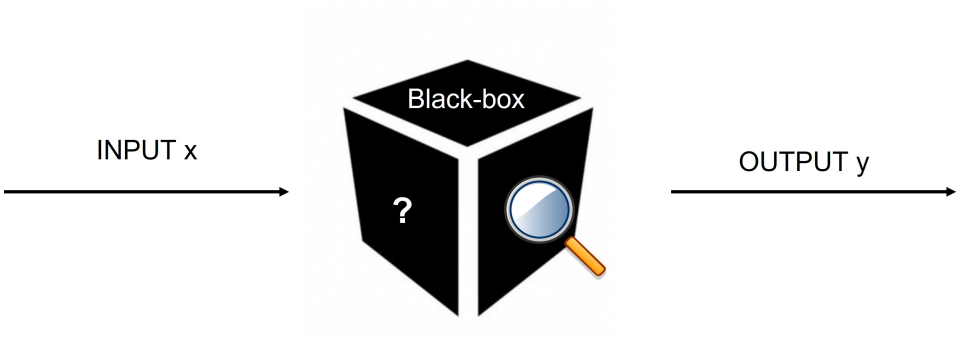
TÉCNICO LISBOA



Building a Benchmark Framework for eXplainable Artificial Intelligence (XAI) Methods

Dulce Marques de Carvalho Martins Canha

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisor(s): Prof. Kary Främling
Prof. Ana Luísa Nobre Fred

Examination Committee

Chairperson: Prof. Maria Margarida Campos da Silveira
Supervisor: Prof. Kary Främling
Member of the Committee: Prof. Bruno Emanuel Da Graça Martins

October 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Preface

The work presented in this thesis was developed integrated in a research team at the Computer Science Department of Aalto University, in Espoo, Finland, during the period February-August 2022, under the supervision of Prof. Kary Främling.

Acknowledgments

The path that culminates with the conclusion of this dissertation was accompanied by several important contributions, at different stages, which I would like to mention in these lines.

First I would like to mention how rewarding it was to have the possibility to develop this dissertation integrated in a research team in the department of computer science at Aalto University, in Finland, within the Erasmus program. I thank Professor Kary Främling, for accepting my collaboration and helping me with my goals. I am particularly grateful to have met Professor Sylvain Kubler, with whom I had the opportunity to discuss my work, who guided me through it, and helped me when I encountered difficulties. Finally, I would like to thank professor Ana Fred for helping me in the revision of this work and giving me advice.

I am very happy to have met the “cool” group in Erasmus, who gave me moments that I will never forget and who made me laugh and encouraged me when I was unmotivated. My special thanks to my friends Sara, Sigrid, and my roomie Ariana to whom, in addition to all the wonderful times we spent together, I could always unburden my frustrations. During the last summer it was very important to have my friends, yogi Kneev and “my dear” Jasper (read with Frodo voice), by my side. I also want to thank Pol, who supported me during (almost) all this journey that was the realization of this thesis, and who always believed I could do it.

To my university friends, here’s to us! I am very grateful to have spent these 5 years by their side, and to have had the opportunity to grow with each one of them. A special thanks to MOF, Caramelo, Marta, Jorginho, Francisco, Diogo, and to my best friend Tiago.

To my girls, Mariana, Simoninha, Xucas, and Ritão, many many thanks. They have been with me for over 10 years (Mariana, since I know myself), and I know I can always count on them. They never let me give up, and have always advised me towards the best decisions, even if I did not make some of them.

Finally, I want to thank my sister Sara for always rooting for me and who inspires me a lot. A special thanks to my mother Silvana, who supported me tirelessly and lived with me every moment of joy and despair, who gave me energy, love, and encouragement to reach the end of this path, and who carefully read my work. Lôviú.

Resumo

A inteligência artificial (IA), especificamente as suas sub-áreas aprendizagem automática e aprendizagem profunda, têm obtido resultados impressionantes numa variedade de domínios de investigação científica, tais como medicina, segurança, e economia. Contudo, sistemas complexos de IA, embora demonstrem ótimos desempenhos de precisão, são vistos como caixas negras que carecem de explicabilidade. Com o aumento do número de sistemas de IA, torna-se importante para os seres humanos compreender como é que cada caixa negra chega a um resultado. A área da inteligência artificial explicável (XAI) surgiu assim da necessidade de resolver o problema da caixa negra. Esta área tem vindo a crescer rapidamente, mas em direções diferentes, revelando a dificuldade que a comunidade científica atualmente enfrenta para chegar a um consenso sobre definições e critérios de avaliação comuns, muitas vezes formulados de forma subjetiva. Para ultrapassar esta lacuna na investigação, a presente dissertação propõe um quadro de referência para os métodos XAI, concebido com base numa revisão metodológica sistemática da literatura, de modo a obter indicadores de desempenho objetivos e mensuráveis de uma forma abrangente e consensual. Este quadro é posteriormente aplicado para comparar métodos XAI conhecidos ou promissores, considerando um dataset tabular do domínio da medicina (previsão de doença cardíaca). O estudo comparativo realizado mostrou a relevância do método CIU, que abrange com mais eficácia as propriedades selecionadas de explicabilidade, quando comparado com outros métodos. Adicionalmente, o quadro proposto contribui para o estabelecimento de formalismo e taxonomia comuns, promovendo a uniformidade que está em falta na área de XAI.

Palavras-chave: Inteligência Artificial Explicável, Aprendizagem Automática, Inteligência Artificial de Confiança, Critérios de Avaliação, Quadro de Referência

Abstract

Artificial intelligence (AI), namely its sub-fields machine learning and deep learning, have demonstrated impressive outcomes in a variety of scientific research domains, such as medicine, security, and finance. However, complex AI systems, despite demonstrating great results and accuracy performances, are seen as black-boxes that suffer from lack of explainability. Therefore, as AI systems continue to grow, it becomes important for humans to understand how each black-box arrived to a certain result. This way, the field of eXplainable artificial intelligence (XAI) arose from the necessity of solving the black-box problem. XAI field has been growing fast, but in different directions, revealing the difficulty the scientific community faces to agree on common definitions and evaluation criteria, which are often formulated in a subjective manner. To overcome this gap in research, the present dissertation proposes a benchmark framework for XAI methods, which is designed based on a methodological systematic literature review in order to derive objective and measurable performance indicators in a comprehensive and consensual manner. This framework is then applied to compare 9 well-known or promising XAI methods considering a tabular dataset from the medicine domain (heart disease prediction). This benchmark study showed the relevancy of the CIU method, which covers to a better extent the 10 selected properties of explainability, when compared to other methods. Moreover, the proposed framework contributes to the settlement of common formalism and taxonomy, which promotes the uniformity that is lacking in the XAI field.

Keywords: eXplainable Artificial Intelligence, Machine Learning, Trustworthy Artificial Intelligence, Black-boxes, Evaluation Criteria, Benchmark Framework

Contents

Declaration . . . iii
Preface . . . v
Acknowledgments . . . vii
Resumo . . . ix
Abstract . . . xi
List of Tables . . . xv
List of Figures . . . xvii
List of Acronyms . . . xix

1 Introduction **1**
1.1 Motivation and Topic Overview . . . 1
1.2 Objectives and Contributions . . . 3
1.3 Thesis Outline . . . 4

2 State-Of-The-Art and Theoretical Background **5**
2.1 SoTa Methodology . . . 5
2.2 XAI: What, Why, and Where? . . . 6
2.2.1 From AI to XAI . . . 6
2.2.2 The need for XAI and Application Domains . . . 7
2.2.3 Terminology clarification . . . 11
2.3 XAI: How? . . . 12
2.3.1 Review Settings . . . 12
2.3.2 Methods and Trends . . . 14
2.3.3 Selected Methods . . . 16
2.4 XAI: Evaluation . . . 30
2.4.1 Property Identification . . . 31

3 XAI Benchmark Framework Formalization **33**
3.1 Property Selection . . . 33
3.1.1 Representativeness . . . 33
3.1.2 Structure & Speed . . . 35
3.1.3 Selectivity . . . 36

3.1.4 Contrastivity . . . 37
3.1.5 Interactivity . . . 37
3.1.6 Fidelity . . . 39
3.1.7 Faithfulness . . . 39
3.1.8 Truthfulness . . . 41
3.1.9 Stability . . . 42
3.1.10 (Un)Certainty . . . 43
3.1.11 XAI Goals and Summary . . . 43
3.2 Metrics Formalization for Tabular Data . . . 44
3.2.1 Quantitative metrics . . . 44

