (prediction inversion)*
3. *There is no partial instance $\hat{x} \subset \tilde{x}$ satisfying 1 and 2 (minimality).*

In Definition 25, the term $x[\tilde{x}]$ denotes the data instance x where variables included in $\tilde{x}$ are inverted.

Example 23 (Example 21 continued)**.** *Keeping up with the task of classifying loan applications as accepted/rejected and giving the same input x and the same prediction from the previous example, a counterfactual explanation to change the reject decision "loan denied" into "loan granted" is CF_x= (Income=1000, accounts=3). This means that the loan will be approved if the user increases his income by 200€ and closes two accounts.*

Most approaches proposed within the literature (e.g. [DCL+18, MCV+18, MST20, WMR17]) to find counterfactual explanations are based on solving optimization problems. Thus, the generated explanations may be considered heuristic and not fully trustable. Recently, there has been work on a applying formal reasoning to generate counterfactuals. Authors in [INAMS20] address the "Why Not?" question and proposed an approach based on abductive reasoning to find contrastive explanations (CXp), i.e. find a change of feature values that guarantees a change of prediction.
Following our modeling, an MCS for $\Sigma_f \cup \Sigma_x$ comes down to a subset of soft clauses denoted $\tilde{x}$, namely a part of x that is enough to remove (or reverse in our case) in order to restore the consistency (flip the prediction $f(x)$=0 to $f(x[\tilde{x}])$=1). Recall that the term $x[\tilde{x}]$ denotes the data instance x where variables included in the counterfactual $\tilde{x}$ are inverted.

Proposition 5. *Let f be the decision function of a ML classifier, Σ_f be its CNF representation, x be a data instance predicted negatively and $\Sigma_f \cup \Sigma_x$ the corresponding Partial Max-SAT encoding of $f(x)$=0. Let $CF(x, f)$ be the set of counterfactuals of x w.r.t $f(x) = 0$. Let MCS($\Sigma_{f,x}$) the set of MCSes of $\Sigma_f \cup \Sigma_x$. Then,*

$$\forall \tilde{x} \subseteq x, \tilde{x} \in CF(x, f) \iff \tilde{x} \in MCS(\Sigma_{f,x}) \tag{3.5}$$

Proposition 5 states that each MCS of the CNF $\Sigma_f \cup \Sigma_x$ represents a CF $\tilde{x} \subseteq x$ for the prediction $f(x)$=0 and vice versa.

Proof. Recall that MCSes can only include soft clauses in our modeling. For the first implication (each MCS $\acute{x}$ is a counterfactual explanation), it is easy to check that if $\acute{x}$ is an MCS then it satisfies the 3 properties of Definition 25 (counterfactual explanation). Namely, (1) $\acute{x}$ is part of the data instance x (since the MCS are limited to soft clauses in our modeling), (2) Since $f(x)$=0 iff ($\Sigma_f \cup \Sigma_x$) is unsatisfiable and $\acute{x} \in MCS$ then $f(x[\acute{x}])$= 1-$f(x)$, (3) $\acute{x}$ is minimal.
For the implication in the opposite direction, it is enough to see that for each counterfactual $\acute{x}$, there exists an MCS containing only the elements of $\acute{x}$ as unit clauses. From Definition 25 we have $\acute{x} \subseteq x$, $f(x)$=0 and $f(x[\acute{x}])$=1. We know that $f(x)$=0 iff ($\Sigma_f \cup \Sigma_x$) is unsatisfiable. Hence, since $f(x[\acute{x}])$=1 then ($\Sigma_f \cup \Sigma_{x[\acute{x}]}$) is satisfiable. Consequently, $\acute{x}$ is an MCS ($\acute{x}$ is minimal and restores the consistency). □

Example 24 (Example 19 continued)**.** *Given the CNF $\Sigma_f \cup \Sigma_x$ associated to RF_f from Example 19 and the input x=(1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1), we enumerate the counterfactual explanations to identify the minimal changes to alter the outcome vote Democrat to Republican. The reading of such explanations is as follows : in order to force the prediction to be Republican (1), it is enough to alter x by activating/deactivating the variables included in CF_x. There are four $\boldsymbol{CF}_x$ such that :*

- *CF_x^1="X_4=0 AND X_{12}=0" : activate (set to 1) physician-fee-freeze (X_4) and education-spending (X_{12}) while keeping the remaining values unchanged);*
- *CF_x^2="X_5=0 AND X_{12}=0" : activate el-salvador-aid (X_5) and education-spending (X_{12}) while keeping the remaining values unchanged);*
- *CF_x^3="X_5=0 AND X_9=1" : activate (set to 1) el-salvador-aid (X_5) and deactivate (set to 0) mx-missile (X_9) while keeping the remaining values unchanged);*

- CF_x^4="X_4=0 *AND* X_5=0" : *activate physician-fee-freeze* (X_4) *and el-salvador-aid* (X_5) *while keeping the remaining values unchanged);*

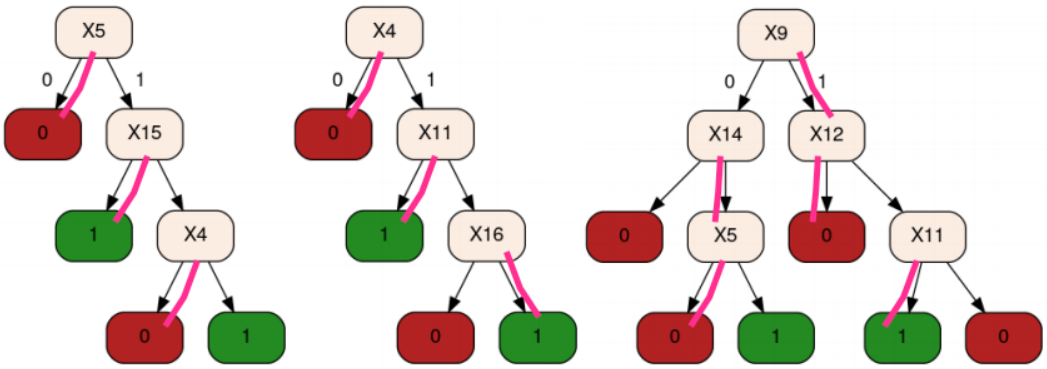


Figure 3.6: The random forest paths set by x

It is easy to see that the four $\mathbf{CF}_x$ *allow to flip the negative prediction associated with* x*. Indeed, in Figure 3.6, the pink lines show the branches of the trees that are fixed by the current input instance* x*. Clearly, according to* CF_x^1="X_4=0 *AND* X_{12}=0"*, if we set* X_4=1 *and* X_{12}=1 *then this will force* DT_2 *and* DT_3 *to predict* 1 *making the prediction of the random forest flips to* 1*.*

3.3.4 On enumerating sufficient reasons and counterfactuals

The results presented in Section 3.3.2 and 3.3.3 establishing that counterfactual and sufficient reason notions correspond to MCS and MUS respectively within Partial MaxSAT-based setting allow us to exploit the state-of-the-art algorithms for the extraction and enumeration of MUSes and MCSes. The decision problem of checking if a propositional formula is satisfiable (SAT) is known to be NP-complete [Coo71]. An unsatisfiable CNF formula can have an exponential number of MCSes and MUSes of the instance. The time needed to this complete computation is exponential as well. However, this number in practice is often relatively small, allowing for complete enumeration of the MCSes of a given instance. From a computational point of view, the enumeration of MCSes presents a real challenge. Yet, the task of finding MCSes is often easier than the one of finding MUSes directly because in practice, it is easier to find satisfiable subsets of constraints than unsatisfiable subsets [LS08].

However, advances in constraint solver technologies have allowed the development of more efficient algorithms [BL03, DRGM10, MSHJ+13, GLM14, BDTK14, MPMS15, MIPMS16] which can handle problems with several million clauses and variables, allowing them to be efficiently used in many applications and made the resolution of some previously impossible instances feasible. Approaches such as [LS09, MSHJ+13, BDTK14, PMJMS17, PMJMS18] calculate all MCSes. The main principle of these methods is to iteratively block the MCS found in order to avoid recalculating it. To do so, these methods match to each MCS found, a new clause formed by the disjunction of the literals of the clauses forming it. The aim is to ensure that at least one of the MCS clauses will be activated in the solutions of the following enumerations.

A second category of methods proposed to enumerate MCSes is the one based on the duality hitting set. For instance, the `CAMUS` system [LS08] uses this relation to compute all MUSes in two steps. The first step is to compute all the MCSes of the unsatisfiable formula Σ. The second one is to find MUSes by computing the minimal hitting sets of the MCSes.

Duality of sufficient reasons and counterfactuals

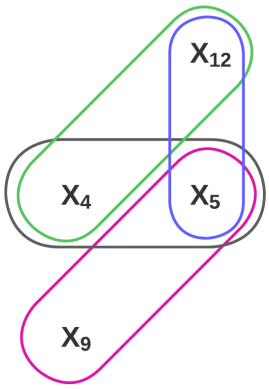
Given a collection Ω of sets from some finite domain D, a hitting set (denoted HS) of Ω is a set of elements from D that "hits" every set in Ω by having at least one element in common with it. A formal definition is given in Definition 26.

Definition 26. *Given $\Omega \subseteq 2^D$, a hitting set H of Ω is $H \subseteq D$ such that $\forall S \in \Omega, H \cap S \neq \emptyset$.*

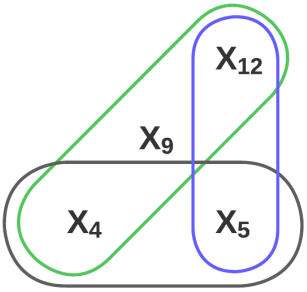
A hitting set H is said to be *minimal*, denoted MHS, if no subset of H is a hitting set of Ω. In other words, no element can be removed from H without losing the property of being a hitting set. A hitting set H is *minimum* if it has the smallest size over all hitting sets.

Given a CNF formula Σ, the set of its MUSes(Σ) and the set of its MCSes(Σ) are "hitting set duals" of one another [13]. The minimal hitting set duality between MUSes and MCSes is a well known relationship [Rei87, BL03] used in many works (e.g. [LS08],[BS05]). Concretely, every MUS hits every MCS. Dually, every correction subset hits every unsatisfiable subset. The duality of explanations and their enumeration has been also investigated in [INAMS20] where they exploit it to establish a duality relationship between abductive and contrastive explanations. The same relation stands between SR_x and CF_x explanations which allows us to exploit any algorithm for computing MUSes/MCSes to enumerate both kinds of explanations. The enumeration of *sufficient reasons explanations* of $\Sigma_f \cup \Sigma_x$ can be done using the hitting set duality over the counterfactuals explanations.

Example 25. *Given a set of MCSes enumerated for the CNF Σ of Example 24 and composed of four reparation sets : MCSes(Σ)= {{X_4, X_{12}},{X_5, X_{12}} ,{X_4, X_5}, {X_5, X_9}}, we select the variable X_9 to be contained in a growing MUS. It appears only in the last MCS of the set. To ensure that X_9 is not redundant, we remove X_5 from the remaining MCSes. This leaves {{X_4}, {X_{12}}} as the remaining sub-problem. The set of MUSes(Σ) computed as the hitting set of MCSes(Σ) as described above and illustrated in Figures 3.7a and 3.7b is presented as follows :*



(a) Set of MCSes from Example 24.



(b) Set of MUSes from Example 22.

Figure 3.7: Set of MUSes and MCSes of the CNF Σ from Example 19.

An example of minimal hitting set (yet not minimum) in the set of MUSes(Σ) is {X_4, X_9, X_{12}}. In the same set MUSes(Σ), {X_4, X_5} and {X_5, X_{12}} are the only minimum hitting sets.

[13] A subset α of Σ is MUS iff α is a minimal hitting set of MCSes(Σ), a subset p of Σ is an MCS iff p is a minimal hitting set of MUSes(Σ)

$$\text{MCSes}(\Sigma) = \{\{X_4, X_{12}\},\{X_5, X_{12}\},\{X_4, X_5\}, \{X_5, X_9\}\}$$

$$\text{MUSes}(\Sigma) = \{\{X_4, X_5\},\{X_5, X_{12}\},\{X_4, X_9, X_{12}\}\}$$

In our case, we can use any state-of-the-art algorithm to enumerate MCSes for Partial Max-SAT formulae. We use the `EnumELSRMRCache` tool that implements the boosting algorithm for MCSes enumeration proposed in [GIL18], which introduces a technique that improves the performance of the best approaches for enumerating the MCSes of an inconsistent CNF formula. This method implements the model rotation paradigm [BMS11, NRS14, BK15, NBMS18] allowing it to recursively compute sets of MCSes in a heuristic and efficient way. The `EnumELSRMRCache` tool is implemented in C++ and uses Minisat[14] as backend SAT solver. We also use the `LBX-like MCS` enumerator module from the `PySAT` toolkit [IMMS18][15] framework implements a prototype of the LBX algorithm for the computation of a minimal correction subset (MCS) and/or MCS enumeration for partial MaxSAT formulae. The LBX abbreviation stands for literal-based MCS extraction algorithm, which was proposed in [MPMS15]. The module can use any SAT solver available in `PySAT` toolkit [IMMS18]. As for the enumeration of sufficient reasons, it is associated to the one of MUSes. In our case, the enumeration of MUSes is achieved by computing all the minimal hitting sets of all the MCSes enumerated.

Complexity of SR_x and CF_x enumeration

The complexity associated with the enumeration of a sufficient reason or a counterfactual explanations amounts to the complexity associated respectively to the enumeration of a MUS or an MCS (Propositions 4 and 5). These results already exist in the literature. The extraction of a MUS has a complexity of FP^{NP} and checking whether there exists a MUS of size $\leq k$ is of complexity Σ_2^P-complete [Gup06, Lib05]. The task of finding the smallest MUS has been studied in several papers such as [IPLMS15, MLA+05] and its complexity is $FP^{\Sigma_2^P}$.

Lemma 2. *Let $\Sigma_f \cup \Sigma_x$ the CNF encoding of the classifier f and the data x. Finding a sufficient reason explanation is a problem of complexity FP^{NP} and finding the smallest SR explanation is in $FP^{\Sigma_2^P}$.*

Proof. According to proposition 4 that states that a MUS is a sufficient reason explanation, finding a sufficient reason is equivalent to finding a MUS, hence, it has the same complexity. The same reasoning applies for finding the smallest SR explanation. □

Checking whether a subset of a CNF formula is an MCS is DP-complete[16] [CT95] and the computation of an MCS is a problem of complexity FP^{NP}.

Lemma 3. *Let $\Sigma_f \cup \Sigma_x$ the CNF encoding of the classifier f and the data x. Let $f(x) = 0$, then finding a counterfactual explanation is a problem of complexity FP^{NP}.*

Proof. Due to Proposition 5 which states that counterfactual explanations corresponds to MCSes, finding a counterfactual explanation is equivalent to finding an MCS, hence, it has the same complexity. □

To sum up, we model the enumeration of symbolic explanations as a Partial Max-SAT problem which allows us to exploit any existing algorithms for the enumeration of MUSes and MCSes such as [LS08, BMS12, NRS13, MSHJ+13, BK15, WH13]. Since we are using MUSes and MCSes as a basis

[14] MiniSat is a open-source SAT solver `http://minisat.se/`

[15] `PySAT` toolkit is designed for simple, fast, and effective Python-based prototyping using SAT oracles. It is composed of four modules which are wrappers for the code originally implemented in the C/C++ and Python languages.

[16] A problem $\mathcal{P}$ belongs to the class DP if it can be written as $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ with $\mathcal{P}_1 \in$ NP and $\mathcal{P}_2 \in$ coNP.

for providing interpretable explanations and thanks to the hitting set duality between them, it is possible to first enumerate MCSes and then use the minimal hitting set duality for computing the MUSes of a formula such as in [BS05, LS08]. Such enumeration involves calling SAT solvers for each explanation. This constraint might cause scalability problems since a call to a SAT-solver amounts to the resolution of an NP-complete decision problem. Recent works [SSDC20, AKM20a, MSGC+20, IIM20, ABB+21, Ign20, CSGD20b, IM21] study same families of classifiers whose explanations can be enumerated either in a polynomial time or at least, efficiently in practice.

3.3.5 Beyond SR_x and CF_x explanations

The symbolic explanations (SR_x and CF_x) directly enumerated are complementary to the feature attribution explanations in the sense that they answer questions beyond the contribution of input features into a decision. In addition, once enumerated, such explanations can be exploited to answer requests such as *what are the irrelevant features for a given decision ?* and *what are the features to have complete explanations?* For instance, such requests are useful for system builders (data scientists, developers, etc) to check whether an algorithm is correctly relying on the right variables. In addition, this kind of information is necessary to have confidence and gain trust in the model before it is deployed. It is also useful to regulatory bodies (e.g public organization or government agencies) to know if a change in a legally sensitive feature (e.g. race, gender, country of origin) produces a change in the model's outcome.

Based on SR_x and CF_x explanations, one can answer other XAI queries such as what are the relevant/irrelevant features for a given $f(x)$ prediction.

Irrelevant features *Irrelevant features* are meant to answer the question *what are the irrelevant features for a given decision ?* An irrelevant feature is a feature that do not appear in any explanation of a prediction $f(x)$.

Definition 27. *(Irrelevant features) Let $f(x)$ be the prediction of a sample x by a model f, X the input feature space of f and $E(x, f)$ the set of explanations associated to $f(x)$ (either SR_x or CF_x explanations). The set of irrelevant features $X^{\mathcal{IRL}}$ is formally defined as : $\{X_i \in X \mid X_i \notin e_i, \forall e_i \in E(x, f)\}$.*

Example 26 (Example 19 continued)**.** *Let us continue with the United Stated Congressional Voting records dataset. The input space X is composed of 16 features X=$\{X_1, \ldots, X_{16}\}$. Given the set of SR_x and CF_x explanations in Examples 22 and 24, and the sample x from Example 19, the set of irrelevant features is : $X^{\mathcal{IRL}} = \{X_1, X_2, X_3, X_6, X_7, X_8, X_{10}, X_{11}\}$.*

Relevant features A relevant feature is defined as a feature involved in at least one explanation $e_i \in E(x, f)$.

Definition 28. *(Relevant features) Let $f(x)$ be the prediction of a sample x by a model f, X the input feature space of f and $E(x, f)$ the set of explanations associated to $f(x)$ (either SR_x or CF_x explanations). The set of relevant features $X^{\mathcal{RLV}}$ is formally defined as : $\{X_i \in X \mid \exists e_i \in E(x, f)$ s.t $X_i \in e_i\}$.*

Example 27 (Example 26 continued)**.** *Let us continue with the United Stated Congressional Voting records dataset, given the set of irrelevant features from the previous Example 26, the set of relevant features would be : $X^{RLV} = \{X_4, X_5, X_9, X_{12}\}$.*

Remark 6. *It is clear that the set of relevant features $X^{\mathcal{RLV}}$ is the complement of $X^{\mathcal{IRL}}$ in X.*

Necessary features A necessary feature is a feature appearing in every explanation.

Definition 29. *(Necessary features) Let $f(x)$ be the prediction of a sample x by a model f, X the input feature space of f and $E(x, f)$ the set of explanations associated to $f(x)$ (either SR_x or CF_x explanations). The set of necessary features X^{NCS} is formally defined as :* $\{X_i \in X \mid X_i \in e_i, \forall e_i \in E(x, f)\}$.

A necessary feature denotes a highly relevant feature since it is involved in every explanation.

3.4 Experimental study

This section presents an experimental evaluation to verify the tractability of the proposed approach in this chapter. The evaluation will follow the processing flow proposed in this chapter (surrogate modeling, encoding into CNF and explanation enumeration). The experimental protocol followed is described in the following: We considered a selection of datasets known from the literature and publicly available and can be found on Kaggle (https://www.kaggle.com/) or UCI (http://archive.ics.uci.edu/ml/). The studied datasets deal with binary classification and are listed in Table 3.1. No pre-processing was performed on the data except the binarization of variables. We transform the data using a binary threshold[17]. The binarization threshold (T) used was defined empirically as follows: given a starting set of values (such as the mean, the median, etc.) associated to each variable, we choose the threshold T which optimizes the accuracy of the model trained on binarized data using T. For instance, the widely used standard "MNIST" dataset [18] composed of handwritten digit images of size 28 × 28 pixels was binarized using a threshold set to $T = 127$.

Dataset	#instances	#features	data type
MNIST	70000	784	Images
MONK's Problems	181	16	Numerical
Spect Heart	160	22	Numerical
Congressional US Voting	435	16	Numerical
Breast Cancer	287	48	Numerical

Table 3.1: Properties of the datasets used.

All experiments were performed on machines equipped with an Intel Core i7-7700 (3.60GHz ×8) processors, with 32 Gb of RAM and under the Linux Cen-tOS operating system. The time-out has been set to 600 seconds for each execution of an enumeration algorithm.

To enumerate the explanations, our approach principally needs the black-box (f), the instance to explain (x) and the radius r to define the neighborhood of x. We trained our own black-box models (f) for the different dataset and used Multi-layer Perceptron (MLP), Decision Tree (DT) and Logistic Regression (LR) classifiers from the Python library Scikit-Learn in its version v0.22.1. The hyper-parameters of different classifiers were set to their default values. We also used *"one-vs-all"* Binary Neural Network (BNN)[19] classifiers on the MNIST dataset to recognize digits (0 to 9) using the pytorch implementation[20] of the Binary-Backpropagation algorithm BinaryNets [HCS+16]. As for the surrogate models, we used random forest (RF) from the Scikit-Learn in its version v0.22.1. A wide range of hyper-parameters for RFs were explored in attempts to reach relatively the best fidelity of the surrogate models

[17] All values above the threshold are marked as 1 and all values equal to or below the threshold are marked as 0.

[18] MNISTdatasetavailableathttp://yann.lecun.com/exdb/mnist/

[19] defined as a neural networks with binary weights and activations at run-time

[20] available at: https://github.com/itayhubara/BinaryNet.pytorch

associated to the different classifiers. The search for the main hyper-parameter to (best cross validation score) was done through a grid search procedure using the GridSearchCV[21] from Scikit-Learn. The values for the number of decision trees and maximum depth of the RF tested were ranging in [3,40] and [4,100] respectively.

3.4.1 Results

We report the following results by setting the following parameters nb_trees=10 and max_depth=24 for the surrogate classifier. Each surrogate model was trained on the vicinity of an input sample x using different values of radius (r). For the MNIST dataset, the experiments were conducted on an average of 1500 to 2500 instances picked randomly. As for the rest of the datasets, all instances were used to test the approach.

Impact of hyper-parameters on the fidelity of surrogate model

We present in this section the results of experiments conducted on the MNIST dataset to show the impact of the different hyper-parameters used on the fidelity of the surrogate model.

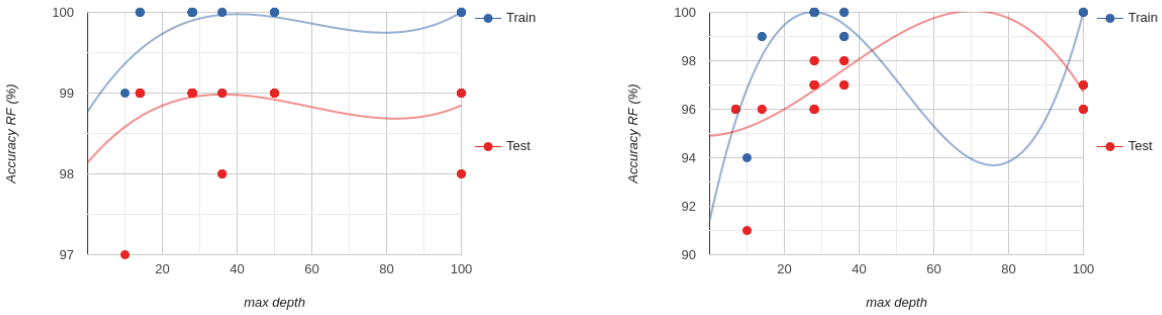
The max_depth represents the maximum allowed depth of each tree in the forest. Note that the deeper the tree, the more splits it has (i.e it captures more information about the data). Figure 3.8 shows the results of experiments conducted to find a suitable max_depth to improve the accuracy. The blue curve represents the accuracy when the classification model is built on the training dataset and the red one corresponds to the accuracy of the model on the test data. As shown on the different plots of Figure 3.8, the accuracy of RF is best for values of max_depth between 10 and 20.

The $ntree$ parameter (also referenced to as nb_trees) represents the number of decision trees in RF. The classification accuracy of a RF can be improved when using a large number of trees but it also results in a more important training time. Figure 3.11 shows the results of experiments conducted to find a suitable $ntree$ to balance accuracy and computation time. The green curve represents the accuracy of the classification model on the training dataset and the orange one corresponds to the accuracy of the model on the test data. As shown on Figure 3.11, the accuracy of classification is not significantly improved when $ntree$ is greater than 10 but still reached at least 92%. Thus, the $ntree$ was set as 10 during the evaluation.

Figure 3.9 shows the number of CF_x explanations with respect to the neighborhood size $|V_x|$. The blue and red bars correspond respectively to the case when a maximum size of $|V_x|$ is used and when not. As shown in Figure 3.9, the instances having a large number of neighbors have a bigger total number of explanations, in comparison to when an upper bound for the vicinity size is used. Since in this experiment we propose local explanations, a maximum number of neighbors was set during the experimentation to preserve the "local" aspect of the explanations as well as to optimize their enumeration (all explanations).

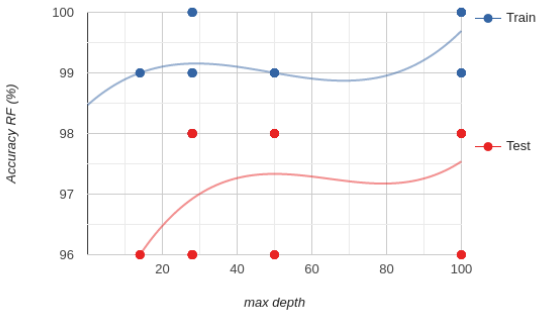
Figure 3.10 shows the performance of RF classification model trained on a set of neighbors obtained using different radius values. We can observe that the model has good performance for the different sizes of V_x and seems to not be significantly improved for neighborhoods composed of more than 200 instances in general. As shown in Figure 3.9, a large neighborhood generates a large set of explanations. Thus, to balance the accuracy of RF and the size of the output set of explanations, we set the maximum size of the locality of an instance to 200.

[21] GridSearchCV exhaustively generates candidates from a grid of parameter values specified with the param_grid parameter.

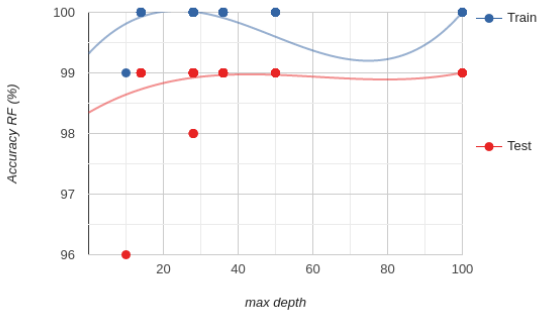


(a) RF trained on MNIST_1

(b) RF trained on MNIST_8



(c) RF trained on MNIST_9



(d) RF trained on MNIST_0

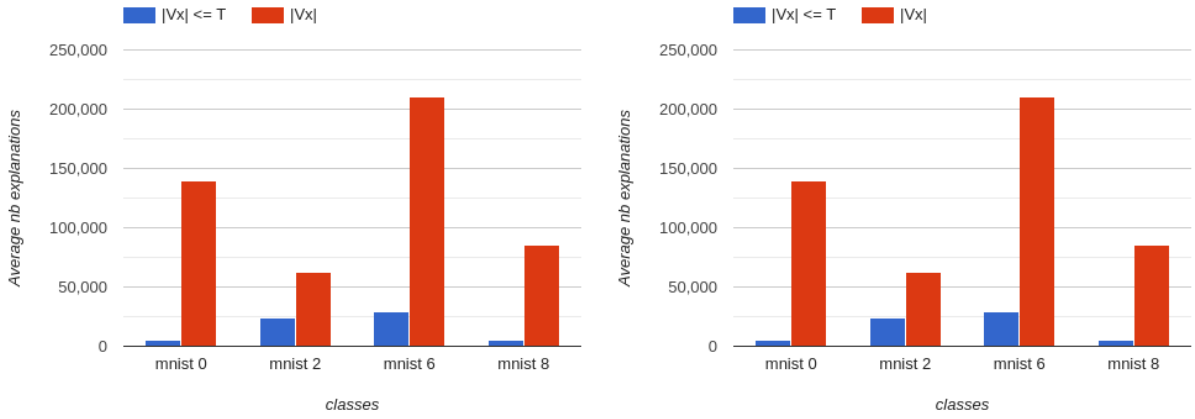
Figure 3.8: The classification accuracy of RF models with respect to the max_depth parameter.

Evaluating the CNF encoding feasibility

We report our results regarding the encoding phase to generate the CNF formulae in terms of its size represented in number of variables and clauses (Vars/CLs) and also the encoding time (in seconds). We use the Tseitin Transformation [Tse83] to encode the propositional formulae associated to our surrogate model into an equisatisfiable CNF formulae, the input format required to use the SAT solvers. For the MNIST dataset, the predictions are made using [22] the "one-vs-all" BNN classifiers trained to recognize the digits. The results reported here concern the "0","2","6" and "8" digits but they are similar for the other digits. For the other datasets, a MLP classifier is used. Note that a maximum number of neighbor instances set to 200 is used to limit our region of interest around an instance x.

Table 3.2 shows that the generated random forest classifiers provide interesting results in terms of fidelity (high accuracy of the surrogate models) and tractability (size of the CNF encoding). This ensures that the explanations generated locally are relevant for the original model having the same behavior. In Table 3.2, the size of CNF is expressed as *number of variables/number clauses*. We can see that the number of variables and clauses of CNF formulae remains reasonable and easily handled by the current SAT-solvers which confirms the feasibility of the approach.

[22] the results for the other digits are similar but can not be reported here because of space limitation



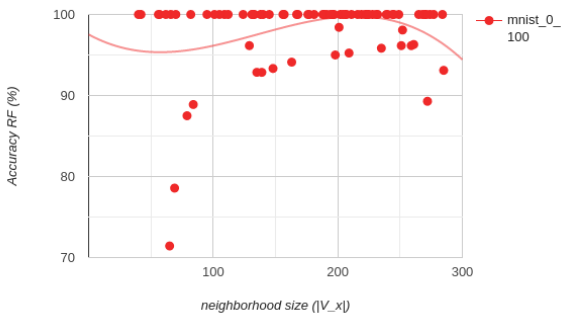
(a) Radius set to 150.

(b) Radius set to 250.

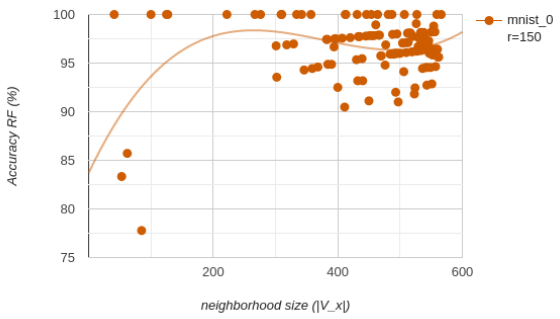
Figure 3.9: Example to compare the average total number of CF_x explanations with respect to the neighborhood size when it is limited to a maximum number ($|V_x| \leq T$) or not limited.

Radius (r)	$\|V_x\|$	Dataset name	Acc (%)	MIN $\|\Sigma\|$ (Vars/CLs)	AVG $\|\Sigma\|$ (Vars/CLs)	MAX $\|\Sigma\|$ (Vars/CLs)	MIN $runtime$ (s)	AVG $runtime$ (s)	MAX $runtime$ (s)
100	60	MNIST_0	97.4	141/837	449/930	841/1079	0.8916	0.9739	1.1667
	69	MNIST_2	95.25	98/825	689/1025	1359/1624	0.904	1.1	1.53
	70	MNIST_6	96.7	124/833	783/1060	1443/1874	0.913	1.14	1.62
	169	MNIST_8	95.21	1219/1230	1615/2318	2043/3499	1.26	1.78	3.68
150	160	MNIST_0	97.7	147/843	1475/1934	2083/3657	0.8991	1.1643	1.6446
	167	MNIST_2	91.19	1205/1211	1579/2263	2069/3672	1.32	1.87	3.71
	165	MNIST_6	93.88	1211/1216	1739/2702	2173/3950	1.17	1.6	3.01
	200	MNIST_8	95.1	1529/2070	1795/2798	2133/3725	1.38	1.87	2.92
250	201	MNIST_0	98.87	1744/4944	1979/5540	2176/6066	0.83	1.05	1.51
	>200	MNIST_2	93	1941/5452	2172/6050	2429/6760	0.88	1.06	1.92
	>200	MNIST_6	96	1978/5534	2270/6293	2558/7028	0.82	0.92	1.31
	>200	MNIST_8	95.12	1837/5178	2059/5727	2330/6408	0.74	0.86	1.32
1	33	SPECT	100	1296/3866	1565/4576	2163/6154	2.71	3.06	3.86
2	52		99.04	1317/3918	1776/5160	3139/8846	1.01	2.08	4.09
3	63		97.3	1297/3862	1933/5581	3519/9850	0.89	1.29	2
4	75		93.47	1322/3928	1891/5493	33003/9264	0.752	1.11	1.872
7	112		90.98	1676/4886	2315/6683	2638/7588	85.7	1.11	1.29
22	160		99	2495/7174	2758/7921	3088/8844	1.07	1.214	1.5
5	36	MONKS	91.21	1314/3908	1576/4595	2353/6668	1.09	2.124	4.62
16	181		98	2351/6714	2883/8146	3451/9694	1.66	1.56	2.03
5	26	BREAST CANCER	78.68	1291/3836	1757/5080	2995/8454	1.21	1.76	2.38
10	147		79.82	1327/3942	4605/12891	5868/16416	1.18	2.18	3.84
48	>200		82	5094/14184	6069/16907	7053/19586	2.02	2.5	3.42
2	22	CONGRES. VOTING	100	128/280	1368/4042	1494/4372	0.9	1.73	3.7
5	70		94.67	1291/3836	1853/5347	2997/8462	1.15	1.74	2.64
16	>200		90.82	1313/3893	2024/5783	2873/8122	0.91	1.85	2.45

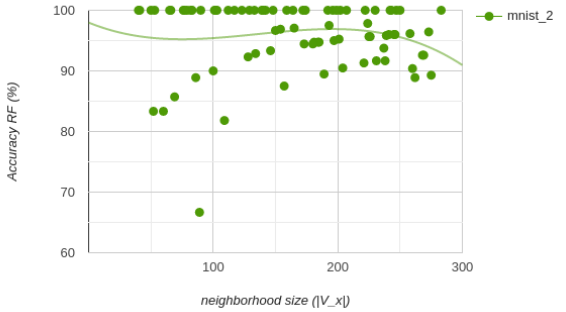
Table 3.2: Evaluating the scalability of the CNF encoding in terms of size (number of variables and clauses) and encoding time in seconds for different datasets using different radius values.



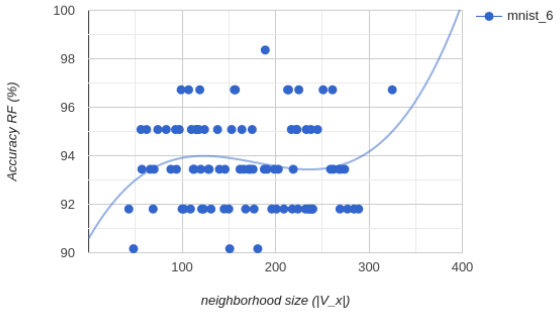
(a) MNIST_0 (r=100)



(b) MNIST_0 (r=150)



(c) MNIST_2 (r=100)



(d) MNIST_6 (r=100)

Figure 3.10: The classification accuracy of RF models with the respect to the neighborhood size.

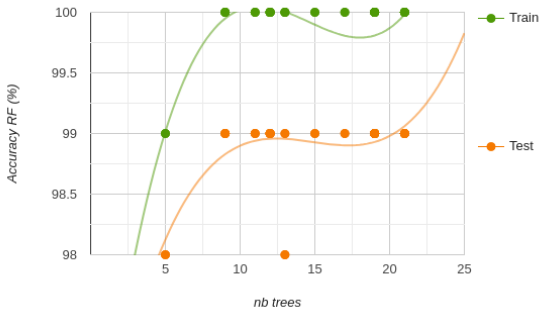
Evaluating the enumeration of symbolic explanations

The objective here is to assess the practical feasibility of the enumeration (scalability) of SR_x and CF_x explanations. For the enumeration of CF_x, we use the *EnumELSRMRCache tool*[23] implementing the boosting algorithm for MCSes enumeration proposed in [GIL18] with a timeout set to 600s. As for the SR_x explanations, their computation is easily done by exploiting the minimal hitting set duality relationship between MUSes and MCSes. The different values used for the radius r during the experiments has a direct impact on determining the locality of x, and therefore selecting the data used to train the surrogate model.

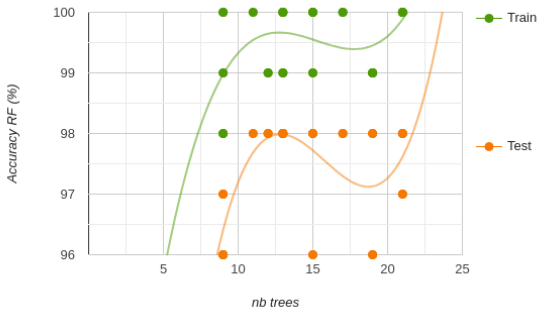
We observe within **Table 3.3** that the average runtime remains reasonable (note that the times shown in Table 3.3 relate to the time taken to list all the explanations. The solver starts to find the first explanations very promptly) and that the approach is efficient in practice for medium sized classifiers (as shown in the experiments for BNNs with around 800 variables). We also observe that the number of CF_x may be challenging, hence, this emphasizes the need for scoring metrics to filter them out and identify the ones with the strongest influence on the prediction.

Table 3.4 presents the evaluation of the sufficient reasons generation (SR_x). Those explanations are enumerated using the duality between sufficient reasons and counterfactuals. Knowing that the runtime

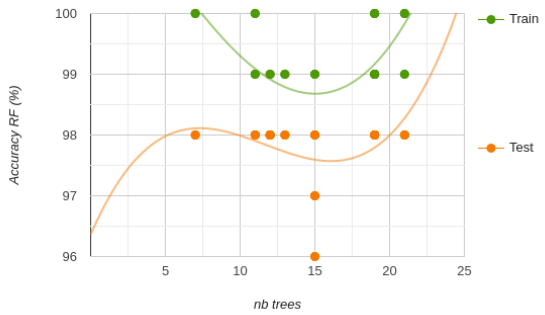
[23] available at http://www.cril.univ-artois.fr/enumcs/



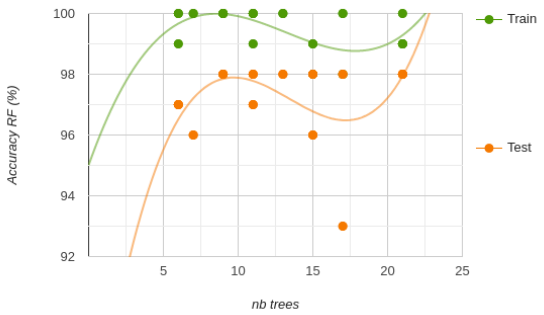
(a) RF trained on MNIST_0



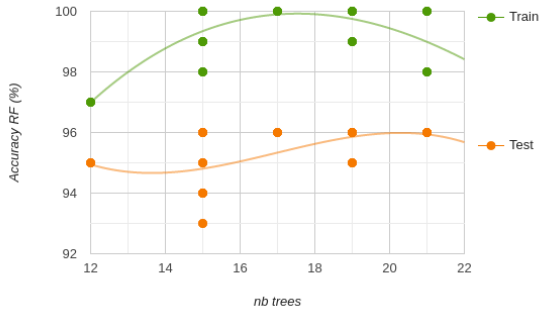
(b) RF trained on MNIST_3



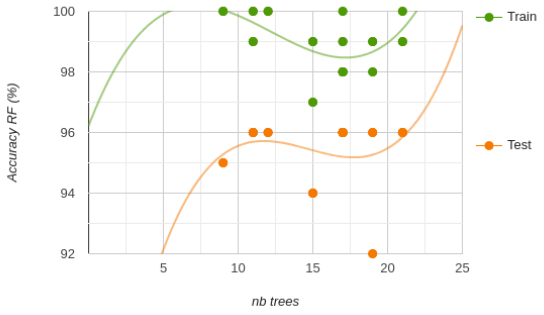
(c) RF trained on MNIST_4



(d) RF trained on MNIST_5



(e) RF trained on MNIST_6



(f) RF trained on MNIST_8

Figure 3.11: The classification accuracy of RF models with respect to the $ntree$ parameter.

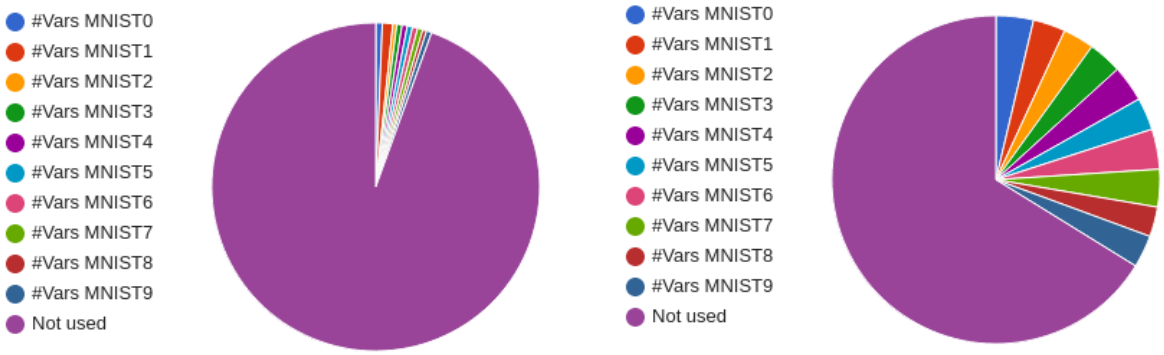
reported in Table 3.4 is the time needed for the enumeration of all possible SR_x explanations, the results remain of the same order of magnitude as that of the counterfactuals presented in Table 3.3.

The pie charts in Figure 3.12 are used to illustrate the portion of relevant features on average (Figure 3.12a) and at most (Figure 3.12b) composing the explanations enumerated for the different classifiers trained on the MNIST dataset. As shown on the charts, the number of relevant features remains low

Radius (r)	Dataset name	MIN #CF	AVG #CF	MAX #CF	MIN *runtime* (s)	AVG *runtime* (s)	MAX *runtime* (s)	One *CF runtime* (s)
100	MNIST_0	1	80	1047	$8*10^{-4}$	$4.09*10^{-2}$	$54*10^{-1}$	
	MNIST_2	1	2198	222180	$3*10^{-4}$	$32.42*10^{-2}$	43.24	
	MNIST_6	1	13753	225456	$2.4*10^{-3}$	12.39	275.52	
	MNIST_8	1	5343	85407	$1.1*10^{-3}$	2.74	35.08	
150	MNIST_0	1	4733	261504	10^{-4}	0.88	67.05	$\leq 10^{-4}$
	MNIST_2	4	23503	346288	$3*10^{-3}$	13.36	315.34	
	MNIST_6	1	29527	401117	$6.6*10^{-3}$	19.1	328.73	
	MNIST_8	2	5703	739433	$7*10^{-4}$	1.86	57.38	
250	MNIST_0	10	35790	285219	0.005	21.49	234.18	
	MNIST_2	13	63916	546005	0.11	42.11	600	
	MNIST_6	15	79520	640868	0.11	50.86	531.16	
	MNIST_8	6	69678	1210349	$33*10^{-4}$	32.3	315.34	
3	SPECT	1	2	11	0	$64*10^{-4}$	$137*10^{-4}$	0
4		2	19	107	$2*10^{-4}$	$1.58*10^{-2}$	$5.75*10^{-2}$	
7		2	71	446	$2.7*10^{-3}$	$5.24*10^{-2}$	$2.88*10^{-1}$	
22		15	204	700	0.01	0.12	0.42	
5	MONKS	1	3	20	$2*10^{-4}$	$5.3*10^{-3}$	$1.4*10^{-2}$	0
16		3	15	41	0.01	0.03	0.06	
5	BREAST CANCER	1	18	202	0	$1.57*10^{-2}$	$1.54*10^{-1}$	0
10		1	368	3118	$2*10^{-4}$	$6.97*10^{-1}$	6.05	
48		11	947	5541	0.02	1.5	17.7	
2	CONGRES. VOTING	1	1	3	$3*10^{-4}$	$6.6*10^{-3}$	$9.2*10^{-3}$	0
5		1	3	11	$5*10^{-4}$	$7.9*10^{-3}$	$3.28*10^{-2}$	
16		1	5	44	$1.9*10^{-3}$	$1.21*10^{-2}$	$4*10^{-2}$	

Table 3.3: Evaluating the enumeration of all the counterfactual explanations.

compared to the size of the input feature space (composed of 784 variables). This is directly related to the property of minimality of our explanations, which translates into concise symbolic explanations.



(a) Average number of variables involved

(b) Maximum number of variables involved

Figure 3.12: The percentage of relevant features w.r.t the initial feature space composed of 784 variables for the MNIST dataset.

Radius (r)	Dataset name	MIN #SR	AVG #SR	MAX #SR	MIN *runtime* (s)	AVG *runtime* (s)	MAX *runtime* (s)
100	MNIST_0	1	230	1808	$1.1* 10^{-3}$	$2.94* 10^{-2}$	$3.38* 10^{-1}$
	MNIST_2	1	9911	116700	$1.2* 10^{-3}$	47.6	595
	MNIST_6	3	1900	19129	$1.1* 10^{-3}$	$7.018* 10^{-1}$	7.06
	MNIST_8	1	13032	126764	$9* 10^{-4}$	31	488
150	MNIST_0	12	31665	172530	$2.2 * 10^{-3}$	77.87	519
	MNIST_2	1	11851	119937	$7* 10^{-4}$	36.58	594
	MNIST_6	5	17218	144640	$1* 10^{-3}$	63.09	558
	MNIST_8	2	20189	227446	$8* 10^{-4}$	29.1	578
250	MNIST_0	3	32135	246428	$1.4* 10^{-3}$	100	595
	MNIST_2	16	27519	155956	$3.1* 10^{-3}$	104.67	592
	MNIST_6	65	38323	179112	$7.2* 10^{-3}$	121	577
	MNIST_8	4	33659	315753	$1.4* 10^{-3}$	35.98	536
3	SPECT	1	3	13	$9*10^{-4}$	$1.8*10^{-3}$	$4.9*10^{-3}$
4		1	21	113	$1.1*10^{-3}$	$5*10^{-3}$	$2.32*10^{-2}$
7		1	87	390	$1.2*10^{-3}$	$2.47*10^{-2}$	$12.36*10^{-2}$
22		1	223	1290	$9*10^{-4}$	$4.94*10^{-2}$	$41.97*10^{-2}$
5	MONKS	1	3	13	$8*10^{-4}$	$1.3*10^{-3}$	$3.4*10^{-3}$
16		1	28	102	$8*10^{-4}$	$3.9*10^{-3}$	$1.67*10^{-2}$
5	BREAST CANCER	1	23	308	$1*10^{-3}$	$4.4*10^{-3}$	$6.69*10^{-2}$
10		1	1502	10817	$9*10^{-4}$	$6.66*10^{-2}$	7.63
48		4	4737	69838	$1.3*10^{-3}$	2.16	55.25
2	CONGRES. VOTING	1	2	3	$1.1*10^{-3}$	$1.5*10^{-3}$	$2.4*10^{-3}$
5		1	2	5	$8*10^{-4}$	$1.4*10^{-3}$	$2.4*10^{-3}$
16		1	8	46	$9*10^{-4}$	$2.5*10^{-3}$	$1.1*10^{-2}$

Table 3.4: Evaluating the enumeration of all the sufficient reasons explanations.

Figure 3.13 represents box plots indicating the range in which the size of explanations (number of variables) is located. We observe that in general, the symbolic explanations are of concise sizes which help a human to understand the reasons of a model's decision easily.

Illustrating SR_x and CF_x explanations for MNIST dataset

We show examples of both types of symbolic explanations Figure 3.14. Recall that we trained two "one-vs-all"[24] BNNs for each digit ranging from 0 to 9 and we will be using them as black-boxes on some test samples. Although the performance of the classifiers is not the key objective to this study, we use well-trained models that can predict the output well. The classification models f_8 and f_0 have respectively achieved an accuracy of 97% and 99% on test data. Figure 3.14 shows some relevant data samples and the symbolic explanations generated to explain their predictions. The first column shows the input test images representing different digits, their prediction and the targeted outcome that can be reached using a counterfactual explanation. Results are depicted on three columns. The column "Relevant features" represents all the pixels included in the set of explanations enumerated for the test input. The "sufficient

[24] A "one-vs-all" BNN f_i returns a positive prediction for an input image representing the "i" digit, and negative one otherwise.

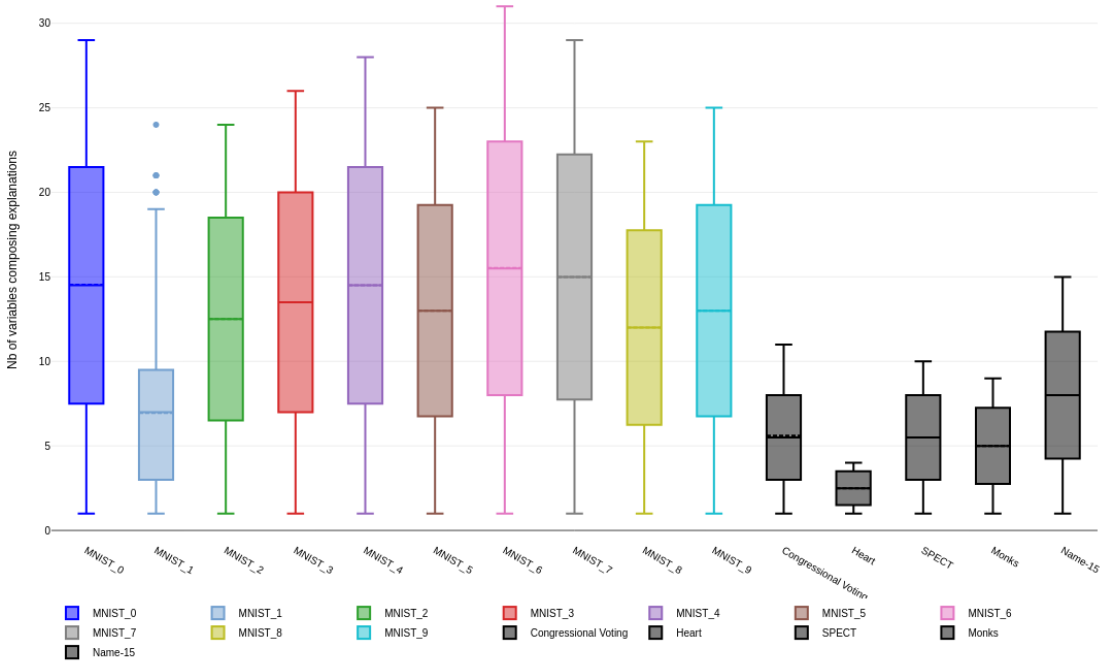


Figure 3.13: The range of explanations size enumerated for the different datasets.

reason" column corresponds to a picture depicting the set of pixels composing an example of a single sufficient reason explanation that justifies the model's decision. Finally, the last column represents an example of a single counterfactual explanation highlighting the set of pixels to activate/deactivate in order to provide users with actionable explanations in the form of data sample that would have received a different outcome. The first row associated to a test input shows images with the pixels composing the different explanations. The second row represents the overlay of test input and the different explanations. Let us consider the first test input (x), Figure 3.14 shows an example of a single SR_x explanation highlighting the sufficient pixels for the models f_8 to trigger a negative prediction for the image x. It also provides an example of a CF_x explanation showing the pixels to invert in the test input to make the model f_8 predict x positively. Another example is presented for a test input illustrating a 1-digit and it is recognized as non 0-digit by f_0. The last row is an example of a test input positively predicted by f_0 as a 0-digit.

3.5 Conclusion

In this chapter, we have proposed a post-hoc interpretability method in order to explain the decisions of black-box classification models. We proposed a novel model agnostic generic approach to explain individual outcomes by providing two complementary types of symbolic explanations (*sufficient reasons* and *counterfactuals*). The generation of local symbolic explanations is done within the framework of satisfiability solving. It takes advantage of the strengths of already existing and proven solutions, and of the powerful practical tools for the generation of MCSes/MUSes. We also introduced a substitution

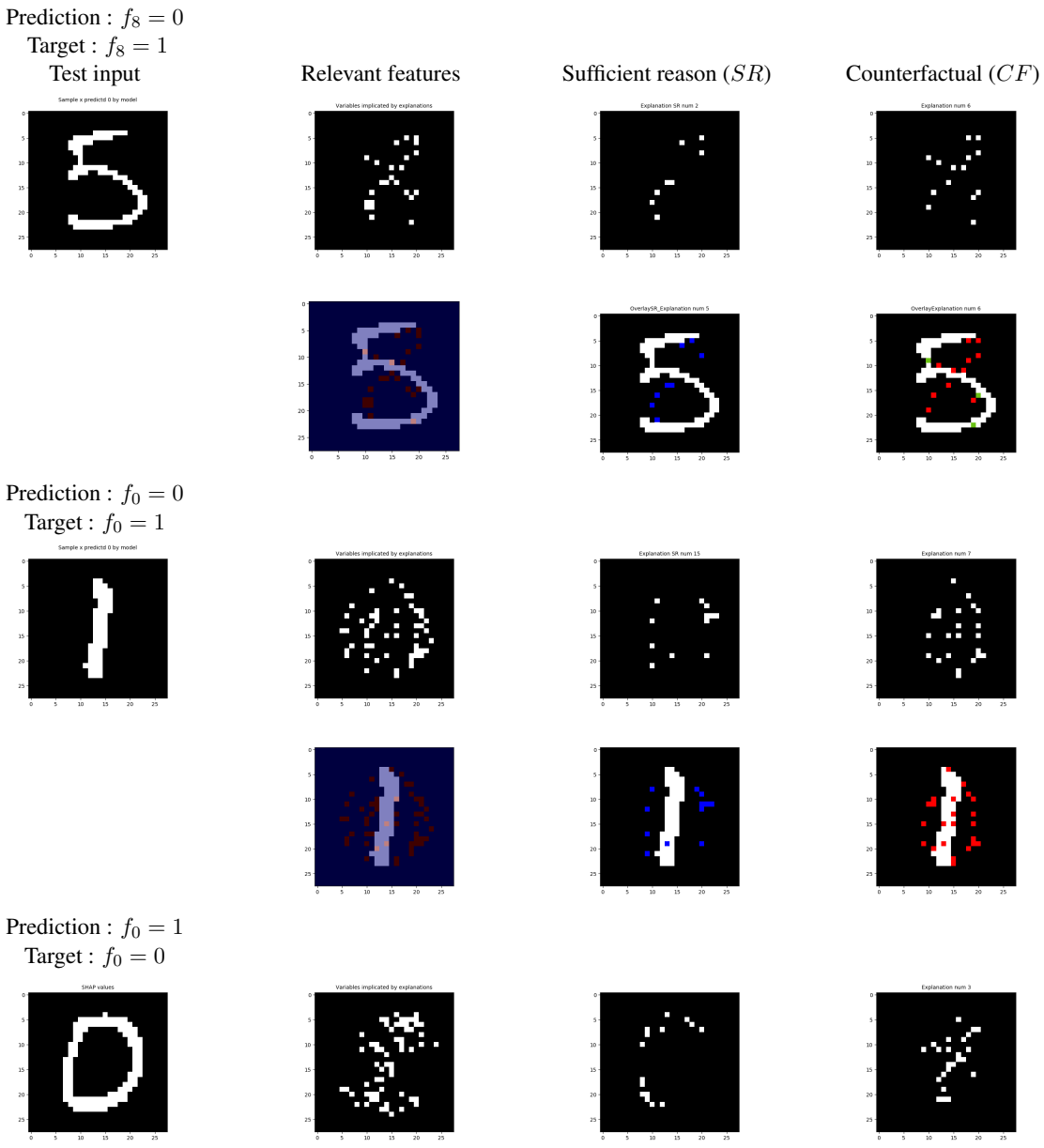


Figure 3.14: Example inputs from MNIST dataset and their respective symbolic explanations.

approach as a solution to overcome the complexity of encoding a ML classifier into an equivalent logical representation. It consists in approximating the original model locally using a surrogate model which will subsequently be used to explain individual predictions of black-box ML models. This approach initially proposed for single-label classification constitutes the cornstone on which the rest of our work is based and has been extended for multi-label classification in the next chapter.

Chapter 4

Symbolic explanations for multi-label classification

Multi-label classification is a well-known predictive task encountered in a wide range of applications such as text categorization, object recognition in images, classification of genes and bio-informatics. For example, in a text categorization problem, each document can be simultaneously assigned to multiple labels or classes (for example, a conference paper may at the same time be tagged as *Machine learning*, *XAI* and *bio-informatics*). In single-label classification, classes are mutually exclusive. At the opposite, in multi-label classification, classes do not exclude each other allowing the same input instance to be simultaneously classified into multiple classes.

This chapter is dedicated to the presentation of an agnostic and declarative approach to provide different types of symbolic explanations for multi-label classifiers. It extends the declarative and model-agnostic approach we presented to explain the predictions of single-label classifiers. More precisely, in addition to sufficient reason and counterfactual explanations presented in Chapter 3, the proposed approach in this chapter makes it possible to generate explanations at different granularity levels which go from structural relationships between labels to the selection of features allowing to force the prediction to any desired target prediction.

We first present in Section 4.1 a brief refresher of the main methods that have been proposed by the community. In Section 4.2, we deal with the definitions of explanations (sufficient reasons and counterfactual explanations) in a multi-label setting on two different levels of granularity (the whole prediction or parts of the prediction). We will present in Section 4.3 a new type of explanations specific to multi-label problems where we rely on the relationships among the considered classes to infer and present explanations. We present a SAT-based setting to enumerate those explanations in Section 4.4. Finally, we tackle in Section 4.5 the issue of evaluating our approach on different types of data.

4.1 Brief review of related works

Unlike single-label classification problems, very few studies have focused on explaining multi-label classifiers. There is mainly a couple of simple feature attribution methods based on aggregating feature importance scores of the different predicted labels computed individually. Explanation approaches in multi-label classification can mainly be categorized into feature importance explanations and decision rules explanations. In [PGMP19], the authors propose "MARLENA", a model-agnostic method to explain the decisions of a multi-label black-box. It generates a synthetic neighborhood around the sample to be explained and learns a multi-label decision tree on it. The explanations are simply the decision rules derived from those decision trees. In [CGG+20], the authors propose an approach to explain neural

network-based systems by learning first-order logic rules from the outputs of the multi-label model. This approach completely ignores the features when providing explanations. The explaining functions give a description in terms of first-order logic where each rule expresses the validity of a certain explanation w.r.t. the output of the model on a given input. In [SB21], the authors focus on multi-label model explainability and propose a method to merge multiple feature importance explanations corresponding to each class into a single list of feature contributions. The aggregation of the feature weights is simply the average feature weights over the k labels. The same idea is used in [Che21] except that they compute Shapley values over the dataset using kernel-SHAP and then compute a global feature importance per label.

Such methods are limited when it comes to the explanation types they provide. For instance, one can not identify which part of the features is responsible for a given part of the multi-label prediction. Moreover, from a user point of view, it can be hard to interpret what a given feature importance means in the case of a multi-label decision given that a feature may have strong influence for predicting a given label while it is irrelevant for predicting other labels. In the majority of related works, there are even no clear and formal definitions of what a multi-label explanations are. To our knowledge, there is no symbolic explanation for the classification multi-label problems. We propose different types of explanations that take into account several possible situations where the interest of the user is directed towards an explanation of the entire prediction or of a specific part of it. Our contribution proposes not only explanations which justify the output obtained, but also actionable explanations in the form of counterfactuals.

4.2 Feature-based explanations

A feature-based explanation involves only features, i.e. aims to explain the predictions of a classifier based only on the features of the input data. It can be associated with different semantics and different granularity levels. We focus on two complementary types of feature-based explanations that are the *sufficient reasons* and *counterfactuals*.

	Entire-outcome	**Fine-grained**
Sufficient reasons (Which features cause the current prediction)	*Why* $f(x)=y$?	*What causes a subset of labels* $y' \subseteq y$ *to be predicted by* f ?
Counterfactuals (Which features to modify to have an alternative prediction)	*Which* x' *st.* $f(x')=y'$?	*Which* x' *forces* f *to make a desired partial prediction ?*

Table 4.1: The symbolic-based multi-label explanations

As for single-label classification, *sufficient reason* explanations correspond to the minimal part of the input data that is sufficient to trigger the current prediction while *counterfactual* explanations refer to the minimal changes in the input data to get an alternative target outcome. Depending on the problem under study, it may be relevant to have different types of explanations. Assume that we have a multi-label classifier (MLC) with a large output set (e.g. hundreds). We may care little about all the labels as it may be irrelevant to provide an explanation for the entire outcome of the model, especially for datasets with very low density, as it represents an inherent challenge for multi-label classification [WXH+14, BH17, BCPdI19, Nig16]. This is especially true since in most cases, the user is interested in the few classes predicted positively. For example, in document categorization tasks, a user may want to understand why a document is (or not) classified in such classes. Explaining "Why the document was not classified in all the remaining classes?" may be irrelevant. Based on this observation, the approach we propose called

SYMCA for SYmbolic explanations for Multi-label ClAssification provides explanations for both the entire prediction (all classes) and explanations for parts of the prediction that are of interest to the user. We sum up in Table 4.1 the different cases we consider for feature-based explanations.

Let us introduce a running example that will be used within this chapter to illustrate the different concepts covered.

Example 28 (Running example)**.** *Classifying Yelp reviews into 5 categories. The "Yelp reviews classification"*[25] *is a categorization problem of reviews about businesses and services into relevant categories to help the users understand the rating of a given restaurant. The objective here is to know whether a review positively comments on certain aspects such as the* `Food`*,* `Service`*,* `Ambience`*,* `Deals` *and* `Worthiness`*. The dataset contains more than 10000 reviews from food and restaurant areas. There are three nominal features: Good (4-5 stars), Moderate (3 stars) and Bad (1-2 stars). The output classes (labels) are :* `(Food (F))`*,* `(Service (S))`*,* `(Ambience (A))`*,* `(Deals (D))`*,* `(Worthiness (W))`*. We consider the MLC* f *depicted in Figure 4.1 and consisting in a Binary Relevance classifier*[26] *using decision trees as base classifiers. Given the following review*[27] *:* ***"We went out with friends to have mexican food, the quesadillas was delicious and came with a lot of cheese. We find the place a little boring but the dining room seemed nice"*** *and a* ***4 stars rating*** *as inputs, the review was classified positively commenting about label* `F` *, and negatively about the rest (S,A,D and W). It corresponds to* y=$(1, 0, 0, 0, 0)$ *where* $y = f(x)$*. The extracted features used in this example correspond to the highlighted paths in Figure 4.1.*

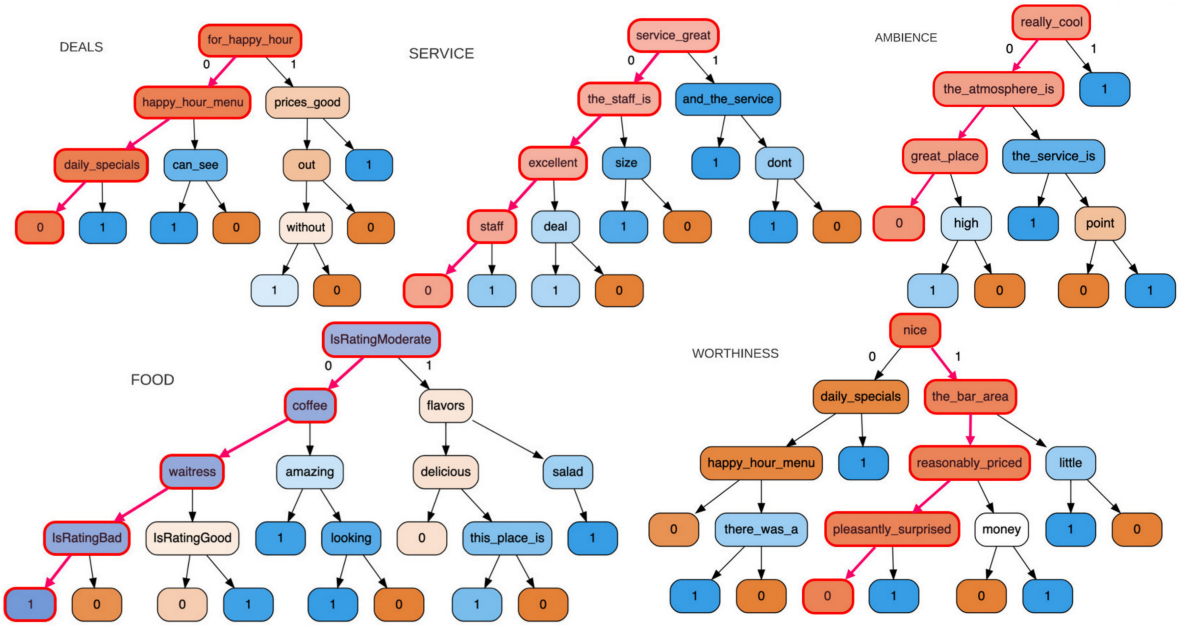


Figure 4.1: Binary Relevance classifier trained on the Yelp dataset using decision trees as base classifiers

[25]The dataset is available at https://www.ics.uci.edu/~vpsaini/.

[26]The binary relevance method (BR for short) is about training one binary classifier independently for each label in order to perform multi-label classification. It is the most commonly used strategy for multi-label classification.

[27]Of course, this is the raw text of the review. Raw data is first pre-processed and two types of features are extracted : i) textual features consisting of unigrams, bigrams and trigrams and ii) binary features representing rating 1-2 stars, 3 stars, and 4-5 stars respectively. Classification is achieved on extracted features.

4.2.1 Entire-outcome explanations

An entire-outcome explanation explains all the predicted labels simultaneously. We recall that in the multi-label setting, the output is a vector y of k binary values assigned to each class of the output space Y. Our feature-based explanations are based on the definition of sufficient reasons and counterfactuals proposed initially for the binary case (see Chapter 3).

Entire-outcome sufficient reasons explanations An entire-outcome sufficient reason explanation (SR for short) identifies the minimal part of a data sample x (namely, the subset of features) sufficient to trigger the current multi-label outcome. Formally, a SR is defined as follows:

Definition 30 (SR explanations)**.** *Let x be a data instance and y=$f(x)$ be its prediction by the multi-label classifier f. An entire-outcome sufficient reason explanation $\tilde{x}$ is such that:*

1. *$\tilde{x} \subseteq x$ ($\tilde{x}$ is a part of x),*
2. *$\forall \acute{x} \in X$, $\tilde{x} \subset \acute{x} : f(\acute{x})$=$f(x)$ ($\tilde{x}$ suffices to trigger y=$f(x)$),*
3. *There is no partial instance $\hat{x} \subset \tilde{x}$ satisfying 1 and 2 (minimality).*

While the two first conditions in Definition 30 search for parts of x allowing to fire the same prediction, the minimality condition allows to find parsimonious explanations (in terms of the number of features involved in the explanation). As we will see in our experiments, the explanations are not unique in the general case. Note that this is the same definition as the one of the SR for the single-label classification case except that y represents a vector of labels instead of one label. In fact, multi-label classification is a generalization of the single-label classification.

Example 29 (Example 28 continued)**.** *(SR **explanations**) The explanations can concern the words present but also the words absent in a given comment entry x. The presence of a word is represented by a binary variable set to 1 if the word appears in x and 0 otherwise. Let us introduce an example of a SR explanation. We want to explain the prediction y=$(1,0,0,0,0)$ for the review in hand. Given Table 4.2 showing examples of one instance of SR explanation per label, an example of sufficient reason for the entire-outcome is* ***('IsRatingBad:0', 'waitress:0', 'looking:0', 'daily_specials:0', 'this_place_is:0', 'delicious:1', 'the_staff_is:0', 'service_great:0', 'great_place:0', 'really_cool:0', 'staff:0', 'excellent:0', 'happy_hour_menu:0', 'prices_good:0', 'for_happy_hour:0', 'the_bar_area:0', 'the_atmosphere_is:0', 'pleasantly_surprised:0', 'reasonably_priced:0')****. *It is easy to check that this SR involves parts forcing the five decisions trees to predict y=$(1,0,0,0,0)$ as detailed in Table 4.2.*

Labels	y	**Sufficient reason explanation**
`Food` (Y_1)	1	['IsRatingBad : 0', 'waitress : 0', 'looking : 0', 'this_place_is : 0', 'delicious:1']
`Service` (Y_2)	0	['the_staff_is:0', 'staff:0', 'excellent:0', 'service_great:0']
`Ambience` (Y_3)	0	['great_place:0', 'really_cool:0', 'the_atmosphere_is:0']
`Deals` (Y_4)	0	['daily_specials : 0', 'happy_hour_menu : 0', 'for_happy_hour : 0']
`Worthiness` (Y_5)	0	['daily_specials : 0', 'happy_hour_menu : 0', 'the_bar_area : 0', 'pleasantly_surprised : 0', 'reasonably_priced : 0']

Table 4.2: A SR explanation for the multi-label prediction y=$(1,0,0,0,0)$.

Entire-outcome counterfactual explanations Given a target outcome $\acute{y}$, a counterfactual entire-outcome explanation (CF for short) is the minimal changes to be done in x in order to obtain $\acute{y}$ instead of y. In other words, if for some reason, one wants to force the classifier to predict $\acute{y}$, then a counterfactual explanation is those minimal changes $\acute{x}$ needed to make on x such that $f(x[\acute{x}])$=$\acute{y}$. The notation $x[\acute{x}]$ denotes the instance x where the variables involved in $\acute{x}$ are inverted.

Definition 31 (CF Explanations)**.** *Let x be a complete data instance and y=$f(x)$ be its prediction by the MLC f. Given a target outcome $\acute{y} \neq y$, an entire-outcome counterfactual explanation $\tilde{x}$ of x is such that:*

1. *$\tilde{x} \subseteq x$ ($\tilde{x}$ is part of x),*
2. *$f(x[\tilde{x}])$=$\acute{y}$ ($\tilde{x}$ fires the target prediction),*
3. *There is no partial instance $\hat{x} \subset \tilde{x}$ satisfying 1 and 2 (minimality).*

Example 30 (Example 28 continued)**.** *(CF **explanations**) Let's assume that the initial prediction y of the review in hand was $(1, 0, 0, 0, 0)$. We want to know the modifications needed on x such that the review is classified as positively commenting on `Service (S)`, `Ambience (A)`, `Deals (D)` and `Worthiness (W)` and negatively commenting the `Food (F)`. Namely, the target prediction $\acute{y}$ is $(0, 1, 1, 1, 1)$. Table 4.3 shows examples of one instance of CF explanation per label, an example of entire-outcome counterfactual is : **['delicious:1', 'IsRatingModerate:0' 'staff:0', 'great_place:0', 'little:1', 'daily_specials:0', 'the_bar_area:0']**. Table 4.3 shows how the above entire-outcome CF involves parts forcing each decision tree to trigger the target outcome $\acute{y}$. For example, to force the model to predict l_2 positively for x, it is enough to add the word 'staff' to x, in other words, change 'staff:0' into 'staff:1' in the decision tree associated with label `S` (l_2).*

Labels	y	$\acute{y}$	**Counterfactual explanations**
`Food` (Y_1)	1	0	['delicious : 1', 'IsRatingModerate : 0']
`Service` (Y_2)	0	1	['staff : 0']
`Ambience` (Y_3)	0	1	['great_place : 0']
`Deals` (Y_4)	0	1	['daily_specials : 0']
`Worthiness` (Y_5)	0	1	['little : 1', 'the_bar_area : 0']

Table 4.3: An entire-outcome CF explanation for the target prediction $\tilde{y}$= $(0, 1, 1, 1, 1)$.

4.2.2 Fine-grained explanations

In practice, it may be more useful to get explanations about a label or a subset of labels of interest rather than an explanation for the entire prediction (a vector of k labels). We say that the label l_j is positively predicted if $l_j = 1$, and negatively predicted if $l_j = 0$.

Remark 7. *In fact, an entire-outcome explanation is a particular case of fine-grained explanation when the target set of interest corresponds to the whole multi-label output vector.*

Fine-grained sufficient reasons explanations Similar to the definition of sufficient reasons for the entire-outcome prediction, a fine-grained sufficient reason ($SR_{\tilde{y}}$ for short) is limited to explaining the part of $\tilde{y} \subset y$ that is of interest for the user.

Definition 32 ($SR_{\tilde{y}}$ explanations)**.** *Let x be a data instance, y=$f(x)$ be its multi-label (entire) prediction by the classifier f and $\tilde{y}$ a subset of y representing the labels of interest ($\tilde{y}$ can involve labels that are predicted positively or negatively). A fine-grained sufficient reason explanation $\tilde{x}$ of x is such that:*

1. *$\tilde{x} \subseteq x$ ($\tilde{x}$ is a part of x),*
2. *$\forall \acute{x} \in X, \tilde{x} \subset \acute{x} : \tilde{y} \subseteq f(\acute{x})$ ($\tilde{x}$ suffices to trigger labels in $\tilde{y}$),*
3. *There is no partial instance $\hat{x} \subset \tilde{x}$ satisfying 1 and 2 (minimality).*

Example 31 (Example 29 continued)**.** *($SR_{\tilde{y}}$ **explanations**) Assume we are only interested in the $SR_{\tilde{y}}$ explanations regarding the labels "Food", "Service" and "Ambience" (explain why we have $(l_1 = 1, l_2 = 0, l_3 = 0)$). An example of a fine-grained $SR_{\tilde{y}}$ explanation for the labels F, S and A is : **['IsRatingBad:0', 'waitress:0', 'looking:0', 'staff:0', 'this_place_is:0', 'delicious:1', 'the_staff_is:0', 'excellent:0', 'service_great:0', 'great_place:0', 'really_cool:0', 'the_atmosphere_is:0']**.*

Fine-grained counterfactual explanations

Definition 33 ($CF_{\tilde{y}}$ Explanations)**.** *Let x be a data instance, y=$f(x)$ be its multi-label prediction by the classifier f. Let $\tilde{y}$ be a subset of y representing the labels of interest (namely, the labels to flip). A fine-grained counterfactual explanation $\tilde{x}$ of x is such that:*

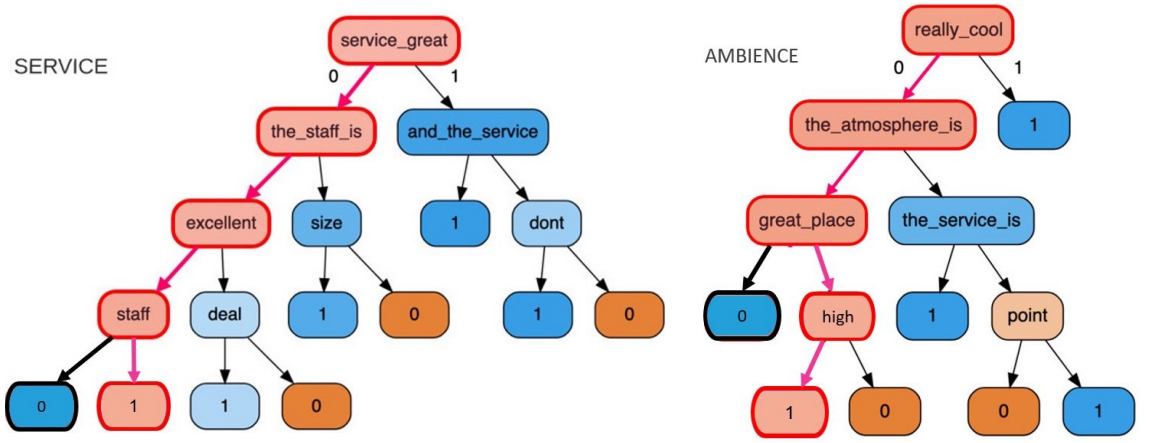
1. *$\tilde{x} \subseteq x$ ($\tilde{x}$ is a part of x),*
2. *$f(x[\tilde{x}]) = y[\tilde{y}]$ (inversion of labels into $\tilde{y}$),*
3. *There is no partial instance $\hat{x} \subset \tilde{x}$ satisfying 1 and 2 (minimality)*

The term $x[\tilde{x}]$ denotes the data instance x where variables included in $\tilde{x}$ are inverted, and $y[\tilde{y}]$ denotes the prediction y where labels included in $\tilde{y}$ are inverted (set to the target outcome).

A fine-grained counterfactual explanation means that instead of a prediction y, we want $\tilde{x}$ which need to be modified in x such that $f(x[\tilde{x}]) = y[\tilde{y}]$ (partial targeted outcome).

Example 32 (Example 30 continued)**.** *($CF_{\tilde{y}}$ **explanations**) Let us assume that we want to invert the prediction of the labels "Service" and "Ambience", corresponding to the partial target prediction $\tilde{y} = (l_2 : 1, l_3 : 1)$. An example of a fine-grained $CF_{\tilde{y}}$ explanation that allows us to reach $\tilde{y}$ is : ['staff:0', 'great_place:0']. We can easily see that changing the value of the variables in $CF_{\tilde{y}}$ will modify the prediction of x for the concerned labels (S, A). The new decision rule paths within the decision trees $DT_{Service}$ and $DT_{Ambience}$ are presented respectively in Figures 4.2a and 4.2b.*

As highlighted in Figure 4.2a, it is sufficient to change the value of the variable staff to 1 to reach the leaf presenting a positive prediction within the decision tree encoding the classifier associated to label Service. The same way, changing the value of variable great_place to 1 leads to the node representing the variable high. It is predicted 0 since it is not present in the sample x as shown in the decision path highlighted in Figure 4.2b.



(a) Decision path set by $CF_{\bar{y}}$ within $DT_{service}$

(b) Decision path set by $CF_{\bar{y}}$ within $DT_{ambience}$

Figure 4.2: Illustration of a fine-grained counterfactual

Special cases of fine-grained counterfactual explanations Given an input data instance x, a MLC f and its prediction y=$f(x)$, one may be interested in practice in the following explanations :

- *Counterfactual explanation for expansion :* By expansion, it is meant adding to the current prediction y more labels predicted positively. For example, if an application (represented by a set of features) is rejected because it was deemed to meet few criteria (corresponding to the labels), a natural question in this case is what needs to be changed in the application to meet more criteria (more labels predicted positively). Expansion requires that at least one label l_i=0 in the current prediction y will be inverted while preserving those labels already predicted positively.

- *Counterfactual explanation for contraction :* This is the opposite scenario to the expansion. It is rather a matter of having fewer positive labels. Clearly, at least one label l_i=1 in the current prediction y needs to be inverted. Thus, the number of negative labels in the new multi-label prediction $\acute{y}$ is greater than the ones in y. The idea is, in addition to the already negatively predicted labels, look for the minimal counterfactual explanations such that the maximum labels positively predicted are deactivated (turns to 0).

Of course, expansion and contraction are special cases of fine-grained counterfactual explanations.

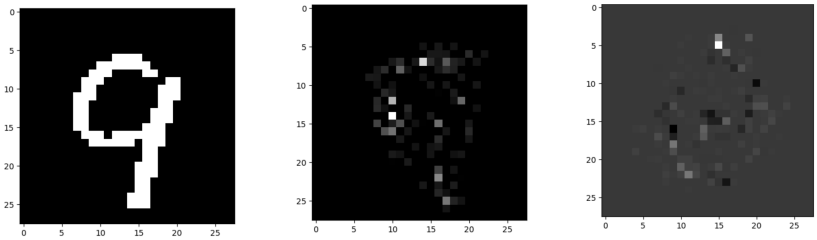
4.3 Label-based explanations

Up to now, we explain the predictions of a classifier only using the features of the input data. Relying solely on features to form symbolic explanations can be problematic in terms of the clarity and relevance of explanations to the user. For instance, authors in [CGG+20] have experimentally shown that local explanations based on the relationships between classes may reveal rules that are able to capture relation between classes such as being Attractive and Young, or being Old and with Gray Hair. As shown in figures of the Example 33, explaining a complex concept (or label) based solely on features can be difficult for the user to understand. In some cases, this aspect can be greatly improved by exploiting relationships or structures between the labels. For instance, if a label l_i is subsumed (in the sense of concept subsumption) by a label l_j according to the MLC f, then clearly sufficient reasons of l_i are also sufficient reasons for l_j. Other examples of relations that can be easily extracted and exploited are label equivalence and disjointedness. The main advantage is that we will have parsimonious explanations

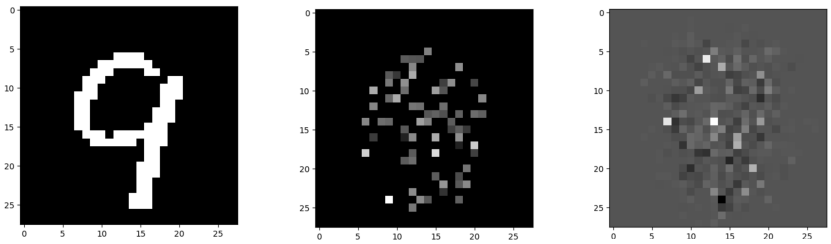
which will be easier for a user to understand, and by reducing the number of the explanations generated, it will simplify their enumeration and presentation.

Example 33. *Let us consider the handwritten digit recognition task performed on the well-known MNIST dataset [LBBH98]. MNIST is composed of 10 classes corresponding to digits from 0 to 9, that we extended by adding the labels* `"Odd"`, `"Even"` *and* `"Prime"`. *Each input image is associated to a vector of thirteen labels*[28]. *Assume we have the following : a MLC model, an input image* x *and its multi-label prediction* y=$(0,0,0,0,0,0,0,0,0,1,1,0,0)$. *Namely,* x *is predicted positively for the labels* `9-digit` *(*l_9*) and* `"Odd"` *(*l_{10}*).*

$$x = \begin{bmatrix} 1*1 & 0 \\ 1*2 & 0 \\ \vdots & \vdots \\ 1*28 & 0 \\ 2*1 & 0 \\ 2*2 & 0 \\ \vdots & \vdots \\ 28*26 & 0 \\ 28*27 & 0 \\ 28*28 & 0 \end{bmatrix} \qquad y = \begin{bmatrix} l_0 & 0 \\ l_1 & 0 \\ l_2 & 0 \\ \vdots & \vdots \\ l_5 & 0 \\ l_6 & 0 \\ l_7 & 0 \\ l_8 & 0 \\ l_9 & 1 \\ l_{10} & 1 \\ l_{11} & 0 \\ l_{12} & 0 \end{bmatrix}$$



(a) Explanations for the label l_9 highlighting the pixels sufficient for the model to recognize the image x as 9



(b) Explanations for the label l_{10} highlighting the pixels sufficient for the model to recognize the image x as Odd

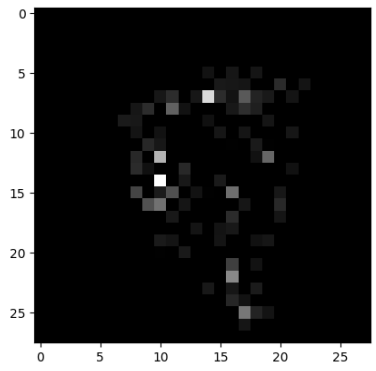
Figure 4.3: Feature-based explanation for a sample from augmented MNIST dataset.