4 XAI Benchmark Framework Application in the Medical Domain 49
4.1 Experimental Results . . . 49
4.1.1 Heart Failure Prediction Dataset . . . 49
4.1.2 Machine Learning Models . . . 51
4.1.3 Explanations . . . 53
4.2 XAI Benchmark Framework Results . . . 64
4.2.1 Baseline Comparison . . . 64
4.2.2 Enhanced Solution . . . 78

5 Conclusions 79
5.1 Limitations and Future Work . . . 80

Bibliography 81

A Literature Review 99

B Results 103

List of Tables

3.1 Property Selection and Description. . . . 34
3.2 Metrics Formalization for Tabular Data. . . . 45

4.1 Model predictions for patient A. . . . 54
4.2 Results of representativeness property. . . . 64
4.3 Results of structure & speed property 65
4.4 Results of selectivity property. . . . 67
4.5 Results of contrastivity property. . . . 69
4.6 Results of fidelity property. . . . 73
4.7 Results of WBC metric for a linear regression model. . . . 75
4.8 Results of stability property. . . . 75

A.1 XAI SoTA papers. . . . 99
A.2 XAI application domains. . . . 99
A.3 XAI SoTA Methods 99
A.3 XAI SoTA Methods 100
A.3 XAI SoTA Methods 101
A.3 XAI SoTA Methods 102

B.1 Local attribution values for patient A. . . . 103
B.2 Global feature importance values. . . . 105
B.3 Extra results of WBC metric. . . . 105

List of Figures

1.1 White-box vs Black-box vs XAI. 2
1.2 Thesis outline. 4

2.1 AI as a sub-field of XAI. 7
2.2 The need for XAI. 8
2.3 XAI leads to trustworthy AI. 9
2.4 XAI terminology clarification. 12
2.5 Trends in XAI methods. 15
2.6 XAI methods: possible inputs and outputs. 16
2.7 Trends in XAI publications. 16
2.8 Intuition behind LIME. 21
2.9 Sparse Linear Explanations using LIME. 23
2.10 Intuition behind anchors vs LIME. 23
2.11 SoTa Property Identification. 32

3.1 Husky vs Wolf example. 41

4.1 First 6 rows of the heart dataset. 51
4.2 Instance of interest: patient A. 54
4.3 PDP explanations for numerical feature Age. 55
4.4 PDP explanations for categorical feature Sex. 55
4.5 PDPs for 2 features. 56
4.6 ICE explanations for numerical feature Age. 57
4.7 ICE explanations for categorical feature RestingBP. 57
4.8 ICE explanation for categorical feature ST_Slope. 57
4.9 PFI explanation. 58
4.10 LIME bar plots. 59
4.11 LIME prediction. 59
4.12 Anchors explanations. 60
4.13 Shapley values explanations. 60
4.14 KernelSHAP explanations. 61
4.15 MOC explanations (CFEs). 62

4.16 CIU calculations. 62
4.17 CIU: CI and CU plots. 63
4.18 CIU: textual explanations. 63
4.19 CIU: CInfl plots. 64
4.20 LIME: selective explanation. 68
4.21 CIU: selective explanation. 69
4.22 Adversarial attack simulation. 70
4.23 LIME and Anchors after adversarial attack. 70
4.24 Shap(ley) after adversarial attack. 71
4.25 CIU after adversarial attack. 71
4.26 LR coefficients. 73
4.27 Faithfulness plots. 74

B.1 LIME heatmap. 104
B.2 Fidelity histograms for LIME and Anchors. 104

List of Acronyms

AI Artificial Intelligence

AI HLEG High-Level Expert Group on Artificial Intelligence

AUC Area Under the Curve

BP Blood Pressure

BS Blood Sugar

CAM Class Activation Mapping

CFEs CounterFactual Explanations

CI Contextual Importance

CInfl Contextual Influence

CIU Contextual Importance and Utility

CNN Convolutional Neural Network

CP Ceteris Paribus

CU Contextual Utility

CVD Cardiovascular Disease

DARPA Defence Advanced Research Project Agency

DL Deep Learning

ECG Electrocardiogram

EDA Exploratory Data Analysis

FS Feature Summary

GLM Generalized Linear Model

Grad-CAM Gradient-weighted Class Activation Mapping

HR Heart Rate

ICE Individual Conditional Expectation

ID Incremental Deletion

LASSO Least Absolute Shrinkage and Selection operator

LIME Local Interpretable Model-agnostic Explanations

LR Logistic Regression

LRP Layer-wise Relevance Propagation

MAUT Multi-Attribute Utility Theory

MCDM Multiple Criteria Decision Making

ML Machine Learning

MOC Multi-Objective Counterfactuals

NN (Artificial) Neural Network

PC Preservation Check

PDP Partial Dependence Plot

PFI Permutation Feature Importance

RF Random Forest

ROAR Remove And Retrain

SA Surrogate Agreement

SHAP SHapley Additive exPlanations

SoTa State-of-The-art

SVM Support Vector Machine

WBC White-Box Check

XAI eXplainable Artificial Intelligence

Chapter 1

Introduction

This chapter provides the reader with the motivation behind the present work and an overview of the addressed topic in section 1.1., followed by the main objectives and main contributions in section 1.2. Finally, section 1.3. describes the thesis outline.

1.1 Motivation and Topic Overview

Artificial intelligence (AI) and machine learning (ML) have demonstrated impressive outcomes in a variety of scientific research domains, such as medicine, forensics, and finance, especially with the emergence of deep learning (DL) [1]. Simple models like a linear regression or a decision tree show a clear relationship between input data and model output. These are called white-box models (see Top Figure 1.1) , as they are seen as transparent and understandable by humans. Complex models like convolutional neural networks (CNNs), a deep neural network architecture, usually outperform the previous ones, showing significantly higher performance in terms of model accuracy [2]. However, these are considered black-box models (see Middle Figure 1.1), as they suffer from a lack of explainability, meaning they lack interpretable tools for humans to understand the model working logic and outputs [3]. This is a huge barrier for their application in real world systems, particularly in medical systems. In the case of clinical decision-making systems, for example, it is important to explain the decisions made by the underlying ML model in order to justify the outcome. Black-boxes are usually not appreciated by healthcare professionals because they want to understand how the system generates a prediction, and these models are frequently viewed as unreliable and a loss of control. In this sense, explainability has been the most desirable attribute of a decision support system in the medical field [4].

Explainable artificial intelligence (XAI) is an emergent field that refers to methods and techniques in AI application which focuses on solving the lack of explainability present in black-box models (see Bottom Figure 1.1). It implements several approaches to better understand a system's underlying mechanisms and outputs. If a ML model can explain its decisions, it is closer to achieve fairness (ensuring that predictions are unbiased), reliability, and transparency, which are some of the principles that should be met when implementing AI [5]. Ultimately, XAI leads to trust and reliance, as it is easier for humans

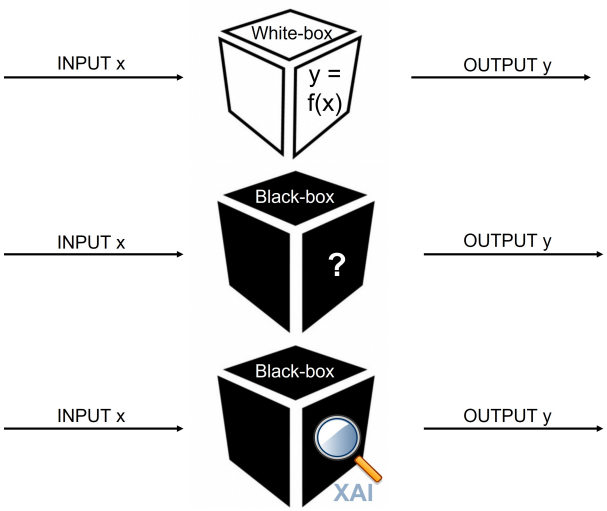


Figure 1.1: Top: White-box model, transparent. Middle: Black-box model, opaque. XAI implements techniques that "look" into the black-box and try to explain the underlying mechanisms.

to trust a system that explains its decisions compared to a black-box [6]. These traits can aid in the usability of AI and ML systems in a large range of scientific domains. Reliability is particularly important for healthcare professionals, as if an AI system can clarify its decisions, the former can start using it as an assisting tool for clinical decision, for example, the making of the correct diagnosis [2].