Figure 4.3 shows a sample image x *from the MNIST dataset and its corresponding feature-based explanations. The first column corresponds to the input sample* x *represented as a matrix of 28x28 pixels*

[28]The labels having an index $i \in [0,9]$ indicate whether the input image x is recognized as an `i-digit` while the labels having an index $i \in [10,12]$ indicate respectively whether the represented digit is being classified as an "odd", "even" or a "prime".

binarized (784 variables). The second column represents a heatmap corresponding to the frequency of involvement of the variables appearing in SR_x[29] in the explanation sets of instances from V_x. The last column is the feature-attribution explanation provided by the well-known SHAP[LL17] approach [30]. In Figure 4.3, the sufficient reasons were provided by the post-hoc symbolic explainer we presented previously in Chapter 3. While sufficient reasons for label l_9 clearly display the pixels that the model relies on for classifying x to be "9", it is hard to understand what makes the classifier predict x as "Odd". Clearly, the feature-based explanations for label "Odd" are not intuitive since the representation of an object is very different from its visual representation. Indeed, classifying the MNIST digits into Odd, Even, Prime classes appears to be somewhat confusing because the way a symbol (e.g. 0,2,7) is written has got nothing to do with the properties of the number. Same observation holds for SHAP explanations. Let us now build another explanation for x being predicted as "Odd" based on the fact that x is predicted as "9" and that the subsumption relation between "9" and "Odd" holds over the predictions of the classifier f. The result is presented in Figure 4.4. Clearly, the explanation showing why the image x is predicted as the digit "9" and that label "9" is necessarily associated with label "Odd" is more intuitive and easier to understand.

SR$_x$for $(l_9 = 1)$



Label-based explanations for $(l_9 = 1, l_{10} = 1)$

$\forall x, l_9 \rightarrow l_{10}$ i.e. since the prediction of the instance x is such that l_9=1 then necessarily l_{10}=1.
Hence, the input image *"x is predicted as an Odd number since it is also classified as a '9' digit*. Then, a sufficient label-based explanation of the output $(l_{10}) = True$ is the sufficient condition $(l_9) = True$

Figure 4.4: Combining feature and label-based explanation for a sample from augmented MNIST.

Types of label-based explanations

In theory, many relationships between labels can be exploited. In practice, two major questions arise: 1) How to extract the relations between the labels from the predictions of the classifiers ? and 2) How to exploit the relations extracted during the generation and presentation of explanations ?

Relations between the labels For the first question, we can limit ourselves to certain types of relationships that are easy to extract and easy to understand for the user. Intuitive and useful examples of label relations are :

Class subsumption (class implication) : Let us consider a subsumption relationship between two class labels Y_1 and Y_2 denoted $Y_1 \subseteq Y_2$ (the notation $Y_1 \subseteq Y_2$ means that samples belonging to Y_1 necessarily belong to Y_2). If the relation $Y_1 \subseteq Y_2$ holds, then each time the prediction of an instance x is such that

[29] Recall that sufficient reasons explanations justify why the trained model has positively predicted labels l_9 and l_{10} for x.

[30] The heatmaps represent the value of the pixels contributing positively to the current prediction (having a positive SHAP value). The value 0 is assigned to all the pixels which are unfavorable to the current predictions. (having a negative SHAP value).

Y_1=1 then necessarily Y_2=1. Hence, knowing Y_1=1 is sufficient to assert that Y_2=1. We chose to focus on the positive predictions since the number of positively predicted labels is very often small compared to the size of the output space of multi-label datasets.

Proposition 6. *Given a multi-label classifier f and two class variables Y_1 and Y_2 where $Y_1 \subseteq Y_2$, then, the explanations of $Y_1 = 1$ are explanations for $Y_2 = 1$ and satisfy the conditions (1)-(2) of Definition 24 but do not guarantee to satisfy the condition (3) of the same definition.*

Proof. Let $\tilde{x} \in SR(x, f_1)$, $f_1(x) = 1$ and $Y_1 \subseteq Y_2$. It is easy to see that $\tilde{x} \subset x$ since $x \in SR(x, f_1)$ **(1)**. Since $f_1(x) = 1$ and $Y_1 \subseteq Y_2$ then $f_2(x) = 1$ **(A)**. $\tilde{x} \in SR(x, f_1)$ implies that condition (2) from Definition 24 is satisfied. More precisely, $\forall \acute{x} \in X, \tilde{x} \subset \acute{x} : f_1(\acute{x})=f_1(x)$. Let $\acute{x}$ s.t $\tilde{x} \subset \acute{x}$. Since $f_1(\acute{x}) = 1$ and $Y_1 \subseteq Y_2$ then $f_2(\acute{x}) = 1$ **(B)**. **(A)** and **(B)** imply that $f_2(\acute{x}) = f_2(x)$. Consequently, $\forall \acute{x} \in X, \tilde{x} \subset \acute{x} : f_2(\acute{x})=f_2(x)$ **(2)**. **(1)** and **(2)** imply that $\tilde{x}$ satisfies conditions (1)-(2) of Definition 24 for f_2.
As for condition 3 of Definition 24 (minimality), there is no guarantee to obtain a minimal explanation for $f_2(x) = 1$ given a sufficient reason for $f_1(x) = 1$ (see counter-example 34).

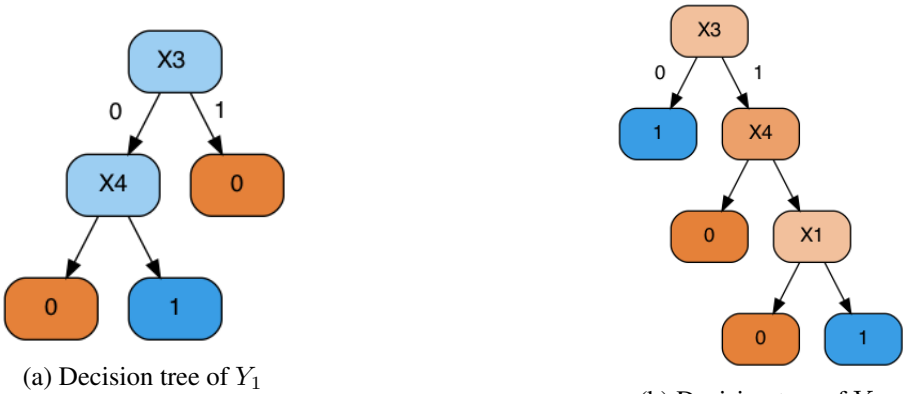
□

An illustration of this proposition is given in Example 34. In order to explain positively predicted instances, we can simply work on the negation of the symbolic representation (CNF) of f (namely $\neg\Sigma_f$). The enumeration of the explanations is done in the same way as for negative predictions.

Example 34. *Let us consider the following set of data. From the data, we can see that there is a subsumption relationship between the labels Y_1 and Y_2 ($Y_1 \subset Y_2$) and between Y_3 and Y_4 ($Y_4 \subset Y_3$).*

$X = \{X_1, X_2, X_3, X_4\}$	$Y = \{Y_1, Y_2, Y_3, Y_4\}$
1101	1111
0101	1111
0001	1111
1001	1110
1011	0100
0111	0010
0011	0011

Assume a decision tree classifier is trained on these data for each label (see Figure 4.5):
We want to explain positive predictions and thus, each classifier $\neg f$ encoded into a CNF formula as shown above.



(a) Decision tree of Y_1

(b) Decision tree of Y_2

Figure 4.5: Decision tree classifiers trained on different labels.

DT_1	$\Sigma_{\neg f_1} \Leftrightarrow (X_3 \vee \neg X_4)$
DT_2	$\Sigma_{\neg f_2} \Leftrightarrow (X_3) \wedge (\neg X_3 \vee \neg X_4 \vee \neg X_1)$
$Input\ sample$	$\Sigma_x \Leftrightarrow (x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4)$

Given the sufficient reason explanations associated to Y_1 and Y_2, we can see that the $SR(x, \neg f_1) \nsubseteq SR(x, \neg f_2)$. We have $\{\neg x_3, \neg x_4\} \subset x$ and $\forall \acute{x}, \{\neg x_3, \neg x_4\} \subset \acute{x}$ implies $f_2(\acute{x}) = 0$. Nevertheless, $\{\neg x_3, \neg x_4\}$ is not minimal for $\neg f_2$ because of $\neg x_3$.

$$\text{SR}(\Sigma_{\neg f_1} \cup \Sigma_x) \quad = \quad \{\ \{\neg x_3, x_4\}\ \}$$

$$\text{SR}(\Sigma_{\neg f_2} \cup \Sigma_x) \quad = \quad \{\{\neg x_3\}, \{x_1, x_4\}\ \}$$

Class equivalence : Two labels Y_1 and Y_2 are said equivalent and denoted $Y_1 \equiv Y_2$ if each time an instance x where Y_1=1 is such that Y_2=1 and if Y_2=1 then Y_1=1. This means that one can explain Y_1 reusing the explanations of Y_2 and vice versa. Note that class equivalence relationships may be rare over whole multi-label data-sets but they are likely to be found when explaining in the neighborhood of a given sample x.

Proposition 7. *Given a multi-label classifier f and two class labels $Y_1 = 1$ and $Y_2 = 1$ where $Y_1 \equiv Y_2$, then, the SR explanations of $Y_1 = 1$ are also SR explanations for $Y_2 = 1$ and vice-versa.*

Proof. Given $Y_1 \equiv Y_2$, proving that $SR(x, f_1) = SR(x, f_2)$ amounts to prove the implication in two directions. For the first implication $SR(x, f_1) \subseteq SR(x, f_2)$, we have the following. Let $\tilde{x} \in SR(x, f_1)$, $f_1(x) = 1$ and $Y_1 \equiv Y_2$. From Proposition 6, since $f_1(x) = 1$ and $Y_1 \subseteq Y_2$, then conditions (1) and (2) of Definition 24 are satisfied. Here it is a proof by contradiction for the condition (3) of Definition 24. Let $x^* \subset \tilde{x}$ s.t $x^* \subset x$ **(1)** and $\forall \acute{x}, x^* \subset \acute{x}$: $f_2(\acute{x}) = f_2(x)$ **(2)**. Thanks to $f_1(x) = 1$, **(2)** and $Y_1 \equiv Y_2$, $\forall \acute{x}, x^* \subset \acute{x} : f_1(\acute{x}) = f_1(x)$ **(2')**. **(1)** and **(2')** imply a contradiction with $\tilde{x} \in SR(x, f_1)$. Thus, $\nexists x^* \subset x$ s.t $\forall \acute{x}, x^* \subset \acute{x}, f_2(\acute{x}) = 1$. Consequently, $\tilde{x} \in SR(x, f_2)$.

For the implication in the opposite direction ($SR(x, f_2) \subseteq SR(x, f_1)$), it is an easy exercise to check that the same reasoning applies. Consequently, given the relation $Y_1 \equiv Y_2$ then $SR(x, f_1) = SR(x, f_2)$. □

Class disjointedness : The disjointedness relationship between two labels Y_1 and Y_2 denoted $Y_1 \cap Y_2 = \emptyset$ denotes the fact that the two classes are mutually exclusive. Namely, for any instance x, if Y_1=1 then necessarily Y_2=0 and conversely if Y_1=0 then necessarily Y_2=1. Then, explaining the label Y_1 allows to explain Y_2 given the disjointedness relation. For instance, using again our digit classification example, it is enough to explain why a digit is labeled `Odd`=1 to understand why it is labeled `Even`=0 given that $\texttt{Odd} \cap \texttt{Even} = \emptyset$ holds for f.

Proposition 8. *Given a multi-label classifier f and two class labels Y_1 and Y_2 where $Y_1 \cap Y_2 = \emptyset$, then, the explanations of Y_1 are also explanations for $\neg Y_2$ and vice-versa.*

Proof. It is clear that the relation $Y_1 \cap Y_2 = \emptyset$ can be written as follows : $Y_1 \subseteq \neg Y_2$ and $Y_2 \subseteq \neg Y_1$. Therefore, $Y_1 \equiv \neg Y_2$ and vice-versa. Following Proposition 7, it is enough to show that $SR(x, f_1) \subseteq SR(x, \neg f_2)$ and $SR(x, f_2) \subseteq SR(x, \neg f_1)$. The demonstration is provided in Proof 4.3. □

The relations between labels can be used independently of the instances to explain the general functioning of the classifier. Note that the extraction of some relations may require the use of dedicated tools or the implementation of specific programs. For instance, testing the equivalence between CNFs comes down to testing if a CNF is a logical consequence of a premise CNF.

4.3.1 Impact of presence of relationships on explanations

To explain the prediction associated with a specific instance, one can combine feature-based explanations with relations between classes as in the example of Figure 4.4. How to combine such explanations is a very broad issue.
Logical relationships can be extracted in several ways, whether using the logical representation (e.g CNF) or the set of predictions of these models. We seek to find the relationship (if any) between every pair of labels. There are different ways to extract certain relationships between two classes. One can consider the propositional logic language with the aim of using the efficient SAT solvers as the problem solving engine. For instance, checking the equivalence between two Boolean formulae can be done as presented in [Dar20] for CNFs and in [WZWY19] for Boolean functions. An important advantage in using such formal modeling is their correctness properties (deductions, model checking, etc), in addition to the speed and availability of SAT solvers. However, some of these properties can be too expensive to compute, which limit their use in practice. Another way to do it is empirically, where the identification of relations between two classes will be reduced to a comparison of their predictions on a set of data. Such an ad-hoc approach has the advantage of being fast and efficient in terms of time and space. However, this does not guarantee that all relations are found nor that those found are valid for the whole data space. Another option is to model the problem of finding relationships that bind classes as an association rule learning problem. The aim is to discover strong rules between an antecedent class Y_1 and a consequent class Y_2 using for instance the confidence measure[31]. For instance, in Example 34 where a class subsumption exists between Y_4 and Y_3 on a given neighborhood, the confidence measure of the association rule $Y_4 \Rightarrow Y_3$ would be equal to 100%.

[31] The confidence value of an association rule denoted as $Y_1 \Rightarrow Y_2$ in data mining is the ratio of transactions containing both Y_1 and Y_2 to the total number of Y_1.

Once these relations are extracted, they can be exploited to determine the sets of final explanations as well as their presentation. In the case where a label Y_i implies a label Y_j, it is possible to directly use the explanations of Y_i to explain Y_j (see Proposition 6). An example is shown in Figure 4.4. Implicitly, this means that the variables that contributed to the prediction of the i^{th} class are also favorable to the j^{th} class given the subsumption relationship between them. When it comes to the equivalence and disjunction between a pair labels (Y_i,Y_j), several scenarios are possible and a choice arises : *should we generate explanations for Y_i and use them to explain Y_j as well, or should we do it in the opposite direction?* and *what are the criteria to consider when making such a choice?* For instance, we can consider the rate of involvement of a label in the detected relations, quality of explanations, etc.

4.4 A model-agnostic SAT-based approach for enumerating symbolic explanations

The concepts described so far as well as the different definitions presented describing explanations aim at making the individual predictions of multi-label classifiers more understandable and interpretable for a user. This section presents an extension of the SAT-based model agnostic approach we proposed for the single-label case with the main difference being the output of Step 1 (set of CNFs associated with k labels instead of a single CNF associated with the single output class). Indeed, in addition to locally explaining individual classifier predictions, the goal of the approach proposed in Chapter 3 was to build a base explainer to explain a model individually and then use it to explain multi-label classifiers.

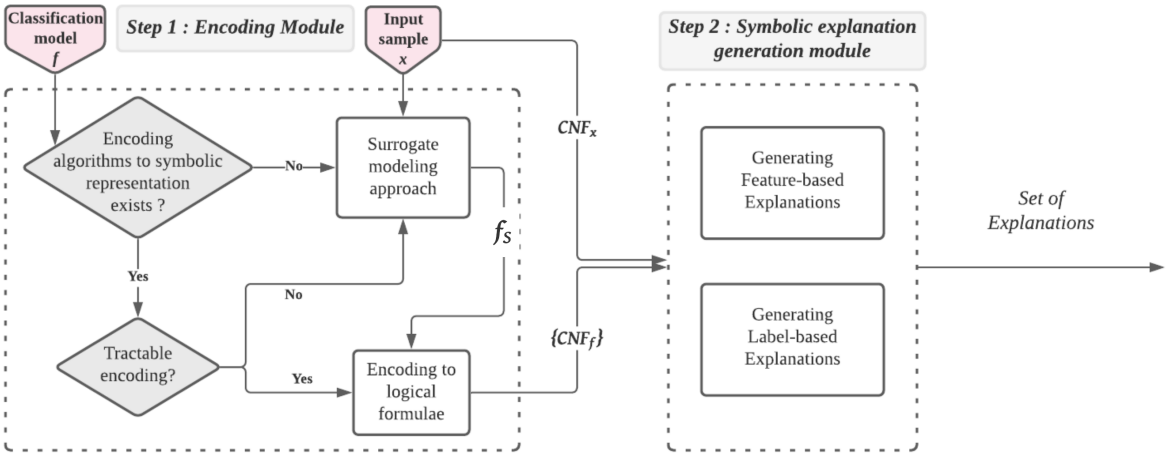


Figure 4.6: Overview of the proposed approach for the multi-label setting

Before diving into more details, Figure 4.6 depicts a general overview of the approach adapted to the multi-label setting. As mentioned in the introduction, our approach for providing symbolic explanations is agnostic and declarative. It is based on modeling the multi-label classifier and our explanation enumeration problems as variants of the propositional satisfiability problem (SAT). It mainly goes through two steps: A first step for encoding the MLC in an "equivalent" (or "faithful" in case of using a surrogate model) canonical symbolic representation. The second step is about enumerating the explanations.

4.4.1 Step 1: Multi-label classifier symbolic modeling

The aim of this step is to associate the multi-label classifier with a symbolic equivalent or faithful symbolic representation that can be processed by a SAT-based oracle to enumerate our symbolic explanations. As shown in Figure 4.6, there are two cases to be considered:

Direct encoding : As mentioned in Chapter 3, some machine learning models (e.g. random forests, Binarized Neural Networks and some Bayesian network classifiers) have direct equivalent encoding into symbolic representation such as OBDDs, SDDs (e.g. [NKR+18, SCD19, SSDC20]) and CNFs which is the standard input format for SAT solvers. More details about such approaches were given in Chapter 3. Hence, in some cases, a multi-label classifier can be directly and equivalently encoded in CNF. For instance, the Binary Relevance classifier using decision trees as base classifiers can be equivalently encoded in CNF as illustrated in our running example. The idea is to associate a CNF Σ_{f_i} to each base classifier f_i such that the binary prediction of f_i for a data instance x is captured by the truth value or the logical consistency of Σ_{f_i} with Σ_x (recall that Σ_x stands for the CNF encoding of the data instance x [32]). Formally, as stated in Definition 23, f_i is said to be equivalent to Σ_{f_i} iff for any data instance x :

$$\Sigma_{f_i} \cup \Sigma_x \models \begin{cases} \top & \text{if } f_i(x) = 1 \\ \bot & \text{otherwise} \end{cases} \tag{4.1}$$

where $\top$ means that the conjunction of Σ_{f_i} and Σ_x is satisfiable, corresponding to a positive prediction of the label l_i. $\bot$ means that the conjunction of Σ_{f_i} and Σ_x is unsatisfiable, corresponding to a negative prediction of the label l_i.

Surrogate modeling : In case the multi-label classifier (MLC) cannot be directly encoded into a CNF or in case such encoding would be intractable, our approach proceeds by associating with the multi-label classifier a faithful surrogate model that can be encoded into a CNF. In addition to allowing the handling of any multi-label classifier, the surrogate modeling offers another useful advantage that is providing local explanations. Indeed, it is challenging to explain a model's prediction over the whole dataset where the decision boundary may not be easily captured. The surrogate model built locally will make it possible to provide explanations in the neighborhood of x. Our approach associates a surrogate model f_{S_i} to each label l_i. The surrogate model f_{S_i} is trained on the vicinity of the data sample x using the original training instances or by generating new data samples by randomly perturbing features of the input instance x. The MLC model f is then used as a predictor on these samples where the predictions $f(x)$ become the targets. Surrogate models should be tuned to be as faithful as possible to the behavior of the original model f in the neighborhood of x. A good surrogate model is the one able to ensure a good trade-off between high faithfulness to the initial model and tractability of its CNF encoding.

Example 35 (Example 28 continued)**.** *Let us continue our running example. The encoding of the five decision trees of Figure 4.1 into five CNFs is straightforward as shown in the following (recall that intuitively, encoding a decision tree in CNF comes down to encode the paths leading to leaves labeled 0).*

[32] An input sample x is directly encoded in CNF in the form of a set of unit clauses.

$$
\begin{array}{ll}
Food & y_1 \Leftrightarrow (IsRatingModerate \vee coffee \vee waitress \vee \neg IsRatingBad) \wedge \\
 & (IsRatingModerate \vee coffee \vee \neg waitress \vee IsRatingGood) \wedge \\
 & (IsRatingModerate \vee \neg coffee \vee \neg amazing \neg looking) \wedge \\
 & (\neg IsRatingModerate \vee flavors \vee delicious) \wedge \\
 & (\neg IsRatingModerate \vee flavors \vee \neg delicious \vee \neg this\_place\_is)
\end{array}
$$

$$
\begin{array}{ll}
Service & y_2 \Leftrightarrow (service\_great \vee the\_staff\_is \vee excellent \vee staff) \wedge \\
 & (service\_great \vee the\_staff\_is \vee \neg excellent \vee \neg deal) \wedge \\
 & (service\_great \vee \neg the\_staff\_is \vee \neg size) \wedge \\
 & (\neg service\_great \vee \neg and\_the\_service \vee \neg dont)
\end{array}
$$

$$
\begin{array}{ll}
Ambience & y_3 \Leftrightarrow (really\_cool \vee the\_atmosphere\_is \vee great\_place) \wedge \\
 & (really\_cool \vee the\_atmosphere\_is \vee \neg great\_place \vee \neg high) \wedge \\
 & (really\_cool \vee \neg the\_atmosphere\_is \vee \neg the\_service\_is \vee point)
\end{array}
$$

$$
\begin{array}{ll}
Deals & y_4 \Leftrightarrow (for\_happy\_hour \vee happy\_hour\_menu \vee daily\_specials) \wedge \\
 & (for\_happy\_hour \vee \neg happy\_hour\_menu \vee \neg can\_see) \wedge \\
 & (\neg for\_happy\_hour \vee prices\_good \vee \neg out) \wedge \\
 & (\neg for\_happy\_hour \vee prices\_good \vee out \vee \neg without)
\end{array}
$$

$$
\begin{array}{ll}
Worth & y_5 \Leftrightarrow (nice \vee daily\_specials \vee happy\_hour\_menu) \wedge \\
 & (nice \vee daily\_specials \vee \neg happy\_hour\_menu \vee \neg there\_was\_a) \wedge \\
 & (\neg nice \vee the\_bar\_area \vee reasonably\_priced \vee pleasantly\_surprised) \wedge \\
 & (\neg nice \vee the\_bar\_area \vee \neg reasonably\_priced \vee money) \wedge \\
 & (\neg nice \vee \neg the\_bar\_area \vee \neg little)
\end{array}
$$

Once the encoding step is achieved, we can rely on SAT-based oracles to provide explanations as presented in the following step.

4.4.2 Step 2: Symbolic explanation enumeration

Recall that in Step 2, we are given as input a set of CNFs ($\{\Sigma_1,..,\Sigma_k\}$ associated to k labels) encoding the MLC f and a CNF encoding the data instance x denoted Σ_x. The aim of this step is to enumerate explanations for the prediction y=$f(x)$. Similarly to the single-label case, we rely on SAT-based oracles in order to provide sufficient reasons and counterfactuals for a given label l_i (see Section 3.3 for how one can use it for binary classifiers). In the following, let $SR(x, f_{S_i})$ (resp. $CF(x, f_{S_i})$) denote the set of sufficient reasons (resp. counterfactuals) to explain individual prediction of a base classifier $f_{S_i}(x)$.

Enumerating feature-based explanations

We present in the following how the proposed approach proceeds depending on the type of symbolic explanations to provide. Note that the entire-outcome explanations are actually a general case of fine-grained explanations where the target prediction $\tilde{y}$ concerns all output classes ($|\tilde{y}| = k$).

Entire-outcome sufficient reasons SR : Since we can provide sufficient reasons for each label l_i, then it suffices to combine (join) an SR_i from each classifier f_i to form an explanation for the whole outcome.

Proposition 9. *(Joint entire-outcome SR) Given a sufficient reason $\tilde{x}_i$ for each classifier i=1,..,k, the explanation $\tilde{x} = \wedge_{i=1}^{k}\tilde{x}_i$ satisfies conditions (1)-(2) of Definition 30 but does not guarantee to satisfy the condition (3) of the same definition.*

Proof. (sketch) An explanation $\tilde{x} = \wedge_{i=1}^{k}\tilde{x}_i$ involves exactly one explanation from each individual base classifier. It is easy to check that $\tilde{x}$ verify the property 1 and 2 of Definition 30. Namely, (1) $\tilde{x}$ is part of the data instance (since each part $\tilde{x}_i \subseteq x$), (2) let $\acute{x} \in X, \tilde{x} \subset \acute{x}$, it is easy to see that $f(\acute{x}) = f(x)$. Indeed, since $\tilde{x} \subset \acute{x}$ then $\tilde{x}_i \subset \acute{x}$ for $i = 1, .., k$. Following Definition 24, $f_i(\acute{x}) = f_i(x)$ for $i = 1, .., k$.

As for the condition (3) of Definition 30 (minimality), combining base classifier SR explanations does not guarantee to obtain a minimal explanation. (see counter-example 36).

□

Example 36. *Assume a multi-label classification problem where data items are labeled in one or more categories (labels). The feature space is $X = \{X_1, X_2, X_3, X_4, X_5\}$ composed of five binary variables and three label variables $Y = \{Y_1, Y_2, Y_3\}$. The classifiers f_1 and f_3 predicted positively for the instance $x = (0, 1, 1, 0, 0)$. Classifier f_1 has three SR explanations for predicting positively x, namely $SR(x, f_1)$=$\{\{\neg x_1, x_2, x_3\}, \{\neg x_1, x_2, \neg x_4\}, \{x_2, x_3, \neg x_4\}\}$ and f_2 has also three SR explanations for $f(x)$ that are $SR(x, f_2)$=$\{\{\neg x_1, x_3\}, \{x_3, \neg x_5\}, \{\neg x_4\}\}$. Classifier f_3 has two SR explanations $SR(x, f_3)$=$\{\{\neg x_1, x_2\}, \{x_3, \neg x_4\}\}$. The set of joint entire-outcome SR explanations built by joining SR explanations of the three classifiers f_1, f_2 and f_3 gives 18 joint explanations including the following $\{\{\neg x_1, x_2, x_3\}, \{\neg x_1, x_2, x_3, \neg x_5\}, \{\neg x_1, x_2, x_3, \neg x_4\}, \{\neg x_1, x_2, x_3, \neg x_4, \neg x_5\}, ..\}$. In this example, explanations $\{\neg x_1, x_2, x_3\}$ and $\{\neg x_1, x_2, x_3, \neg x_5\}$ are explanations obtained by combining SR explanations of $f_1(x), f_2(x)$ and $f_3(x)$. Clearly, explanation $\{\neg x_1, x_2, x_3, \neg x_5\}$ is not minimal since $\{\neg x_1, x_2, x_3\}$ is a joint entire-outcome SR with a smaller size.*

An example of an entire-outcome sufficient reason is shown in Table 4.2. The entire-outcome explanations are required to be conform to certain properties adapted from those defined for the BR explanations in [Tab19]. Hence, an entire-outcome explanation should verify :

- *Unanimity :* The explanation must explain all the parts of the multi-label prediction $f(x) = y$.
- *Decomposability :* The decomposition of the explanation of $f(x) = y$ should explain every label composing y.

Entire-outcome counterfactuals CF : Similar to sufficient reasons, one can form an explanation as far as we have counterfactuals CF_i for each label l_i. More precisely, let the MLC f predict y for x (namely, $f(x)$=y). Let us assume that the user wants to force the prediction to $\acute{y}$ of size k. Then, an explanation is formed by joining a counterfactual CF_i from each classifier f_i.

Proposition 10. *(Joint entire-outcome CF) Given a counterfactual $\tilde{x}_i$ for each classifier i=1,..,k, the explanation $\tilde{x} = \wedge_{i=1}^{k}\tilde{x}_i$ satisfies conditions (1)-(2) of Definition 31 but does not guarantee to satisfy the condition (3) of the same definition.*

Proof. (sketch) It is easy to check that $\tilde{x}$ verify the property 1 and 2 of Definition 31. Namely, (1) $\tilde{x}$ is part of the data instance (since each part $\tilde{x}_i \subseteq x$), (2) let $\acute{x} \in X, \tilde{x} \subset \acute{x}$, it is easy to see that $f(x[\tilde{x}])$=$\acute{y}$. Indeed, since $\tilde{x} \subset \acute{x}$ then $\tilde{x}_i \subset \acute{x}$ for $i = 1, .., k$. Following Definition 25, $f_i(x[\tilde{x}])$=$\acute{y}_i$ for $i = 1, .., k$.

As for the condition (3) of Definition 31 (minimality), combining base classifier CF explanations does not guarantee to obtain a minimal explanation. (see counter-example 37). □

Example 37. *(Example 36 continued) Given the SR explanations of classifiers f_1, f_2 and f_3, the set of CF explanations associated to those classifiers respectively are the following : $CF(x, f_1) = \{\{x_2\}, \{\neg x_1, x_3\}, \{\neg x_1, \neg x_4\}, \{x_3, \neg x_4\}\}$, $CF(x, f_2) = \{\{x_3, \neg x_4\}, \{\neg x_1, \neg x_5, \neg x_4\}\}$ and finally the set $CF(x, f_3) = \{\{x_2, \neg x_4\}, \{\neg x_1, \neg x_4\}, \{x_2, x_3\}, \{\neg x_1, x_3\}\}$. Joining the CF entire-outcome explanations of classifiers f_1, f_2 and f_3 gives the following explanations $\{\{x_2, x_3, \neg x_4\}, \{x_2, \neg x_1, \neg x_4, \neg x_5\}, \{\neg x_1, \neg x_4, \neg x_5\}, \{\neg x_1, x_2, x_3, \neg x_4, \neg x_5\}, ..\}$. It is clear that the explanations $\{\neg x_1, x_2, x_3, \neg x_4, \neg x_5\}$ is not minimal since the joint entire-outcome CF explanation $\{\neg x_1, \neg x_4, \neg x_5\}$ has a smaller size.*

An example of an entire-outcome sufficient reason is shown in Table 4.3.

Fine-grained sufficient reasons $SR_{\tilde{y}}$: For fine-grained explanations, we proceed in a similar way while restricting to the part $\tilde{y} \subseteq y$ of interest to the user. Let $\tilde{y} = \{\tilde{y}_1, .., \tilde{y}_z\}$ be a subset of y representing the labels of interest where $\tilde{y}_j$ is the j^{th} element of $\tilde{y}$ with $j = 1, ..z$. Namely, given sufficient reasons for each label $y_j \in \tilde{y}$, then joining an SR_j from each classifier f_j with $y_j \in \tilde{y}$ is enough to form an explanation for the partial outcome $\tilde{y}$ as shown in the example 31.

Lemma 4. *(Joint fine-grained SR) Given a sufficient reason $\tilde{x}_{\tilde{y}_j}$ for each classifier $j = 1, ..z$, the explanation $\tilde{x} = \wedge_{j=1}^{z} \tilde{x}_{\tilde{y}_j}$ satisfies conditions (1)-(2) of Definition 32 but does not guarantee to satisfy the condition (3) of the same definition.*

Proof. (sketch) An explanation $\tilde{x} = \wedge_{j=1}^{z} \tilde{x}_{\tilde{y}_j}$ involves exactly one explanation from each individual base classifier in $\tilde{y}$. It is easy to check that $\tilde{x}$ verify the property 1 and 2 of Definition 32. Namely, (1) $\tilde{x}$ is part of the data instance (since each part $\tilde{x}_{\tilde{y}_j} \subseteq x$), (2) let $\acute{x} \in X, \tilde{x} \subset \acute{x}$, it is easy to see that $f(\acute{x}) = f(x)$. Indeed, since $\tilde{x} \subset \acute{x}$ then $\tilde{x}_{\tilde{y}_j} \subset \acute{x}$ for $j = 1, .., z$. Following Definition 24, $f_j(\acute{x}) = f_j(x)$ for $j = 1, .., z$.

As for the condition (3) of Definition 32 (minimality), combining base classifier SR explanations does not guarantee to obtain a minimal explanation. (see counter-example 38).

□

Example 38. *Let us reuse the SR explanations of Example 36. Given a target outcome $\tilde{y} = (l_2, l_3)$, the set of joint fine-grained SR explanations formed by joining SR explanations of classifiers f_2 and f_3 is $\{\{\neg x_1, x_3, x_2\}, \{\neg x_1, x_3, \neg x_4\}, \{x_3, \neg x_5, \neg x_1, x_2\}, \{x_3, \neg x_5, \neg x_4\}, \{\neg x_4, \neg x_1, x_2\}, \{\neg x_4, x_3\}\}$. Similarly to Example 36, it is clear that the explanations $\{\neg x_1, x_3, \neg x_4\}$ and $\{x_3, \neg x_5, \neg x_4\}$ are not minimal since the size of the explanation $\{\neg x_4, x_3\}$ is smaller.*

Fine-grained counterfactuals $CF_{\tilde{y}}$: Given counterfactuals for each label $y_j \in \tilde{y}$, then joining an CF_j from each classifier f_j such that $y_j \in \tilde{y}$, j=1,..,z, allows to build an explanation in order to obtain the partial outcome $\tilde{y}$ as shown in the Example 32.

Lemma 5. *(Joint fine-grained CF) Given a counterfactual $\tilde{x}_{\tilde{y}_j}$ for each classifier $j = 1, ..z$, the explanation $\tilde{x} = \wedge_{j=1}^{z} \tilde{x}_{\tilde{y}_j}$ satisfies conditions (1)-(2) of Definition 33 but does not guarantee to satisfy the condition (3) of the same definition.*

Proof. (sketch) An explanation $\tilde{x} = \wedge_{j=1}^{z} \tilde{x}_{\tilde{y}_j}$ involves exactly one explanation from each individual base classifier in $\tilde{y}$. It is easy to check that $\tilde{x}$ verify the property 1 and 2 of Definition 33. Namely, (1) $\tilde{x}$ is part of the data instance (since each part $\tilde{x}_{\tilde{y}_j} \subseteq x$), (2) let $\acute{x} \in X, \tilde{x} \subset \acute{x}$, it is easy to see that $f(\acute{x}) = f(x)$. Indeed, since $\tilde{x}_{\tilde{y}_j} \subset \acute{x}$ then $\tilde{x}_{\tilde{y}_j} \subset \acute{x}$ for $j = 1, .., z$. Following Definition 25, $f_j(x[\tilde{x}]) = \acute{y}_j$ for $j = 1, .., z$.

As for the condition (3) of Definition 33 (minimality), combining base classifier CF explanations does not guarantee to obtain a minimal explanation. (see counter-example 39).

□

Example 39. *Let us reuse the CF explanations of Example 37. Given a target outcome* $\tilde{y} = (l_1, l_2)$*, the set of joint fine-grained CF explanations formed by joining CF explanations of classifiers* f_1 *and* f_2 *would be* $\{\{x_2, x_3, \neg x_4\}, \{x_2, \neg x_1, \neg x_5, \neg x_4\}, \{\neg x_1, x_3, \neg x_4\}, \{\neg x_1, x_3, \neg x_5, \neg x_4\}, \{x_3, \neg x_4\}, \{\neg x_1, x_3, \neg x_4, \neg x_5\}, \{\neg x_1, x_3, \neg x_4\}, \{\neg x_1, \neg x_5, \neg x_4\}\}$*. It is clear that explanation* $\{x_2, x_3, \neg x_4\}$ *is not minimal since the explanation* $\{x_3, \neg x_4\}$ *has a smaller size.*

Remark 8. *It is important to notice that our modeling through associating a set of binary classifiers (surrogate models) encoded in CNF is not the only way to use a SAT-based oracle to provide explanations. For instance, one can explain a MLC by learning a one-vs-rest binary function* f' *that only recognizes the outcome* $f(x)$ *as a positive prediction. It is formally defined as follows:*

$$f'(x) = \begin{cases} 1 & \text{if } f(x) = y \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

However, such modeling provides neither label-based explanations nor fine-grained ones.

Enumerating label-based explanations

Recall that label-based explanations denote structural relationships between labels. In order to extract some relationships, one can also rely on a SAT-based modeling where each individual labels l_i is associated with a CNF Σ_{f_i}. Hence, checking whether some relationships hold between subsets of labels comes down to checking, for example, the corresponding logical relationships between their respective CNF formulas. Alternatively, we can rely on the predictions of labels to search for some relationships.

For instance, assume we are given a multi-label model f and an input x, and the we want to check whether l_1 is equivalently logic to l_2 ($l_1 \equiv l_2$) in the vicinity of x (denoted V_x). We can either check if the Σ_{f_1} is logically equivalent to Σ_{f_2} (in which case they must share the same models), or, we can simply check that l_1=1 iff l_2=1 for any prediction y'=$f(x')$ such that $x' \in V_x$ is a complete instance.

As mentioned in the previous section, what structural relationships between labels or subsets of labels to look for (beyond the three examples cited) and how to exploit them is not a trivial question.

4.5 Experimental analysis

This section presents the experimental study carried out to evaluate our approach. The datasets used in our experiments are publicly available and can be found at Kaggle[33] or at UCI [34]. The Yelp dataset can be found following this link [35]. Numerical and categorical attributes are binarized. The textual datasets used are pre-preprocessed and binarized.

Dataset	#instances	#classes	#features	data type
Augmented MNIST	70000	13	784	Images
Yelp Review Analysis	10806	5	671	Textual
IMDB Movie Genre Prediction	65500	24	30	Textual
Patient Characteristics Survey (NYS 2015)	105099	5	63	Textual/ Numeric

Table 4.4: Properties of the different data-sets used.

[33] www.kaggle.com
[34] archive.ics.uci.edu/ml/
[35] https://www.ics.uci.edu/~vpsaini/

In order to enumerate our symbolic explanations for binary classifiers, we rely on a SAT-based oracle where the enumeration of counterfactuals is done using the enumcs tool[GIL18] and the sufficient reasons are enumerated thanks to the duality hitting set. The time limit for the enumeration of symbolic explanations was set to 300 seconds. All experiments presented in this section have been run on a cluster of computers equiped with quadcore bi-processors Intel Xeon E5-2643 3 (3.3 GHz) and 64 GB of memory running under the CentOS Stream 8.3.

Hyper-parameters tuning

A wide range of hyper-parameters were explored in attempts to reach relatively the best performance of the surrogate models associated to the different multi-label classifiers. We conducted experiments using different radius values for each dataset in order to determine the vicinity of the explained sample that ensures a good faithfulness. The hyper-parameters tuning in this section is done using the randomized search method (RandomizedSearchCV) from Scikit-learn library in its version v0.22.1. Table 4.5 lists the hyper-parameters used to train the surrogate models for each class. Note that some dataset properties may negatively influence the performance of the surrogate model and must be taken into account. For instance, the prediction task becomes harder with an imbalanced data distribution or data with low density. An upper bound for the vicinity size of an explained instance was set to 400 samples.

Table 4.5: Tuned hyper-parameters used for the surrogate models of each of the labels of the different datasets.

	label	$ntree$	max_depth	max features	min samples split	min samples leaf	acc train (%)	acc test (%)
Yelp	l_1	19	100	auto	8	4	97.3	92.7
	l_2	19	28	sqrt	16	1	97.8	87.4
	l_3	21	100	sqrt	40	4	95.7	89.1
	l_4	11	28	auto	2	1	100	97.6
	l_5	9	100	sqrt	40	2	96.6	94.5
MNIST$_{ml}$	l_1	13	28	sqrt	8	1	99.9	98.9
	l_2	19	50	sqrt	8	1	99.9	99.1
	l_3	13	100	auto	16	1	99	98.1
	l_4	9	28	auto	8	1	99.8	96.6
	l_5	13	36	sqrt	2	2	99.2	97.6
	l_6	11	36	auto	2	2	100	96.8
	l_7	11	28	auto	8	4	99.6	98.6
	l_8	21	14	sqrt	16	1	99	97.6
	l_9	9	100	auto	2	1	100	97
	l_{10}	13	28	sqrt	8	4	99	96
	l_{11}	11	14	auto	8	1	99.2	95.5
	l_{12}	19	100	auto	8	2	99.8	95.5
	l_{13}	19	36	sqrt	16	1	99.1	94.1
NYS15	l_1	12	14	auto	8	8	100	100
	l_2	3	28	auto	40	4	97.8	98.4
	l_3	12	7	sqrt	8	8	99.7	99.9
	l_4	12	14	auto	2	1	100	99.4

	l_5	12	28	auto	2	2	99.2	97.6
IMDB	l_1	15	7	sqrt	16	4	99.9	100
	l_2	12	28	auto	16	1	99.9	99.7
	l_3	12	14	auto	8	4	100	100
	l_4	15	28	sqrt	40	1	99.9	99.8
	l_5	9	28	sqrt	16	4	100	100
	l_6	15	28	sqrt	2	1	100	99.7
	l_7	9	7	auto	8	4	100	100
	l_8	15	14	sqrt	2	1	100	100
	l_9	15	7	sqrt	16	1	100	100
	l_{10}	6	7	auto	40	4	100	100
	l_{11}	15	14	sqrt	2	2	100	100
	l_{12}	9	14	sqrt	2	4	100	100
	l_{13}	9	7	sqrt	40	8	100	100
	l_{14}	15	28	sqrt	2	4	99.8	99.8
	l_{15}	15	28	auto	16	8	100	100
	l_{16}	12	28	sqrt	2	1	100	98
	l_{17}	12	7	auto	40	2	99.9	99.8
	l_{18}	15	28	sqrt	16	2	99.9	99.8
	l_{19}	12	14	auto	40	8	100	100
	l_{20}	12	28	sqrt	2	8	99.9	100
	l_{21}	15	28	sqrt	8	8	99.8	100
	l_{22}	12	28	auto	2	4	100	100
	l_{23}	3	28	sqrt	40	1	100	100
	l_{24}	9	7	auto	16	1	100	100

4.5.1 Results

In order to generate entire-outcome explanations, each base classifier of the binary relevance (BR) model is approximated using a random forest and then encoded into a CNF formula. Table 4.6 lists the average size and time of the encoding step computed over surrogate models. We can see that the average accuracy of the surrogate random forest classifiers is high meaning that the surrogate models can achieve high faithfulness levels wrt. the original model. Regarding the size of the generated CNFs expressed as the number of variables (Vars) and number of clauses (CLs), one can see that it is tractable and it is easily handled by the SAT-solver (in Step 2).

Dataset	radius	AVG RF acc (%)	MIN $\|\Sigma\|$	AVG $\|\Sigma\|$	MAX $\|\Sigma\|$	MIN $runtime$ (s)	AVG $runtime$ (s)	MAX $runtime$ (s)
YELP	60	92.67	96/232	4827/13004	13732/36864	0.48	3.29	13.73
Review	180	92.73	4625/12416	6812/18395	15963/428941	2.97	4.64	15.32
Augmented	150	93.97	509/1268	12095/32353	14308/38344	0.68	12.58	16.13
MNIST	250	96.27	423/1119	9556/25455	15105/40530	1.35	7.93	14.41
IMDB	30	99.53	863/2344	1282/3533	3149/8558	0.82	1.09	2.73
NYS15	63	96.73	2446/6615	7887/21370	11305/30594	1.91	6.73	10.12

Table 4.6: Evaluating the CNF encoding over different datasets.

Table 4.7 shows the results of enumerating both sufficient reasons and counterfactuals explanations. Using local surrogate models over multiple values of the radius, the symbolic explanations of each base classifier are enumerated. An average is then computed and is given in Table 4.7 and Table 4.8. We notice over the different datasets that the average time necessary to enumerate all the explanations for a given instance grows linearly and varies from 2 to 20 seconds to find all the possible explanations of all the base classifiers. For instance, if we compare the enumeration time of mono-label classifiers trained on the MNIST dataset versus the multi-label classifier, we can see that the trend is the same. The same finding holds for the number of explanations where one can see that on average this number increases proportionally to the size of the input features set. We also notice that the number of SR explanations is of the same order as the number of CF ones. Interestingly enough, one can notice that the time required to find one sufficient reason (resp. counterfactual) explanation is very negligible, meaning that the proposed approach is feasible in practice and allows to explain medium sized multi-label classifiers efficiently.

Dataset	radius	MIN #CFs	AVG #CFs	MAX #CFs	runtime One CF (s)	MIN $runtime$(s)	AVG $runtime$(s)	MAX $runtime$(s)
YELP	60	1891	2025	6858	$\leq 10^{-3}$	$\leq 10^{-3}$	2.29	13.46
Review	180	2601	3203	9693	$\leq 10^{-3}$	0.009	4.5	29.97
Augmented	150	96	4971	9347	$\leq 10^{-3}$	0.02	15.61	33.27
MNIST	250	1158	5027	11323	$\leq 10^{-3}$	1.77	15.9	45.36
IMDB	30	5	14	22	≈ 0	0.13	2.78	7.47
NYS15	63	134	1052	2399	$\leq 10^{-4}$	0.15	2.83	9.37

Table 4.7: Enumeration of entire-outcome counterfactual explanations.

Dataset	radius	MIN #SRs	AVG #SRs	MAX #SRs	enumtime One SR (s)	MIN enumtime(s)	AVG enumtime(s)	MAX enumtime(s)
YELP Review	60	13116	23167	38620	0.028	10.94	19.37	31.95
Augmented MNIST	150	11292	11956	12621	0.053	12.26	13.06	13.85
IMDB Movie Genre	30	3	41.83	161	0.004	0.003	0.02	0.07

Table 4.8: Enumeration of entire-outcome sufficient reasons explanations.

Fine-grained explanations

The following results concern only fine-grained CF explanations. However, the findings still hold for the enumeration of the fine-grained SR explanations. We vary the size k[36] of the target partial prediction y_{target} and we choose randomly the labels that compose it, i.e. the labels we are going to explain given an instance x. The results are obtained on a BR model trained on each dataset. Both cases of negative and postive predictions are considered and are presented in two parts: we use the acronym POS to refer to the explanations of the positively predicted labels and the acronym NEG for those of the negatively predicted. Similarly to the entire-outcome explanations, we are first interested in measuring the size of the CNF encoding associated to the labels of interest ($l_i \in y_{target}$), and also we are interested in the enumeration times needed and the size of the sets of explanations generated.
Table 4.9 lists the average size of the CNF formula and runtime of the encoding step. We can see that the

[36]Recall that k takes values $\in [1, |Y|]$ with $|Y|$ being the number of classes recognized by the MLC.

Dataset	Pred Type	AVG RF's accuracy	MIN $\|\Sigma\|$ (Vars/CLs)	AVG $\|\Sigma\|$ (Vars/CLs)	MAX $\|\Sigma\|$ (Vars/CLs)	MIN runtime (s)	AVG runtime (s)	MAX runtime (s)
YELP	*NEG*	93.5%	39/94	4663/12572	18616/49938	0.6	3.75	23.16
Review	*POS*	89.95%	28/62	4071/10932	12043/32308	0.5	2.94	9.67
Augmented	*NEG*	97.48%	103/250	6552/17352	15557/41797	0.69	5.32	15.36
MNIST	*POS*	92.36%	117/292	10808/28840	18441/49788	0.97	10.43	21.18
IMDB	*NEG*	99.67%	279/732	1426/3860	7556/20261	0.79	1.56	7.55
Movie	*POS*	98.37	128/296	3002/8264	10504/28378	0.68	2.61	8.69
Genre								

Table 4.9: Evaluationg the CNF encoding size of the different datasets.

size of the CNFs on average is tractable and that it is almost the same result obtained when enumerating the entire-outcome explanations.

Table 4.10 presents the results of enumerating fine-grained counterfactuals. First, we can say that the size of the CNFs has practically not changed compared to the entire-outcome explanations (see Table 4.6). We clearly notice that on average the number of explanations generated has strongly decreased compared to the results of Table 4.7. This finding makes sense since we have reduced the number of labels to explain. Secondly, we notice that the average time necessary for enumerating one counterfactual is negligible and meets the results of Table 4.7.

	Operation type	$\|y_{target}\|$	AVG $\#CFs$	MAX $\#CFs$	enumtime One CF (s)	AVG $\|CF\|$	AVG enum-time(s)	AVG enum-time(s)
YELP	Expansion (NEG to POS)	1	977	5009	$\leq 10^{-4}$	2	0.83	7.16
		2	1678	6716	$\leq 10^{-4}$	2	1.65	11.83
		3	2102	7876	$\leq 10^{-4}$	3	2.17	20.19
		4	3395	8933	$\leq 10^{-4}$	3	4.32	17.4
	Contraction (POS to NEG)	1	848	6550	$\leq 10^{-4}$	2	0.98	12.99
		2	490	5066	$\leq 10^{-4}$	2	0.52	8.01
		3	441	3723	$\leq 10^{-4}$	2	0.4	5.14
MNIST ML	Expansion (NEG to POS)	1	1932	10011	$\leq 10^{-3}$	3	3.18	27.08
		2	2158	9769	$\leq 10^{-3}$	5	3.95	39.56
		3	3895	8395	$\leq 10^{-3}$	5	8.9	28.52
	Contraction (POS to NEG)	1	2799	14759	$\leq 10^{-3}$	2	5.26	54.19
		2	3821	12895	$\leq 10^{-3}$	3	9.54	48.86
		3	4642	12007	$\leq 10^{-3}$	4	12.10	43.79
IMDB	Expansion (NEG to POS)	1	6	30	≈ 0	2	0.0023	0.04
		2	10	22	≈ 0	2	0.0041	0.021
		3	10	43	≈ 0	2	0.0038	0.024
	Contraction (POS to NEG)	1	16	60	≈ 0	1	0.009	0.02
		2	16	62	≈ 0	1	0.008	0.02
		3	23	93	≈ 0	1	0.01	0.03

Table 4.10: Enumerating fine-grained counterfactuals.

4.6 Conclusion

In the literature, very few studies have focused on explaining multi-label classifiers, thus it is not obvious to compare our approach to existing ones. Generating symbolic explanations for multi-label classification is quite straightforward to accomplish within the symbolic framework presented in the Chapter 3. However, it is worth noticing that the contributions of this work are not simple extensions from the multi-class framework to the multi-label one since there are, for example, concepts specific to the multi-label case such as label-based and fine-grained explanations. We introduced the concept of the label-based explanations in order to take advantage of the structural relationships between labels in order to reduce the number of generated explanations and improve their presentation to the user.
The declarative paradigm used in our work has been successfully used in explainable AI [Ber21] but also for other problems such as declarative data mining where one can enumerate for example frequent itemsets using a SAT oracle [JSS15]. Doing so, we take advantage of the strengths of modern SAT solvers used as oracles. In the case of ML models that do not admit direct CNF representation, a crucial component of the proposed approach is to approximate the model with another one that does admit such a representation. We intend in future work to focus on the extraction of label-based explanations and their combination with feature-based ones in order to reduce the number of explanations and improve their presentation. We also go beyond symbolic explanations and address score-based explanations in a multi-label setting in the next part of this manuscript.

Part III

Feature-attribution explanations

5

Feature attribution explanations for single-label classification

Large amounts of efforts have been devoted to developing approaches to explain individual classification decisions such as decision rules [RSG18], counterfactual examples [WMR17] and logic-based approaches [SCD18b, DH22, INMS19a, Ign20, INMS19b, ABB+22a]. We will focus in this part on a type of explanation well known in XAI called feature attribution (e.g. SHAP [LL17], LIME [RSG16]). We start with providing a definition of feature attribution explanation in a general setting as well as the main approaches that generate them in Section 5.1. In section 5.2.1 we will present the set of fine-grained properties we propose to analyze, rank and select explanations and in section 5.2.2 the ones allowing to assess the relevance of features. We also propose some scoring functions to check out the suggested properties. Full examples of explanations produced by the whole approach and an evaluation are provided in section 5.3.

5.1 Feature attribution explanations

Feature attribution is a popular and widely used approach for explaining the predictions of machine learning models and is by far the most well-studied explainability technique [BSH+10, GBY+18]. Depending on the application domains and the nature of the data, they are sometimes referred to as pixel saliency, saliency maps, rationales, attentions, feature-level interpretations, feature importance or simply feature attributions. Intuitively, a feature attribution method associates a numerical score with each input feature reflecting the weight or influence of this feature in the prediction of a machine learning model. It is worth to notice that there is no consensus on the definition or the semantics of these feature attributions nor on the way to systematically evaluate their relevance. In practice, this can be seen through the various existing methods to derive feature importance scores reflecting the influence of each attribute for prediction tasks such as regression or multi-class classification.

5.1.1 Review of related works

As mentionned in the state-of-the-art chapter, explainable AI has focused on two main families of post-hoc approaches when it comes to explaining machine learning models. The first one focuses on formal XAI to provide *symbolic* explanations ([SCD18b],[INMS19b],[Rei87],[Rym94],[INMS19b]) and are also used for verification and diagnosis purposes ([Rei87], [Rym94], [INMS19b]), while the second one provides insights into how much each feature contributed to the outcome decision and focuses on attributing *numerical* importance scores to the input features. These techniques are used across many dif-

ferent domains and are popular amongst machine learning scientists who want to sanity check a model before deploying it [BXS+20]. The main objective of our contribution is to firstly to provide complementary types of symbolic explanations and secondly provide score-based ones for a better understanding and usability of explanations. The first part of our work was presented previously in Chapter 3 of part one of this manuscript. We will get into more details about the second part of our contribution within this chapter and present how we provide feature attribution explanations based on some properties.

Feature attribution consists in associating scores to each input feature X_i of an instance x to reflect to what extent X_i contributed to the prediction $f(x)$. Formally, a feature attribution for multi-class tasks is defined as follows :

Definition 34 (Feature attribution). *Assume we are given a multi-class classifier f and a feature attribution method h. Assume also a data instance $x = (x_1,..,x_n) \in X$. An attribution of the prediction $y = f(x)$ at input instance x is the vector $h(f, x) = (a_1, ..., a_n)$ where each score $a_i \in \mathbb{R}$.*

According to the considered method, a feature attribution a_i for the feature X_i may denote sensitivity [SVZ13b], relevance [BBM+15], local influence [RSG16], Shapley values [LL17], filter activations [NHWF21], etc. Examples for common feature attribution methods are provided in Section 1.2.2 and we can cite SHAP [LL17], LIME [RSG16], saliency maps [SVZ13a, ZF14, STY17, SDBR15, PLPN19], Grad-CAM [SCD+17], integrated gradients [STY17], LRP [BBM+15] and DeepLift [SGK17] (see [LPK20, LPK21] for recent literature surveys).

In the rest of this chapter, we will present how we widened the scope of our investigation to compute feature attribution based on symbolic explanations presented previously (cf Chapter 3). In the case of a single-label classification, we limit ourselves in our study to proposing score-based explanations and feature attribution by relying on such explanations and associating scores w.r.t to some properties. As for the multi-label setting, we will expand this study in Chapter 6 to consider other explainability approaches as oracles and see how to combine or aggregate their output to propose feature attribution for multi-label decisions.

5.2 Feature attribution explanations for single-label classification

The symbolic explanations we proposed so far to explain individual classification decisions based on the observations that MUSes and MCSes are cornerstones of analyzing thus measuring inconsistencies [LS08] and can be obtained in large numbers. This causes a selection problem as the whole set of explanations can be very large. Indeed, considering the SAT-based modeling we adopt in our approach, an inconsistent Boolean formula can potentially have a large set of explanations (MUSes and MCSes). More precisely, for a knowledge base containing p clauses, the number of MUSes and MCSes can be in the worst case exponential in p [LS08]. Thus, defining how to measure the quality of an explanation and to convey how much the model relies on certain features to make a decision at some specific input becomes necessary in order to focus on those providing more insights.
Figure 5.1 depicts the overview of our XAI approach at **Step 3**. Indeed, in addition to the large number of symbolic explanations, the different needs of users and nature of systems from an application to another raise the question of *which explanations to choose ?* or *which explanations and (or) what features are most relevant ?* We address all of the issues raised above by giving at the same time a local score-based explanation with respect to some properties[37], and also, feature attribution providing feature-level importance scores for how much a given input feature contributes to a model's output. These properties can be interpreted as relevance characteristics of an explanation [38] and are presented in Section 5.2. We

[37] Those properties can provide information on certain aspects of the generated explanations.
[38] Of course, the relevance depends on the user's interpretation and the context.

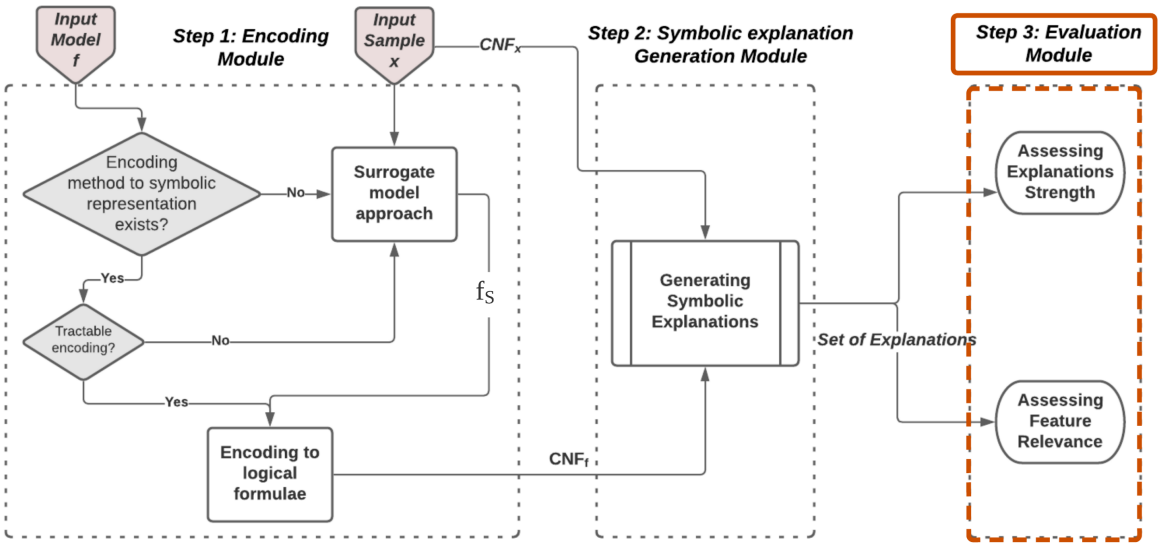


Figure 5.1: Focus on the Step 3 of the proposed approach

also propose some examples of scoring functions to assess them numerically. This corresponds to **Step 3 (Explanation and feature relevance scoring)** in Figure 5.1.

Before we start the presentation of the properties proposed to assess the relevance of both explanations and features, we introduce some notations and definitions essential for understanding the rest of the work:

- Let $E(x, f)$ be a non empty set denoting the set of explanations (either ***SR*** or ***CF***) for an input instance x predicted negatively by the classifier f.
- An explanation is denoted by e_i where $i \in [0, |E(x, f)|]$.
- The neighborhood of x within the radius r denoted $V(x, r)$ (also written V_x for short) is formally defined as $V(x, r)$:$\{v \in X \mid \text{dist}(x, v) \leqslant r\}$[39].
- Given an explanation e_i, let *size(e_i)* denote the number of variables composing it.

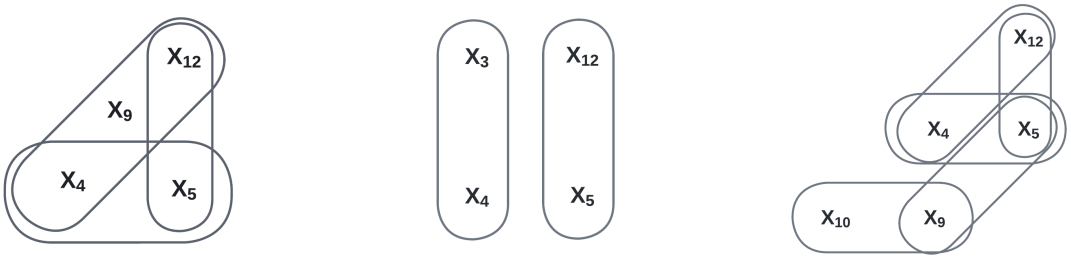
Definition 35 (Extent of an explanation)**.** *Let Extent(e_i, x, r) be the set of data instances defined as follows: $\{v \in V(x, r) \mid f(v)$=$f(x) = 0$ and for $e_i \in E(v, f)\}$.*

Intuitively, *Extent(e_i, x, r)* denotes the set of data instances from the neighborhood of x that are negatively predicted by f and sharing the explanation e_i. We provide illustrative examples to describe the different concepts we introduce in this section.

Example 40. *Let's continue with the United Stated Congressional Voting Records (firstly introduced in example 19) where a trained model f classify instances as 'Republican' or 'Democrat' based on the 16 key votes identified. Given a radius set to 3 ($r = 3$), then the neighborhood of x is $V(x, 3) = \{x', x''\}$. Let the set of sufficient reason explanations associated to each instance from $V(x, 3)$ as follows :*

The extent of the different explanations $\in Exp(f, x)$ are illustrated in Figure 5.3. As shown, an explanation belonging to several set of explanations $E(x_i, f)$ is present in every set highlighted with

[39] dist(x,v) denotes a distance measure that returns the distance between x and v.



(a) MUSes of $\Sigma_{f,x}$. (b) MUSes of $\Sigma_{f,x'}$. (c) MUSes of $\Sigma_{f,x''}$.

$$E(x,f) = \{\ e_1 = \{X_4, X_5\}, e_2 = \{X_5, X_{12}\}, e_3 = \{X_4, X_9, X_{12}\}\ \}$$

$$E(x',f) = \{\ e'_1 = \{X_5, X_{12}\}, e'_2 = \{\ X_3, X_4\}\ \}$$

$$E(x'',f) = \{\ e''_1 = \{X_5, X_{12}\}, e''_2 = \{X_4, X_5\}, e''_3 = \{X_9, X_{10}\}, e''_4 = \{X_5, X_9\}\ \}$$

the same color. For instance, we can see that the explanation e_1 is present in every $E(x_i, f)$ while explanation e_3 is specific to the sample x.

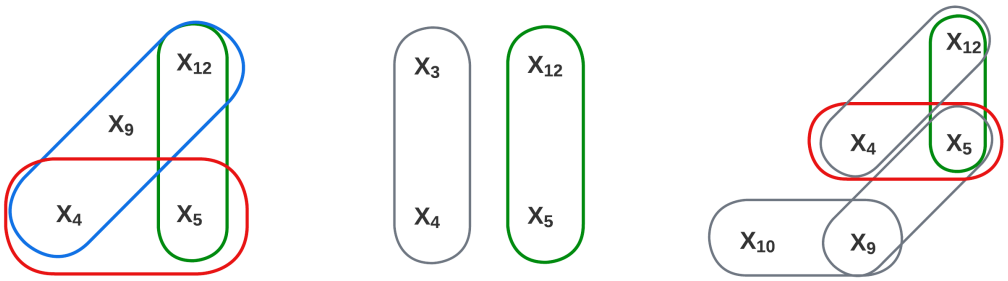


Figure 5.3: Extent of $E(x, f)$ in the neighborhood $V(x, 3) = \{x', x''\}$.

Hence, the extent sets of explanations in $E(x, f)$ are represented in the following :

$$Extent(e_1, x, 3) = \{\ x, x''\}$$

$$Extent(e_2, x, 3) = \{\ x, x', x''\ \}$$

$$Extent(e_3, x, 3) = \{\ x\ \}$$

We define now a similar notion to the extent of an explanation but at a feature-level.

Definition 36 (Cover of a feature)**.** *Let Cover(X_k,x) be the set of explanations from $E(x, f)$ where the feature X_k is involved (namely Cover(X_k,x)=$\{e_i \mid X_k \in e_i$ for $e_i \in E(x, f)\}$).*

A cover of a feature designates the set of explanations including the variable X_i. It can be computed within set of explanations associated to x (i.e. $E(x, f)$) or the set of explanations of instances from the locality of x (i.e. $\bigcup_{v \in V(x,r)} E(v, f)$).

Example 41 (Example 40 continued'). *The cover of the different variables composing the explanations of x within $E(x, f)$ is illustrated in Figure 5.4 and given in the following :*

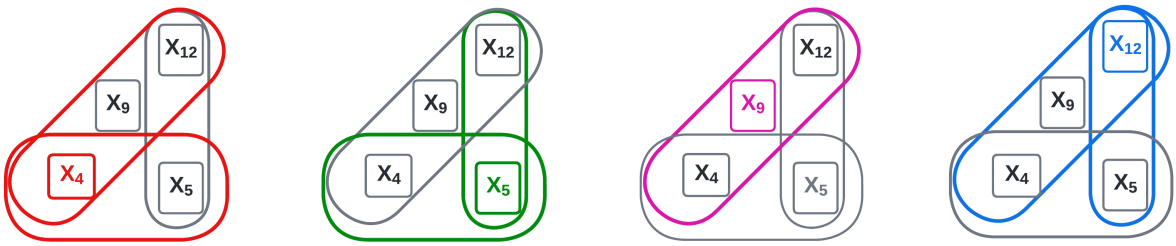


Figure 5.4: Cover of the variables in the explanation set $E(x, f)$.

$$
\begin{aligned}
Cover(X_4, x) &= \{ e_1, e_3 \} \\
Cover(X_5, x) &= \{ e_1, e_2 \} \\
Cover(X_9, x) &= \{ e_1 \} \\
Cover(X_{12}, x) &= \{ e_2, e_3 \}
\end{aligned}
$$

The cover of the different variables composing the explanations of x within $E(v, f)$ are illustrated in Figure 5.5 and given in the following : $v \in V(x,3)$

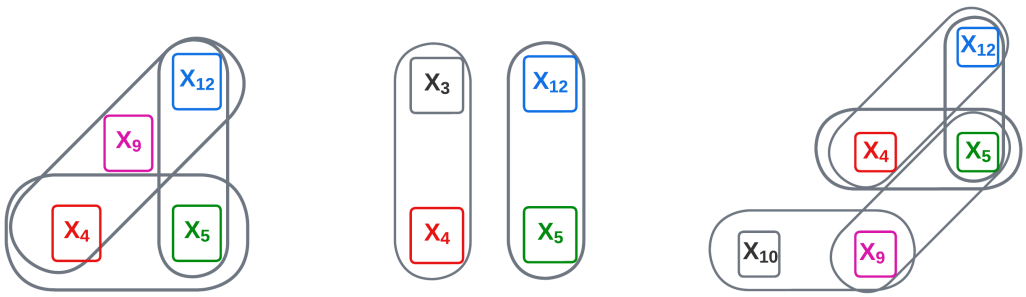


Figure 5.5: Cover of variables in $E(x, f)$ in the neighborhood $V(x, 3)$.

$$\underset{v \in V_x}{Cover}(X_4, v) = \{ e_1, e_3, e'_2, e''_2 \}$$

$$\underset{v \in V_x}{Cover}(X_5, v) = \{ e_1, e_2, e'_1, e''_1, e''_2, e''_4 \}$$

$$\underset{v \in V_x}{Cover}(X_9, v) = \{ e_3, e''_3, e''_4 \}$$

$$\underset{v \in V_x}{Cover}(X_{12}, v) = \{ e_2, e_3, e'_1, e''_1 \}$$

5.2.1 Properties of symbolic explanations and scoring functions

In this section, we propose three natural properties to evaluate explanations and capture some of the aspects allowing their analysis, ranking and selection :

Parsimony ($\mathcal{PAR}$) : The parsimony is a natural property allowing to select the simplest or shortest explanations (namely, explanations involving less features). The intuition behind a simpler explanation is that it is easier to focus on understanding or changing few variables instead of trying to change many features. Hence, the parsimony score of an explanation e_i should be inversely proportional to it's size. Formally, given a data instance x, its set of explanations $E(x, f)$: For two explanations e_1 and e_2 from $E(x, f)$: $\mathcal{PAR}(e_1) > \mathcal{PAR}(e_2)$ iff size(e_1)<size(e_2). An example of a scoring function satisfying the parsimony property is :

$$S_{\mathcal{PAR}}(e_i) = \frac{1}{size(e_i)} \tag{5.1}$$

Generality ($\mathcal{GEN}$) : This property aims to reflect how much an explanation can be general to a multitude of data instances, or in the opposite, reflect how much an explanation is specific to the instance x. Intuitively, the generality of an explanation should be proportional to the number of data instances it explains. Given a data instance x, its set of explanations $E(x, f)$, its neighborhood $V(x, r)$ and two explanations e_1 and e_2 from $E(x, f)$: $\mathcal{GEN}(e_1) > \mathcal{GEN}(e_2)$ iff $|\textit{Extent}(e_1, x, r)| > |\textit{Extent}(e_2, x, r)|$. An example of a scoring function capturing this property is :

$$S_{\mathcal{GEN}}(x, r, e_i) = \frac{|Extent(e_i, x, r)|}{|V(x, r)|} \tag{5.2}$$

Intuitively, this scoring function assesses the proportion of data instances in the neighborhood of the instance x that are negatively predicted and that share the explanation e_i.

Explanation responsibility ($\mathcal{RESP}$) : This property allows to answer the question how much an explanation is responsible for the current prediction. Intuitively, if there is a unique explanation, then this latter is fully responsible of the decision (only cause of the decision). Hence, the responsibility of an explanation should be inversely proportional to the number of explanations in $E(x, f)$. Given two different data instances x_1 and x_2, their respective explanation sets $E(x_1, f)$ and $E(x_2, f)$ and an explanation $e_i \in E(x_1, f)$ and $e_j \in E(x_2, f)$, we have:
$\mathcal{RESP}(x_1, e_i) < \mathcal{RESP}(x_2, e_j)$ iff $|E(x_1, f)| > |E(x_2, f)|$. For a given data instance x, the responsibility of $e_i \in E(x, f)$ could be evaluated using the following scoring function :

$$S_{\mathcal{RESP}}(x, e_i) = \frac{1}{|E(x, f)|} \tag{5.3}$$

Remark 9. *The scoring function of Eq. 5.3 assigns the same score to every explanation in $E(x, f)$. It means that it can be used to compare two different sets of explanations but it can not be used to evaluate explanations within the same set.*

To remedy the problem raised in the previous remark, we propose to define a more granular evaluation of the responsibility of explanations in $E(x, f)$ by calculating a responsibility score for each e_i in the neighborhood of x. An example of a scoring function capturing this property, would be :

$$S_{\mathcal{RESP}}(x, r, e_i) = \max_{v \in V(x,r) | e_i \in E(v,f)} (S_{\mathcal{RESP}}(v, e_i)) \tag{5.4}$$

Example 42 (Example 40 continued)**.** *Given the set of sufficient reason explanations of x and the instances from its neighborhood, we evaluate the relevance of explanations $e \in E(x, f)$ w.r.t to the properties and by the means of the scoring functions presented above. These score-based explanations are then ranked in order to choose the more convenient one w.r.t to the need of the user for whom these explanations are intended.*

explanation \ *criteria*	$S_{\mathcal{PAR}}(e_i)$	$S_{\mathcal{GEN}}(x, r, e_i)$	$S_{\mathcal{RESP}}(x, e_i)$	$S_{\mathcal{RESP}}(x, r, e_i)$
e_1	**1/2**	2/3	1/3	1/3
e_2	**1/2**	**3/3**	1/3	**1/2**
e_3	1/3	1/3	1/3	1/3

Table 5.1: Evaluating explanations w.r.t to the $\mathcal{PAR}$, $\mathcal{GEN}$ and $\mathcal{RESP}$ properties.

These properties make it possible to analyze and if necessary select or order the symbolic explanations according to a particular property. Of course, we can define other properties or variants of these properties (e.g. relative parsimony to reflect the parsimony of one explanation compared to the parsimony of the rest of the explanations). The properties can have a particular meaning or a usefulness depending on the applications and users. It would be interesting to study the links and the interdependence between these properties. Let us now see properties allowing to assess the relevance of the features reflecting their contribution to the prediction.

5.2.2 Properties of features-based explanations and scoring functions

It is possible to have inside the set of explanations the same cardinality and extent for different explanations, and therefore, they cannot be distinguished by properties such as *parsimony, generality and responsibility*. Hence, we consider properties at a feature-level by taking a look into the variables composing the explanation and trying to assess their relevance. We propose the following properties for the features:

Feature involvement ($\mathcal{FI}$) : This property is intended to reflect the extent of involvement of a feature within the set of explanations. The intuition is that a feature that participates in several explanations of the same instance x should have a higher importance compared to a less involved feature. Given a data instance x, its set of explanations $E(x, f)$, and two features X_1 and X_2:
$\mathcal{FI}(X_1,x) > \mathcal{FI}(X_2,x)$ iff $|Cover(X_1,x)| > |Cover(X_2,x)|$. An example of a scoring function capturing this property is :

$$S_{\mathcal{FI}}(X_k, x) = \frac{|Cover(X_k, x)|}{|E(x, f)|} \tag{5.5}$$

Feature generality ($\mathcal{FG}$) : This property captures to what extent a feature is frequently involved in explaining instances in the vicinity of the sample to explain x. Given a sample x, its vicinity $V(x,r)$ and their explanation set $E(V(x,r),f)$ defined as $\bigcup_{v \in V(x,r)} E(v,f)$ and two features X_1 and X_2 : $\mathcal{FG}(X_1) > \mathcal{FG}(X_2)$ iff $|\bigcup_{v \in V(x,r)} Cover(X_1,v)| > |\bigcup_{v \in V(x,r)} Cover(X_2,v)|$. An example of a scoring function capturing this property could be :

$$S_{\mathcal{FG}}(X_k) = \frac{|\bigcup_{v \in V(x,r)} Cover(X_k,v)|}{|\acute{x} \in E(V(x,r),f)|} \tag{5.6}$$

Feature responsibility ($\mathcal{FR}$) : This property is intended to reflect the responsibility or contribution of a feature X_i within the set of symbolic explanations of x. Intuitively, the responsibility of a feature should be inversely proportional to the size of the explanations where it is involved (the shortest the explanation, the highest the responsibility value of its variables). Given two features X_1, X_2 with non empty covers: $\mathcal{FR}(X_1) > \mathcal{FR}(X_2)$ iff $\underset{e_j \in Cover(X_1,x)}{aggr}(size(e_j)) < \underset{e_j \in Cover(X_2,x)}{aggr}(size(e_j))$ where $aggr$ stands for an aggregation function (e.g. $\min$, $\max$, AVG, etc.). An example of a scoring function satisfying this property is :

$$S_{\mathcal{FR}}(X_k) = \frac{1}{\underset{e_j \in Cover(X_k,x)}{AVG}(size(e_j))} \tag{5.7}$$

Example 43. *Similarly to Example 42 on score-based explanations, we will take an example for the computation of the different feature attributions w.r.t to the Feature involvement ($\mathcal{FI}$), Feature generality ($\mathcal{FG}$) and Feature responsibility ($\mathcal{FR}$).*

feature \ *criteria*	$S_{\mathcal{FI}}(X_k,x)$	$S_{\mathcal{FG}}(X_k)$	$S_{\mathcal{FR}}(X_k)$
X_4	2/3	4/9	1/5
X_5	2/3	**2/3**	1/4
X_9	1/3	1/3	**1/2**
X_{12}	2/3	4/9	1/5

Table 5.2: Evaluating explanations w.r.t to the $\mathcal{PAR}$, $\mathcal{GEN}$ and $\mathcal{RESP}$ properties.

In addition to the different scores associated to properties presented above, we can aggregate them (e.g., by averaging) to get an overall score depending on the user needs. To the best of our knowledge, our **agnostic** and **declarative** approach was the first that generates different types of symbolic explanations and **fine-grained** score-based ones. Note that this is not an exhaustive list of properties that one could be interested in in order to select and rank explanations or rank features according to their influence on the prediction. For instance, many desiderata can be required for an explanation in order to explain predictions in understandable terms to a user and to its expectations such as plausibility (in the sense that explanations need to be sufficiently convincing to users) [LBJ16, LCH+19, SZM19] and readability [WR15, YRS17, ALSA+17]. However, it is unclear how to choose since standard tests and benchmarks to evaluate such requirements are lacking and such evaluations still remain an open problem [DVK17, JG20].

5.3 Experimental results

This section presents some experiments of the score-based and feature attribution explanations generated according to the approach depicted within this chapter. We considered a selection of datasets from the literature and publicly available and can be found on Kaggle[40] or UCI[41]. The studied datasets are associated with binary classification tasks and are listed in Table 5.3. No pre-processing was performed on the data except the binarization of variables.

Dataset	#instances	#features	data type
MNIST	70000	784	Images
MONK's Problems	181	16	Numerical
Spect Heart	160	22	Numerical
Congressional US Voting	435	16	Numerical
Heart disease	303	9	Numerical
Breast Cancer	287	48	Numerical

Table 5.3: Properties of the datasets used.

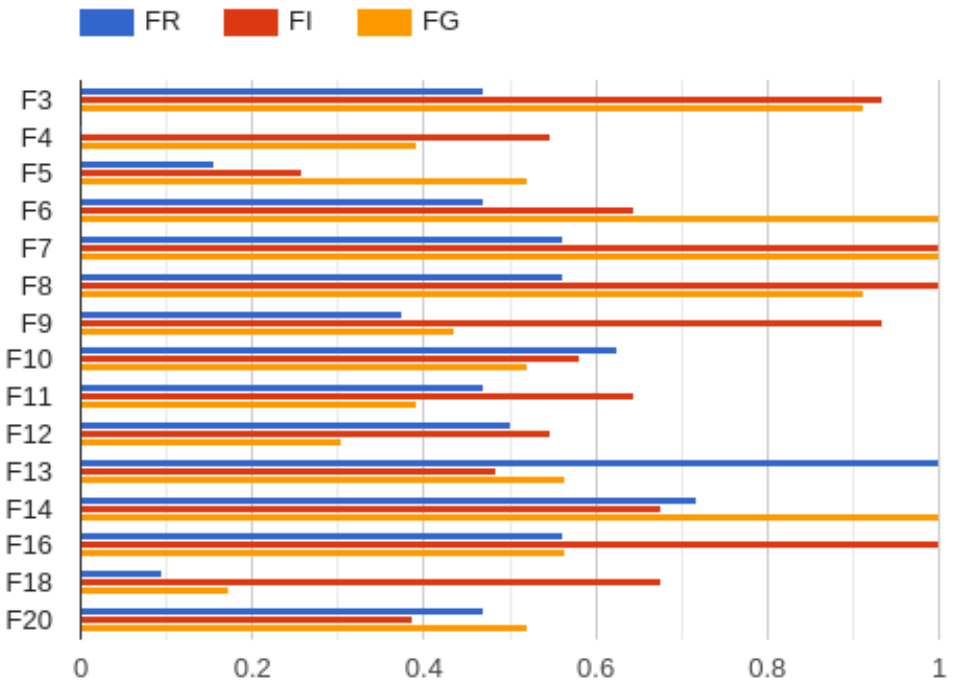
The experimental protocol used to train surrogate models is already presented in Chapter 3 and remains applicable for this section. All experiments were performed on machines equipped with an Intel Core i7-7700 (3.60GHz ×8) processors, with 32 Gb of RAM and under the Linux Cen-tOS operating system. The time-out has been set to 300 seconds for each execution of an enumeration algorithm.

We present in the following some examples of our score-based explanations provided to explain individual predictions of black-boxes trained on different datasets. In addition, we compare the relevance of our numerical explanations to the ones of similar "feature-attribution" approaches where we consider an important method in Explainable AI : the SHAP[42] tool implementing the SHAP (SHapley Additive exPlanations) approach [LL17].

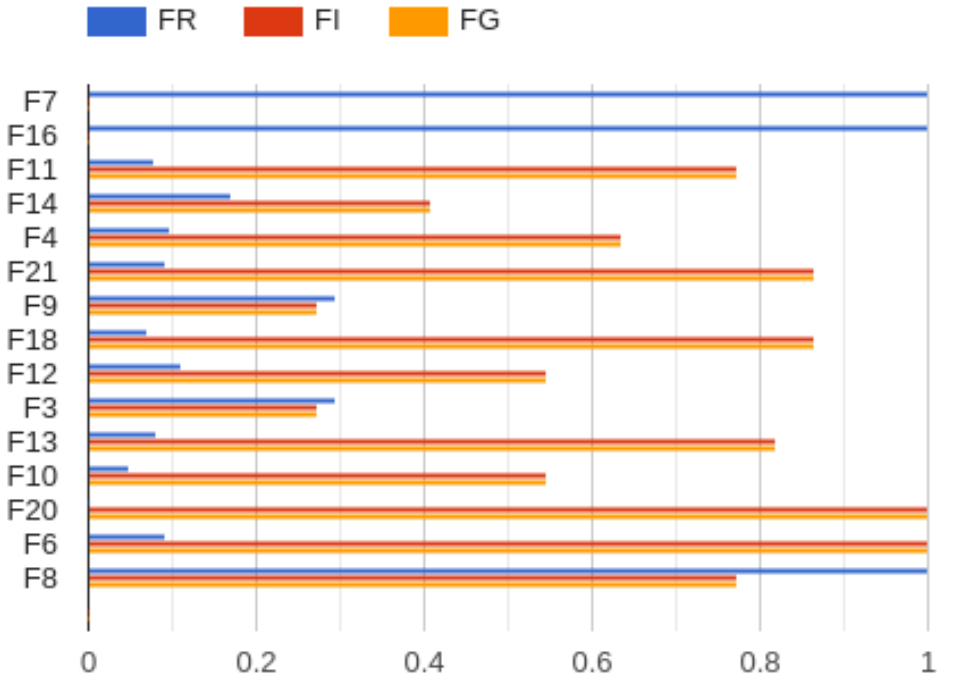
Figures 5.6a and 5.6b are an example of the feature scores associated to sufficient reasons (SR_x) and counterfactuals explanations (CF_x) enumerated to explain an instance negatively predicted from the SPECT dataset. The score-based explanations attributed to the features composing the explanations are plotted into bar charts. The blue bar represents the Feature responsibility ($\mathcal{FR}$), the red bar represents the Feature involvement ($\mathcal{FI}$) and finally, the orange bar represents the Feature generality ($\mathcal{FG}$). As observed, the order of importance of features varies depending on the desiderata chosen. An example of the SHAP explanation generated for the instance is also provided in Figure 5.7. Note that the representation of explanations can be done in different ways. For instance, it can be more convenient to represent our explanations in the form of heatmaps when the inputs are images. Such a representation allows to visualize and highlight the part of the input having the highest scores w.r.t to the different criteria. For instance, Figure 5.8 shows heatmaps corresponding to the *Feature involvement (FI)* scores (column "b-c") and *Feature rxesponsibility (FR)* (column "d-e") scores of the different input variables involved in SR_x and CF_x. Compared to SHAP explanations (Figure 5.7), ours are most often visually simpler, clearer and easier to understand and use.

To compare our results with existing methods and test if they correspond with those of SHAP, we follow the following protocol : Given a dataset, here we used the Monk's Problems, Spect Heart, Congressional US Voting, Heart disease and Breast Cancer datasets, we enumerate explanations for all instances to using a radius equal to the half of the input space size ($r = |X|/2$). We enumerate symbolic

[40] Kaggle dataset: (https://www.kaggle.com/).
[41] UCI dataset: (http://archive.ics.uci.edu/ml/).
[42] Available at https://github.com/slundberg/shap



(a) Feature attribution for SR explanations.



(b) Feature attribution for CF explanations.

Figure 5.6: Examples of explanations on a test input negatively predicted from the SPECT dataset.

explanations using our approach named ASTERYX and compute the different score-based explanations for the different features. We also generate the SHAP values for each instance. We look for the size

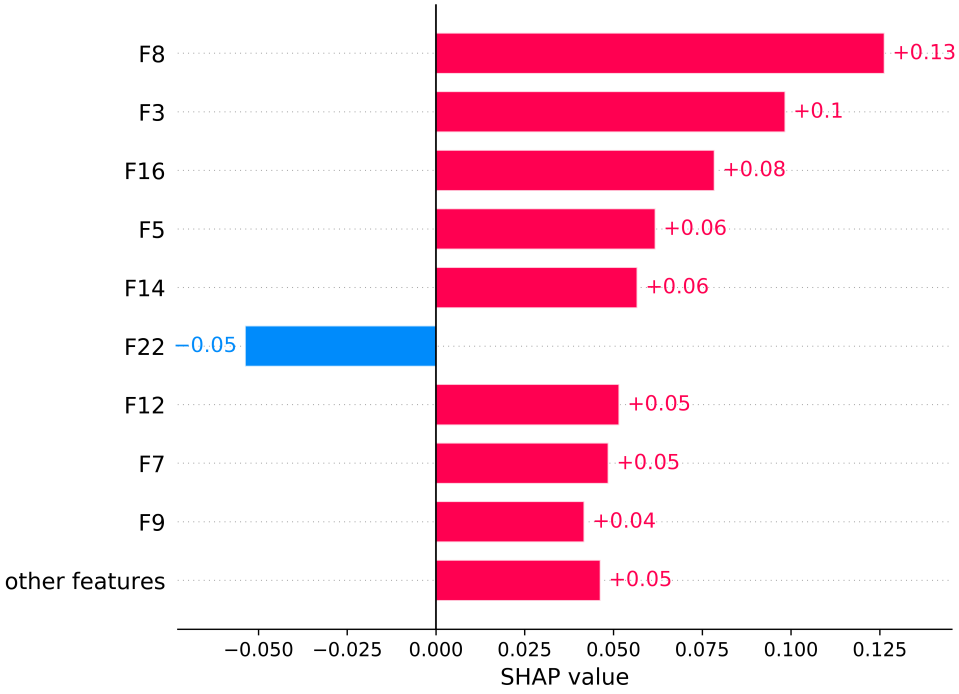


Figure 5.7: SHAP values for test input explained in Figure 5.6.