Many governmental, non-governmental and standards organizations have launched initiatives to establish ethical principles for the development of AI. In the EU, this step was taken by the High-Level Expert Group on Artificial Intelligence (AI HLEG), who wrote and published "Ethics Guidelines for Trustworthy AI" [7]. This document lists four ethical principles that should be adhered when developing, deploying and using AI systems: respect for human autonomy, prevention of harm, fairness and explicability. Moreover, it lists seven key requirements that AI system's entire life cycle should meet in order to achieve trustworthy AI and concludes with an assessment list that offers guidance on each requirement's practical implementation. These requirements should be considered in line with the specific application and are applicable to different groups of stakeholders, namely developers, who should implement and apply them, deployers (e.g., a hospital), who should ensure that the systems they use meet the requirements, and end-user (e.g., a doctor) and broader society, who should be informed. In this document, the principle of explicability is listed as one of the ethical principles in the context of AI systems. Also, transparency is presented as one of the seven key requirements for trustworthy AI, where traceability, explainability, and communication are shown to be all necessary to reach it. Although explainability is included in the transparency requirement, most of these trustworthy AI requirements guide directly the XAI approach as a crucial component to consider and include in AI systems. Also, XAI methods are stated as one of the technical methods vital for trustworthy AI. The AI HLEG authors state that "for a system to be trustworthy, we must be able to understand why it behaved a certain way and why it provided a given interpretation". Humans require clear explanations, arguments and evidence to be able to self-assess the quality of the decision/suggestion and decide when and how to trust and use an AI system.

In the last few years, a large number of different XAI methods have been proposed in the literature, existing the need to define a set of evaluation criteria that allow researchers to compare them. Accordingly, the research activity in this field has been growing very fast, but in different directions, demonstrating a lack of common formalism for defining XAI related concepts and identifying the essential properties scholars should consider when developing or choosing methods for explainability. It is crucial that XAI methods themselves are understandable and easily accessible for end-users, and, most importantly, non-experts [8–10]. The task of evaluating the (predictive) performance of a ML model is simple, as there is a ground truth label to compare the test data with. On the other hand, the task of evaluating the explainability of that model is not simple, as there is no accessible ground truth explanation and therefore no direct way of evaluating and comparing different explanations. Moreover, this task becomes even more challenging due to the lack of consensus among the research community on the definition of the term explainability and other related concepts (e.g., interpretability and transparency) [11, 12]. In this sense, there is the need to design a comprehensive and consensual benchmark framework for XAI methods that can integrate ML workflows and allow for their comparison and, ultimately, selection of the most appropriate method(s) to use. This should be considered in line with specific audiences and contexts. XAI methods can be applied to different ML or DL models and different types of data, and therefore can provide effective decision support for a variety of tasks, in relevant domains such as transportation, security, medicine, finance, legal, and military.

1.2 Objectives and Contributions

As mentioned in the Preface, the work presented in this thesis was developed integrated in a research team at the Computer Science Department of Aalto University, focusing on XAI, and the team supervisor has developed an explainability method that is called Contextual Importance and Utility (CIU) [13]. During the period February-August 2022, investigation about XAI in general, main challenges and available approaches was conducted.

This Master's Thesis aims to investigate the XAI field and study different approaches. XAI is becoming a very wide subject area, with a lot of research directions. Hence, the specific target objectives, which emphasize the scope of the present dissertation, are the following:

1. **Systematic literature review:**

   Discover the current state of scientific research in XAI, together with its main characteristics, approaches, challenges, misunderstandings, and gaps. Relevant (evaluation) criteria identification.

2. **XAI Benchmark Framework Formalization:**

   Selection of a comprehensive and consensual list of properties and respective formalization with structured and measurable performance indicators to be used as a benchmark framework for XAI.

3. **Application of the Benchmark Framework in the medical domain:**

Selection of 9 well-known XAI methods, considering a tabular scientific dataset from the healthcare domain for a classification problem (heart disease prediction), with the purpose of validating the defined framework and, subsequently, compare the explainability methods and assess the relevance of CIU.

The successful achievement of these objectives leads to the following contributions in the XAI field:

- Categorization of XAI papers and XAI methods.
- Settlement of clear and unambiguous, yet comprehensive, XAI terminology.
- Settlement of common criteria for XAI evaluation.
- XAI benchmark framework which concretely addresses how to evaluate different methods considering 10 comprehensive and consensual properties.
- Ready to use implementation code available as opensource on Github.

1.3 Thesis Outline

This dissertation is organized into five main chapters. The first and current chapter 1 comprises an introduction to the work, including the motivation for the conducted study, the objectives proposed to be accomplished, the main contributions, and the thesis outline.

The thesis outline is summarized and detailed in Figure 1.2. In Chapter 2, a systematic analysis of literature review is carried on, together with relevant background knowledge. Subsequently, XAI classification criteria, methods, challenges, and most importantly, an extensive list of properties is spotted and identified to be used as a referent taxonomy for the proceeding development of a comprehensive and consensual benchmark framework for XAI in Chapter 3. In Chapter 4, a tabular scientific dataset from the healthcare domain (heart disease classification problem) is considered, with the purpose of validating the defined framework and, subsequently, compare 9 well-known XAI methods and assess the relevance of CIU. Finally, Chapter 5 highlights the main conclusions to be drawn from this work, along with the main limitations and suggestions regarding future work.

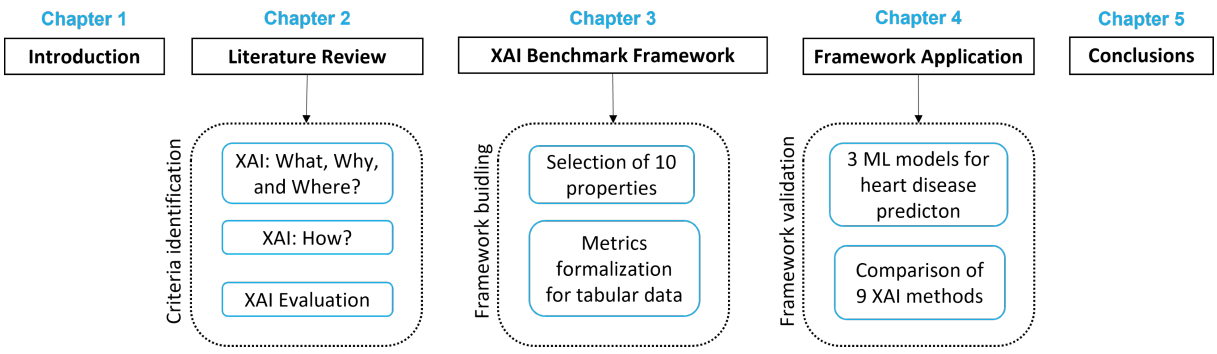


Figure 1.2: Thesis outline scheme.

Chapter 2

State-Of-The-Art and Theoretical Background

This chapter provides the reader with the state-of-the-art (SoTa) of XAI and relevant theoretical background. Section 2.1 presents the methodology adopted for the SoTa. Next, section 2.2 provides an insight into AI and how XAI arises from the former, answering the question "What is XAI?". Then, section 2.3 approaches the questions "Why is XAI needed?" and "Where is XAI applied?". Next, section 2.3 provides a summary of available XAI methods and a theoretical background regarding their taxonomy, answering the question "How is XAI applied?". Still in this section, a detailed description of CIU and 8 other well-known XAI methods is performed. Finally, section 2.4 describes the SoTa review of XAI evaluation and associated challenges.

2.1 SoTa Methodology

Explainable Artificial Intelligence (XAI) is becoming a very wide subject area, with a lot of research directions. Hence, here it is emphasized the scope which this works focuses on, by explaining the followed methodology.

To realize the current state of scientific research in this subject, a search in Google Scholar was carried out to identify articles published in indexed journals, books or newspapers between the years 2015-2022. Key words used in the initial search include "Explainable AI", "Interpretable AI", "Explainability ML" and "Interpretability ML". Several papers were found through this refining selection phase, being excluded all the papers wrote in a language different than English. A sum of 2 books and 96 papers were chosen for the final review stage, selected after evaluating the titles, abstracts, and their main contributions. From these, 26 were classified as XAI surveys/reviews, 19 as discussion/theoretical papers, 20 as evaluation/comparison studies of XAI methods, and 12 as frameworks or new approaches for XAI. Moreover, 7 user studies and 19 case/use studies were selected. Table A.1 presents the paper distribution of this systematic literature review - note that the papers are placed in the section in which they are predominantly inserted, and may have characteristics of other(s). The remaining bibliography

was found from the books and papers mentioned, or through relevant research that arose from reading them for the development of this work.

A systematic analysis of literature review was carried on as a first step. Accordingly, a detailed study of existing SoTa papers (see Table A.1) was conducted and, subsequently, classification criterion was identified, which includes its terminology (**what** is XAI?), its motivations (**Why** is XAI needed?) and its domains (**Where** is XAI used?). The results from this analysis are discussed in Section 2.2. Next, to understand **how** XAI is being applied in the literature and how to evaluate it, a depth analysis of the SoTa surveys and reviews was performed. In this sense, in Section 2.3, SoTa methods are identified and related trends and conclusions are spotted. Moreover, a theoretical background of CIU and other selected methods that are implemented and compared in Chapter 4 are provided. The last Section of this Chapter approaches XAI evaluation, its associated challenges, and, most importantly, an extensive list of properties is spotted and identified to be used as a referent taxonomy for the proceeding development of a benchmark framework for XAI methods in Chapter 3.

2.2 XAI: What, Why, and Where?

2.2.1 From AI to XAI

The first work in the field of Artificial Intelligence (AI) was published in 1950 by Alan Turing [14]. Known as one of the founding fathers of AI, Turing proposed, in this paper, what subsequently became known as the Turing test, where he questions whether an artificial computer can think [15]. However, the term “artificial intelligence” was introduced in 1955 by John McCarthy [16]. In 2007, McCarthy redefined AI as “the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable” [17].

Intelligence can be defined, for example, as the “ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation” [18]. This definition is linked to machine learning (ML), a sub-field of AI, as an AI system can be developed without ML algorithms, i.e., with no trained mathematical model. There are three sub-categories of machine learning: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the algorithm relies on labelled data with known inputs and corresponding outputs (e.g., labels for classification problems) to give predictions, which is usually divided into a training and a testing dataset. In unsupervised learning, the algorithm relies on unlabelled data (contains only inputs) to find a structure in the data. In reinforcement learning, the algorithm continuously learns from feedback to achieve a certain goal.

Deep Learning (DL), often used interchangeably with ML, is actually a sub-field of ML, as the way the algorithms learn is different. While standard ML algorithms build on data representations obtained by feature engineering, exploring domain knowledge, data representation and feature extraction are most often intrinsically learned by DL algorithms exploring layered architectures of computational units. A

"deep" ML algortihm involves the existence of a large number of layers. The multilayer perceptron is an example of a simple neural network (NN), composed of more than three layers (one input layer, one or more hidden layers, and one output layer). Due the complexity and structure of the DL algorithms, many (not necessarily all) do not require "feature engineering" (extraction of features based on expert domain knowledge - a first pre-processing step). This is one of the biggest advantages of DL over ML, as the first "can ingest unstructured data in its raw form (e.g., text, images), and it can automatically determine the hierarchy of features which distinguish different categories of data from one another" [19], therefore not needing human intervention for data processing.

Artificial intelligence, namely its sub-fields machine learning and deep learning, have demonstrated impressive outcomes in a variety of scientific research domains, such as medicine, forensics, and finance, especially with the emergence of DL [1]. However, AI systems, despite demonstrating great results and accuracy performances, are often referred as black-box algorithms that cannot be explained (see Middle Figure 1.1. Therefore, as AI systems continue to grow, it becomes important for humans to understand how the black-box arrived to a result. From this necessity arose the field of eXplainable artificial intelligence (XAI), that focuses on solving the black-box problem (see Bottom Figure 1.1. The term XAI was introduced by DARPA (Defence Advanced Research Project Agency), as a research program that focuses in producing more explainable models, while maintaining a high level of learning performance (prediction accuracy) and consequently enabling their understanding by human users so that they can gain trust and effectively manage the emerging of AI [20]. Based on the definition of AI given above, XAI can be defined as "the science and engineering of making (self)-explanatory intelligent machines". Lastly, Bibal et al. [21] stated that XAI should cover four levels: "(i) providing the main features used to make a decision, (ii) providing all the processed features, (iii) providing a comprehensive explanation of the decision and (iv) providing an understandable representation of the whole model".

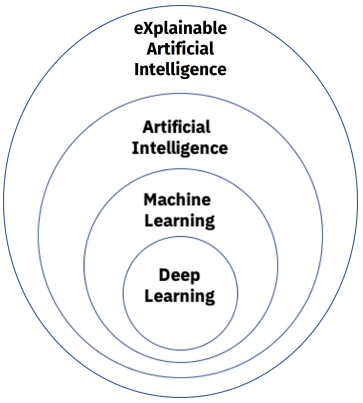


Figure 2.1: Machine learning and deep learning are sub-fields of artificial intelligence, which is introduced here a sub-field of explainable artificial intelligence. This figure is adapted from [19].

2.2.2 The need for XAI and Application Domains

Simple ML models like linear regression or decision trees show the relationship between input data and model output and therefore are seen as transparent and self-explainable/understandable - white-box models (see Top Figure 1.1). Complex ML models like random forests and especially DL models, like deep neural networks, usually outperform the previous ones, showing significantly higher performance in terms of prediction accuracy [2]. However, they are black-box models (see Middle Figure 1.1) that do not provide an explanation for their outcomes and, due to their complex structure, are not understandable

to human users. Indeed, white-box algorithms are more understandable than black-boxes, and there is often a trade-off between accuracy and explainability: the former group is usually less accurate and the latter is usually less explainable [8]. For low-stake decisions, like movie recommendations on Netflix, this trade-off is not important, and a low explainable model can be used. For high-stake decisions, like a criminal justice system, this trade-off is very important, and usually a high explainable system is preferred, bearing the cost of a low(er) predictive performance. XAI is crucial in such situations, focusing on research that tries to avoid this trade-off, or at least making it more dynamic, where accuracy can be achieved along with explainability.

Figure 2.2 shows the need for XAI in a wrap, putting together an image taken from DARPA [20] and Adadi et al. [8], who listed four main reasons why explanations are needed: justify, control, improve, and discover. Justification is one key reason for XAI, as it allows the user to understand why (and why not) a certain output was given (or not), especially when unexpected decisions are made. Control is important, as having a greater understanding about a system behavior helps to rapidly identify when the system might fail and correct errors, which leads to the next reason for XAI: the need to continuously improve the AI system. The more explainable and understandable a model is, the easier it is to correct it and improve it. Finally, explanations can aid in the discover of new (hidden) insights, particularly when humans' knowledge of causal systems is incomplete, like in medicine [22].

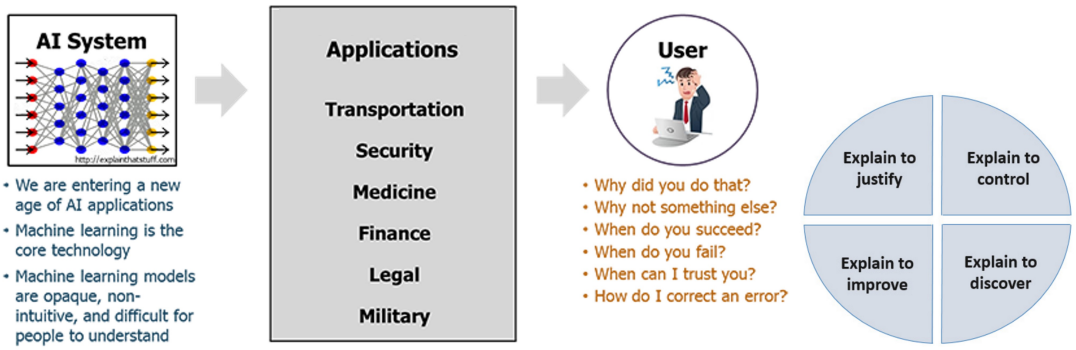


Figure 2.2: The need for XAI. This figure is adapted from [20] and [8].