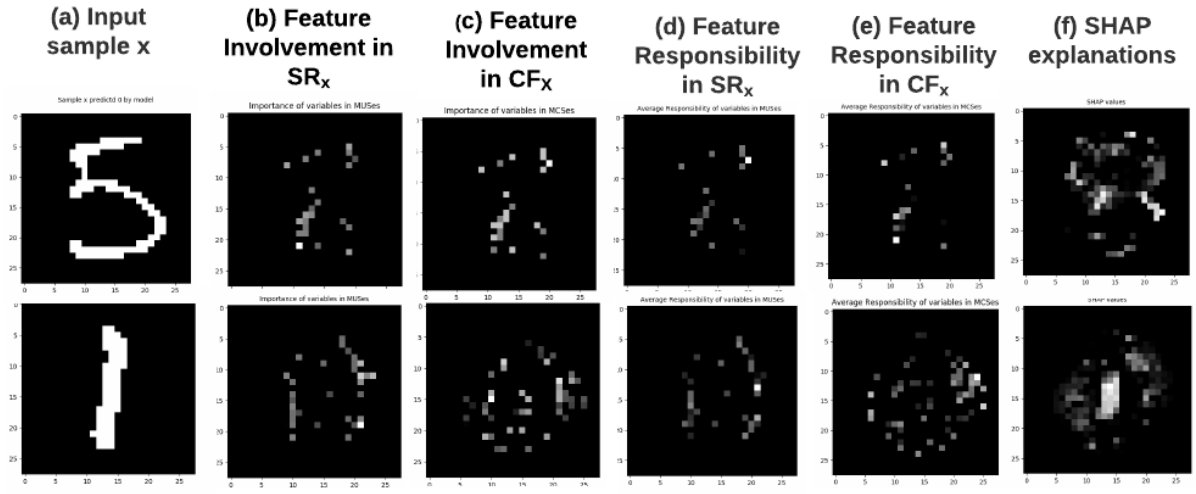


Figure 5.8: Heatmaps in columns (b-c) representing the (FI) score, and (d-e) the (FR) computed over the SR_x and CF_x of the samples data from MNIST (column a) in comparison to heatmaps of the SHAP values (column f).

of shortest SR_x explanations (m) and select the m important features having the highest SHAP values. A first observation was that finding common variables shared between the explanations of the two approaches per run was systematic. We calculated the fraction : *Number of time there are common variables* / *Number of total runs* and we obtained 100% for all the datasets used. The second observation is about the number of features shared given the m important SHAP values. The average and maximum

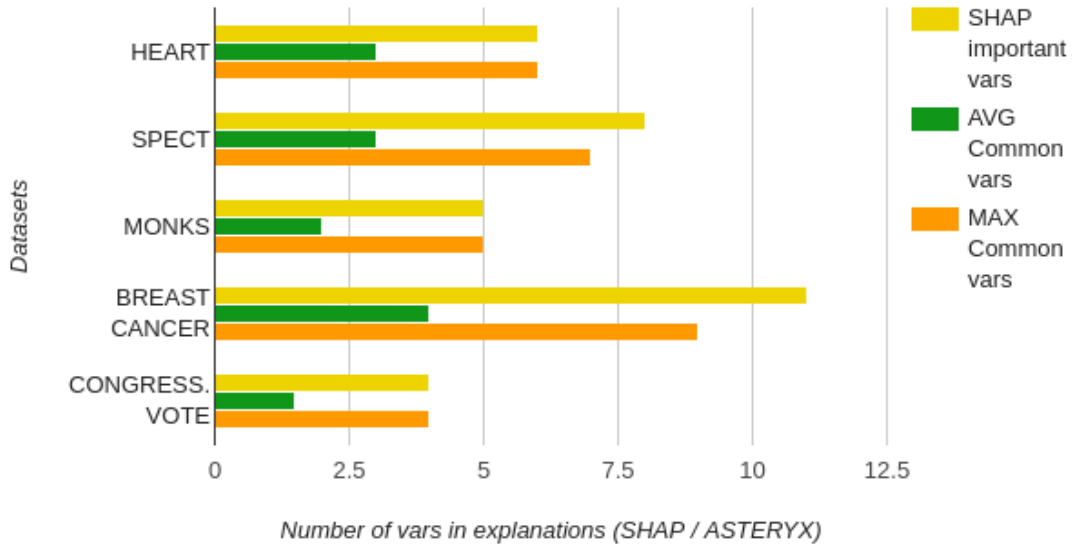


Figure 5.9: Average (and maximum) proportion of common important variables between SHAP explanations and those of our ASTERYX approach.

number of input features shared between ASTERYX's explanations and most important SHAP's features is represented in Figure 5.9. The yellow bar corresponds to the average number of most important SHAP values influencing the prediction (having a high score) and presented as an explanation. The green bar corresponds to the average number of common variables between our symbolic explanations ASTERYX and the SHAP explanations. Finally, the orange bar corresponds to the maximum number of variables shared between ASTERYX and SHAP. As observed from the results, we can approximately identify a ratio of 0.5 between the size of the set of common variables and the size of the m most important SHAP values.
As for the MNIST dataset, we compared our most important features according to the $\mathcal{FI}$ score of our approach and those of SHAP and the results on a sample of images coincide in 46% of cases, which is visually confirmed in Figure 5.8. We proceed as follows to compute such kind of correlation coefficient between the different scores proposed and the explanations calculated by SHAP: In a descending order, we consider third of the pixels of the explanations (let P be the cardinality of this set) and we compare them with P pixels having the highest SHAP values. The correlation coefficient corresponds to the result of the fraction between the size of the intersection set and P.

5.4 Conclusion

In this chapter, we have presented another type of explanations. Such explanations are numerical feature-based that aim to quantify the contribution of each feature in a prediction. In addition to the generation of two types of symbolic explanations which are *sufficient reasons* and *counterfactuals*, we associate scores reflecting the relevance of the explanations and the features w.r.t to some properties. To the best of our knowledge, our approach is the first that generates different types of symbolic explanations and **fine-**

grained score-based ones. Our experimental results show the effectiveness of the proposed approach in providing both symbolic and score-based explanations. Finally, we were able to identify that a large part of the features composing our explanations corresponded to the most influential SHAP values. In next chapter, we will see how feature attribution explanations can be used to explain multi-label predictions and what are the specificities related to such setting.

6

Feature-attribution explanations for multi-label classification

In this chapter, we will present the last part that we have explored during this thesis. We shift the attention to multi-label classification where an instance is associated with a non-empty subset of labels and we are interested in defining what a feature attribution explanation represents for such setting. We introduce the problem studied in Section 6.1. We will see in Section 6.2 how we can generalize some feature attribution techniques initially proposed in the literature for mono-label classifiers to the multi-label ones. We describe two desirable criteria to evaluate aggregation of those explanations methods. We also propose a criteria specific to multi-label classification and suggest to use it to infer feature attributions based on relationships between labels. We introduce an alternative to aggregation in Section 6.3 and propose a new attribution method based on symbolic explanations in Section 6.4.

6.1 Introduction

A lot of the post-hoc XAI methods have focused on feature-level importance scores for how much a given input feature contributes to a model's output. As stated before, depending on the considered method, a feature attribution may denote sensitivity [SVZ13b], relevance [BBM+15], local influence [RSG16], Shapley values [LL17], filter activations [NHWF21], etc. For multi-label classification where a prediction is a subset of labels, there is only a very simple feature attribution methods based on aggregating feature importance scores of the different predicted labels computed individually.

In this chapter, we are interested in feature attribution for multi-label classification tasks. The goal is to associate to each input feature X_i a score a_i reflecting the influence of X_i on the prediction $y{=}f(x)$ (or any subset y' of y) as illustrated in Figure 6.1.

Example 44. *In the picture below, the black-box is the multi-label classifier. Input instances are vectors of n feature values. The vectors on the right-hand side show all the feature attributions computed using explanation functions denoted $h(x, y_i)$ for each label. The problem studied is how to find the global feature attributions for a multi-label prediction based on the ones generated for individual labels.*

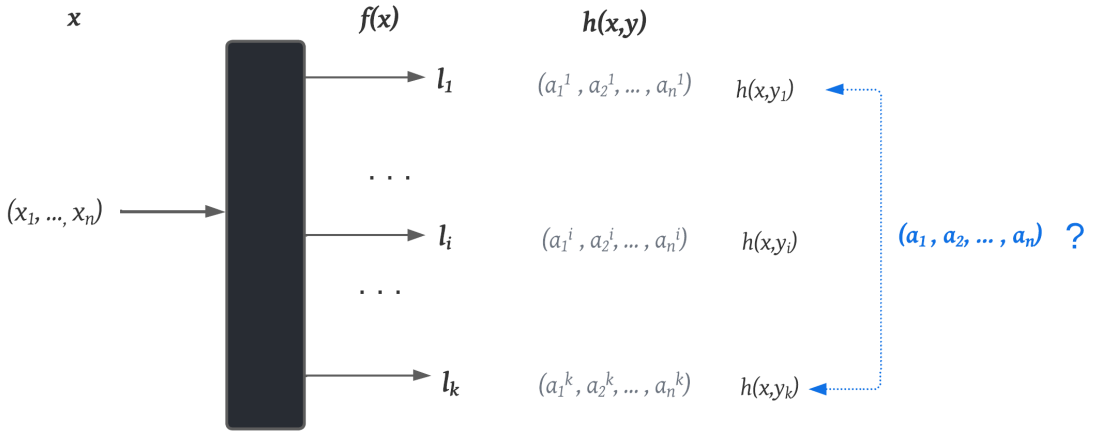


Figure 6.1: Illustrative figure of the problem studied

We propose schemes for achieving feature attribution for multi-label classification which are all model-agnostic and which are based on existing feature attribution methods used as oracles.

1. We first point out some deficiencies in attribution methods based on aggregation. To do this, we define three desirable properties such as sensitivity (features actually leading to different predictions should have positive scores), data stability (similar data items predicted equally should lead to similar explanations) and label-explanation correlation (correlated labels should be associated with correlated explanations). We show empirically that some oracles and aggregation operators capture these properties to different extents.

2. The second main contribution of this chapter is a framework based on problem transformation allowing to provide global feature attributions capturing the above properties while using existing attribution methods as an oracle.

3. The third contribution consists in a new attribution method based on symbolic explanations such as sufficient and counterfactual reasons, from which attribution scores are generated.

Examples of the existing feature attribution approaches proposed to explain the multi-label classification such as [SB21, Che21, PGMP19] were previously introduced in Section 4.1 of Chapter 4. In the following, we propose and evaluate schemes for feature attribution specifically designed to explain multi-label predictions in an agnostic way while taking advantage of existing feature attribution methods used as oracles.

6.2 Aggregation-based feature attribution

The aim of this section is to compare some aggregation-based attribution methods using existing attribution approaches such as SHAP and LIME considered as oracles employed to generate the set of explanations. In an aggregation-based scheme, the global attribution aggregates feature attributions relative to labels individually into a global attribution for the considered multi-label prediction. This may be useful to get an overall idea of the influence of each feature on the whole (or a subset) predicted labels. Let h denote the attribution oracle used to achieve feature attribution of labels (individually), and let

$$Att(x_i) = Agg_{y_j \in y'}(h(x_i, y_j)), \tag{6.1}$$

where Agg is an aggregation function[43] and y' is the part of the multi-label prediction to explain.

Given that the generated local feature attributions will provide insights into predicting each label y_j individually, then intuitively, one should choose aggregation operators allowing to capture complementary information (contrary to aggregation measures capturing confirmation or consensus as it is the case in ensemble classifiers). In our study, we compare aggregation operators for redundant information on the one side with operators suited for complementary information on the other side.

There are basically two questions when it comes to aggregation: *i) what is the meaning or semantics to give to aggregated scores (for example, is the aggregation of Shapley values for an attribute X_i a Shapley value ?)* and *ii) how to assess relevance of aggregated scores?* Let us first provide three natural properties that we want to capture in a multi-label setting, then, assess in the experimental study to what extent such properties are captured by some common aggregation operators.

6.2.1 Three basic properties for feature attribution in multi-label classification

Sensitivity

This property, also called separability, indicates that if there are two different predictions for two instances of data that differ only in the value of a single feature X_i, then the feature X_i must have a positive attribution. Also, if a feature is never used by a model to make a prediction, then the attribution score for that feature should always be zero. To illustrate and motivate the interest of such property, we provide the following example for the case where a change in a feature value alters the prediction and how we expect the importance of the feature to evolve.

Example 45. *In the figure below, the black-box is a multi-label classifier. Input instances have three features and have similar values except for the variable X_3. The model outputs for the instances x and x' two different predictions where the difference lies in the prediction of the label l_2. The vectors on the right-hand side correspond to the feature attributions of each instance. The importance of X_3 is expected to increase since it represents the only difference between two instances predicted differently.*

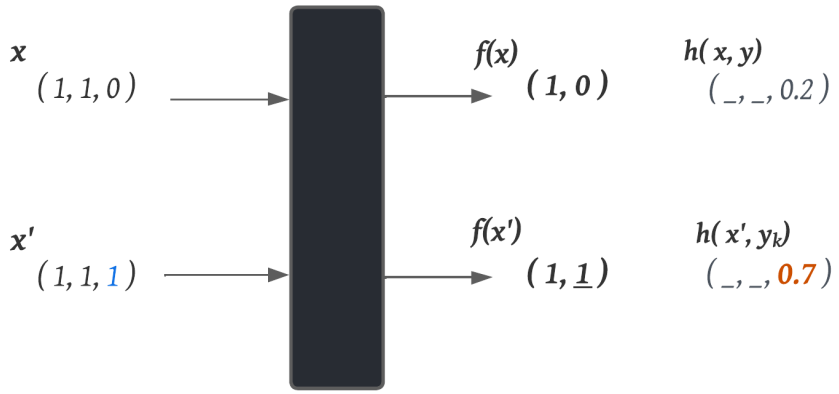


Figure 6.2: Example illustrating the sensitivity property evaluated on the explanations of a multi-label prediction.

In order to evaluate the sensitivity property, the two separability scenarios set out above should be considered empirically. We provide details about how we do it in Section 6.5.1.

[43] In practice, one can use depending on needs $average$, $median$, max, min, etc.

Data-explanation stability

Data stability, called simply stability in some papers [LPK21], intuitively states that similar data instances predicted equally should have similar explanations. Therefore, small changes in the input data (that do not change the predictions) should result in small changes in the explanations associated with the predictions made on those data. In other words, explanation functions should be insensitive to perturbations in the model inputs, especially if the model output does not change. This aims to capture a kind of continuity, i.e. for data points close to each other in the feature space, we expect their explanations to be close to each other if such data points share the same prediction.

Example 46. *Similarly to Example 45, the input instances have three features and have similar values except for the variable X_1. The model outputs the same prediction for the two instances x and x'. The vectors on the right-hand side correspond to the feature attributions of each instance. The importance of X_1 is expected to remain the same or undergo minimal changes (low impact) as shown below.*

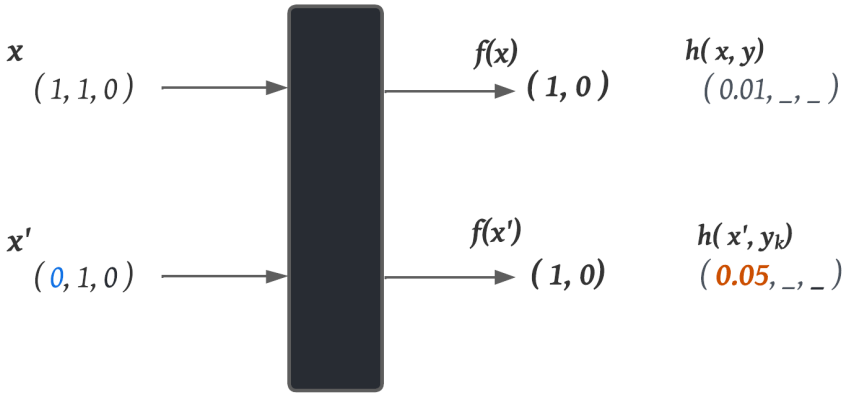


Figure 6.3: Example illustrating the data-explanation stability evaluated on the explanations of a multi-label prediction.

Stability evaluation can be achieved similar to the way we assess sensitivity. In the current work, we assess the stability property of a feature attribution function h as its ability to generate feature attributions close to each other for an instance x and a perturbed instance $\acute{x}$ which have the same prediction (namely $f(\acute{x})$=$f(x)$). We give more insights on how we assess it empirically in Section 6.5.1.

Label-explanation correlation

This property refers to the consistency of the explanations provided for pairs of labels, especially in case the labels are strongly correlated. Intuitively, if two labels (predicted by the model to be explained), are strongly correlated then their explanations should also be strongly correlated. In an extreme case, if two labels are equivalent (predicted with the same value regardless of the input data), then the explanations must also be strongly correlated. Similarly, if two labels are predicted independently of each other then their explanations should also be independent.

To illustrate this property, suppose a multi-label classification problem consisting in making hypotheses on possible diseases (representing the labels) from a certain number of observed symptoms (representing the features). If two diseases are often always predicted simultaneously, then their explanations (which can be assimilated to the associated symptoms) must be correlated, otherwise the two diseases will not be predicted simultaneously.

While sensitivity and stability properties also apply for multi-class classification, the third property is specific to multi-label classification and it is the first time it is proposed to our knowledge. The correlation between pairs of labels or between pairs of attribution vectors can directly be assessed by commonly used statistical correlation measures such as mutual information, Pearson coefficient, etc.

6.2.2 Aggregation operators

In our case, aggregation consists of the operation of merging information from oracles in order to summarize or obtain global information. More precisely, the input to our aggregation operation is a set of k attribution vectors v_j=($a^j{}_1, a^j{}_2, .., a^j{}_n$) such that v_j=$h(x, y_j)$ is the attribution vector obtained by the oracle given the instance x for the label y_j. The output is a vector of scores v_{agg}=($a_1, a_2, .., a_n$) where agg is a user defined aggregation operator. Note that the vectors v_j share the same scale and are provided by the same oracle h.

Example 47. *Assume we have a data instance of interest x=$(0, 1, 0, 1, 0)$ to be explained by a feature attribution function h and that its prediction by a model f is $(1, 0, 0)$ (i.e. $f(0, 1, 0, 1, 0)$=$(1, 0, 0)$). v_j=$h(x, y_j)$ is the attribution vector obtained by the oracle on the instance x for the label y_j. The generated feature attribution explanations for each class $l_{j|j=1,2,3}$ of the prediction $y = (1, 0, 0)$ are represented in Table 6.1.*

	class	Att_1	Att_2	Att_3	Att_4	Att_5
v_1	$l_1 = 1$	0.093	0.55	0.27	0.043	0.044
v_2	$l_2 = 0$	-0.37	0.12	-0.48	0.15	0.58
v_3	$l_3 = 0$	-0.078	0.46	-0.75	0.196	0.172

Table 6.1: Feature attribution explanations of x per label.

The vectors of feature attribution explanation for the prediction of x obtained using the aggregation operators average, maximum and minimum are presented as follows :

	Att_1	Att_2	Att_3	Att_4	Att_5
v_{mean}	-0.118	0.376	-0.32	0.129	0.265
v_{max}	0.093	0.55	0.27	0.196	0.58
v_{min}	-0.37	0.12	-0.75	0.043	0.044

Table 6.2: Feature attribution explanation of $f(x) = (1, 0, 0)$.

From an aggregation point of view, it is natural that aggregation has some natural properties like commutativity, fairness and insensitivity to vacuous information. From a feature attribution point of view, an aggregation should satisfy some intuitive properties such as unanimity (resp. majority) (e.g. if a feature X_i is considered influential (resp. not influential) for each (most of) label, then it should be considered influential (resp. non influential) by the feature attribution obtained after aggregation). Note that our objective is not to study and analyse aggregation operators for feature attribution purposes. Rather, we will focus on properties that are important from a feature attribution point of view and to what extent aggregation preserves, improves or degrades these properties. It is worth noticing that aggregation-based feature attribution needs k calls to a feature attribution oracle. The following provides a solution based on reusing and inferring feature attributions relative to some labels based on the feature attributions of other correlated labels.

Feature attribution based on aggregation is intuitive and simple to implement but suffers from some drawbacks, in particular, the number of calls to the oracle. Indeed, if the multi-label classification problem includes k labels, it will be necessary to make k calls to the oracle h, which can be very computationally expensive, especially if we use an oracle that is not computationally efficient like SHAP in its standard version[44]. One way to improve this is to exploit the following fact: if the oracle used captures the property of *label-explanation correlation* for two labels y_i and y_j then this correlation can potentially make it possible to infer a feature attribution $h(x, y_i)$ from feature attribution $h(x, y_j)$. This is all the more true as this correlation is very strong. The idea is to predict $h(x, y_j)$ from $h(x, y_i)$ by exploiting correlations (especially linear ones) that may exist between predictions of y_i and y_j.

The following section provides an alternative scheme to aggregation-based feature attributions that allows to only make one call to a feature attribution oracle.

6.3 Multi-label feature attribution through problem transformation

In this section, we provide a setting allowing to rely on existing feature attribution methods considered as oracles to ensure feature attribution without aggregating attributions relative to labels individually. Recall that in multi-label classification tasks, training examples are couples of vectors (x,y) where $x \in X$ is the data input, and $y \in \{0,1\}^k$ is the corresponding output label vector. In order to explain the prediction $f(x)$=y of a multi-label classifier f, we associate a binary classifier f' with f as follows:

$$f'(x) = \begin{cases} 1 & \text{if f(x)=y} \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

Then, in order to achieve feature attribution for a data instance x, we can simply rely on $h(x, f')$. Namely, we achieve feature attribution using the oracle h on a binary classifier f'. For instance, if h performs feature attribution for $f(x)$ using a neighborhood around x then in the transformation-based scheme, we will use h to attribute features for f' around x.

Example 48. *(Example 47 continued) In this example, we transform a multi-label problem into a single-label one according to Equation 6.2. Assume that the neighborhood of an instance x=$(0,1,0,1,0)$ to be explained by a feature attribution function h is given in Table 6.3 and assume that the prediction is $f(0,1,0,1,0)$=$(1,0,0)$.*

X_1	X_2	X_3	X_4	X_5	$f(x)$			$f'(x)$
0	1	0	1	0	1	0	0	1
1	1	0	0	0	0	1	0	0
1	0	1	1	0	0	1	1	0
1	1	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1

Table 6.3: From multi-label predictions into mono-label ones.

Explaining $f(x) = (1,0,0)$ amounts to explaining $f'(x) = 1$ by the means of $h(x, f')$ as shown in Table 6.4.

[44] Computationally efficient versions are KernelSHAP or TreeSHAP.

	Att_1	Att_2	Att_3	Att_4	Att_5
$v_{f'}$	-0.13	0.76	0.45	-0.29	0.21

Table 6.4: Feature attribution explanation generated for the transformed problem $f'(x)$.

Hence, instead of explaining $f(x)$ with h, our transformation-based scheme explains $f'(x)$ with h. This transformation approach has several advantages:

- It can be used with any feature attribution oracle h such as SHAP and LIME.
- The semantics of scores is inherited from the one of the used oracle. For instance, using this scheme, it is clear that the numerical score associated to a feature while making a prediction is a Shapley value in case the used oracle is SHAP. This is not necessarily the case when aggregating feature attributions.
- It requires only one single call to an oracle h for the feature attribution while aggregation-based scheme requires k calls to an oracle.

So far, we have proposed a first scheme based on aggregation and a second one based on problem transformation. In the next section, we propose to achieve feature attribution for multi-label classification by first transforming the problem as in the second scheme and then use an oracle providing symbolic explanations.

6.4 Multi-label feature attribution through symbolic explanations

In addition to popular attribution methods like SHAP and LIME, we propose to use an alternative oracle that provides different types of symbolic explanations. More precisely, in this scheme, we first transform the problem, then generate symbolic explanations (in the form of sufficient reasons and counterfactuals) and finally derive feature attributions from the computed symbolic explanations. Given that problem transformation is presented in the previous section, let us directly present the remaining steps.

6.4.1 Generating symbolic explanations

In order to provide symbolic explanations, we rely on our generic and agnostic approach `ASTERYX` that allows to generate two main forms of symbolic explanations that are *sufficient reasons* and *counterfactuals* along with score-based explanations (cf Chapter 3 and Chapter 5). To recall, the approach is based on encoding a classifier into an equivalent symbolic representation and using a surrogate approach, `ASTERYX` relies on SAT-based solvers such as [GIL18] and [IMM18] to enumerate the two types of symbolic explanations.

6.4.2 From symbolic explanations to feature attributions

Once the symbolic explanations are enumerated, they are used to calculate scores that will be associated with the variables forming these explanations. These scores represent the feature attributions that are derived from the initial set of symbolic explanations. We recall that the scores are calculated with respect to certain properties (previously defined in Chapter 5) that we briefly recall in the following :

- **Feature involvement** ($\mathcal{FI}$) : This property is intended to reflect the extent of involvement of a feature within the set of explanations.

- **Feature generality** ($\mathcal{FG}$) : This property captures at what extent a feature is frequently involved in explaining instances in the vicinity of the sample to explain.

- **Feature responsibility** ($\mathcal{FR}$) : This property is intended to reflect the responsibility or contribution of a feature X_i within the set of symbolic explanations of x.

6.5 Experimental study

This section presents the empirical study carried out to evaluate the three schemes provided for achieving feature attribution for multi-label tasks.

Dataset	#instances	#classes	#features	data type	density
Augmented MNIST	70000	13	784	Images	0.184
Yelp Review Analysis	10806	5	671	Textual	0.328
IMDB Movie Genre Prediction	65500	24	30	Textual	0.07
Foodtruck	408	12	102	Categorical /Numerical	0.191
Patient Characteristics Survey (NYS15)	105099	5	63	Textual / Numerical	0.41

Table 6.5: Properties of the datasets used.

We considered a selection of multi-label datasets known from the literature and publicly available and can be found on Kaggle[45] or UCI[46]. The details of the datasets, such as the number of examples, the type of attributes, the number of classes and their label density are given in Table 6.5. The initial version of the MNIST dataset is composed of 10 classes corresponding to digits from 0 to 9. We extended that version by adding the labels `"Odd"`,`"Even"` and `"Prime"` and called it "Augmented MNIST". Each input image is associated to a vector of thirteen labels[47]. Experiments presented in this section have been carried out on a cluster of computers equiped with quadcore bi-processors Intel Xeon E5-2643 3 (3.30 GHz) and 64 GB of memory running under the CentOS Stream 8.3 and on Intel Core i7-7700 (3.60GHz ×8) processors with 32Gb memory on Linux OS.

Two popular multi-class feature attribution methods have been used within the experiments as oracles: the widely-used SHAP [LL17] and LIME [RSG16]. As for symbolic explainer, we used our approach `ASTERYX` (cf Chapter 3 and 5). We used the Binary relevance (BR) multi-label classifiers with Logistic Regression or Decision Tree as base classifiers from the Scikit-multilearn library. Such BR models are trained on the different datasets and their predictions are being explained. We observed during the experiments that as the label-density decreases, the prediction task becomes harder. Given the neighborhood of a sample test x, very few samples are predicted as the non-majority class, which causes learning problems where the trained model recognizes systematically the majority class. To address this problem, we operate an under sampling on the neighborhood V_x where we randomly pick samples from majority class equal to the number of samples in the minority class so that both the classes will have approximately the same number of samples. All the multi-class feature attribution oracles used in our

[45]Kaggle dataset : (`https://www.kaggle.com/`).

[46]UCI dataset : (`http://archive.ics.uci.edu/ml/`).

[47]The labels having an index $i \in [0, 9]$ indicate whether the input image x is recognized as an `i-digit` while the labels having an index $i \in [10, 12]$ indicate respectively whether the represented digit is being classified as an "odd", "even" or a "prime" numbers.

experiments are local methods relying on the neighborhood of the instance to explain to generate explanations.

In order to provide an empirical evaluation comparing the three schemes proposed in this paper, let us first present the methodology and means used for each scheme, then give the empirical results of the three schemes.

6.5.1 Evaluating aggregation-based feature attribution scheme

In the following, we assess the gain or loss in terms of sensitivity and stability due to the aggregation operation.

Sensitivity In our work, the sensitivity property of a feature attribution function h is assessed quantitatively as follows :
Let f be a classifier, h be a feature attribution function, x be a data point and assume a set of perturbed instances $\rho(x)$=$\{\acute{x} : \text{dist}(x, \acute{x}) \leqslant r$ and $f(\acute{x}) \neq f(x)\}$, we assess the sensitivity property at x as follows:

$$\eta_{SNS}(h, x) = \underset{\acute{x} \in \rho(x)}{AVG}(dist(h(x, f), h(\acute{x}, f))) \tag{6.3}$$

where $dist$ denotes a distance (or dissimilarity) function between x and a perturbed instance $\acute{x}$. One can sample the data and have a representative assessment of the sensitivity property. According to Equation 6.3, the larger the output, the better is h in terms of sensitivity.

Since our objective is to assess the gain or loss in terms of sensitivity due to the aggregation operation, we compare on average the sensitivity of individual label feature attributions (before aggregation) w.r.t the sensitivity of resulting global feature attribution (after aggregation). The results are presented in Table 6.6. Several findings are possible from Table 6.6. In particular, we notice on the different datasets that most of the time, we obtain the best (here highest) values (highlighted in red) of sensitivity at the level of individual label feature attributions (before aggregation), except for the MNIST dataset.

Data-explanation stability For any feature attribution function h, we can measure its stability following Definition 6.4. Given a classifier f, a feature attribution method h, a data point x and a set of perturbed instances $\rho(x)$ defined as $\rho(x)$ =$\{$ $\acute{x}$: dist$(x, \acute{x}) \leqslant r$ and $f(\acute{x})$=$f(x)$ $\}$, we assess the stability

	Oracle	Before AVG $(\eta_{SNS}(f_i, x))$	After AVG $(\eta_{SNS}(f_i, x))$	Before MAX $(\eta_{SNS}(f_i, x))$	After MAX $(\eta_{SNS}(f_i, x))$	Before MIN $(\eta_{SNS}(f_i, x))$	After MIN $(\eta_{SNS}(f_i, x))$
YELP	SHAP	3.143	3.0	3.44	3.54	2.69	3.16
	LIME	2.2	0.96	2.31	0.97	2.06	0.92
MNIST	SHAP	0.37	1.19	0.38	1.39	0.361	1.25
	LIME	0.294	0.44	0.3	0.46	0.291	0.40
FOOD TRUCK	SHAP	2.09	1.1	2.14	1.48	2.03	1.24
	LIME	0.92	0.37	0.94	0.38	0.9	0.34
NYS15	SHAP	1.29	1.1	1.54	1.483	1.34	1.15
	LIME	0.76	0.36	0.85	0.37	0.66	0.33
IMDB	SHAP	0.43	1.0	0.44	1.17	0.42	1.03
	LIME	0.228	0.373	0.232	0.38	0.224	0.35

Table 6.6: Evaluation of aggregation-based feature attribution wrt the sensitivity property.

	Oracle	Before AVG ($\eta_{STB}(f_i, x)$)	After AVG ($\eta_{STB}(f_i, x)$)	Before MAX ($\eta_{STB}(f_i, x)$)	After MAX ($\eta_{STB}(f_i, x)$)	Before MIN ($\eta_{STB}(f_i, x)$)	After MIN ($\eta_{STB}(f_i, x)$)
YELP	SHAP	0.92	1.75	1.4	3.05	0.7	2.65
	LIME	2.11	0.68	2.29	0.7	1.79	0.54
MNIST	SHAP	0.39	1.73	0.41	1.979	0.36	1.19
	LIME	0.24	0.485	0.25	0.498	0.22	0.401
FOOD	SHAP	0.75	1.36	0.82	1.48	0.66	1.08
TRUCK	LIME	0.91	0.389	0.94	0.396	0.88	0.347
NYS15	SHAP	0.82	1.26	0.83	1.37	0.80	1.03
	LIME	0.22	0.33	0.24	0.345	0.2	0.3
IMDB	SHAP	0.48	1.52	0.49	1.73	0.4	1.07
	LIME	0.16	0.41	0.169	0.422	0.15	0.342

Table 6.7: Evaluation of aggregation-based feature attribution wrt the data-explanation stability property.

property between $h(x, f)$ and attributions of close instances $\in \rho(x)$ as follows :

$$\eta_{STB}(h, x) = \underset{\acute{x} \in \rho(x)}{AVG}(dist(h(x, f), h(\acute{x}, f))), \tag{6.4}$$

where $dist$ denotes a distance (or dissimilarity) function between x and a perturbed instance $\acute{x}$. Equation 6.4 quantitatively assesses the stability property at a data instance x. One can sample the data and have a representative assessment. According to our way of evaluating stability, the lower the result, the better the feature attribution in terms of stability.

For the data-explanation stability property, we compare on average the data-explanation stability of individual label feature attributions (before aggregation) w.r.t the data-explanation stability of resulting global feature attribution (after aggregation). The results are presented in Table 6.7.

We obtain the best (here smallest) values (highlighted in green) at the level of the individual label feature attributions, meaning that the property of stability is better preserved before aggregation and merging the individual label feature attributions degrades it. Regarding the best aggregation operator, it can be seen that $\max$ aggregation selects the highest values and thus benefits the sensitivity to the detriment of stability. Conversely, for stability, $\min$ aggregation is more suitable because it ensures high stability but this is to the detriment of sensitivity. We also observe that the SHAP method seems better than LIME with a higher average value for sensitivity and a lower average value for stability (underlined values per dataset).

In addition to sensitivity and stability, we assess label-explanation correlation in order to choose a feature attribution oracle. In this experiment, we assessed this property by the mean of the Pearson correlation coefficient and Mutual Information (MI). Note that the main difference is that the Pearson's correlation coefficient aims to capture linear relationships between variables while mutual information measures general dependence. Given the base classifiers of a multi-label classifier, we measure the correlation of predictions for each pair of labels and their respective SHAP and LIME explanations.

The results on the different datasets are presented in the Figures 6.4 and 6.5. where the X axis groups the pairs of labels sharing the same *MI* or Pearson's R interval. For example, the last bin will represent all the pairs of labels y_i and y_j having *MI*(y_i, y_j) or Pearson's R belonging to the interval $]0.9, 1]$. On the Y axis, the mean of *MI* or Pearson's R of the scores corresponding to the instances predicted with the pair (y_i, y_j) is represented. We can see that both SHAP and LIME manage to capture strong label-explanation correlation when the labels are strongly correlated. However, the results diverge when it comes to weakly correlated labels where we observe that the mean values of the MI and Pearson's R metrics are still high for weakly correlated labels, indicating strong correlation between the explanations. Indeed, for the IMDB and the augmented MNIST, we notice that LIME, closely followed by SHAP, tends to capture

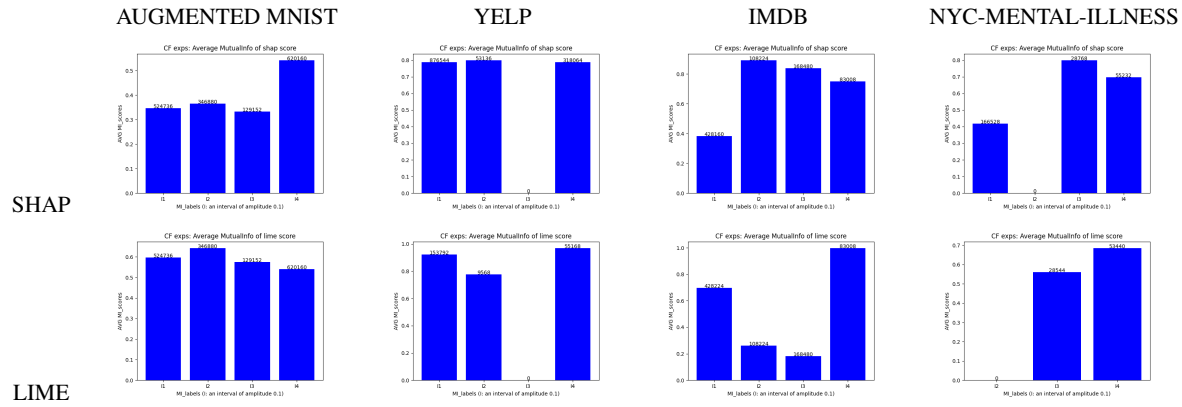


Figure 6.4: Evaluating label-explanation correlation using the mutual information (MI) coefficient.

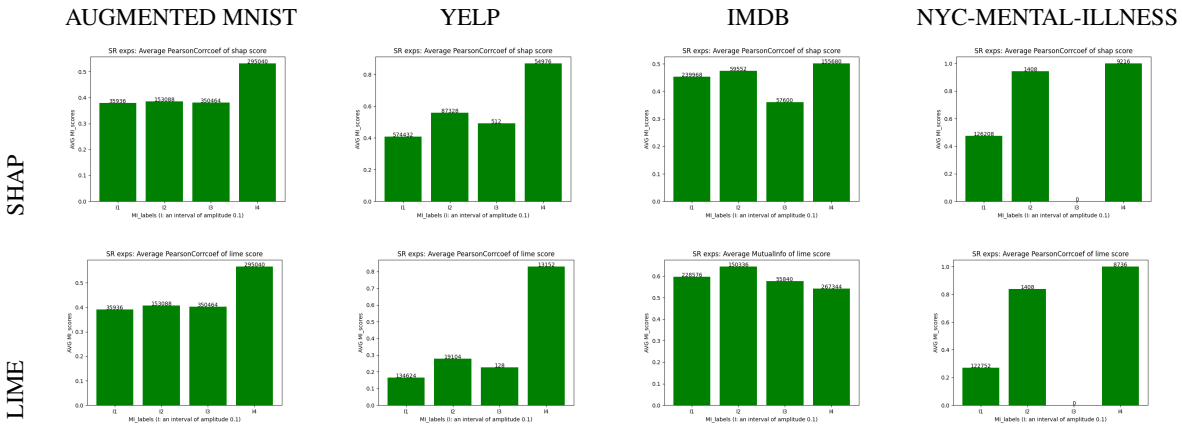


Figure 6.5: Evaluating label-explanation correlation using the Pearson's R coefficient.

strong correlation regardless of the relationship that exists between the labels. This may be due to the assessment in our experiments of this property on restricted neighborhoods (therefore with very similar instances, possibly similar attributions due to the stability property).

6.5.2 Evaluating problem transformation-based feature attribution scheme

We keep up with the same multi-label classifiers and the same oracles (SHAP and LIME) to evaluate multi-label feature attribution through problem transformation. Table 6.8 gives the results of the evaluation wrt the sensitivity and stability properties. The main finding is that the problem transformation approach ensures the smallest values for the stability property over the different datasets. Such results are expected since the approach by definition reduces the task of explaining a multi-label classifiers to explaining a single classifier, which makes it possible to minimize the discrepancies that there could be between the explanations of similar instances.

6.5.3 Evaluating symbolic explanation-based feature attribution scheme

The objective here is to evaluate our three basic properties using a symbolic explanation-based oracle. The same multi-label models used for the evaluation of aggregation-based attribution scheme are used in this part.

Property	Dataset	SHAP	LIME
Sensitivity	YELP	0.28	0.75
	AUG. MNIST	0.18	0.15
	FOODTRUCK	0.29	0.5
	NYS15	0.25	0.75
Data-explanation stability	YELP	0.05	0.4
	AUG. MNIST	0.38	0.09
	FOODTRUCK	0.6	0.33
	NYS15	0.05	0.4

Table 6.8: Evaluation of multi-label feature attribution through problem transformation.

The feature attribution oracles used are : Feature Involvement (*FI*), Feature Responsibility (*FR*) and Feature Generality (*FG*) from ASTERYX.

		Before AVG $(\eta_{SNS}(f_i,x))$	After AVG $(\eta_{SNS}(f_i,x))$	Before MAX $(\eta_{SNS}(f_i,x))$	After MAX $(\eta_{SNS}(f_i,x))$	Before MIN $(\eta_{SNS}(f_i,x))$	After MIN $(\eta_{SNS}(f_i,x))$
YELP	FR_{CF}	2.51	2.43	2.86	3.15	2.18	2.02
	FG_{CF}	1.61	1.41	1.82	1.87	1.41	1.11
	FI_{CF}	1.26	1.05	1.41	1.46	1.26	0.78
MNIST	FR_{CF}	1.77	2.49	1.8	2.83	1.75	2.29
	FG_{CF}	1.7	1.65	1.74	1.9	1.66	1.49
	FI_{CF}	1.53	1.23	1.56	1.46	1.5	1.09
FOOD TRUCK	FR_{CF}	2.05	1.71	2.13	2.37	1.98	1.46
	FG_{CF}	1.77	1.41	1.82	2.02	1.71	1.14
	FI_{CF}	1.48	1.15	1.51	1.75	1.44	0.9
NYS15	FR_{CF}	2.22	1.71	2.45	2.33	1.97	1.48
	FG_{CF}	1.8	1.44	1.88	2.0	1.71	1.18
	FI_{CF}	1.39	1.17	1.45	1.71	1.32	0.93
IMDB	FR_{CF}	1.29	2.2	1.33	2.49	1.24	2.05
	FG_{CF}	1.51	1.62	1.55	1.85	1.47	1.5
	FI_{CF}	1.33	1.28	1.34	1.5	1.32	1.15

Table 6.9: Evaluation of multi-label feature attribution through aggregation by assessing sensitivity for counterfactuals (CF).