Ultimately, XAI is needed to help building trust around AI systems. "For a system to be trustworthy, we must be able to understand why it behaved a certain way and why it provided a given interpretation". This phrase is taken from the EU report mentioned in Chapter 1, "Ethics Guidelines for Trustworthy AI" [7]. This study publishes many critical requirements, beyond XAI, included within the different AI principles guidelines. However, those requirements are not completely detached from XAI; in fact, they are intertwined, like proposed in Figure 2.3. This is succinctly discussed in the following paragraphs, where it is highlighted how XAI is linked to trustworthy AI, by contributing, at different levels, to each of the requirements. Moreover, XAI methods should themselves cover some of the requirements:

1. **Human agency and oversight**

   XAI methods should be developed with a human-in-the-loop approach, enhancing human oversight. Moreover, explanations allow target users to be able to make informed autonomous decisions regarding AI systems. Users should be given the knowledge and tools to understand and

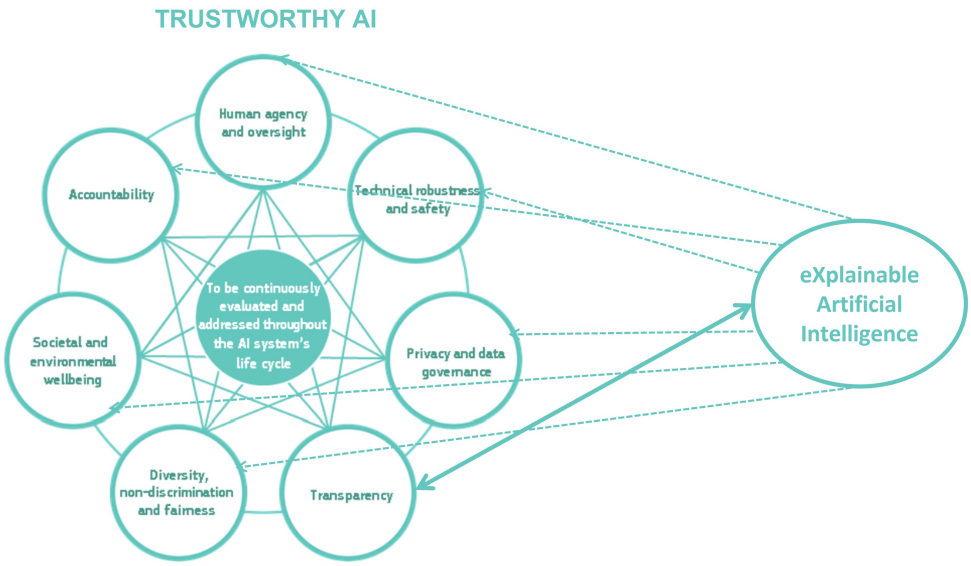


Figure 2.3: Interrelationship of the seven requirements and XAI: XAI supports each of the requirements (at different levels) and the requirements support each other, and should be implemented and evaluated throughout the XAI system's life-cycle. Adapted from [7].

interact with AI systems to a satisfactory degree and, when possible, be enabled to reasonably self-assess or challenge the system, which is possible with an explainable ML model, specifically one that provides an explanation interactive interface.

2. **Technical robustness and safety**

   By assessing how the system behavior can be changed, leading the system to make different decisions (or different explanations), the technical robustness of (X)AI methods can be improved. By giving robust and accurate explanations, XAI approaches can act in the minimization of unintended consequences and errors of AI systems. When occasional inaccurate decisions (or explanations) cannot be avoided, it is important that the (X)AI system can indicate how likely these errors are, so that the target user knows when to trust them. Furthermore, it is critical that the results of AI systems are reproducible, as well as reliable, which can be assessed from the respective explanations.

3. **Privacy and data governance**

   An XAI method that does not require access to the data or the model is attractive for companies (or situations) where privacy data is necessary.

4. **Transparency**

   Included in this requirement, is explainability, which concerns the ability to explain both the technical processes of an AI system and the related human decisions. To achieve this, XAI methods have been proposed. XAI also enables identification of the reasons why an AI-decision was erroneous which, in turn, could help prevent future mistakes. This traceability is also important in XAI and the identification of erroneous decisions should be performed alongside with the identification of erroneous explanations. Explanations can reveal AI system's capabilities and limitations, allowing

them to be communicated to AI deployers or end-users in a manner appropriate to the use case at hand. XAI "opens" the black-box, moving the system from an opaque to a transparent one.

5. **Diversity, non-discrimination and fairness**

   XAI proposals can be used for bias detection, preventing AI systems from the unintended (in)direct prejudice and discrimination against certain groups or people, potentially exacerbating prejudice and marginalization. An explanation interface enhances the active participation of all people in existing and emerging computer-mediated human activities and with regard to assistive technologies. Moreover, dissecting the internals of a black-box model via XAI techniques can help identifying the capability of the model to maintain the input data diversity at its output.

6. **Societal and environmental wellbeing**

   XAI methods, by making the AI systems more human understandable and interactive, can be used to enhance social skills, contributing to the social impact of AI.

7. **Accountability**

   XAI contributes to auditability as it can help explaining and assessing AI systems for different profiles, including regulatory ones. Also, accountability is closely linked to fairness, and XAI can contribute to the minimization and report of negative impacts, ensuring that necessary changes can be made to the system where and when needed. Furthermore, the minimization and report of negative impacts can be assessed inside XAI, by developing an internal evaluation and validity of the method, reporting on actions or decisions that contribute to a certain explanation. Finally, to ensure responsibility and accountability for AI systems and their outcomes, explainability is crucial.

Concluding, explainable AI leads to trustworthy AI, and the "implementation of these requirements should occur throughout an AI system's entire life cycle and depends on the specific application". The application domains presented in Figure 2.2 for AI systems represent potential domains where there is a need for research activity on explainable models: transportation, security, medicine, finance, legal, and military. Table A.2 distributes the case studies introduced in Table A.1 through each of these applications domains, showing how XAI is being deployed in real case scenarios. XAI approaches are particularly relevant in areas of social impact, such as medicine and healthcare [23, 24] criminal justice (legal domain) and autonomous vehicles (transportation domain) [25]. The DARPA research program [20] that coined the term XAI was made by military researchers, showing that the military domain also suffers from the AI (lack of) explainability problem [8].

Therefore, XAI can bring great benefit to several specific-application domains [8], as more complex and difficult-to-interpret AI approaches (e.g. DL models) are being adopted. Besides the previously mentioned domains, XAI strategies have become increasingly important in areas like software analytics [26], biology & chemistry [27–29], energy & power [30], and others.

Besides being considered in line with the specific application, the development of models and methods in XAI that aim to contribute to the achievement of (all) the requirements for trustworthy AI should also consider and be applicable to different groups of stakeholders. These groups are: the developers,

who should implement and apply explainability methods, the deployers (e.g., an hospital), who should ensure that the systems they use meet the requirements, and the end-user (e.g., a doctor or a patient) and broader society, who should be informed. Concluding, the XAI stakeholders includes everyone who "either want a model to be 'explainable', will consume the model explanation, or are affected by decisions made based on model output" [31]. This is inline with the idea that, beyond improving model understandability as a goal in itself, it is necessary to integrate the deployers and end-users (specially domain experts) in the design of explainability strategies. Otherwise, machine learning is unlikely to become a part of real-word applications, such as routine clinical and healthcare practice [32].

2.2.3 Terminology clarification

After carefully analyzing the SoTa literature, it became clear that, despite its fast emerging, XAI is still not a well-established field, demonstrating a lack of common formalism and taxonomy. Scientific research around XAI has produced many different definitions of explainability and has identified various concepts related to it that most often overlap with each other, namely interpretability, transparency, intelligibility, comprehensibility and understandability. Specifically, there is a heated debate about whether interpretability and explainability should be used interchangeably. There is a clear division between scholars, as about 50% of the reviewed surveys use the terms interchangeably and the other half consider they are different. Therefore, the first challenge arising from the rapid growth of the research activity in XAI is the establishment of a common formalism to define XAI related concepts. Scholars should work on an agreement regarding what explainability is and start to use the same words and concepts, so that research around this subject becomes clearer and organized.

That being said, this section provides succinct and unambiguous definitions, in the XAI context, of transparency, interpretability, and explainability, which are related to the ability to observe the processes that lead to the decision making of the model [2]:

- **Transparency**: A model is considered transparent if its decision making is by itself understandable [33], meaning a user can see and understand the mathematical mechanisms that map inputs to outputs [34]. This applies to white-box models (see Top Figure 1.1), such as linear regression. Black-box models, such as CNNs, are the opposite, being seen as opaque systems (see Middle Figure 1.1).

- **Interpretability**: The ability to provide the meaning in understandable terms to humans [35]. A model is considered interpretable if it is described in a way that can be further explained. The more interpretable the model, the deeper the extent to which cause-effect relationships can be observed within a system [2], i.e., the user can relate properties of the input to their output [34].

- **Explainability**: The ability to provide the functioning of a ML system in understandable terms to humans [5]. The more explainable a model, the deeper the understanding that humans achieve in terms of the internal procedures that take place while the model is training or making decisions [2].

The concepts above are introduced here as similar, yet distinct concepts. Transparency is about

being able to automatically understand the decision making of an AI system; interpretability is about being able to discern the internal mechanics without necessarily knowing why; explainability is being able to explain what is happening, i.e., the system's reasoning. This is illustrated in Figure 2.4, using an illustrative example of an AI system that predicts the profession based on an input image (here the decision is correct, the image corresponds to the class doctor). Explainability adds a reasoning line, which consists in explaining the decision making of an AI system using, for example, understandable features of the input data. On the contrary, transparent and interpretable models do not have this reasoning line, and are not able to provide explanations. However, they are described in a way that enable the explanations of its decisions [34]. Other concepts related to XAI that appear in the literature are not introduced here because they are not so commonly used, so that there are no overlaps between them, and to contribute to clear and unambiguous, yet comprehensive, terminology.

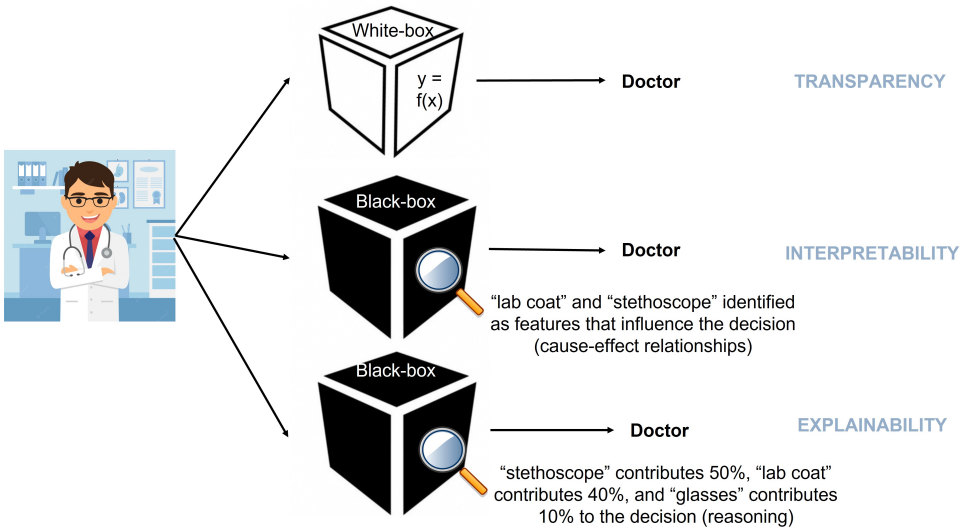


Figure 2.4: Difference between transparency (top), interpretability (middle), and explainability (bottom). The latter shows that the black-box model is biased towards the use of glasses.

2.3 XAI: How?

2.3.1 Review Settings

As discussed in the previous Section, methods and techniques involving XAI research are necessary not only to explain the system's behaviour and results to users, but also to deploy reliable and trustworthy technology [7]. In this Section, the focus is on how these methods and techniques are being proposed and used by researchers, i.e., how XAI is being deployed.

The complexity of a ML model is directly related to its interpretability and explainability. Generally, the more complex the model, the more difficult it is to interpret and explain [8]. This is related to the accuracy vs. explainability trade-off, which led to the establishment of two explainability strategies: intrinsic and post-hoc methods. Intrinsic methods correspond to explainable by design methods, where explainability is directly achieved through constraints imposed on the model during training (white-box models are

intrinsically explainable). Post-hoc methods are used to provide black-box explanations after model training [33, 36], therefore avoiding the explainability vs. accuracy trade-off. Some researchers support the first (classic) approach for high-stake decisions [37], but there has been an exponential interest and demand for the alternative (novel) approach. A common view is that "as long as the model is accurate for the task, and uses a reasonably restricted number of internal components, intrinsic interpretable models are sufficient. If otherwise the prediction target involved complex and highly accurate models, considering post-hoc interpretation models is necessary" [8].

The latter strategy is the focus of the second stage of this systematic review and, from the SoTa surveys, a total of 131 post-hoc XAI methods published between 2010 on-wards (except PDP[1] and CIU[2]) were identified. They are displayed in Table A.3 where the first three columns identify the method by its *Name*, *Reference* paper, and publication *Year*. The fourth column, *% surveys*, is a measure of popularity of the method, as it corresponds to the percentage of the reviewed surveys (from a total of 26) that appears in. The methods are displayed in decreasing order of popularity. The last column, *Software*, checks whether the code for reproducing the explanations is available (Y - with hyperlink) or not (N). Note that the given software is not always the original one, when this is not available by the authors. In this case, another implementation of the method is provided. Furthermore, in some cases (e.g., LIME), there is more than one available software, in both Python and R programming languages (and sometimes Java). Furthermore, in order to make the output of this review easily and practically accessible to readers, the most widely used distinctions adopted to annotate the methods are also summarized in each of the remaining columns (from fifth to eighth column):

- *Portability* indicates the range of ML models to which the explanation method can be applied, distinguishing between model-agnostic (A) and model-specific (S) methods. Model-Agnostic (A) methods can be used to explain any type of model, treating all ML models as black-boxes, even if they are not. Model-Specific (S) methods can be used to explain only a specific type of black-box model. Regarding the latter, most of the encountered methods are specific for deep neural networks, meaning they usually need access to parameters of the network layers. The portability of DNN-specific methods can of course depend on the type of layers it needs access. For example, Grad-CAM (Gradient-weighted Class Activation Mapping) [38] is applicable to any CNN-based model, whereas CAM (Class Activation Mapping) [39] requires a particular CNN architecture because it uses information taken out of the last fully-connected layer of the network.

- *Scope* indicates the extent to which the method addresses the entire model behavior, distinguishing between global (G) and local (L) explanations. A global (G) method aims at explaining the entire model behavior, i.e., the overall logic of the (black-box) model. This way, a global explanation is valid for any instance. A local (L) method aims at explaining the reasons for a single prediction, being valid for a specific instance. It is important to mention that some local methods provide some kind of global overview. LIME [40] and Anchors [41], for example, explain several individual pre-

[1] PDP is from 2001, included here due to its high popularity and close relation to ICE method. Both methods will be further described.

[2] CIU is included here because it was proposed, in 1996, by the supervisor of this thesis, and has been continuously studied and improved by the CS research team at Aalto University in Espoo, Finland, where this work was developed.

dictions (submodular pick, as named by the authors) of the black-box model as a way to provide a global explanation overview. Other methods provide both local and global explanations (L/G), such as CIU [42], which provides a unified definition of global and local feature importance.

- The seventh column distinguishes the methods considering the type of *Data* they can be applied to. It covers the three principal data types recognized in the literature: tabular data (TAB), images (IMG) and text (TXT). The XAI techniques can be data-specific, being restrict to one or two of these data types, or data-agnostic, meaning they can be used for the three of them (ANY). Other data types include time series (TS) or videos (VID), which are still in the very early stage regarding explainability approaches. Only the last three methods, out of the 131 XAI methods presented in table A.3, are specific to these data types, being included here just to show that explainability can be extend to other data types.

- Type of *Problem* distinguishes methods designed to be applied to regression (R) or classification (C) problems/tasks. The first works to predict continuous values such as housing prices and the second is used to classify discrete values such as “benign” or “malign” in a tumor classification problem. XAI methods can be specific to either regression or classification, or possible to use in both situations. Note that this factor of distinction concerns tabular data only, as for text and images data types, the problem type is always classification.

2.3.2 Methods and Trends

From Table A.3, which presents an overview of the latest publications of (new) XAI methods, the main trends regarding their approaches and characteristics are taken.