In Tables 6.9 and 6.10, we notice that we mainly obtain a higher sensitivity at the level of the global feature attribution obtained after the aggregation for the sufficient reasons, while the opposite is observed for the explanations counterfactuals, where we find a greater sensitivity before aggregation. In Table 6.10, we notice a strong stability at the level of the explanations before the aggregation for the SRs. For CFs, the results are more mixed with a slight advantage for aggregation (cf Table 6.9).
In terms of the oracle, none of the three oracles stand out in a striking way, with a slight advantage for the *FR* for sensitivity in CF explanations and for the *FI* for SR explanations which seem to achieve better scores for the Sensitivity property (underlined in Tables 6.9 and 6.10) i.e. tend to be more sensitive to

		Before AVG $(\eta_{SNS}(f_i, x))$	After AVG $(\eta_{SNS}(f_i, x))$	Before MAX $(\eta_{SNS}(f_i, x))$	After MAX $(\eta_{SNS}(f_i, x))$	Before MIN $(\eta_{SNS}(f_i, x))$	After MIN $(\eta_{SNS}(f_i, x))$
YELP	FR_{SR}	3.27	3.34	3.51	4.08	3.04	2.82
	FG_{SR}	3.6	3.56	3.8	4.47	3.41	2.96
	FI_{SR}	3.81	3.7	3.99	4.62	3.63	3.1
MNIST	FR_{SR}	1.4	2.23	1.43	2.65	1.38	1.94
	FG_{SR}	2.69	2.97	2.78	3.55	2.5	2.6
	FI_{SR}	2.56	3.09	2.63	3.67	2.5	2.71
FOOD TRUCK	FR_{SR}	2.627	1.8	2.7	2.66	2.56	1.4
	FG_{SR}	3.08	2.32	3.12	3.31	3.03	1.85
	FI_{SR}	3.09	2.31	3.15	3.24	3.05	1.87
NYS15	FR_{SR}	2.51	1.88	2.75	2.7	2.27	1.49
	FG_{SR}	2.73	2.32	2.36	3.29	2.59	1.91
	FI_{SR}	2.64	2.34	2.82	3.22	2.46	1.92
IMDB	FR_{SR}	1.97	2.21	2.01	2.58	1.94	1.97
	FG_{SR}	2.07	2.72	2.09	3.2	2.04	2.47
	FI_{SR}	1.92	2.77	1.95	3.23	1.89	2.52

Table 6.10: Evaluation of multi-label feature attribution through aggregation by assessing sensitivity for sufficient reasons explanations (SR).

perturbations of data causing a change in prediction. With regard to Data-explanation stability, the *FR* stands out with the lowest scores (underlined in Tables 6.12 and 6.11) reflecting a similarity in the feature attributions attributed to instances that have undergone little change that does not impact their prediction. For the choice of aggregation functions, the same analysis made at the level of aggregation-based feature attribution applies to symbolic-based feature attribution oracles (max aggregation for sensitivity and min aggregation for Data-explanation stability)

		Before AVG $(\eta_{STB}(f_i,x))$	After AVG $(\eta_{STB}(f_i,x))$	Before MAX $(\eta_{STB}(f_i,x))$	After MAX $(\eta_{STB}(f_i,x))$	Before MIN $(\eta_{STB}(f_i,x))$	After MIN $(\eta_{STB}(f_i,x))$
YELP	FR_{SR}	2.58	1.95	2.99	3.93	2.25	1.03
	FG_{SR}	3.66	2.67	4.008	5.45	3.34	1.39
	FI_{SR}	3.75	2.76	4.08	5.61	3.44	1.45
MNIST	FR_{SR}	1.02	1.95	1.11	3.93	0.94	1.s03
	FG_{SR}	2.26	2.67	2.43	5.45	2.105	1.39
	FI_{SR}	2.01	2.76	2.17	5.61	1.85	1.45
FOOD TRUCK	FR_{SR}	2.11	1.85	2.17	2.47	2.04	1.55
	FG_{SR}	3.31	2.59	2.954	3.48	2.84	2.19
	FI_{SR}	3.34	2.62	2.96	3.53	2.79	2.2
NYS15	FR_{SR}	0.46	1.68	0.48	2.2	0.4	1.43
	FG_{SR}	1.27	2.44	1.64	3.17	1.1	2.1
	FI_{SR}	1.08	2.42	1.42	3.17	0.97	2.07
IMDB	FR_{SR}	2.72	1.82	2.76	2.84	2.1	1.34
	FG_{SR}	2.70	2.53	2.91	3.97	2.4	1.84
	FI_{SR}	2.81	2.52	2.92	4.01	2.1	1.83

Table 6.11: Evaluation of multi-label feature attribution through aggregation by assessing data-explanation stability for sufficient reasons explanations (SR).

		Before AVG $(\eta_{STB}(f_i,x))$	After AVG $(\eta_{STB}(f_i,x))$	Before MAX $(\eta_{STB}(f_i,x))$	After MAX $(\eta_{STB}(f_i,x))$	Before MIN $(\eta_{STB}(f_i,x))$	After MIN $(\eta_{STB}(f_i,x))$
YELP	FR_{CF}	0.8	1.39	1.19	2.88	0.62	0.82
	FG_{CF}	0.48	0.96	0.72	2.4	0.37	0.43
	FI_{CF}	1.23	1.03	1.47	2.75	1.005	0.39
MNIST	FR_{CF}	0.41	1.39	0.51	2.88	0.32	0.82
	FG_{CF}	0.49	0.96	0.61	2.4	0.35	0.43
	FI_{CF}	1.03	1.03	2.75	2.12	0.396	0.619
FOOD TRUCK	FR_{CF}	0.84	1.04	0.9	1.5	0.78	0.86
	FG_{CF}	0.69	0.89	0.79	1.34	0.62	0.72
	FI_{CF}	1.31	1.14	1.37	1.69	1.24	0.92
NYS15	FR_{CF}	0.38	0.94	0.89	1.325	0.4	0.79
	FG_{CF}	0.69	0.88	0.78	1.26	0.62	0.74
	FI_{CF}	0.4	1.14	1.03	1.26	0.2	0.74
IMDB	FR_{CF}	0.27	1.12	1.34	1.88	0.2	0.83
	FG_{CF}	0.6	0.89	0.886	1.63	0.3	0.6
	FI_{CF}	1.27	1.1	1.37	2.005	1.24	0.75

Table 6.12: Evaluation of multi-label feature attribution through aggregation by assessing data-explanation stability for counterfactuals (CF).

		YELP	AUG. MNIST	FOODTRUCK	NYC15
Sensitivity	FR_{CF}	2.46	2.75	3.6	2.46
	FR_{SR}	2.6	1.64	2.11	2.6
	FG_{CF}	2.26	2.18	2.69	2.25
	FG_{SR}	3.8	2.47	3.53	3.81
	FI_{CF}	2.13	1.83	2.25	2.13
	FI_{SR}	3.82	2.43	3.54	3.82
Data-explanation stability	FR_{CF}	0.83	1.11	1.67	0.83
	FR_{SR}	1.63	1.64	1.9	1.4
	FG_{CF}	1.4	1.42	3.42	1.39
	FG_{SR}	2.39	2.47	3.01	3.26
	FI_{CF}	1.81	0.97	2.32	1.81
	FI_{SR}	2.4	2.44	2.58	3.3

Table 6.13: Evaluating symbolic explanation-based feature attribution scheme through problem transformation with respect to the Sensitivity and Data-explanation stability properties.

Table 6.13 presents the evaluation results of the symbolic-based feature attribution through problem transformation scheme. SRs explanations seem to be more sensitive to change with higher values compared to those of CFs (underlined in table 6.13). On the other hand, CF explanations seem more stable with smaller values than those of SRs (underlined in Table 6.13). In terms of which oracle to choose, the *FR* seems to provide the most stable explanations and along with *FI*, providing the most sensitive explanations too.

The last property we check on the symbolic-based feature attribution explanations is the Label-explanation stability. The results are presented in Figures 6.6 and 6.7. The green bar-charts represent the results of SRs explanations while the blue ones represent the results of the CFs. As shown in the figures, all of the oracles manages to keep a strong stability between the explanations of related labels. We notice on the graph of certain datasets (in particular MNIST, Yelp and Nyc-mental-illness) that weakly correlated labels are associated with weakly correlated explanations, which reflects a low stability between explanations of labels that are not related. We also note that label-explanation stability is better preserved at the level of the explanation SRs compared to the CFs.

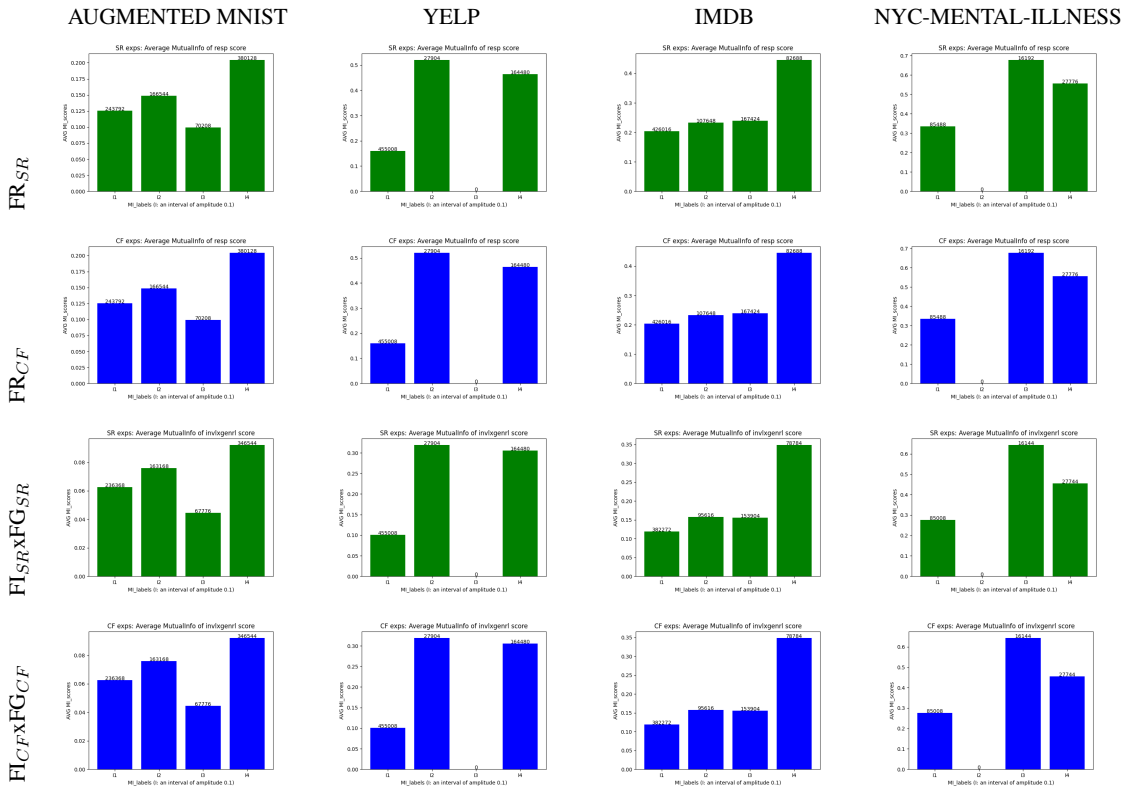


Figure 6.6: Evaluating label-explanation stability using the mutual information (MI) coefficient.

6.5.4 Comparative study

Now that we have seen the results for the different schemes proposed to define feature attribution explanations for a multi-label predictor, we compare them to see if we can identify the best scheme or even the best oracle feature attribution to use. We summarize the main points as follows:

- **Speed (runtime) :** it is obvious that the problem transformation scheme is faster than the aggregation ones. In the first scheme we call the feature attribution oracle only once while we will have to do k calls in the second scheme.

- **Representation of the explanation :** the final representation is the same for all schemes. The multi label explanation will be presented as a vector of size n representing the influence of each feature on the whole (or a subset) predicted labels

- **Nature of explanations :** the problem transformation scheme retains the nature of the explanations generated, by transforming the multi-label problem into a binary classification problem, we can directly use the multiple interpretability methods defined for single-label problems and thus, preserve the nature of the explanations and their properties. This is not the case with aggregation.

- **Feature attribution oracle :** there is no clear explanation function that stands out more than the others to best measure the properties according to the different approaches. We notice that SHAP often makes it possible to obtain the smallest values for the criterion of Data-explanation stability while the symbolic-based feature attribution methods seem to get out of it better if we consider the property of sensitivity. Regarding the Label-explanation stability property, symbolic-based oracles seem to stand out compared to SHAP and LIME.

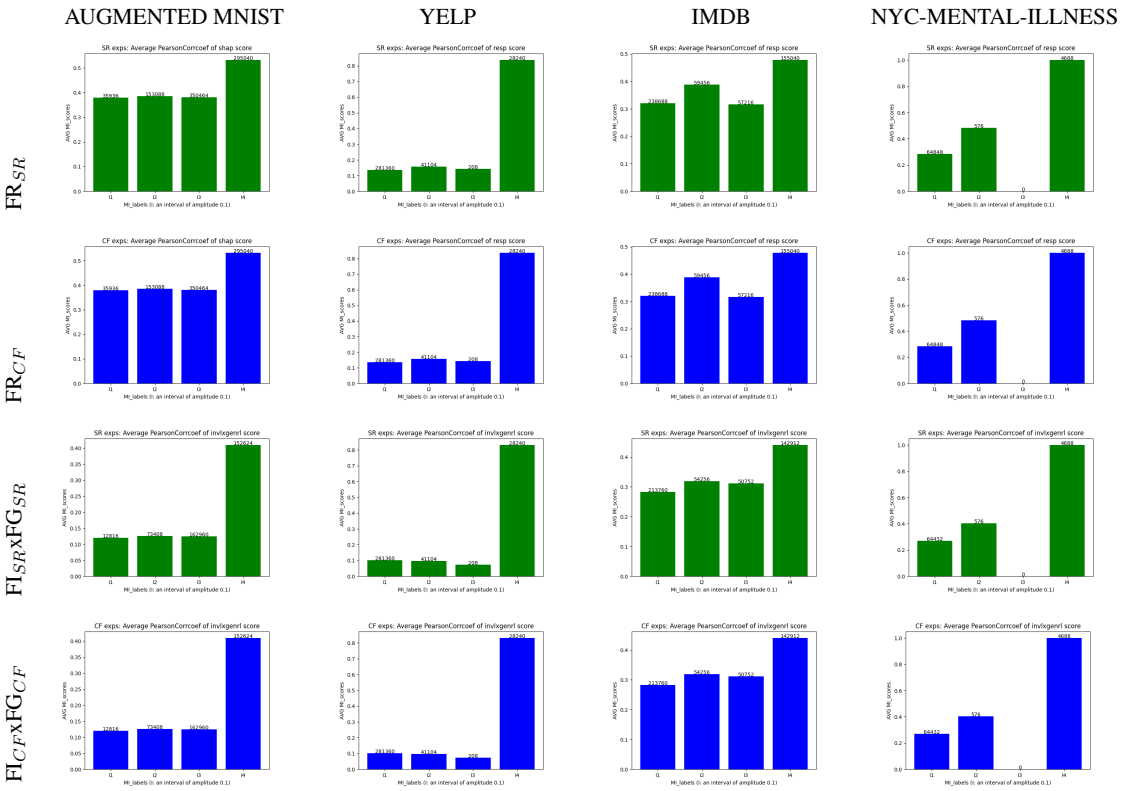


Figure 6.7: Evaluating label-explanation stability using the Pearson's R coefficient.

6.5.5 Evaluating feature attribution inference

The goal here is to infer explanations given related classes of a multi-label predictor. To do so, we compare the difference between scores that we will call "deduced" and the real explanations calculated for the label in question. This evaluation is done on different multi-label predictors having MLP Classifier, Logistic Regression and Random Forest Classifier as base classifiers. The feature attribution oracles used are SHAP, LIME and ASTERYX.

Different metrics such as euclidean distance and mean squared error (MSE) were used to compute the difference between the real feature attribution explanations and the deduced ones and are represented on the Y axis. They give a relatively high weight to large difference between real and deduced scores, which means the smaller the weight, the closer the fit is to the real feature attribution scores.
The X axis represents the *MI* (resp *Pearsons's R*) coefficient between pairs of labels of the same data instances. For example, pairs of labels y_i and y_j having *MI*(y_i, y_j)=1 (resp *Pearsons's R*(y_i, y_j)=1) are strongly correlated around $V(x, r)$. On the Y axis, the mean of distances between real feature attribution scores and the *MI*-deduced (resp *Pearson's R*-deduced) scores corresponding to the instances predicted with the pair (y_i, y_j) is represented. To sum up, we have on the X axis of Figures 6.8 and 6.9, the *MI*(y_i, y_j) of different pairs of labels. On the Y axis, the average distance between the real and alpha-score deduced from MI(scores) where alpha is the coefficient between the related pair of labels.

The curves in Figures 6.8 and 6.9 show that all the explanation methods used tend to generate explanations very similar to those inferred from a relationship between two classes. These latter are obtained by multiplying the real scores with the coefficient between the related pair of labels. Note that the greater the correlation coefficient between the labels, the smaller the difference between the explanation vectors and those for the two measurements used (MSE and Euclidean distance).

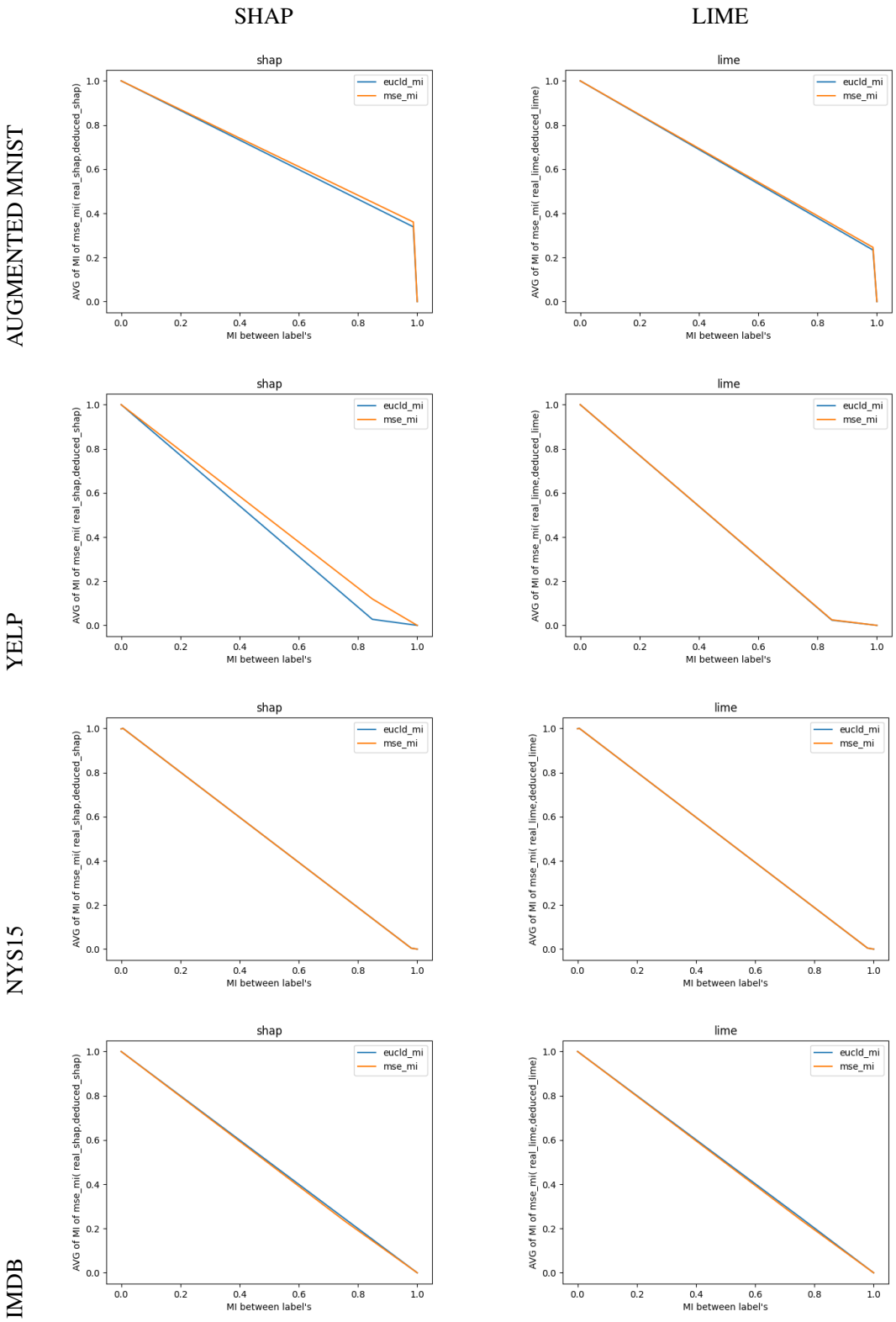


Figure 6.8: Average difference between real vs deduced feature attribution scores (SHAP/LIME) given the MI between labels.

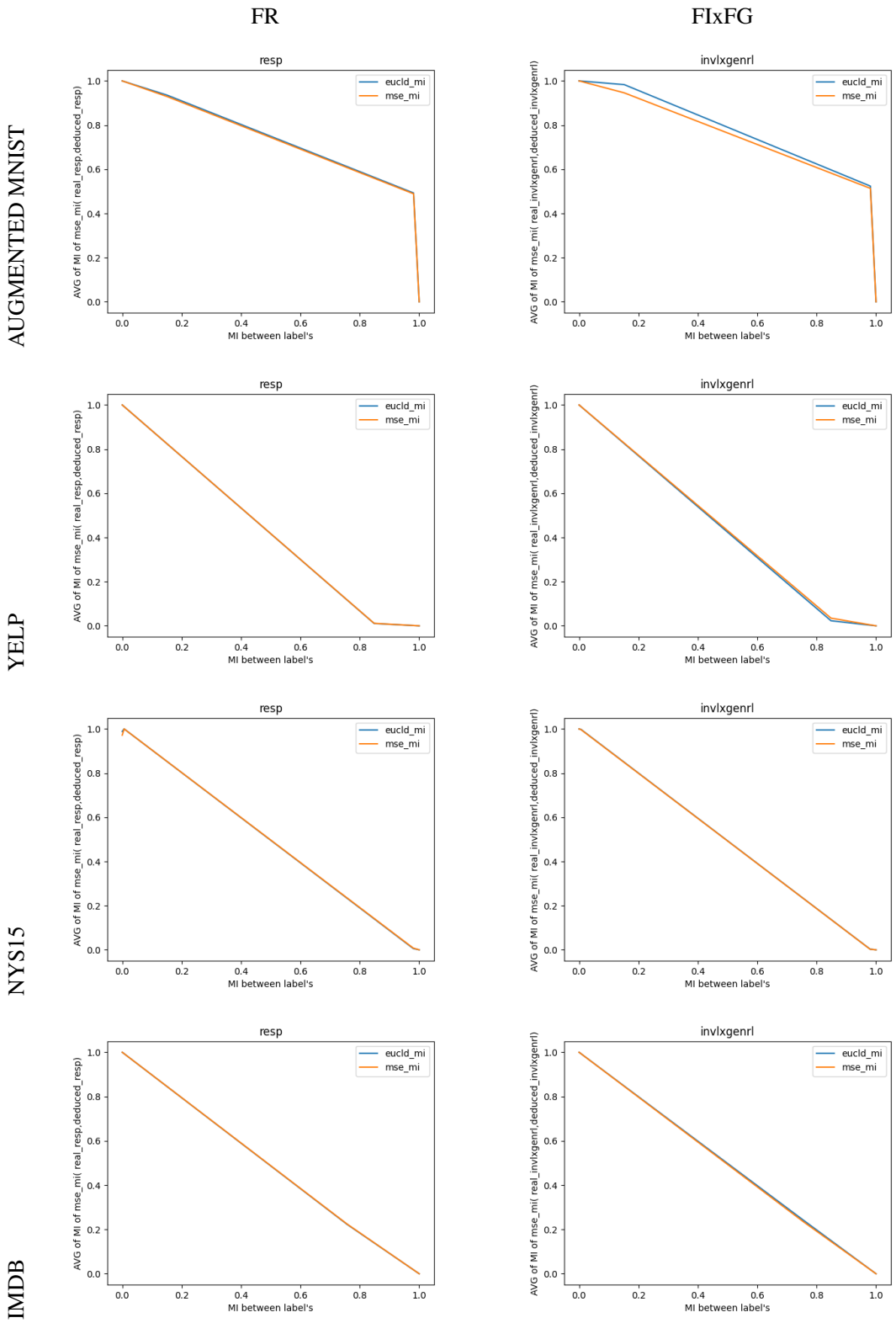


Figure 6.9: Average difference between real vs deduced feature attribution scores(FR/FIxFG) given the MI between labels.

This track is interesting because in addition to the power of optimization that there may be in terms of the number of calls to a multi-class oracle and the number of explanations to be processed, the inference of explanations can be used to "learn to explain" by training a model on the generated explanations.

6.6 Conclusion

The literature reports many approaches for explaining binary and multi-class classifiers but only a few are dedicated to the multi-label setting. In this chapter, we addressed feature attribution for multi-label classification problems. The main objective is to take advantage of existing feature attribution methods for multi-class classification and provide schemes to use them as oracles to provide features attributions in a multi-label classification setting. We proposed three schemes for achieving this task : i) an aggregation-based scheme, ii) a problem transformation-based scheme and iii) symbolic explanation-based one. In order to assess the relevance of feature attributions obtained using our three schemes, we first highlighted three desirable properties : sensitivity, data-explanation stability and label-explanation correlation, then used such properties to assess empirically the quality of our feature attribution schemes.We proposed to go further concerning the property of label-explanation correlation by exploiting it to infer feature attributions relative to a label by using the explanations already calculated for another label with which it is correlated. Clearly, the preliminary results we have obtained confirm our intuition regarding the new property of label-explanation correlation. In addition to the feature attribution inference, it would be interesting to exploit the correlations to better present explanations. This is a track that we will investigate in our future work.

Conclusion and future work avenues

This thesis brings different contributions on explaining individual predictions of multi-class and multi-label classifiers.

Let's first summarize our results and some perspectives that we think are most relevant and promising. Note that the work of this thesis has been the subject of international publications (see ([BCAMT20]), ([BCAMT21a]), and ([BCAMT21a])).

In chronological order, we first worked on proposing a symbolic approach to generate two types of complementary explanations to explain predictions of a multi-class classification problem. We proposed a generic and declarative approach based on the encoding of the model to be explained in an equivalent symbolic representation. This latter serves to generate in particular two types of symbolic explanations which are *sufficient reasons* and *counterfactuals*. We rely on SAT-solving where we encode the problems of generating our symbolic explanations as two common problems related to satisfiability testing which are enumerating *minimal reasons* of why a formula is inconsistent (MUSes) and *minimal changes to restore the consistency of a formula* (MCSes).
For instance, our contribution makes it possible to equip the symbolic approach proposed in [SCD19] with a module for counterfactual explanations. Our work presented in [BCAMT20] takes advantage of well-defined concepts and proven tools for the MCSes enumeration. Moreover, it is specifically designed to provide exact, valid and complete explanations with a rigorous foundation since it is based on the encoding of a classifier into an **equivalent** and **tractable** symbolic representation. However, such approach suffered from some limitations. The main issue faced with this contribution is that it required a compilation process to get the symbolic representation of a classifier where the compilation algorithms (e.g [SCD19, SSDC20]) remain specific to the type of model studied and thus, limit the type of models that could be explained. The other issue was the complexity of exact methods where the size of the symbolic encoding such as Ordered Decision Diagram (ODD) associated with the classifier becomes intractable (exponential) for problems with a few dozen input variables, which makes the explanations too expensive to compute. This is not new as it is known that scalability is a weak point of all exact methods developed for full-precision or binarized networks. Moreover, another issue that arose was the question of how to choose an explanation and on what basis given the large number of explanations generated. Recall that an inconsistent Boolean formula formed by p clauses, can potentially have a large set of explanations (the number of MUSes and MCSes can be in the worst case exponential in p [LS08]).

In a second time and in order to overcome the scalability limitation mentioned above, we introduced the surrogate modeling to the encoding step of our approach. Henceforth, the encoding of a classifier is done either : (1) using model encoding algorithms if available (compilers already exist for Bayesian networks [SCD19], decision trees and some neural nets) and if the encoding is tractable (non agnostic case); (2) Or using a surrogate approach consisting in the approximation of the classifier's decisions by the mean of a surrogate model trained on the locality of an instance (agnostic case). The aim of such proposal is to balance between the guarantees of using a formal method and its feasibility in practice. Thus, we keep a rigorous symbolic formalism and we introduce the surrogate modeling to enhance the scalability of the approach and makes it more general as it assumes no knowledge whatsoever about the

model (model-agnostic). The surrogate model should guarantee to be i) as faithful as possible to the initial model (ensures same predictions) and ii) allows to obtain a tractable CNF encoding. We used the Random Forest classifiers as it showed a good trade-off between the desiderata just mentioned. The experiments showed interesting results in reducing the time and the size of the encoding representation confirming that introducing a symbolic approximation through a surrogate model makes the approach more scalable.

Afterwards, we shifted our attention to the score-based explanations where we equipped our generic model-agnostic approach to explain individual outcomes with a third module (Explanation and feature relevance scoring) that aims to evaluate the explanations by assessing their relevance w.r.t a set of natural properties. Moreover, such module allows to assess the relevance of features and to evaluate their individual contributions to the outcome using scoring functions w.r.t to a set of intuitive properties. The objective of the approach is to explain the predictions of a black-box model by providing both symbolic and score-based explanations with the help of Boolean satisfiability concepts. To the best of our knowledge, our approach is the first that generates different types of symbolic explanations and **fine-grained** score-based ones. It allows on the one hand to exploit the strength of modern SAT-solvers and on the other hand to consider other forms of symbolic explanations.

Our work then turned to the multi-label tasks, where typically many labels are predicted for each instance. First, we extended our proposed approach from a multi-class setting to explain predictions in a multi-label setting by adapting the definitions of the symbolic explanations. We defined several symbolic explanation types and showed how we can enumerate them using the existing SAT-based oracles. By taking advantage of the structural relationships between labels, a new concept for label-based explanations is introduced resulting in a reduction of the number of generated explanations and leading to a better presentation. The contributions of this work, namely, developing concepts specific to the multi-label case such as label-based and fine-grained explanations are not simple extensions from the multi-class framework to the multi-label one.

Finally, we propose a novel model-agnostic feature-based approach based on widely-used feature attribution methods and symbolic explainers. We propose two techniques to generate multi-label explanations: (1) by combining label's explanation using aggregation functions and (2) by learning a binary classifier that only recognizes the outcome to explain (transformation problem). Furthermore, we propose to infer explanations from the relationships between the output classes of a multi-label classifier. We extend the properties of sensitivity, stability to the multi-label setting in addition to a new property specific to multi-label classification that we called label-explanation correlation. We show the method's effectiveness with an empirical analysis on real-world data, yet, none of the XAI methods significantly outperforms the others.

Prospects and future work

Different avenues for future work are still open after this thesis. Firstly, our analysis of the practical feasibility of the approach has been carried out on few different application cases, but all of them designed to use binary classifiers. It would be interesting to test the applicability of our approach extended to the multi-class case. We also want to check empirically the impact of fidelity of a surrogate model on the generated explanations and their quality. For example, we would check their consistency by testing if the explanations of the surrogate model and the original are approximately the same. We can also evaluate the number of explanations generated and see if it increases/decreases with respect to the fidelity of the surrogate model.

As for numerical explanations, we plan in the future to study the relationships between score-based explanations w.r.t the different properties of sufficient reasons and counterfactuals and check if the minimal hitting set duality between MUSes and MCSes is reflected at the scores level. Another track would also be to check the consistency between the scores calculated for the variables and those calculated for the explanations. Intuitively, a variable that has an important weight w.r.t to some property is expected to participate in explanations that also have a high score and vice versa. Thus, if such a relationship were to be confirmed, one could, for example, infer the score of an explanation from the variables that compose it, and conversely, assign weights to variables from the scores of the explanations in which they participate. We also intend to explore the use of inconsistency measurements to assign scores to the different explanations obtained.
Another interesting avenue would be to check experimentally whether the proposed approach is efficient in verifying the robustness of a model and finding adversarial examples. We stress that even if for the existing approaches proposed to generate adversarial examples, the optimization problem is similar to the one posed in the generation of counterfactuals, the desiderata are different. For example, in adversarial learning (often applied to images), the goal is an imperceptible change in the input image enough to fool models into producing incorrect predictions, while for counterfactual enumeration, the goal is provide users with actionable explanations in the form of data instances that would have received a different outcome.

There are several relevant directions for future work for the multi-label setting as well. In particular, we intend to focus on how to extract the relations between the labels from the predictions of the classifiers and how to exploit the relations extracted during the generation of explanations. In theory, any relationship between labels can be exploited. In practice, we limited ourselves to certain types of relationships that are easy to extract and easy to understand for the user, but it can be interesting to explore other types of relationships linking the output classes.
Another idea for a multi-label setting would be to consider the explanations set generated using a XAI oracle (e.g LIME or SHAP) for a label as an argumentation system, and multi-label explainability as a problem fusion of argumentation systems. For example, a variable from the test input may have a positive influence (score) for a given label and a negative influence (score) for another one, thus, the question about how to attribute a final score to such variable w.r.t to the whole multi-label prediction arises. Such idea is motivated by the objective of using the concepts of argumentation for the resolution of conflicts between the beliefs of a rational agent (here explanations of a data instance input).

Another possible avenue would be the integration of preferences and knowledge of the domain upstream of the enumeration, given that SAT modeling allows us to do so.

Bibliography

[AALC21] Emre ATES, Burak AKSAR, Vitus J LEUNG, and Ayse K COSKUN. « Counterfactual Explanations for Multivariate Time Series ». In *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, pages 1–8. IEEE, 2021. 1.2, 1.2.2

[AB18] Amina ADADI and Mohammed BERRADA. « Peeking inside the black-box: a survey on explainable artificial intelligence (XAI) ». *IEEE access*, 6:52138–52160, 2018. 1.1.1

[ABB+21] Gilles AUDEMARD, Steve BELLART, Louenas BOUNIA, Frédéric KORICHE, Jean-Marie LAGNIEZ, and Pierre MARQUIS. « On the Computational Intelligibility of Boolean Classifiers ». In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, pages 74–86, 11 2021. 1.2.1, 2.2, 2.2.1, 3.3.4

[ABB+22a] Gilles AUDEMARD, Steve BELLART, Louenas BOUNIA, Frédéric KORICHE, Jean-Marie LAGNIEZ, and Pierre MARQUIS. « Les raisons majoritaires: des explications abductives pour les forêts aléatoires ». *Extraction et Gestion des Connaissances: EGC'2022*, 38, 2022. 2.2, 2.2.2, 5

[ABB+22b] Gilles AUDEMARD, Steve BELLART, Louenas BOUNIA, Frédéric KORICHE, Jean-Marie LAGNIEZ, and Pierre MARQUIS. « On the explanatory power of Boolean decision trees ». *Data Knowledge Engineering*, Page 102088, 2022. 2.2

[ABD+18] Alekh AGARWAL, Alina BEYGELZIMER, Miroslav DUDÍK, John LANGFORD, and Hanna WALLACH. « A reductions approach to fair classification ». In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018. 1.1.2

[ADLPR22] Nicholas ASHER, Lucas DE LARA, Soumya PAUL, and Chris RUSSELL. « Counterfactual Models for Fair and Adequate Explanations ». *Machine Learning and Knowledge Extraction*, 4(2):316–349, 2022. 2.1

[ADRDS+20] Alejandro Barredo ARRIETA, Natalia DÍAZ-RODRÍGUEZ, Javier DEL SER, Adrien BENNETOT, Siham TABIK, Alberto BARBADO, Salvador GARCÍA, Sergio GIL-LÓPEZ, Daniel MOLINA, Richard BENJAMINS, and OTHERS. « Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI ». *Information fusion*, 58:82–115, 2020. 1, 1.1

[ADWF15] Babak ALIPANAHI, Andrew DELONG, Matthew T WEIRAUCH, and Brendan J FREY. « Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning ». *Nature biotechnology*, 33(8):831–838, 2015. 1.1

[AET18] Muhammad Aurangzeb AHMAD, Carly ECKERT, and Ankur TEREDESAI. « Interpretable machine learning in healthcare ». In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018. 2.3

[AFM02] Jérôme AMILHASTRE, Hélene FARGIER, and Pierre MARQUIS. « Consistency restoration and explanations in dynamic CSPs—application to configuration ». *Artificial Intelligence*, 135(1-2):199–234, 2002. 2.2.1

[AGM+18] Julius ADEBAYO, Justin GILMER, Michael MUELLY, Ian GOODFELLOW, Moritz HARDT, and Been KIM. « Sanity checks for saliency maps ». *Advances in neural information processing systems*, 31, 2018. 2.3

[AJ18] David ALVAREZ-MELIS and Tommi S. JAAKKOLA. « On the Robustness of Interpretability Methods ». *CoRR*, abs/1806.08049, 2018. 2.3

[AJM12] Muhammad Naeem AYYAZ, Imran JAVED, and Waqar MAHMOOD. « Handwritten character recognition using multiclass svm classification with hybrid feature extraction ». *Pakistan Journal of Engineering and Applied Sciences*, 2012. 1.1

[AKM20a] Gilles AUDEMARD, Frédéric KORICHE, and Pierre MARQUIS. « On tractable XAI queries based on compiled representations ». In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, pages 838–849, 2020. 2.2, 2.2.1, 3.3.4

[AKM20b] Gilles AUDEMARD, Frédéric KORICHE, and Pierre MARQUIS. « On Tractable XAI Queries based on Compiled Representations ». In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, pages 838–849, 9 2020. 1.2.2

[ALSA+17] Elaine ANGELINO, Nicholas LARUS-STONE, Daniel ALABI, Margo SELTZER, and Cynthia RUDIN. « Learning certifiably optimal rule lists for categorical data ». *arXiv preprint arXiv:1704.01701*, 2017. 2.3, 5.2.2

[Alt92] Naomi S ALTMAN. « An introduction to kernel and nearest-neighbor nonparametric regression ». *The American Statistician*, 46(3):175–185, 1992. 1.1

[AMJ18] David ALVAREZ MELIS and Tommi JAAKKOLA. « Towards robust interpretability with self-explaining neural networks ». *Advances in neural information processing systems*, 31, 2018. 1.3

[ANORC13] Ignasi ABÍO, Robert NIEUWENHUIS, Albert OLIVERAS, and Enric RODRÍGUEZ-CARBONELL. « A parametric approach for smaller and better encodings of cardinality constraints ». In *International Conference on Principles and Practice of Constraint Programming*, pages 80–96. Springer, 2013. 3.2.1

[AP94] Agnar AAMODT and Enric PLAZA. « Case-based reasoning: Foundational issues, methodological variations, and system approaches ». *AI communications*, 7(1):39–59, 1994. 1.2.2

[APR21] Nicholas ASHER, Soumya PAUL, and Chris RUSSELL. « Fair and adequate explanations ». In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 79–97. Springer, 2021. 2.2

[ARAB+17] P ALLISY-ROBERTS, P AMBROSI, DT BARTLETT, BM COURSEY, LA DEWERD, E FANTUZZI, and JC MCDONALD. « The 11th Annual State of Agile Report ». *Journal of the ICRU*, 6(2):7–8, 2017. 1.1

[ARLG20] Kasun AMARASINGHE, Kit T. RODOLFA, Hemank LAMBA, and Rayid GHANI. « Explainable Machine Learning for Public Policy: Use Cases, Gaps, and Research Directions ». *CoRR*, abs/2010.14374, 2020. 2.3

[AS+94] Rakesh AGRAWAL, Ramakrishnan SRIKANT, and OTHERS. « Fast algorithms for mining association rules ». In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. Citeseer, 1994.

[AZ20] Daniel W APLEY and Jingyu ZHU. « Visualizing the effects of predictor variables in black box supervised learning models ». *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020. 2.1

[BAL+21] Clara BOVE, Jonathan AIGRAIN, Marie-Jeanne LESOT, Charles TIJUS, and Marcin DETYNIECKI. « Contextualising local explanations for non-expert users: an XAI pricing interface for insurance. ». In *IUI Workshops*, 2021. 1.2.3

[BAL+22] Clara BOVE, Jonathan AIGRAIN, Marie-Jeanne LESOT, Charles Albert TIJUS, and Marcin DETYNIECKI. « Contextualization and Exploration of Local Feature Importance Explanations to Improve Understanding and Satisfaction of Non-Expert Users ». *27th International Conference on Intelligent User Interfaces*, 2022.

[BB03] Olivier BAILLEUX and Yacine BOUFKHAD. « Efficient CNF encoding of boolean cardinality constraints ». In *International conference on principles and practice of constraint programming*, pages 108–122. Springer, 2003. 3.2.1

[BB13] Richard A BERK and Justin BLEICH. « Statistical procedures for forecasting criminal behavior: A comparative assessment ». *Criminology & Pub. Pol'y*, 12:513, 2013. 1.1

[BBM+15] Sebastian BACH, Alexander BINDER, Grégoire MONTAVON, Frederick KLAUSCHEN, Klaus-Robert MÜLLER, and Wojciech SAMEK. « On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation ». *PloS one*, 10(7):e0130140, 2015. (document), 1.2.2, 1.7, 1.2.2, 1.2.3, 5.1.1, 6.1

[BCAMT20] Ryma BOUMAZOUZA, Fahima CHEIKH-ALILI, Bertrand MAZURE, and Karim TABIA. « A Symbolic Approach for Counterfactual Explanations ». In *International Conference on Scalable Uncertainty Management*, pages 270–277. Springer, 2020. 6.6

[BCAMT21a] Ryma BOUMAZOUZA, Fahima CHEIKH-ALILI, Bertrand MAZURE, and Karim TABIA. « ASTERYX: A model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations ». In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 120–129, 2021. 6.6

[BCAMT21b] Ryma BOUMAZOUZA, Fahima CHEIKH-ALILI, Bertrand MAZURE, and Karim TABIA. « A ModelAgnostic SAT-based Approach for Symbolic Explanation Enumeration ». In *Proceedings of the 23rd International Conference on Artificial Intelligence*, 2021.

[BČB18] Jaroslav BENDÍK, Ivana ČERNÁ, and Nikola BENEŠ. « Recursive online enumeration of all minimal unsatisfiable subsets ». In *International symposium on automated technology for verification and analysis*, pages 143–159. Springer, 2018. 3.1, 3.3.1

[BCC+16] Mariusz BOJARSKI, Anna CHOROMANSKA, Krzysztof CHOROMANSKI, Bernhard FIRNER, Larry JACKEL, Urs MULLER, and Karol ZIEBA. « Visualbackprop: visualizing cnns for autonomous driving ». *arXiv preprint arXiv:1611.05418*, 2, 2016. 1.2.2

[BCNM06] C BUCILUA, R CARUANA, and A NICULESCU-MIZIL. « Model compression, in proceedings of the 12 th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ». *New York, NY, USA*, 3, 2006. 1.2.2

[BCPdI19] Alberto BLANCO, Arantza CASILLAS, Alicia PÉREZ, and Arantza Diaz de ILARRAZA. « Multi-label clinical document classification: Impact of label-density ». *Expert Systems with Applications*, 138:112835, 2019. 4.2

[BdSRM14] Flávia Cristina BERNARDINI, Rodrigo Barbosa da SILVA, Rodrigo Magalhaes RODOVALHO, and Edwin B. Mitacc MEZA. « Cardinality and Density Measures and Their Influence to Multi-Label Learning Methods ». *Learning and Nonlinear Models*, 12:53–71, 2014. 1.2

[BDTK14] Fahiem BACCHUS, Jessica DAVIES, Maria TSIMPOUKELLI, and George KATSIRELOS. « Relaxation search: A simple way of managing optional clauses ». In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014. 3.3.4

[Ber21] Leopoldo BERTOSSI. « Declarative approaches to counterfactual explanations for classification ». *Theory and Practice of Logic Programming*, pages 1–35, 2021. 4.6

[BFOS84] Leo BREIMAN, JH FRIEDMAN, RA OLSHEN, and CJ STONE. « Classification and regression trees. Wadsworth & Brooks ». *Cole Statistics/Probability Series*, 1984. 1.2.1

[BH17] Stefan BERNDORFER and Aron HENRIKSSON. « Automated diagnosis coding with combined text representations ». *Stud Health Technol Inform*, 235:201–205, 2017. 4.2

[BHvM09] Armin BIERE, Marijn HEULE, and Hans van MAAREN. *Handbook of satisfiability*, volume 185. IOS press, 2009. 3.2, 3.1, 3.3.1

[BJL09] Ruth MJ BYRNE and Philip N JOHNSON-LAIRD. « 'If'and the problems of conditional reasoning ». *Trends in Cognitive Sciences*, 13(7):282–287, 2009. 1.2.1

[BK15] Fahiem BACCHUS and George KATSIRELOS. « Using minimal correction sets to more efficiently compute minimal unsatisfiable sets ». In *International Conference on Computer Aided Verification*, pages 70–86. Springer, 2015. 3.1, 3.3.1, 3.3.4, 3.3.4

[BK16] Fahiem BACCHUS and George KATSIRELOS. « Finding a collection of MUSes incrementally ». In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 35–44. Springer, 2016. 3.1, 3.3.1

[BL03] Elazar BIRNBAUM and Eliezer L LOZINSKII. « Consistent subsets of inconsistent systems: structure and behaviour ». *Journal of Experimental & Theoretical Artificial Intelligence*, 15(1):25–46, 2003. 3.1, 3.3.4, 3.3.4

[BL22] Isabelle BLOCH and Marie-Jeanne LESOT. « Towards a Formulation of Fuzzy Contrastive Explanations ». *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, 2022. 1.2.2

[BLS+18] Geuntae BAE, Hojae LEE, Sunghoon SON, Doha HWANG, and Jongseok KIM. « Secure and robust user authentication using partial fingerprint matching ». In *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6. IEEE, 2018. 1.1

[BLSB04] Matthew R BOUTELL, Jiebo LUO, Xipeng SHEN, and Christopher M BROWN. « Learning multi-label scene classification ». *Pattern recognition*, 37(9):1757–1771, 2004. 1.2

[BMPS20] Pablo BARCELÓ, Mikaël MONET, Jorge PÉREZ, and Bernardo SUBERCASEAUX. « Model interpretability through the lens of computational complexity ». *Advances in neural information processing systems*, 33:15487–15498, 2020. 1.2.1

[BMS11] Anton BELOV and Joao MARQUES-SILVA. « Accelerating MUS extraction with recursive model rotation ». In *2011 Formal Methods in Computer-Aided Design (FMCAD)*, pages 37–40. IEEE, 2011. 3.3.4

[BMS12] Anton BELOV and Joao MARQUES-SILVA. « MUSer2: An efficient MUS extractor ». *Journal on Satisfiability, Boolean Modeling and Computation*, 8(3-4):123–128, 2012. 3.3.4

[BP19] Ismaïl BAAJ and Jean-Philippe POLI. « Natural language generation of explanations of fuzzy inference decisions ». In *2019 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2019. 2.3

[BRB18] Wieland BRENDEL, Jonas RAUBER, and Matthias BETHGE. « Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models ». In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 1.1.2

[Bre01a] Leo BREIMAN. « Random forests ». *Machine learning*, 45(1):5–32, 2001. 1.2.2

[Bre01b] Leo BREIMAN. « Statistical modeling: The two cultures (with comments and a rejoinder by the author) ». *Statistical science*, 16(3):199–231, 2001. 1.2.1

[Bry86] Randal E BRYANT. « Graph-based algorithms for boolean function manipulation ». *Computers, IEEE Transactions on*, 100(8):677–691, 1986. 2.2.1

[BS05] James BAILEY and Peter J STUCKEY. « Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization ». In *International Workshop on Practical Aspects of Declarative Languages*, pages 174–186. Springer, 2005. 3.3.4, 3.3.4

[BSH+10] David BAEHRENS, Timon SCHROETER, Stefan HARMELING, Motoaki KAWANABE, Katja HANSEN, and Klaus-Robert MÜLLER. « How to explain individual classification decisions ». *The Journal of Machine Learning Research*, 11:1803–1831, 2010. 5.1

[BTD+16] Mariusz BOJARSKI, Davide Del TESTA, Daniel DWORAKOWSKI, Bernhard FIRNER, Beat FLEPP, Prasoon GOYAL, Lawrence D. JACKEL, Mathew MONFORT, Urs MULLER, Jiakai ZHANG, Xin ZHANG, Jake ZHAO, and Karol ZIEBA. « End to End Learning for Self-Driving Cars ». *CoRR*, abs/1604.07316, 2016. 1.1

[BVKV+18] Reuben BINNS, Max VAN KLEEK, Michael VEALE, Ulrik LYNGS, Jun ZHAO, and Nigel SHADBOLT. « 'It's Reducing a Human Being to a Percentage' Perceptions of

Justice in Algorithmic Decisions ». In *Proceedings of the 2018 Chi conference on human factors in computing systems*, pages 1–14, 2018. 1.1.2

[BWM20] Umang BHATT, Adrian WELLER, and José MF MOURA. « Evaluating and aggregating feature-based model explanations ». *arXiv preprint arXiv:2005.00631*, 2020. 2.3

[BXS+20] Umang BHATT, Alice XIANG, Shubham SHARMA, Adrian WELLER, Ankur TALY, Yunhan JIA, Joydeep GHOSH, Ruchir PURI, José MF MOURA, and Peter ECKERSLEY. « Explainable machine learning in deployment ». In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 648–657, 2020. 1.1.3, 5.1.1

[Byr19] Ruth MJ BYRNE. « Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. ». In *IJCAI*, pages 6276–6282, 2019. 1.2.2

[CD03] H. CHAN and Adnan DARWICHE. « Reasoning about Bayesian Network Classifiers ». In *UAI*, 2003. 2.2.1

[CD07] Mark CHAVIRA and Adnan DARWICHE. « Compiling Bayesian Networks Using Variable Elimination. ». In *IJCAI*, volume 2443, 2007. 2.2.1

[CE13] Paulo CORTEZ and Mark J EMBRECHTS. « Using sensitivity analysis and visualization techniques to open black box data mining models ». *Information Sciences*, 225:1–17, 2013. (document), 1.2.2, 1.11

[CG16] Tianqi CHEN and Carlos GUESTRIN. « Xgboost: A scalable tree boosting system ». In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[CGG+20] Gabriele CIRAVEGNA, Francesco GIANNINI, Marco GORI, Marco MAGGINI, and Stefano MELACCI. « Human-Driven FOL Explanations of Deep Learning. ». In *IJCAI*, pages 2234–2240, 2020. 4.1, 4.3

[Che21] Shikun CHEN. « Interpretation of multi-label classification models using shapley values ». *CoRR*, abs/2104.10505, 2021. 4.1, 6.1

[Chr94] H CHRISTOS. « PAPADIMITRIOU: Computational complexity ». *Addison-Wesley*, 2(3):4, 1994. 3.1

[CK01] Amanda CLARE and Ross D KING. « Knowledge discovery in multi-label phenotype data ». In *European conference on principles of data mining and knowledge discovery*, pages 42–53. Springer, 2001. 2

[CLG+15] Rich CARUANA, Yin LOU, Johannes GEHRKE, Paul KOCH, Marc STURM, and Noemie ELHADAD. « Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission ». In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015. 1.1, 1.3

[CLP+18] Jiaoyan CHEN, Freddy LÉCUÉ, Jeff Z PAN, Ian HORROCKS, and Huajun CHEN. « Knowledge-based transfer learning explanation ». In *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2018. 1.2, 1.3

[CLR+18] Chaofan CHEN, Kangcheng LIN, Cynthia RUDIN, Yaron SHAPOSHNIK, Sijia WANG, and Tong WANG. « An interpretable model with globally consistent explanations for credit risk ». *arXiv preprint arXiv:1811.12615*, 2018. 1.1

[CNC+16] Jie-Zhi CHENG, Dong NI, Yi-Hong CHOU, Jing QIN, Chui-Mei TIU, Yeun-Chung CHANG, Chiun-Sheng HUANG, Dinggang SHEN, and Chung-Ming CHEN. « Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans ». *Scientific reports*, 6(1):1–13, 2016. 1.1

[CNQ03] Gianpiero CABODI, Sergio NOCCO, and Stefano QUER. « Improving SAT-based bounded model checking by means of BDD-based approximate traversals ». In *2003 Design, Automation and Test in Europe Conference and Exhibition*, pages 898–903. IEEE, 2003. 3.2.1

[Coo71] Stephen A COOK. « The complexity of theorem-proving procedures ». In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971. 3.1, 3.3.4

[CRT98] Alessandro CIMATTI, Marco ROVERI, and Paolo TRAVERSO. « Automatic OBDD-based generation of universal plans in non-deterministic domains ». In *AAAI/IAAI*, pages 875–881, 1998. 2.2.1

[CS95] Mark CRAVEN and Jude SHAVLIK. « Extracting tree-structured representations of trained networks ». *Advances in neural information processing systems*, 8, 1995. 1.2.1, 1.2.2

[CSGD20a] Arthur CHOI, Andy SHIH, Anchal GOYANKA, and Adnan DARWICHE. « On Symbolically Encoding the Behavior of Random Forests ». In *3rd Workshop on Formal Methods for ML-Enabled Autonomous Systems (FoMLAS)*, 2020. 2.2.2

[CSGD20b] Arthur CHOI, Andy SHIH, Anchal GOYANKA, and Adnan DARWICHE. « On symbolically encoding the behavior of random forests ». *arXiv preprint arXiv:2007.01493*, 2020. 3.3.4

[CSHB18] Aditya CHATTOPADHAY, Anirban SARKAR, Prantik HOWLADER, and Vineeth N BALASUBRAMANIAN. « Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks ». In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 1.2, 1.2.2, 1.2.3, 1.3

[CSSD17] Arthur CHOI, Weijia SHI, Andy SHIH, and Adnan DARWICHE. « Compiling neural networks into tractable Boolean circuits ». *intelligence*, 2017. 2.2.1

[CT95] Zhi-Zhong CHEN and Seinosuke TODA. « The complexity of selecting maximal solutions ». *Information and Computation*, 119(2):231–239, 1995. 3.1, 3.3.4

[CV95] Corinna CORTES and Vladimir VAPNIK. « Support-vector networks ». *Machine learning*, 20(3):273–297, 1995. 1.1

[CVKRBA12] Binu P CHACKO, VR VIMAL KRISHNAN, G RAJU, and P BABU ANTO. « Handwritten character recognition using wavelet energy and extreme learning machine ». *International Journal of Machine Learning and Cybernetics*, 3(2):149–161, 2012. 1.1

[CW17] Nicholas CARLINI and David WAGNER. « Towards evaluating the robustness of neural networks ». In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1.1.1, 1.1.2

[CW18] Nicholas CARLINI and David WAGNER. « Audio adversarial examples: Targeted attacks on speech-to-text ». In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018. 1.1.1

[Dar99] Adnan DARWICHE. « Compiling knowledge into decomposable negation normal form ». In *IJCAI*, volume 99, pages 284–289. Citeseer, 1999. 2.2.1

[Dar01] Adnan DARWICHE. « Decomposable negation normal form ». *Journal of the ACM (JACM)*, 48(4):608–647, 2001. 2.2.1

[Dar11] Adnan DARWICHE. « SDD: A new canonical representation of propositional knowledge bases ». In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. 2.2.1

[Dar20] Adnan DARWICHE. « Three Modern Roles for Logic in AI ». In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS'20, Page 229–243, New York, NY, USA, 2020. Association for Computing Machinery. 1.2.2, 3, 4.3.1

[Das18] Jeffrey DASTIN. Amazon scraps secret AI recruiting tool that showed bias against women. In *Ethics of Data and Analytics*, pages 296–299. Auerbach Publications, 2018. 1.1

[dCF09] André CPLF de CARVALHO and Alex A FREITAS. « A tutorial on multi-label classification techniques ». *Foundations of computational intelligence volume 5*, pages 177–195, 2009. 1

[DCL+18] Amit DHURANDHAR, Pin-Yu CHEN, Ronny LUSS, Chun-Chen TU, Paishun TING, Karthikeyan SHANMUGAM, and Payel DAS. « Explanations based on the missing: Towards contrastive explanations with pertinent negatives ». *Advances in neural information processing systems*, 31, 2018. 1.2, 1.2.2, 3.3.3

[DF18] Julia DRESSEL and Hany FARID. « The accuracy, fairness, and limits of predicting recidivism ». *Science advances*, 4(1):eaao5580, 2018. 1.1

[DFL+18] Lieyun DING, Weili FANG, Hanbin LUO, Peter ED LOVE, Botao ZHONG, and Xi OUYANG. « A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory ». *Automation in construction*, 86:118–124, 2018. 1.1

[DG17] D DUA and C GRAFF. « UCI machine learning repository. University of California, School of Information and Computer Science », 2017.

[DGW18] Sanjeeb DASH, Oktay GUNLUK, and Dennis WEI. « Boolean decision rules via column generation ». *Advances in neural information processing systems*, 31, 2018. 1.1, 1.3

[DH22] Adnan DARWICHE and Auguste HIRTH. « On the (Complete) Reasons Behind Decisions ». *Journal of Logic, Language and Information*, pages 1–26, 2022. 1.2, 1.2, 1.3, 2.2, 5

[DHP+12] Cynthia DWORK, Moritz HARDT, Toniann PITASSI, Omer REINGOLD, and Richard ZEMEL. « Fairness through awareness ». In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012. 1.2, 1.3

[DLM+22] Sanghamitra DUTTA, Jason LONG, Saumitra MISHRA, Cecilia TILLI, and Daniele MAGAZZENI. « Robust Counterfactual Explanations for Tree-Based Ensembles ». In *International Conference on Machine Learning*, pages 5742–5756. PMLR, 2022. 1.2, 1.3

[DLP+18] Yinpeng DONG, Fangzhou LIAO, Tianyu PANG, Hang SU, Jun ZHU, Xiaolin HU, and Jianguo LI. « Boosting adversarial attacks with momentum ». In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1.1.2

[DLT+18] Hanjun DAI, Hui LI, Tian TIAN, Xin HUANG, Lin WANG, Jun ZHU, and Le SONG. « Adversarial attack on graph structured data ». In *International conference on machine learning*, pages 1115–1124. PMLR, 2018. 1.1.2

[DM02] Adnan DARWICHE and Pierre MARQUIS. « A knowledge compilation map ». *Journal of Artificial Intelligence Research*, 17:229–264, 2002. 2.2.1, 3.2.1

[DPB+19] Amit DHURANDHAR, Tejaswini PEDAPATI, Avinash BALAKRISHNAN, Pin-Yu CHEN, Karthikeyan SHANMUGAM, and Ruchir PURI. « Model Agnostic Contrastive Explanations for Structured Data ». *CoRR*, abs/1906.00117, 2019. 1.2, 1.2.2

[DR20] Arun DAS and Paul RAD. « Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey ». *CoRR*, abs/2006.11371, 2020.

[DRGM10] Emanuele DI ROSA, Enrico GIUNCHIGLIA, and Marco MARATEA. « Solving satisfiability problems with preferences ». *Constraints*, 15(4):485–515, 2010. 3.3.4

[DSZ16] Anupam DATTA, Shayak SEN, and Yair ZICK. « Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems ». In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE, 2016. 1.1.2, 1.2.2

[DVK17] Finale DOSHI-VELEZ and Been KIM. « Towards a rigorous science of interpretable machine learning ». *arXiv preprint arXiv:1702.08608*, 2017. 1.1, 2.3, 5.2.2

[DVKB+17] Finale DOSHI-VELEZ, Mason KORTZ, Ryan BUDISH, Chris BAVITZ, Sam GERSHMAN, David O'BRIEN, Kate SCOTT, Stuart SCHIEBER, James WALDO, David WEINBERGER, and OTHERS. « Accountability of AI under the law: The role of explanation ». *arXiv preprint arXiv:1711.01134*, 2017. 1.1

[EMW97] Michael D ERNST, Todd D MILLSTEIN, and Daniel S WELD. « Automatic SAT-compilation of planning problems ». In *IJCAI*, volume 97, pages 1169–1176, 1997. 2.2.1

[ERLD17] Javid EBRAHIMI, Anyi RAO, Daniel LOWD, and Dejing DOU. « Hotflip: White-box adversarial examples for text classification ». *arXiv preprint arXiv:1712.06751*, 2017. 1.1.1

[EW01] André ELISSEEFF and Jason WESTON. « A kernel method for multi-labelled classification ». *Advances in neural information processing systems*, 14, 2001. 1.2

[FdHvE22] Hidde FOKKEMA, Rianne de HEIDE, and Tim van ERVEN. « Attribution-based Explanations that Provide Recourse Cannot be Robust ». *CoRR*, abs/2205.15834, 2022. 2.1

[FdSRGL22] Joao FERREIRA, Manuel de SOUSA RIBEIRO, Ricardo GONÇALVES, and Joao LEITE. « Looking Inside the Black-Box: Logic-based Explanations for Neural Networks ». In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, pages 432–442, 2022. 1.2, 1.3

[FH17] Nicholas FROSST and Geoffrey E. HINTON. « Distilling a Neural Network Into a Soft Decision Tree ». In Tarek R. BESOLD and Oliver KUTZ, editors, *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 16th and 17th, 2017*, volume 2071 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017. 1.2.2

[Fis36] Ronald A FISHER. « The use of multiple measurements in taxonomic problems ». *Annals of eugenics*, 7(2):179–188, 1936. (document), 1.5

[FJ18] Matteo FISCHETTI and Jason JO. « Deep neural networks and mixed integer linear optimization ». *Constraints*, 23(3):296–309, 2018.

[Fre14] Alex A FREITAS. « Comprehensible classification models: a position paper ». *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014. 1.2.1, 1.2.1

[Fri01] Jerome H FRIEDMAN. « Greedy function approximation: a gradient boosting machine ». *Annals of statistics*, pages 1189–1232, 2001. 1.2, 1.2.2, 1.3

[FSZ12] Alexander FELFERNIG, Monika SCHUBERT, and Christoph ZEHENTNER. « An efficient diagnosis algorithm for inconsistent constraint sets ». *AI EDAM*, 26(1):53–62, 2012.

[GA19] David GUNNING and David AHA. « DARPA's explainable artificial intelligence (XAI) program ». *AI magazine*, 40(2):44–58, 2019. (document), 1.1, 1.1, 1.1.1, 1.2

[GBY+18] Leilani H GILPIN, David BAU, Ben Z YUAN, Ayesha BAJWA, Michael SPECTER, and Lalana KAGAL. « Explaining explanations: An overview of interpretability of machine learning ». In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018. 5.1

[GDL03] Muriel GEVREY, Ioannis DIMOPOULOS, and Sovan LEK. « Review and comparison of methods to study the contribution of variables in artificial neural network models ». *Ecological modelling*, 160(3):249–264, 2003. 1.2.2

[GIL18] Éric GRÉGOIRE, Yacine IZZA, and Jean-Marie LAGNIEZ. « Boosting MCSes Enumeration. ». In *IJCAI*, pages 1309–1315, 2018. 3.3.4, 3.4.1, 4.5, 6.4.1

[GKBP15] Alex GOLDSTEIN, Adam KAPELNER, Justin BLEICH, and Emil PITKIN. « Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation ». *journal of Computational and Graphical Statistics*, 24(1):44–65, 2015. (document), 1.2.2, 1.12

[GKC+18] Olivier GOUDET, Diviyan KALAINATHAN, Philippe CAILLOU, Isabelle GUYON, David LOPEZ-PAZ, and Michele SEBAG. Learning functional causal models with generative neural networks. In *Explainable and interpretable models in computer vision and machine learning*, pages 39–80. Springer, 2018. 1.2.2

[GLB+10] Michael L GATZA, Joseph E LUCAS, William T BARRY, Jong Wook KIM, Quanli WANG, Matthew D CRAWFORD, Michael B DATTO, Michael KELLEY, Bernard MATHEY-PREVOT, Anil POTTI, and OTHERS. « A pathway-based classification of human breast cancer ». *Proceedings of the National Academy of Sciences*, 107(15):6994–6999, 2010. 1.1

[GLM14] Éric GRÉGOIRE, Jean-Marie LAGNIEZ, and Bertrand MAZURE. « An experimentally efficient method for (MSS, CoMSS) partitioning ». In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. 3.3.4

[GMP07] Eric GRÉGOIRE, Bertrand MAZURE, and Cédric PIETTE. « Boosting a Complete Technique to Find MSS and MUS Thanks to a Local Search Oracle. ». In *IJCAI-07*, volume 7, pages 2300–2305, 2007. 3.1, 3.3.1

[GMR+18] Riccardo GUIDOTTI, Anna MONREALE, Salvatore RUGGIERI, Franco TURINI, Fosca GIANNOTTI, and Dino PEDRESCHI. « A survey of methods for explaining black box models ». *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[GMX+18] Wenbo GUO, Dongliang MU, Jun XU, Purui SU, Gang WANG, and Xinyu XING. « Lemna: Explaining deep learning based security applications ». In *proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 364–379, 2018. 1.2, 1.2.3, 1.3

[GPM+17] Kathrin GROSSE, Nicolas PAPERNOT, Praveen MANOHARAN, Michael BACKES, and Patrick MCDANIEL. « Adversarial examples for malware detection ». In *European symposium on research in computer security*, pages 62–79. Springer, 2017. 1.1.1

[GPSS19] Sachin GROVER, Chiara PULICE, Gerardo I SIMARI, and VS SUBRAHMANIAN. « Beef: Balanced english explanations of forecasts ». *IEEE Transactions on Computational Social Systems*, 6(2):350–364, 2019. 1.2, 1.3

[GR22] Niku GORJI and Sasha RUBIN. « Sufficient Reasons for Classifier Decisions in the Presence of Domain Constraints ». *AAAI, February*, 2022. 2.2

[GSS15] Ian J. GOODFELLOW, Jonathon SHLENS, and Christian SZEGEDY. « Explaining and Harnessing Adversarial Examples ». In Yoshua BENGIO and Yann LECUN, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1.1.1, 1.1.2

[GT99] Fausto GIUNCHIGLIA and Paolo TRAVERSO. « Planning as model checking ». In *European Conference on Planning*, pages 1–20. Springer, 1999. 2.2.1

[Gup06] Anubhav GUPTA. « *Learning abstractions for model checking* ». PhD thesis, Carnegie Mellon University, 2006. 3.1, 3.3.4

[GWE+19] Yash GOYAL, Ziyan WU, Jan ERNST, Dhruv BATRA, Devi PARIKH, and Stefan LEE. « Counterfactual visual explanations ». In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019. 1.2, 1.2.2

[Hau85] John HAUGELAND. « Artificial intelligence: the very idea », 1985. 1

[Hay94] Simon HAYKIN. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994. 1.1

[HCS+16] Itay HUBARA, Matthieu COURBARIAUX, Daniel SOUDRY, Ran EL-YANIV, and Yoshua BENGIO. « Binarized Neural Networks ». In *Advances in Neural Information Processing Systems*, volume 29, 2016. 3.4

[HD05] Jinbo HUANG and Adnan DARWICHE. « On compiling system models for faster and more scalable diagnosis ». In *AAAI*, pages 300–306, 2005. 2.2.1

[HDM+11] Johan HUYSMANS, Karel DEJAEGER, Christophe MUES, Jan VANTHIENEN, and Bart BAESENS. « An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models ». *Decision Support Systems*, 51(1):141–154, 2011. 1.2.1, 1.2.1

[HFS+18] Holger A HAENSSLE, Christine FINK, Roland SCHNEIDERBAUER, Ferdinand TOBERER, Timo BUHL, Andreas BLUM, A KALLOO, A Ben Hadj HASSEN, Luc THOMAS, A ENK, and OTHERS. « Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists ». *Annals of oncology*, 29(8):1836–1842, 2018.

[HH18] Satoshi HARA and Kohei HAYASHI. « Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach ». In Amos J. STORKEY and Fernando PÉREZ-CRUZ, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 77–85. PMLR, 2018. 1.2.1

[HHDA18] Lisa Anne HENDRICKS, Ronghang HU, Trevor DARRELL, and Zeynep AKATA. « Generating Counterfactual Explanations with Natural Language ». *CoRR*, abs/1806.09809, 2018. 1.2, 1.2.2

[HIIM21] Xuanxiang HUANG, Yacine IZZA, Alexey IGNATIEV, and João MARQUES-SILVA. « On Efficiently Explaining Graph-Based Classifiers ». In Meghyn BIENVENU, Gerhard LAKEMEYER, and Esra ERDEM, editors, *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021*, pages 356–367, 2021. 1.2.1, 2.2

[Hin19] Michael HIND. « Explaining explainable AI ». *XRDS: Crossroads, The ACM Magazine for Students*, 25(3):16–19, 2019. (document), 1.1.3, 1.3

[HKWL21] Shu HU, Lipeng KE, Xin WANG, and Siwei LYU. « TkML-AP: Adversarial Attacks to Top-k Multi-Label Learning ». In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7649–7657, October 2021.

[HM22] Xuanxiang HUANG and João MARQUES-SILVA. « On Deciding Feature Membership in Explanations of SDD & Related Classifiers ». *CoRR*, abs/2202.07553, 2022. 2.2

[Ho95] Tin Kam HO. « Random decision forests ». In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995. 3.2.1

[HPS16] Moritz HARDT, Eric PRICE, and Nati SREBRO. « Equality of opportunity in supervised learning ». *Advances in neural information processing systems*, 29, 2016. 1.1.2

[HRS19] Xiyang HU, Cynthia RUDIN, and Margo SELTZER. « Optimal sparse decision trees ». *Advances in Neural Information Processing Systems*, 32, 2019. 1.1

[HSJ+04] Tarik HADZIC, Sathiamoorthy SUBBARAYAN, Rune M JENSEN, Henrik R ANDERSEN, Jesper MØLLER, and Henrik HULGAARD. « Fast backtrack-free product configuration using a precompiled solution space representation ». *small*, 10(1):3, 2004. 2.2.1

[HVD+15] Geoffrey HINTON, Oriol VINYALS, Jeff DEAN, and OTHERS. « Distilling the knowledge in a neural network ». *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 1.2.2

[HY01] David J HAND and Keming YU. « Idiot's Bayes—not so stupid after all? ». *International statistical review*, 69(3):385–398, 2001. 1.1

[HYHI21] Kazuaki HANAWA, Sho YOKOI, Satoshi HARA, and Kentaro INUI. « Evaluation of Similarity-based Explanations ». In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2.3

[Ign20] Alexey IGNATIEV. « Towards Trustable Explainable AI. ». In *IJCAI*, pages 5154–5158, 2020. 1.2.2, 2.1, 2.2, 2.2.2, 3.3.4, 5

[IIM20] Yacine IZZA, Alexey IGNATIEV, and João MARQUES-SILVA. « On Explaining Decision Trees ». *CoRR*, abs/2010.11034, 2020. 1.2.2, 3.3.4

[IIM22] Yacine IZZA, Alexey IGNATIEV, and João MARQUES-SILVA. « On Tackling Explanation Redundancy in Decision Trees ». *J. Artif. Intell. Res.*, 75:261–321, 2022. 2.2, 2.2.2

[IIN+21] Yacine IZZA, Alexey IGNATIEV, Nina NARODYTSKA, Martin C. COOPER, and João MARQUES-SILVA. « Efficient Explanations With Relevant Sets ». *CoRR*, abs/2106.00546, 2021. 2.2

[IIN+22] Yacine IZZA, Alexey IGNATIEV, Nina NARODYTSKA, Martin C COOPER, and Joao MARQUES-SILVA. « Provably Precise, Succinct and Efficient Explanations for Decision Trees ». *arXiv preprint arXiv:2205.09569*, 2022. 2.2

[IM21] Yacine IZZA and João MARQUES-SILVA. « On Explaining Random Forests with SAT ». In Zhi-Hua ZHOU, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 2584–2591. ijcai.org, 2021. 1.2.2, 3.3.4

[IMM18] Alexey IGNATIEV, Antonio MORGADO, and Joao MARQUES-SILVA. « PySAT: A Python Toolkit for Prototyping with SAT Oracles ». In *SAT*, pages 428–437, 2018. 6.4.1

[IMMS18] Alexey IGNATIEV, Antonio MORGADO, and Joao MARQUES-SILVA. « PySAT: A Python toolkit for prototyping with SAT oracles ». In *International Conference on Theory and Applications of Satisfiability Testing*, pages 428–437. Springer, 2018. 3.3.4

[IMS21] Alexey IGNATIEV and Joao MARQUES-SILVA. « SAT-based rigorous explanations for decision lists ». In *International Conference on Theory and Applications of Satisfiability Testing*, pages 251–269. Springer, 2021. 1.2.2, 2.2, 2.2.2

[INAM20] Alexey IGNATIEV, Nina NARODYTSKA, Nicholas ASHER, and João MARQUES-SILVA. « On Relating 'Why?' and 'Why Not?' Explanations ». *CoRR*, abs/2012.11067, 2020. 1.2.2

[INAMS20] Alexey IGNATIEV, Nina NARODYTSKA, Nicholas ASHER, and Joao MARQUES-SILVA. « From contrastive to abductive explanations and back again ». In *International Conference of the Italian Association for Artificial Intelligence*, pages 335–355. Springer, 2020. 3, 3.3.3, 3.3.4

[INM19] Alexey IGNATIEV, Nina NARODYTSKA, and João MARQUES-SILVA. « On Validating, Repairing and Refining Heuristic ML Explanations ». *CoRR*, abs/1907.02509, 2019. 2.1

[INMS19a] Alexey IGNATIEV, Nina NARODYTSKA, and Joao MARQUES-SILVA. « Abduction-based explanations for machine learning models ». In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1511–1519, 2019. 1.2, 1.2, 1.2.2, 1.2.2, 1.2.2, 1.3, 2.2, 2.2.2, 3, 5

[INMS19b] Alexey IGNATIEV, Nina NARODYTSKA, and Joao MARQUES-SILVA. « On Relating Explanations and Adversarial Examples ». In *Advances in Neural Information Processing Systems*, volume 32, 2019. 1.2.2, 2.2, 2.2.2, 5, 5.1.1

[IPLMS15] Alexey IGNATIEV, Alessandro PREVITI, Mark LIFFITON, and Joao MARQUES-SILVA. « Smallest MUS extraction with minimal hitting set dualization ». In *International Conference on Principles and Practice of Constraint Programming*, pages 173–182. Springer, 2015. 3.3.4

[IPNMS18] Alexey IGNATIEV, Filipe PEREIRA, Nina NARODYTSKA, and Joao MARQUES-SILVA. « A SAT-based approach to learn explainable decision sets ». In *International Joint Conference on Automated Reasoning*, pages 627–645. Springer, 2018. 1.1

[JBB+21] Sérgio JESUS, Catarina BELÉM, Vladimir BALAYAN, João BENTO, Pedro SALEIRO, Pedro BIZARRO, and João GAMA. « How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations ». In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 805–815, 2021. 2.3

[JCS+20] Jongbin JUNG, Connor CONCANNON, Ravi SHROFF, Sharad GOEL, and Daniel G GOLDSTEIN. « Simple rules to guide expert classifications ». *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 183(3):771–800, 2020. 1.2

[JG20] Alon JACOVI and Yoav GOLDBERG. « Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? ». In Dan JURAFSKY, Joyce CHAI, Natalie SCHLUTER, and Joel R. TETREAULT, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4198–4205. Association for Computational Linguistics, 2020. 2.3, 5.2.2

[JKN04] Ulf JOHANSSON, Rikard KÖNIG, and Lars NIKLASSON. « The truth is in there-rule extraction from opaque models using genetic programming. ». In *FLAIRS Conference*, pages 658–663. Miami Beach, FL, 2004. 1.2, 1.3

[JKV+19] Shalmali JOSHI, Oluwasanmi KOYEJO, Warut VIJITBENJARONK, Been KIM, and Joydeep GHOSH. « Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making Systems ». *CoRR*, abs/1907.09615, 2019. 1.2.2

[JMD+05] Aleks JAKULIN, Martin MOŽINA, Janez DEMŠAR, Ivan BRATKO, and Blaž ZUPAN. « Nomograms for visualizing support vector machines ». In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 108–117, 2005. 1.2.2

[JSS15] Said JABBOUR, Lakhdar SAIS, and Yakoub SALHI. « Decomposition based SAT encodings for itemset mining problems ». In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 662–674. Springer, 2015. 4.6

[Jun04] Ulrich JUNKER. « Preferred explanations and relaxations for over-constrained problems ». In *AAAI-2004*, 2004.

[JV00] Rune M JENSEN and Manuela M VELOSO. « OBDD-based universal planning for synchronized agents in non-deterministic domains ». *Journal of Artificial Intelligence Research*, 13:189–226, 2000. 2.2.1

[KAAS12] Toshihiro KAMISHIMA, Shotaro AKAHO, Hideki ASOH, and Jun SAKUMA. « Fairness-aware classifier with prejudice remover regularizer ». In *Joint European conference on machine learning and knowledge discovery in databases*, pages 35–50. Springer, 2012. 1.1.2

[Kas21] Atoosa KASIRZADEH. « Reasons, Values, Stakeholders: A Philosophical Framework for Explainable Artificial Intelligence ». In Madeleine Clare ELISH, William ISAAC, and Richard S. ZEMEL, editors, *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, Page 14. ACM, 2021. 1.1.2

[KBBV20] Amir-Hossein KARIMI, Gilles BARTHE, Borja BALLE, and Isabel VALERA. « Model-agnostic counterfactual explanations for consequential decisions ». In *International Conference on Artificial Intelligence and Statistics*, pages 895–905. PMLR, 2020. 1.2, 1.2.2, 1.2.2

[KBD+17] Guy KATZ, Clark BARRETT, David L DILL, Kyle JULIAN, and Mykel J KOCHENDERFER. « Reluplex: An efficient SMT solver for verifying deep neural networks ». In *International conference on computer aided verification*, pages 97–117. Springer, 2017. 2.2.2

[KGJS15] Been KIM, Elena GLASSMAN, Brittney JOHNSON, and Julie SHAH. « iBCM: Interactive Bayesian case model empowering humans via intuitive interaction ». 2015. 1.1, 1.3

[KK19a] Jakko KEMPER and Daan KOLKMAN. « Transparent to whom? No algorithmic accountability without a critical audience ». *Information, Communication & Society*, 22(14):2081–2096, 2019. 1.1.2

[KK19b] Eoin M KENNY and Mark T KEANE. « Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ANN-CBR twins for XAI ». In *Twenty-Eighth International Joint Conferences on Artifical Intelligence (IJCAI), Macao, 10-16 August 2019*, pages 2708–2715, 2019. (document), 1.2.2, 1.14, 1.2.2

[KK21] Ioannis KAKOGEORGIOU and Konstantinos KARANTZALOS. « Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing ». *International Journal of Applied Earth Observation and Geoinformation*, 103:102520, 2021.