Figure 2.5, on the left, shows the total number of methods considering their portability and scope. Regarding the scope of explainability, there seems to be a preference among scholars for local explanations, focusing on single predictions. Nevertheless, methods that can provide both local and global perspectives are ideal. Regarding the portability of explainability, methods that can be applied to all types of black-boxes are preferred, as these agnostic approaches “provide crucial flexibility in the choice of models, explanations, and representations, improving debugging, comparison, and interfaces for a variety of users and models” [43]. However, model-specific methods are more common, which is associated with the fact that images are the most widely used data type, and CNNs are frequently used for image classification. This is why CNN-specific methods are very popular. Images are the main data type studied in the XAI field, followed by tabular data - Figure 2.5 on the right. For tabular data, around half of the methods focus only in classification tasks. The other half can be applied to both regression and classification tasks, which is of course ideal. Text, unlike tabular and image data, does not have a structure, so its related tasks are usually very complex. There is enormous research in this field in literature, which is known as Natural Language Processing (NLP) [33]. However, due to its high complexity, explanations of text data are at the very early stages compared to the former ones. Other data types include time series or video data, which are even more complex and in an earlier stage regarding explainability approaches.

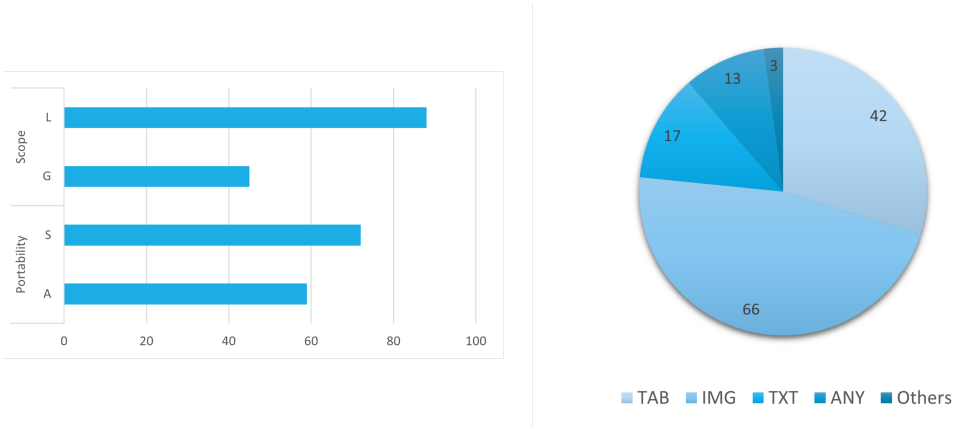


Figure 2.5: Left: Number of XAI methods considering the portability (agnostic (A) vs specific (S)) and scope (global (G) vs local (L)) of explanations. Right: Number of methods applicable to each data type: images (IMG), tabular data (TAB), text (TXT), any of the previous (ANY), or others, namely TS and VID (Others).

For each of the three input data types mostly recognized in the literature - tabular (regression or classification problems), images, and text - different types of explanations can be given as an output. This is illustrated in Figure 2.6, where [6, 33]:

- *Feature Summary (FS)* explanations provide summary values for each feature, usually together with a visualization plot. These values can be a single number per feature (most common), such as feature contribution, a simpler one, like a model prediction, or a more complex one, like a number for each feature pair, representing their pairwise feature interaction strength. For tabular data, feature summary, in particular feature contribution, is the most widely used approach by XAI developers.
- Explanations can be presented in the form of *Rules*, which are a set of conditions that an instance must satisfy in order to meet the rule's decision. This type of explanations is popular, as, due to their logic formalization, they are considered easily understandable.
- Some methods return *Data points* (already existent or newly created). These can be prototypes, which are examples that characterize the predicted outcome, or counterfactuals, which are examples similar to the input data instance, that are found by making the smallest change to some feature values that changes the prediction to a predefined (relevant) output. Note that counterfactuals are gaining a lot of attention because they are seen as human-friendly explanations.
- An *Attention Map* is matrix of scores which reveals how each word in the input text are related to each other.
- A *Saliency Map* highlights the contribution of each pixel/word for a particular class. For image and text data, the most commonly used methods are the ones that create saliency maps. Note that this approach can be seen as the translation of the feature contribution approach for tabular data.
- *Concept Attribution* methods compute attribution to a predefined target "concept" in the image. The explanations show how sensitive is the output (e.g. a prediction of wolf) to a concept (e.g. the presence of snow).

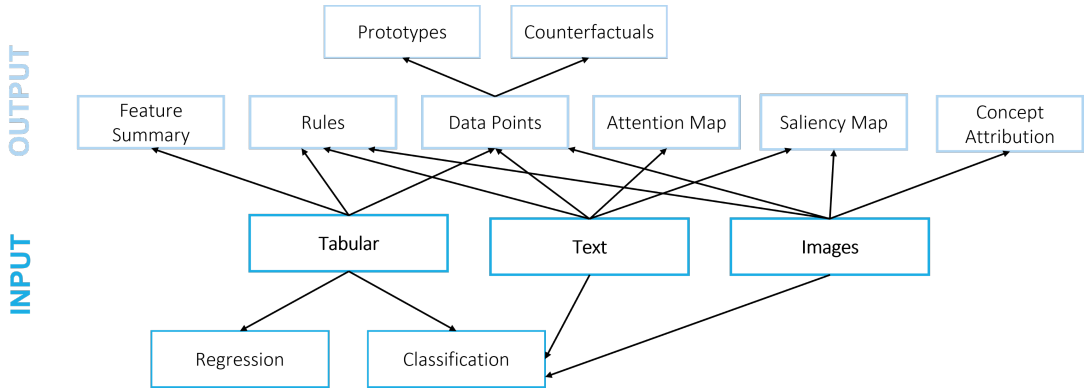


Figure 2.6: XAI methods: possible inputs and outputs.

From the SoTA literature analysis, it is clear that a large number of methods have been proposed, with a peak in publications reached in 2017. However, since then, and as it can be seen in Figure 2.7, with a large number of XAI methods available, the subject of publications has shifted towards categorization and discussion articles, as a consequence of the need for organization in this area of scientific research. In particular, there has been increasing research in the evaluation and comparison of XAI methods, which can be effectively performed if the relevant properties all the methods are meant to cover are correctly identified. This is discussed in detail in Section 2.4.

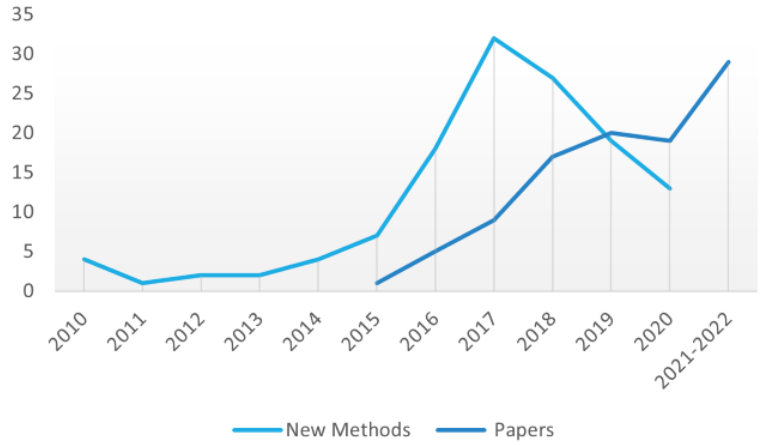


Figure 2.7: Trends in XAI publications. There has been a shift from the development of new XAI methods towards categorization and discussion papers.

2.3.3 Selected Methods

This Subsection provides a description of 9 selected post-hoc methods for explainability. Besides CIU, developed by Kary Främling in 1996, other 8 well-known methods were chosen with the purpose of selecting popular methods that cover all of the possible inputs and outputs in Figure 2.6, and that are going to be used to explain the predictions of the heart classification problem in Chapter 4 (to which the reader is referred to for explanations visualization). Before that, some theoretical foundations are given, to avoid confusion due to ambiguity.