[KKK16] Been KIM, Rajiv KHANNA, and Oluwasanmi O KOYEJO. « Examples are not enough, learn to criticize! criticism for interpretability ». *Advances in neural information processing systems*, 29, 2016. 1.2, 1.2.2, 1.3

[KKS07] Ivo KONDAPANENI, Pavel KORDÍK, and Pavel SLAVÍK. « Visualization techniques utilizing the sensitivity analysis of models ». In *2007 Winter Simulation Conference*, pages 730–737. IEEE, 2007. 1.2, 1.2.2, 1.2.2, 1.3

[KRS14] Been KIM, Cynthia RUDIN, and Julie A SHAH. « The bayesian case model: A generative approach for case-based reasoning and prototype classification ». *Advances in neural information processing systems*, 27, 2014. 1.2.2

[KSB99] R KRISHNAN, G SIVAKUMAR, and P BHATTACHARYA. « Extracting decision trees from trained neural networks ». *Pattern recognition*, 32(12), 1999. 1.2.1

[KSH12] Alex KRIZHEVSKY, Ilya SUTSKEVER, and Geoffrey E HINTON. « Imagenet classification with deep convolutional neural networks ». *Advances in neural information processing systems*, 25, 2012. 1

[KTKA20] Kentaro KANAMORI, Takuya TAKAGI, Ken KOBAYASHI, and Hiroki ARIMURA. « DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. ». In *IJCAI*, pages 2855–2862, 2020. 1.2, 1.2.2

[KWG+18] Been KIM, Martin WATTENBERG, Justin GILMER, Carrie CAI, James WEXLER, Fernanda VIEGAS, and OTHERS. « Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav) ». In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 1.2, 1.3

[LBBH98] Yann LECUN, Léon BOTTOU, Yoshua BENGIO, and Patrick HAFFNER. « Gradient-based learning applied to document recognition ». *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 33

[LBH15] Yann LECUN, Yoshua BENGIO, and Geoffrey HINTON. « Deep learning ». *nature*, 521(7553):436–444, 2015. 1

[LBJ16] Tao LEI, Regina BARZILAY, and Tommi S. JAAKKOLA. « Rationalizing Neural Predictions ». In Jian SU, Xavier CARRERAS, and Kevin DUH, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 107–117. The Association for Computational Linguistics, 2016. 1.2, 1.3, 2.3, 5.2.2

[LBL16] Himabindu LAKKARAJU, Stephen H BACH, and Jure LESKOVEC. « Interpretable decision sets: A joint framework for description and prediction ». In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684, 2016. 1.2.1, 14

[LC01] Stan LIPOVETSKY and Michael CONKLIN. « Analysis of regression in game theory approach ». *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001. 1.2.2

[LCH+19] Isaac LAGE, Emily CHEN, Jeffrey HE, Menaka NARAYANAN, Been KIM, Sam GERSHMAN, and Finale DOSHI-VELEZ. « An evaluation of the human-interpretability of explanation ». *arXiv preprint arXiv:1902.00006*, 2019. 2.3, 5.2.2

[LCLS17] Yanpei LIU, Xinyun CHEN, Chang LIU, and Dawn SONG. « Delving into Transferable Adversarial Examples and Black-box Attacks ». In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1.1.2

[LDB+12] Oscar LUACES, Jorge DÍEZ, José BARRANQUERO, Juan José del COZ, and Antonio BAHAMONDE. « Binary relevance efficacy for multilabel classification ». *Progress in Artificial Intelligence*, 1(4):303–313, 2012. 1

[LEC+20a] Scott M LUNDBERG, Gabriel ERION, Hugh CHEN, Alex DEGRAVE, Jordan M PRUTKIN, Bala NAIR, Ronit KATZ, Jonathan HIMMELFARB, Nisha BANSAL, and Su-In LEE. « From local explanations to global understanding with explainable AI for trees ». *Nature machine intelligence*, 2(1):56–67, 2020. 1.2, 1.3

[LEC+20b] Scott M. LUNDBERG, Gabriel G. ERION, Hugh CHEN, Alex J. DEGRAVE, Jordan M. PRUTKIN, Bala NAIR, Ronit KATZ, Jonathan HIMMELFARB, Nisha BANSAL, and Su-In LEE. « From local explanations to global understanding with explainable AI for trees ». *Nat. Mach. Intell.*, 2(1):56–67, 2020. 1.2.2

[LEL18] Scott M LUNDBERG, Gabriel G ERION, and Su-In LEE. « Consistent individualized feature attribution for tree ensembles ». *arXiv preprint arXiv:1802.03888*, 2018. 1.2.2

[Lib05] Paolo LIBERATORE. « Redundancy in logic I: CNF propositional formulae ». *Artificial Intelligence*, 163(2):203–232, 2005. 3.1, 3.3.4

[Lip18] Zachary C LIPTON. « The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. ». *Queue*, 16(3):31–57, 2018. 1, 1.1, 1.2.2

[LK21] Arnaud Van LOOVEREN and Janis KLAISE. « Interpretable counterfactual explanations guided by prototypes ». In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–665. Springer, 2021. 1.2, 1.2.2

[LKB+17] Geert LITJENS, Thijs KOOI, Babak Ehteshami BEJNORDI, Arnaud Arindra Adiyoso SETIO, Francesco CIOMPI, Mohsen GHAFOORIAN, Jeroen Awm VAN DER LAAK, Bram VAN GINNEKEN, and Clara I SÁNCHEZ. « A survey on deep learning in medical image analysis ». *Medical image analysis*, 42:60–88, 2017. 1.1

[LKCL19] Himabindu LAKKARAJU, Ece KAMAR, Rich CARUANA, and Jure LESKOVEC. « Faithful and customizable explanations of black box models ». In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138, 2019. 1.2.2, 2.3

[LKLH19] Shusen LIU, Bhavya KAILKHURA, Donald LOVELAND, and Yong HAN. « Generative counterfactual introspection for explainable deep learning ». In *2019 IEEE Global*

Conference on Signal and Information Processing (GlobalSIP), pages 1–5. IEEE, 2019. 1.2.2

[LL17] Scott M LUNDBERG and Su-In LEE. « A unified approach to interpreting model predictions ». *Advances in neural information processing systems*, 30, 2017. 1.2, 1.2, 1.2.2, 1.2.2, 1.2.2, 1.2.3, 1.3, 2.1, 3, 33, 5, 5.1.1, 5.3, 6.1, 6.5

[LLCR18] Oscar LI, Hao LIU, Chaofan CHEN, and Cynthia RUDIN. « Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions ». In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1.2.2

[LLM+19] Thibault LAUGEL, Marie-Jeanne LESOT, Christophe MARSALA, X. RENARD, and Marcin DETYNIECKI. « The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations ». In *IJCAI*, 2019. 1.2.2

[LLS+17] Bin LIANG, Hongcheng LI, Miaoqiang SU, Pan BIAN, Xirong LI, and Wenchang SHI. « Deep text classification can be fooled ». *arXiv preprint arXiv:1704.08006*, 2017. 1.1.2

[LNPT18] Francesco LEOFANTE, Nina NARODYTSKA, Luca PULINA, and Armando TACCHELLA. « Automated verification of neural networks: Advances, challenges and perspectives ». *arXiv preprint arXiv:1805.09938*, 2018. 2.2.2

[LO03] Tao LI and Mitsunori OGIHARA. « Detecting emotion in music ». 2003. 1.2

[LOL+18] Bruno LEPRI, Nuria OLIVER, Emmanuel LETOUZÉ, Alex PENTLAND, and Patrick VINCK. « Fair, transparent, and accountable algorithmic decision-making processes ». *Philosophy & Technology*, 31(4):611–627, 2018. 1.1, 1.1.2

[LPK20] Pantelis LINARDATOS, Vasilis PAPASTEFANOPOULOS, and Sotiris KOTSIANTIS. « Explainable ai: A review of machine learning interpretability methods ». *Entropy*, 23(1):18, 2020. 1.2.3, 5.1.1

[LPK21] Pantelis LINARDATOS, Vasilis PAPASTEFANOPOULOS, and Sotiris KOTSIANTIS. « Explainable AI: A Review of Machine Learning Interpretability Methods ». *Entropy*, 23(1), 2021. 5.1.1, 6.2.1

[LPMMS16] Mark H LIFFITON, Alessandro PREVITI, Ammar MALIK, and Joao MARQUES-SILVA. « Fast, flexible MUS enumeration ». *Constraints*, 21(2):223–250, 2016. 3.1, 3.3.1

[LPNC+17] David LOPEZ-PAZ, Robert NISHIHARA, Soumith CHINTALA, Bernhard SCHOLKOPF, and Léon BOTTOU. « Discovering causal signals in images ». In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6979–6987, 2017. 1.2.2

[LR76] Hyafil LAURENT and Ronald L RIVEST. « Constructing optimal binary decision trees is NP-complete ». *Information processing letters*, 5(1):15–17, 1976. 1.2.1

[LRBM08] Marie-Jeanne LESOT, Maria RIFQI, and Bernadette BOUCHON-MEUNIER. Fuzzy prototypes: From a cognitive view to a machine learning principle. In *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, pages 431–452. Springer, 2008. 1.2.2

[LRL+18] Thibault LAUGEL, X. RENARD, Marie-Jeanne LESOT, Christophe MARSALA, and Marcin DETYNIECKI. « Defining Locality for Surrogates in Post-hoc Interpretablity ». *ArXiv*, abs/1806.07498, 2018. 1.2.2

[LRMM15] Benjamin LETHAM, Cynthia RUDIN, Tyler H MCCORMICK, and David MADIGAN. « Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model ». *The Annals of Applied Statistics*, 9(3):1350–1371, 2015. 1.1, 1.3

[LS08] Mark H LIFFITON and Karem A SAKALLAH. « Algorithms for computing minimal unsatisfiable subsets of constraints ». *Journal of Automated Reasoning*, 40(1):1–33, 2008. 3.1, 3.3.1, 3.3.4, 3.3.4, 3.3.4, 5.2, 6.6

[LS09] Mark H LIFFITON and Karem A SAKALLAH. « Generalizing core-guided Max-SAT ». In *International Conference on Theory and Applications of Satisfiability Testing*, pages 481–494. Springer, 2009. 3.3.4

[Mar91] Pierre MARQUIS. « Extending abduction from propositional to first-order logic ». In *International Workshop on Fundamentals of Artificial Intelligence Research*, pages 141–155. Springer, 1991. 2.2.2

[MB18] D Douglas MILLER and Eric W BROWN. « Artificial intelligence in medical practice: the question to the answer? ». *The American journal of medicine*, 131(2):129–133, 2018. 1.1

[MCS+21] Stefano MELACCI, Gabriele CIRAVEGNA, Angelo SOTGIU, Ambra DEMONTIS, Battista BIGGIO, Marco GORI, and Fabio ROLI. « Domain Knowledge Alleviates Adversarial Attacks in Multi-Label Classifiers ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[MCV+18] Rory MCGRATH, Luca COSTABELLO, Chan Le VAN, Paul SWEENEY, Farbod KAMIAB, Zhao SHEN, and Freddy LÉCUÉ. « Interpretable Credit Application Predictions With Counterfactual Explanations ». *CoRR*, abs/1811.05245, 2018. 3.3.3

[MDFF16] Seyed-Mohsen MOOSAVI-DEZFOOLI, Alhussein FAWZI, and Pascal FROSSARD. « Deepfool: a simple and accurate method to fool deep neural networks ». In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1, 1.1.2

[MDFFF17] Seyed-Mohsen MOOSAVI-DEZFOOLI, Alhussein FAWZI, Omar FAWZI, and Pascal FROSSARD. « Universal adversarial perturbations ». In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 1.1.2

[Mel21] Marco MELIS. « Explaining Vulnerability of Machine Learning to Adversarial Attacks ». 2021. (document), 1.4

[MG20] Samaneh MAHDAVIFAR and Ali A GHORBANI. « DeNNeS: deep embedded neural network expert system for detecting cyber attacks ». *Neural Computing and Applications*, 32(18):14753–14780, 2020. 1.2, 1.3

[Mil19] Tim MILLER. « Explanation in artificial intelligence: Insights from the social sciences ». *Artificial intelligence*, 267:1–38, 2019. 1, 1.1, 1.1.1

[Mil21] Tim MILLER. « Contrastive explanation: A structural-model approach ». *The Knowledge Engineering Review*, 36, 2021. 1.2.2

[MIPMS16] Carlos MENCÍA, Alexey IGNATIEV, Alessandro PREVITI, and Joao MARQUES-SILVA. « MCS extraction with sublinear oracle queries ». In *International Conference on Theory and Applications of Satisfiability Testing*, pages 342–360. Springer, 2016. 3.1, 3.3.1, 3.3.4

[MLA+05] Maher MNEIMNEH, Inês LYNCE, Zaher ANDRAUS, João MARQUES-SILVA, and Karem SAKALLAH. « A branch-and-bound algorithm for extracting smallest minimal unsatisfiable formulas ». In *International Conference on Theory and Applications of Satisfiability Testing*, pages 467–474. Springer, 2005. 3.3.4

[MLB+17] Grégoire MONTAVON, Sebastian LAPUSCHKIN, Alexander BINDER, Wojciech SAMEK, and Klaus-Robert MÜLLER. « Explaining nonlinear classification decisions with deep taylor decomposition ». *Pattern recognition*, 65:211–222, 2017. 1.2.2, 1.2.3

[Mol22] Christoph MOLNAR. *Interpretable Machine Learning*. 2 edition, 2022. 1, 1.2, 1.2.1, 1.2.2, 1.2.2, 1.2.2

[MPMS15] Carlos MENCÍA, Alessandro PREVITI, and Joao MARQUES-SILVA. « Literal-based MCS extraction ». In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 3.1, 3.3.1, 3.3.4, 3.3.4

[MSGC+20] Joao MARQUES-SILVA, Thomas GERSPACHER, Martin COOPER, Alexey IGNATIEV, and Nina NARODYTSKA. « Explaining naive bayes and other linear classifiers with polynomial time and delay ». *Advances in Neural Information Processing Systems*, 33:20590–20600, 2020. 2.2, 3.3.4

[MSHJ+13] Joao MARQUES-SILVA, Federico HERAS, Mikolás JANOTA, Alessandro PREVITI, and Anton BELOV. « On computing minimal correction subsets ». In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013. 3.3.4, 3.3.4

[MST20] Ramaravind K MOTHILAL, Amit SHARMA, and Chenhao TAN. « Explaining machine learning classifiers through diverse counterfactual explanations ». In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020. 1.2, 1.2.2, 1.2.2, 3.3.3

[MT07] D Michael MILLER and Mitchell A THORNTON. « Multiple valued logic: Concepts and representations ». *Synthesis lectures on digital circuits and systems*, 2(1):1–127, 2007.

[MTR16] Sameer Singh MARCO TULIO RIBEIRO. « Local interpretable model-agnostic explanations (LIME): An introduction », Aug 2016. (document), 1.10

[MTS19] Divyat MAHAJAN, Chenhao TAN, and Amit SHARMA. « Preserving causal constraints in counterfactual explanations for machine learning classifiers ». *arXiv preprint arXiv:1912.03277*, 2019. 1.2, 1.2.2, 1.2.2

[MVED17] Dmitry M MALIOUTOV, Kush R VARSHNEY, Amin EMAD, and Sanjeeb DASH. Learning interpretable classification rules with boolean compressed sensing. In *Transparent Data Mining for Big and Small Data*, pages 95–121. Springer, 2017. 1.2.1

[MWM18] Daniel L. MARINO, Chathurika S. WICKRAMASINGHE, and Milos MANIC. « An Adversarial Approach for Explainable AI in Intrusion Detection Systems ». In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 3237–3243, 2018. 1.2, 1.3

[MZR21] Sina MOHSENI, Niloofar ZAREI, and Eric D RAGAN. « A multidisciplinary survey and framework for design and evaluation of explainable AI systems ». *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 11(3-4):1–45, 2021. 2.3

[NAC02] Haydemar NÚÑEZ, Cecilio ANGULO, and Andreu CATALÀ. « Rule extraction from support vector machines. ». In *Esann*, pages 107–112, 2002. 1.2.1

[NBMS18] Nina NARODYTSKA, Nikolaj BJØRNER, Maria Cristina MARINESCU, and Mooly SAGIV. « Core-guided minimal correction set and core enumeration ». In *IJCAI International Joint Conference on Artificial Intelligence: Stockholm, 13-19 July 2018*, pages 1353–1361. IJCAI, 2018. 3.1, 3.3.1, 3.3.4

[NDY+16] Anh NGUYEN, Alexey DOSOVITSKIY, Jason YOSINSKI, Thomas BROX, and Jeff CLUNE. « Synthesizing the preferred inputs for neurons in neural networks via deep generator networks ». *Advances in neural information processing systems*, 29, 2016. 1.2.2

[NGC+20] Woo-Jeoung NAM, Shir GUR, Jaesik CHOI, Lior WOLF, and Seong-Whan LEE. « Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks ». In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2501–2508, 2020.

[NHWF21] Anna NGUYEN, Daniel HAGENMAYER, Tobias WELLER, and Michael FÄRBER. « Explaining Convolutional Neural Networks by Tagging Filters », 2021. 5.1.1, 6.1

[Nig16] Priyanka NIGAM. « Applying deep learning to ICD-9 multi-label classification from medical records ». Technical report, Technical report, Stanford University, 2016. 4.2

[NK17] Nina NARODYTSKA and Shiva Prasad KASIVISWANATHAN. « Simple Black-Box Adversarial Attacks on Deep Neural Networks. ». In *CVPR Workshops*, volume 2, Page 2, 2017. 1.1.2

[NKR+18] Nina NARODYTSKA, Shiva KASIVISWANATHAN, Leonid RYZHYK, Mooly SAGIV, and Toby WALSH. « Verifying properties of binarized deep neural networks ». In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2.2.1, 2.2.2, 3.2.1, 4.4.1

[NRS13] Alexander NADEL, Vadim RYVCHIN, and Ofer STRICHMAN. « Efficient MUS extraction with resolution ». In *2013 Formal Methods in Computer-Aided Design*, pages 197–200. IEEE, 2013. 3.3.4

[NRS14] Alexander NADEL, Vadim RYVCHIN, and Ofer STRICHMAN. « Accelerated deletion-based extraction of minimal unsatisfiable cores ». *Journal on Satisfiability, Boolean Modeling and Computation*, 9(1):27–51, 2014. 3.3.4

[NYC15] Anh NGUYEN, Jason YOSINSKI, and Jeff CLUNE. « Deep neural networks are easily fooled: High confidence predictions for unrecognizable images ». In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 1

[NZ17] Robin NIX and Jian ZHANG. « Classification of Android apps and malware using deep neural networks ». In *2017 International joint conference on neural networks (IJCNN)*, pages 1871–1878. IEEE, 2017. 1.1

[OD15] Umut OZTOK and Adnan DARWICHE. « A top-down compiler for sentential decision diagrams ». In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 2.2.1

[OMS17] Chris OLAH, Alexander MORDVINTSEV, and Ludwig SCHUBERT. « Feature visualization ». *Distill*, 2(11):e7, 2017. 1.2.2

[OSC07] Olga OHRIMENKO, Peter J STUCKEY, and Michael CODISH. « Propagation= lazy clause generation ». In *International Conference on Principles and Practice of Constraint Programming*, pages 544–558. Springer, 2007. 3.1

[PGA08] Kemal POLAT, Salih GÜNEŞ, and Ahmet ARSLAN. « A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine ». *Expert systems with applications*, 34(1):482–487, 2008. 1.1

[PGG+19] Dino PEDRESCHI, Fosca GIANNOTTI, Riccardo GUIDOTTI, Anna MONREALE, Salvatore RUGGIERI, and Franco TURINI. « Meaningful explanations of black box AI decision systems ». In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9780–9784, 2019. 1.2.1

[PGMP19] Cecilia PANIGUTTI, Riccardo GUIDOTTI, Anna MONREALE, and Dino PEDRESCHI. « Explaining multi-label black-box classifiers for health applications ». In *International Workshop on Health Intelligence*, pages 97–110. Springer, 2019. 4.1, 6.1

[PHL+17] Timo PEKKALA, Anette HALL, Jyrki LÖTJÖNEN, Jussi MATTILA, Hilkka SOININEN, Tiia NGANDU, Tiina LAATIKAINEN, Miia KIVIPELTO, and Alina SOLOMON. « Development of a late-life dementia prediction index with supervised machine learning in the population-based CAIDE study ». *Journal of Alzheimer's Disease*, 55(3):1055–1067, 2017. 1.1

[PLPN19] Badri N PATRO, Mayank LUNAYACH, Shivansh PATEL, and Vinay P NAMBOODIRI. « U-cam: Visual explanation using uncertainty based class activation maps ». In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7444–7453, 2019. 5.1.1

[PMJ+16] Nicolas PAPERNOT, Patrick MCDANIEL, Somesh JHA, Matt FREDRIKSON, Z Berkay CELIK, and Ananthram SWAMI. « The limitations of deep learning in adversarial settings ». In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 1.1.1

[PMJMS17] Alessandro PREVITI, Carlos MENCÍA, Matti JÄRVISALO, and Joao MARQUES-SILVA. « Improving MCS enumeration via caching ». In *International Conference on Theory and Applications of Satisfiability Testing*, pages 184–194. Springer, 2017. 3.3.4

[PMJMS18] Alessandro PREVITI, Carlos MENCÍA, Matti JÄRVISALO, and Joao MARQUES-SILVA. « Premise set caching for enumerating minimal correction subsets ». In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3.1, 3.3.1, 3.3.4

[Qui87a] J Ross QUINLAN. « Generating production rules from decision trees. ». In *ijcai*, volume 87, pages 304–307. Citeseer, 1987.

[Qui87b] J. Ross QUINLAN. « Simplifying decision trees ». *International journal of man-machine studies*, 27(3):221–234, 1987. 1.2.1

[Rei87] Raymond REITER. « A theory of diagnosis from first principles ». *Artificial intelligence*, 32(1):57–95, 1987. 3.1, 2.2, 2.2.2, 3.3.4, 5.1.1

[RPF+21] Thomas ROJAT, Raphaël PUGET, David FILLIAT, Javier DEL SER, Rodolphe GELIN, and Natalia DÍAZ-RODRÍGUEZ. « Explainable artificial intelligence (xai) on timeseries data: A survey ». *arXiv preprint arXiv:2104.00950*, 2021. 1.2.2

[RPH08] Jesse READ, Bernhard PFAHRINGER, and Geoff HOLMES. « Multi-label classification using ensembles of pruned sets ». In *2008 eighth IEEE international conference on data mining*, pages 995–1000. IEEE, 2008. 1

[RPHF09] Jesse READ, Bernhard PFAHRINGER, Geoff HOLMES, and Eibe FRANK. « Classifier chains for multi-label classification ». In *Joint European conference on machine learning and knowledge discovery in databases*, pages 254–269. Springer, 2009. 1, 3

[RPHF11] Jesse READ, Bernhard PFAHRINGER, Geoff HOLMES, and Eibe FRANK. « Classifier chains for multi-label classification ». *Machine learning*, 85(3):333–359, 2011. 3

[RPR+10] Cynthia RUDIN, Rebecca J PASSONNEAU, Axinia RADEVA, Haimonti DUTTA, Steve IEROME, and Delfina ISAAC. « A process for predicting manhole events in Manhattan ». *Machine Learning*, 80(1):1–31, 2010. 1.1

[RRK90] Dennis W RUCK, Steven K ROGERS, and Matthew KABRISKY. « Feature selection using a multilayer perceptron ». *Journal of Neural Network Computing*, 2(2):40–48, 1990. 1.2.2

[RSE16] Loukmen REGAINIA, Sébastien SALVA, and Cedric ECUHCURS. « A classification methodology for security patterns to help fix software weaknesses ». In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE, 2016. 1.1

[RSG16] Marco Tulio RIBEIRO, Sameer SINGH, and Carlos GUESTRIN. « " Why should i trust you?" Explaining the predictions of any classifier ». In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. (document), 1.2, 1.2.1, 1.2.1, 1.2, 1.2.2, 1.2.2, 1.9, 1.2.3, 1.3, 2.1, 3, 3.2.2, 5, 5.1.1, 6.1, 6.5

[RSG18] Marco Tulio RIBEIRO, Sameer SINGH, and Carlos GUESTRIN. « Anchors: High-precision model-agnostic explanations ». In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 1.2, 1.2, 1.2.2, 1.2.2, 1.3, 2.1, 3.2.2, 5

[RU18] Cynthia RUDIN and Berk USTUN. « Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice ». *Interfaces*, 48(5):449–466, 2018. 1.1

[Rud19] Cynthia RUDIN. « Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead ». *Nature Machine Intelligence*, 1(5):206–215, 2019. 1.1

[Rus19] Chris RUSSELL. « Efficient search for diverse coherent explanations ». In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28, 2019. 1.2, 1.2.2, 1.2.2

[Rym94] Ron RYMON. « An se-tree-based prime implicant generation algorithm ». *Annals of Mathematics and Artificial Intelligence*, 11(1):351–365, 1994. 2.2.2, 5.1.1

[SB75] Edward H SHORTLIFFE and Bruce G BUCHANAN. « A model of inexact reasoning in medicine ». *Mathematical biosciences*, 23(3-4):351–379, 1975. 1.1.1

[SB21] Kushal SINGLA and Subham BISWAS. « Machine learning explanability method for the multi-label classification model ». In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 337–340. IEEE, 2021. 4.1, 6.1

[SCD+17] Ramprasaath R SELVARAJU, Michael COGSWELL, Abhishek DAS, Ramakrishna VEDANTAM, Devi PARIKH, and Dhruv BATRA. « Grad-cam: Visual explanations from deep networks via gradient-based localization ». In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. (document), 1.2.2, 1.8, 1.2.3, 5.1.1

[SCD18a] Andy SHIH, Arthur CHOI, and Adnan DARWICHE. « Formal verification of Bayesian network classifiers ». In *International Conference on Probabilistic Graphical Models*, pages 427–438. PMLR, 2018. 2.2.1

[SCD18b] Andy SHIH, Arthur CHOI, and Adnan DARWICHE. « A symbolic approach to explaining bayesian network classifiers ». *arXiv preprint arXiv:1805.03364*, 2018. 1.2, 1.2, 1.2.2, 1.3, 2.2, 2.2.1, 2.2.2, 8, 3, 3.2.1, 3.2.1, 3.2.1, 5, 5.1.1

[SCD19] Andy SHIH, Arthur CHOI, and Adnan DARWICHE. « Compiling Bayesian Network Classifiers into Decision Graphs ». In *Proceedings of the AAAI-19*, volume 33, pages 7966–7974, 2019. 2.2, 2.2.1, 3, 3.2.1, 3.2.1, 4.4.1, 6.6

[SDBR14] Jost Tobias SPRINGENBERG, Alexey DOSOVITSKIY, Thomas BROX, and Martin RIEDMILLER. « Striving for simplicity: The all convolutional net ». *arXiv preprint arXiv:1412.6806*, 2014. (document), 1.2, 1.2.2, 1.8, 1.2.3, 1.3

[SDBR15] JT SPRINGENBERG, A DOSOVITSKIY, T BROX, and M RIEDMILLER. « Striving for simplicity: The all convolutional net. In arxiv: cs ». *arXiv preprint arXiv:1412.6806*, 2015. 5.1.1

[SDC19] Andy SHIH, Adnan DARWICHE, and Arthur CHOI. « Verifying binarized neural networks by Angluin-style learning ». In *International Conference on Theory and Applications of Satisfiability Testing*, pages 354–370. Springer, 2019. 2.2.2

[SGK17] Avanti SHRIKUMAR, Peyton GREENSIDE, and Anshul KUNDAJE. « Learning important features through propagating activation differences ». In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 1.2.2, 5.1.1

[SGM+21] Mattia SETZU, Riccardo GUIDOTTI, Anna MONREALE, Franco TURINI, Dino PEDRESCHI, and Fosca GIANNOTTI. « Glocalx-from local to global explanations of black box AI models ». *Artificial Intelligence*, 294:103457, 2021. 1.2, 1.3

[SGZS21] Maximilian SCHLEICH, Zixuan GENG, Yihong ZHANG, and Dan SUCIU. « GeCo: quality counterfactual explanations in real time ». *arXiv preprint arXiv:2101.01292*, 2021. (document), 1.13

[SH+08] Sajjad SIDDIQI, Jinbo HUANG, and OTHERS. « Probabilistic sequential diagnosis by compilation ». 2008. 2.2.1

[Sha53] LS SHAPLEY. « QUOTA SOLUTIONS OP n-PERSON GAMES1 ». *Edited by Emil Artin and Marston Morse*, Page 343, 1953. 1.2.2, 2.1

[Sha89] Murray SHANAHAN. « Prediction is Deduction but Explanation is Abduction. ». In *IJCAI*, volume 89, pages 1055–1060, 1989. 2.2.2

[SHG19] 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. 1.2.2

[SHJ+20] Dylan SLACK, Sophie HILGARD, Emily JIA, Sameer SINGH, and Himabindu LAKKARAJU. « Fooling lime and shap: Adversarial attacks on post hoc explanation methods ». In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020. 1.2.2, 2.1

[Sin05] Carsten SINZ. « Towards an optimal CNF encoding of boolean cardinality constraints ». In *International conference on principles and practice of constraint programming*, pages 827–831. Springer, 2005. 3.2.1, 3.2.1

[ŠK14] Erik ŠTRUMBELJ and Igor KONONENKO. « Explaining prediction models and individual predictions with feature contributions ». *Knowledge and information systems*, 41(3):647–665, 2014. 1.2.2

[SKG09] Yang SONG, Aleksander KOŁCZ, and C Lee GILES. « Better Naive Bayes classification for high-precision spam detection ». *Software: Practice and Experience*, 39(11):1003–1024, 2009. 1.1

[SLZS22] Weidi SUN, Yuteng LU, Xiyue ZHANG, and Meng SUN. « DeepGlobal: A framework for global robustness verification of feedforward neural networks ». *Journal of Systems Architecture*, Page 102582, 2022. 1.2, 1.3

[SMA+18] Keiichi SHIMA, Daisuke MIYAMOTO, Hiroshi ABE, Tomohiro ISHIHARA, Kazuya OKADA, Yuji SEKIYA, Hirochika ASAI, and Yusuke DOI. « Classification of URL bitstreams using bag of bytes ». In *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 1–5. IEEE, 2018. 1.1

[SNPS18] Nathan SHONE, Tran Nguyen NGOC, Vu Dinh PHAI, and Qi SHI. « A deep learning approach to network intrusion detection ». *IEEE transactions on emerging topics in computational intelligence*, 2(1):41–50, 2018. 1.1

[SPG+19] Shriansh SRIVASTAVA, J PRIYADARSHINI, Sachin GOPAL, Sanchay GUPTA, and Har Shobhit DAYAL. Optical character recognition on bank cheques using 2D convolution neural network. In *Applications of Artificial Intelligence Techniques in Engineering*, pages 589–596. Springer, 2019. 1.1

[SRA+08] Andrea SALTELLI, Marco RATTO, Terry ANDRES, Francesca CAMPOLONGO, Jessica CARIBONI, Debora GATELLI, Michaela SAISANA, and Stefano TARANTOLA. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008. 1.2, 1.3

[SSDC20] Weijia SHI, Andy SHIH, Adnan DARWICHE, and Arthur CHOI. « On Tractable Representations of Binary Neural Networks ». In Diego CALVANESE, Esra ERDEM, and Michael THIELSCHER, editors, *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pages 882–892, 2020. 2.2, 2.2.1, 3.3.4, 4.4.1, 6.6

[SSF14] Paulo SHAKARIAN, Gerardo I SIMARI, and Marcelo A FALAPPA. « Belief revision in structured probabilistic argumentation ». In *International Symposium on Foundations of Information and Knowledge Systems*, pages 324–343. Springer, 2014. 1.2, 1.3

[SSK+12] Hitesh SAJNANI, Vaibhav SAINI, Kusum KUMAR, Eugenia GABRIELOVA, Pramit CHOUDARY, and Cristina LOPES. « Classifying yelp reviews into relevant categories », 2012.

[STY17] Mukund SUNDARARAJAN, Ankur TALY, and Qiqi YAN. « Axiomatic attribution for deep networks ». In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 1.2.2, 1.2.3, 5.1.1

[SVZ13a] Karen SIMONYAN, Andrea VEDALDI, and Andrew ZISSERMAN. « Deep inside convolutional networks: Visualising image classification models and saliency maps ». *arXiv preprint arXiv:1312.6034*, 2013. 1.2.2, 1.2.2, 5.1.1

[SVZ13b] Karen SIMONYAN, Andrea VEDALDI, and Andrew ZISSERMAN. « Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps », 2013. 5.1.1, 6.1

[SZM19] Julia STROUT, Ye ZHANG, and Raymond J MOONEY. « Do human rationales improve machine explanations? ». *arXiv preprint arXiv:1905.13714*, 2019. 2.3, 5.2.2

[SZS+13] Christian SZEGEDY, Wojciech ZAREMBA, Ilya SUTSKEVER, Joan BRUNA, Dumitru ERHAN, Ian GOODFELLOW, and Rob FERGUS. « Intriguing properties of neural networks ». *arXiv preprint arXiv:1312.6199*, 2013. 1

[Tab19] Karim TABIA. « Towards explainable multi-label classification ». In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1088–1095. IEEE, 2019. 4.4.2

[TBH+18] Richard TOMSETT, Dave BRAINES, Dan HARBORNE, Alun PREECE, and Supriyo CHAKRABORTY. « Interpretable to whom? A role-based model for analyzing interpretable machine learning systems ». *arXiv preprint arXiv:1806.07552*, 2018. 2.3

[TK07] Grigorios TSOUMAKAS and Ioannis KATAKIS. « Multi-label classification: An overview ». *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007. 1

[TKSR16] Jayaraman J THIAGARAJAN, Bhavya KAILKHURA, Prasanna SATTIGERI, and Karthikeyan Natesan RAMAMURTHY. « Treeview: Peeking into deep neural networks via feature-space partitioning ». *arXiv preprint arXiv:1611.07429*, 2016. 1.2.2

[TKV08] Grigorios TSOUMAKAS, Ioannis KATAKIS, and Ioannis VLAHAVAS. « Effective and efficient multilabel classification in domains with large number of labels ». In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, volume 21, pages 53–59, 2008. 3

[TKV09] Grigorios TSOUMAKAS, Ioannis KATAKIS, and Ioannis VLAHAVAS. « Mining multi-label data ». *Data mining and knowledge discovery handbook*, pages 667–685, 2009. 1.2

[TPJR18] Yuchi TIAN, Kexin PEI, Suman JANA, and Baishakhi RAY. « Deeptest: Automated testing of deep-neural-network-driven autonomous cars ». In *Proceedings of the 40th international conference on software engineering*, pages 303–314, 2018. 1.1

[Tse83] Grigori S TSEITIN. On the complexity of derivation in propositional calculus. In *Automation of reasoning*, pages 466–483. Springer, 1983. 3.2.1, 3.4.1

[TSHW20] Sarah TAN, Matvey SOLOVIEV, Giles HOOKER, and Martin T WELLS. « Tree space prototypes: Another look at making tree ensembles interpretable ». In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, pages 23–34, 2020. 1.2.1

[TT04] Gianluca TORTA and Pietro TORASSO. « The role of OBDDs in controlling the complexity of model based diagnosis ». In *Proc. of 15th International Workshop on Principles of Diagnosis (DX-04)*. Citeseer, 2004. 2.2.1

[TV07] Grigorios TSOUMAKAS and Ioannis VLAHAVAS. « Random k-labelsets: An ensemble method for multilabel classification ». In *European conference on machine learning*, pages 406–417. Springer, 2007. 1, 3

[TvdH13] Nikolaj TOLLENAAR and Peter GM van der HEIJDEN. « Which method predicts recidivism best?: a comparison of statistical, machine learning and data mining predictive models ». *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 176(2):565–584, 2013. 1.1

[TVRFOG18] Ruben TOLOSANA, Ruben VERA-RODRIGUEZ, Julian FIERREZ, and Javier ORTEGA-GARCIA. « Exploring recurrent neural networks for on-line handwritten signature biometrics ». *Ieee Access*, 6:5128–5138, 2018. 1.1

[TXT17] Vincent TJENG, Kai XIAO, and Russ TEDRAKE. « Evaluating robustness of neural networks with mixed integer programming ». *arXiv preprint arXiv:1711.07356*, 2017.

[TYRW14] Yaniv TAIGMAN, Ming YANG, Marc'Aurelio RANZATO, and Lior WOLF. « Deepface: Closing the gap to human-level performance in face verification ». In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 1.1, 1

[UR16] Berk USTUN and Cynthia RUDIN. « Supersparse linear integer models for optimized medical scoring systems ». *Machine Learning*, 102(3):349–391, 2016. 1.1, 1.3

[VDH20] Sahil VERMA, John DICKERSON, and Keegan HINES. « Counterfactual explanations for machine learning: A review ». *arXiv preprint arXiv:2010.10596*, 2020. 1.2

[vdWRvD+18] Jasper van der WAA, Marcel ROBEER, Jurriaan van DIGGELEN, Matthieu BRINKHUIS, and Mark NEERINCX. « Contrastive explanations with local foil trees ». *arXiv preprint arXiv:1806.07470*, 2018. 1.2

[VdWSNI+14] Stefan Van der WALT, Johannes L SCHÖNBERGER, Juan NUNEZ-IGLESIAS, François BOULOGNE, Joshua D WARNER, Neil YAGER, Emmanuelle GOUILLART, and Tony YU. « scikit-image: image processing in Python ». *PeerJ*, 2:e453, 2014. 1.1

[Vos16] W Gregory VOSS. « European union data privacy law reform: General data protection regulation, privacy shield, and the right to delisting ». *The Business Lawyer*, 72(1):221–234, 2016. 1.1.1

[Vre19] Mihnea Horia VREJOIU. « Neural Networks and Deep Learning in Cyber Security ». *Romanian Cyber Security Journal*, 1(1):69–86, 2019. 1.1

[Wal12] Joel WALMSLEY. Classical Cognitive Science and "Good Old Fashioned AI". In *Mind and Machine*, pages 30–64. Springer, 2012. 1

[WDGG19] Dennis WEI, Sanjeeb DASH, Tian GAO, and Oktay GUNLUK. « Generalized linear rule models ». In *International Conference on Machine Learning*, pages 6687–6696. PMLR, 2019. 1.1, 1.3

[WGH19] Lior WOLF, Tomer GALANTI, and Tamir HAZAN. « A formal approach to explainability ». In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 255–261, 2019. 2.2

[WH13] Siert WIERINGA and Keijo HELJANKO. « Asynchronous multi-core incremental SAT solving ». In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 139–153. Springer, 2013. 3.3.4

[WHP+18] Mike WU, Michael HUGHES, Sonali PARBHOO, Maurizio ZAZZI, Volker ROTH, and Finale DOSHI-VELEZ. « Beyond sparsity: Tree regularization of deep models for interpretability ». In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 1.2.2

[WKQ+08] Xindong WU, Vipin KUMAR, J Ross QUINLAN, Joydeep GHOSH, Qiang YANG, Hiroshi MOTODA, Geoffrey J MCLACHLAN, Angus NG, Bing LIU, S Yu PHILIP, and OTHERS. « Top 10 algorithms in data mining ». *Knowledge and information systems*, 14(1):1–37, 2008. 1.1

[WMR17] Sandra WACHTER, Brent MITTELSTADT, and Chris RUSSELL. « Counterfactual explanations without opening the black box: Automated decisions and the GDPR ». *Harv. JL & Tech.*, 31:841, 2017. 1.2, 1.2.2, 1.2.2, 3.3.3, 5

[WR15] Fulton WANG and Cynthia RUDIN. « Falling rule lists ». In *Artificial intelligence and statistics*, pages 1013–1022. PMLR, 2015. 5.2.2

[WRDV+17] Tong WANG, Cynthia RUDIN, Finale DOSHI-VELEZ, Yimin LIU, Erica KLAMPFL, and Perry MACNEILLE. « A bayesian framework for learning rule sets for interpretable

classification ». *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017. 1.2.1

[WRK+17] Stephen F WENG, Jenna REPS, Joe KAI, Jonathan M GARIBALDI, and Nadeem QURESHI. « Can machine-learning improve cardiovascular risk prediction using routine clinical data? ». *PloS one*, 12(4):e0174944, 2017. 1.1

[WXH+14] Yunchao WEI, Wei XIA, Junshi HUANG, Bingbing NI, Jian DONG, Yao ZHAO, and Shuicheng YAN. « CNN: Single-label to multi-label ». *arXiv preprint arXiv:1406.5726*, 2014. 4.2

[WYAL19] Danding WANG, Qian YANG, Ashraf ABDUL, and Brian Y LIM. « Designing theory-driven user-centric explainable AI ». In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–15, 2019. 1

[WZLW21] Zhuo WANG, Wei ZHANG, Ning LIU, and Jianyong WANG. « Scalable Rule-Based Representation Learning for Interpretable Classification ». *Advances in Neural Information Processing Systems*, 34, 2021.

[WZWY19] Ziyu WANG, Xiao ZENG, Jinzhao WU, and Guowu YANG. « A Method for Determining the Affine Equivalence of Boolean Functions ». *IEEE Access*, 7:156326–156337, 2019. 4.3.1

[YCN+15] Jason YOSINSKI, Jeff CLUNE, Anh NGUYEN, Thomas FUCHS, and Hod LIPSON. « Understanding neural networks through deep visualization ». *arXiv preprint arXiv:1506.06579*, 2015. 1.2.2

[YRS17] Hongyu YANG, Cynthia RUDIN, and Margo SELTZER. « Scalable Bayesian rule lists ». In *International conference on machine learning*, pages 3921–3930. PMLR, 2017. 2.3, 5.2.2

[Zad96] Lotfi A ZADEH. A note on prototype theory and fuzzy sets. In *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, pages 587–593. World Scientific, 1996. 1.2.2

[ZAG18] Daniel ZÜGNER, Amir AKBARNEJAD, and Stephan GÜNNEMANN. « Adversarial attacks on neural networks for graph data ». In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2847–2856, 2018. 1.1.2

[ZCAW17] Luisa M ZINTGRAF, Taco S COHEN, Tameem ADEL, and Max WELLING. « Visualizing deep neural network decisions: Prediction difference analysis ». *arXiv preprint arXiv:1702.04595*, 2017. 1.2.2

[ZF14] Matthew D ZEILER and Rob FERGUS. « Visualizing and understanding convolutional networks ». In *European conference on computer vision*, pages 818–833. Springer, 2014. 1.2, 1.2.2, 1.3, 5.1.1

[ZH21] Qingyuan ZHAO and Trevor HASTIE. « Causal interpretations of black-box models ». *Journal of Business & Economic Statistics*, 39(1):272–281, 2021. 2.1

[ZKL+16] Bolei ZHOU, Aditya KHOSLA, Agata LAPEDRIZA, Aude OLIVA, and Antonio TORRALBA. « Learning deep features for discriminative localization ». In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 1.2, 1.2.2, 1.2.3, 1.3

[ZVR+17] Muhammad Bilal ZAFAR, Isabel VALERA, Manuel RODRIGUEZ, Krishna GUMMADI, and Adrian WELLER. « From parity to preference-based notions of fairness in classification ». *Advances in Neural Information Processing Systems*, 30, 2017. 1.1.2, 1.2, 1.3

[ZVRG17] Muhammad Bilal ZAFAR, Isabel VALERA, Manuel Gomez ROGRIGUEZ, and Krishna P GUMMADI. « Fairness constraints: Mechanisms for fair classification ». In *Artificial intelligence and statistics*, pages 962–970. PMLR, 2017. 1.1.2

[ZYMW19] Quanshi ZHANG, Yu YANG, Haotian MA, and Ying Nian WU. « Interpreting cnns via decision trees ». In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6261–6270, 2019. 1.2.2

[ZZ05] Min-Ling ZHANG and Zhi-Hua ZHOU. « A k-nearest neighbor based algorithm for multi-label classification ». In *2005 IEEE international conference on granular computing*, volume 2, pages 718–721. IEEE, 2005. 2

Résumé

Cette thèse étudie une méthode d'explicabilité qui allie à la fois le caractère "agnostique" des méthodes numériques et qui propose des explications plus "rigoureuses" qui caractérisent les explications symboliques. Le but étant d'expliquer les prédictions des techniques de classification mono-étiquette et multi-étiquettes. Plusieurs contributions sont apportées dans cette thèse. Premièrement, nous avons travaillé sur le cas mono-étiquette. Nous avons proposé une approche qui va de l'encodage en représentation symbolique du modèle dont on souhaite expliquer les prédictions à la génération d'explication basée sur un oracle SAT. L'idée est de prendre un classifieur, avec une instance, et de produire une formule propositionnelle que nous utiliserons pour générer nos explications. L'inconsistance de cette formule permet d'expliquer les prédictions. Nous considérons les deux cas où nous pouvons avoir la représentation logique du modèle dans son ensemble ou une approximation basée sur un modèle de substitution. Nous nous intéressons à deux types complémentaires d'explications symboliques : les *raisons suffisantes* qui correspondent à un sous-ensemble minimal de l'entrée conduisant à une prédiction spécifique et les *contrefactuelles* qui correspondent à un sous-ensemble de l'entrée permettant de déterminer les modifications minimales à apporter pour obtenir une prédiction différente. Deuxièmement, nous avons proposé des propriétés à considérer afin de prioriser et sélectionner les explications en évaluant leur pertinence ainsi que celle des variables les composants. Par la suite, nous nous sommes intéressés à l'explication des prédictions multi-étiquettes. Nous avons proposé des explications multi-étiquettes à différents niveaux de granularité et étudié la combinaison d'explications mono-étiquette ainsi que les relations structurelles entre classes comme moyen de les générer. Enfin, nous nous sommes intéressés aux scores d'importance au niveau des caractéristiques pour déterminer dans quelle mesure chacune contribue à la sortie d'un modèle multi-étiquettes. Cette contribution examine deux possibilités différentes d'utiliser des méthodes existantes pour le cas mono-étiquette comme oracles ou d'utiliser des attributions de caractéristiques obtenues à partir d'explications symboliques. Afin d'évaluer la qualité des attributions de caractéristiques, nous étendons les propriétés de sensibilité, de stabilité des données au cas multi-étiquettes en plus d'une nouvelle propriété spécifique à la classification multi-étiquettes que nous appelons corrélation label-explication.

Mots-clés: IA explicable, Explications symboliques, Explications basées-score, Modèle-agnostique, Classification multi-étiquettes, Satisfiabilité booléenne, Attribution de caractéristique.

Abstract

This thesis studies the problem of explaining individual predictions of black-box machine learning models. This problem is addressed in both single and multi-label classification. Firstly, we introduce an explanation approach representing a combination of SAT solving and numerical measures to develop a model-agnostic method for providing both symbolic and score-based explanations. The idea is to take a single-label classifier, together with an instance, and produce a propositional formula that we will use to generate our explanations. We consider both cases where we can have the logical representation of the model as a whole or an approximation based on a surrogate model. In the second case, a crucial component of the proposed approach is to approximate the model with another (simpler) one that does admit a tractable logical representation to efficiently enumerate explanations. To comply with the original predictor, the selected surrogate model needs to ensure fidelity. Subsequently, this trained model is used to generate symbolic and numerical explanations. In a second time, we consider a SAT framework with the aim of using SAT solvers as the problem solving engine. Given an unsatisfiable formula corresponding to a negative prediction, modern SAT solvers are able to report the cores generating an inconsistency. In this contribution, we provide two complementary types of symbolic explanations of unsatisfiability called *sufficient reasons* and *counterfactuals* centered around Minimal Unsatisfiable Subsets (MUS) and Minimal Correction Subsets (MCS) respectively. Secondly, we have worked on defining measures of the quality of an explanation and of a variable contribution to properly assess how relevant they are as it becomes necessary to focus on those providing more insights. Next, we have worked on defining possible explanation mechanisms to explain the outcomes of multi-label classifiers. We have introduced explanations at different granularity levels which go from structural relationships between labels to the selection of features. Finally, we were interested in feature-level importance scores for how much a given input feature contributes to a multi-label model's output. This contribution looks into two different possibilities of using existing methods for single-label as oracles or using feature attributions obtained from symbolic explanations. In order to evaluate the quality of feature attributions, we extend the properties of sensitivity, data-stability to the multi-label setting in addition to a new property specific to multi-label classification we called label-explanation correlation.

Keywords: eXplainable AI (XAI), Symbolic explanations, Score-based explanation, Model-agnostic, Multi-label classification, Satisfiability testing, Feature attribution.