Explaining the outcomes of a model f usually means explaining how each feature influences the pre-

diction of the instance(s) being explained and by how much [44]. For a linear model, a white-box model and an intrinsic method for explainability, its output can be directly explained. The learned relationships are linear and the prediction for a single instance x is the weighted sum of its features (also called input variables) [6] and it is given by Equation 2.1. The feature values are represented by x_i, with $i = 1, ..., p$, where p is the total number of features. The small omega w_i represents the learned feature weight for feature x_i. The first weight w_0 is not multiplied with a feature and it is called the intercept, which can be seen as a baseline[3].

$$f(x) = y = w_0 + w_1 x_1 + ... + w_p x_p \tag{2.1}$$

In Decision Theory and related sub-domains such as multiple criteria decision making (MCDM) and multi-attribute utility theory (MAUT), feature $importance$ and $utility$ concepts are clearly defined [45]. For a linear model like the one given by Equation 2.1, a numerical weight w_i expresses the importance of an input feature and a numerical score x_i expresses the utility of different possible input values for different outcomes, i.e., how good or favorable a value is [46]. So, how is the feature $influence$ (for a prediction $f(x)$) expressed? The answer is in Equation 2.2, where $\phi_i(x)$ represents the difference between what a feature contributes when its value is x_i and what it is expected to contribute - feature influence[4]. If the influence $\phi_i(x)$ "is positive, then the feature has a positive contribution (increases the prediction for this particular instance),if it is negative, then the feature has a negative contribution (decreases the prediction), and if it is 0, it has no contribution" [47] to the desired output (baseline).

$$\phi_i(x) = w_i x_i - wiE(X_i) \tag{2.2}$$

So that all the introduced concepts are clearly understood and distinguished, an illustration is provided, using an example of how the weighted average grade of a university student is calculated (range from 10 to 20, as only students with grades above 10 are approved). Considering Paul has 4 courses, where the number of credits is 6 for the first course, 9 for the second, and 15 for the third, giving a total of 30 credits. Paul's weighted average grade $f(x)$ can be represented in the form of equation 2.1 as $f(x) = 0.2x_1 + 0,3x_2 + 0,5x_3$. If $x = (10, 19, 15)$ (meaning Paul's final grade is 15.2) and the average grade of all students is 15 (baseline), then:

- The weight w_i of each course is the number of credits for that course and corresponds to the $importance$ of the course (the courses are the features). The courses have an importance of 0.2 (=$6/30$), 0.3 (= $9/30$), and 0.5 (= $15/30$), respectively, for the final weighted grade. So, the most important course is the third, with an importance value of 0.5 (it contributes in 50% to the output prediction).

- The $utility$ value of each course is the obtained grade for that course. The utility values are 10, 19, and 15 which in percent (considering the range from 10 to 20) are 0, 0.9, and 0.5. So, the most

[3] In fact, when the features have been standardized (mean of zero, standard deviation of one), w_0 is the prediction of the instance x with all features x_i at their mean value [6].

[4] Note that such influence is independent of the values of other features. This is because the linear model is additive (meaning the features do not interact), making it and other additive models easy to understand [47].

favorable value is 19, the grade obtained for the second course.

- The feature (course) $influence$ can be calculated only when a baseline or reference level is considered, which here is the average grade of all students, 15 (in percent is 0.5). The feature influence $\phi_i(x)$ can be directly calculated from equation 2.2, using the normalized values and 0.5 as the expected value. So, the first course value has a negative influence compared to the reference with a magnitude of 0.1[5]. The second has a positive influence of 0.12[6] for the final grade 15.2, when comparing with a baseline of 15. For the third course, the influence is 0[7], which makes sense because Paul had a grade of 15, and the comparison is with the reference which is also 15. This is very important to keep in mind, as although the third course is the most important for the final grade, in this case, it has 0 influence due to what was mentioned just before.

Computing the importance, utility, and influence values for this illustrative example was simple, because the model is known (white-box model) and the features do not interact [47]. However, when dealing with black-boxes, $f(x)$ is unknown, which restricts the way of observing the model behavior. Then, to "get inside" a black-box, the adopted approach of perturbation-based methods is to change the input space and observe what happens to the outputs. From here, the same (and other) concepts can be calculated and studied, although their calculation becomes more complex. The following XAI methods perform this study in different ways.

Partial Dependence Plot (PDP)

The partial dependence plot (PDP) [48] provides a visualization tool, which the authors consider to be one of the most powerful explanation tools. The PDP is a global and feature summary (FS) method for tabular data, as it considers all instances and shows the marginal effect that one or two features have on the predicted outcome of a ML model. It is also a model-agnostic method, as it can be used to explain models produced by any black-box prediction algorithm.

Thinking of a single instance x, $f_i(x)$, which formula is displayed in 2.3, represents the average prediction for that instance when feature i is varied over its range (taken from the training data, using a grid), while the values of other input features remain fixed [47]. This can also be thought of globally, i.e., for all inputs x_n with $n = 1, ...N$ (size of the training dataset), and considering a subset of features S (with complementary subset of features C), instead of a single feature i, which leads to the partial function 2.4 that is displayed in a PDP. The feature vectors $\mathbf{x}_S$ and $\mathbf{x}_C$ together make up the total input feature space $\mathbf{x}$. By marginalizing the output over the distribution of the features in set C, the acrshortpd function depends only on the features of interest in set S (interactions with other features are included, which is a problem when features are correlated), showing the relationship between them and the predicted outcome [6]. As $dP(\mathbf{x}_C)$ is not known, the approximation taken in the last step of equation 2.4 is straightforward: "we estimate the true model with f, the output of a statistical learning algorithm, and we estimate the integral over x_C by averaging over the $N\mathbf{x}_C$ values observed in the training set" [49].

[5] $\phi_1 = w_1x_1 - w1E(X_1) = 0.2 * 0.0 - 0.2 * 0.5 = -0.1$
[6] $\phi_2 = w_2x_2 - w2E(X_2) = 0.3 * 0.9 - 0.3 * 0.5 = 0.12$
[7] $\phi_3 = w_3x_3 - w3E(X_3) = 0.5 * 0.5 - 0.5 * 0.5 = 0$

$$f_i(x) = E[f(x_1, ..., X_i, ..., x_p] \tag{2.3}$$

$$f_S = E[f(\mathbf{x}_S, \mathbf{x}_C] = \int f(\mathbf{x}_S, \mathbf{x}_C dP(\mathbf{x}_C) \approx \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_S, \mathbf{x}_{C_n}) \tag{2.4}$$

The visualization of the partial dependence (PD) function described above is limited to low-dimensional arguments, in fact no more than 2-dimensional. Functions of a single real-valued variable, i.e., the PD function for one numerical feature, can be plotted as a graph of the values of f_i against each corresponding value of i (example in Figure 4.3). Functions of a single categorical variable, i.e., the PD function for one categorical feature, can be represented by a bar plot, each bar representing one category and the bar height the value of the function (example in Figure 4.4). Functions of two real-valued variables can be represented using mesh plots or correlation plots [6, 48] (see Figure 4.5).

PDP plots can be useful for finding linear, non-linear (more complex), and no relationships (monotonic) between the features and the target outcome. Recently, Greenwell et al. [50] proposed a method for calculating a feature importance measure based on PDP. The idea behind it is that a flat line (i.e., a monotonic relationship) on a PDP indicates that the feature is not important, and the more the function varies, the more important the feature is [6].

Individual Conditional Expectation (ICE)

Individual Conditional Expectation (ICE) plots [49] provide a visualization tool, that consists in the disaggregation of the output of classical PDPs introduced above. ICE is the equivalent to PDP for N individual data instances, generating N conditional expectation curves. Then, it is a local (although also providing a global overview) and FS method for tabular data. It is also a model-agnostic method, as it can be used to explain models produced by any black-box prediction algorithm.

The values for one line (for one instance x) are computed using formula 2.3, but without considering the average prediction, i.e., the predictions for that instance when feature i is varied over its range are plotted while the values of other input features remain fixed:

$$f_i(x) = f(x_1, ..., X_i, ..., x_p) \tag{2.5}$$

Like in PDP, instead of an individual feature i, a subset of features S (with complementary subset of features C) can be considered, although usually $|S| = 1$ (only one feature is considered at a time, as more than one would not be possible to clearly visualize in the plot). The lines for N observations constitute the final ICE plot. This way, "the ICE algorithm gives the user insight into the several variants of conditional relationships estimated by the black box" [49]. In contrast to PDPs, where the average effect is provided (global), in ICE it is possible to see individual effects, and therefore heterogeneous relationships and interaction effects can be revealed [6]. Both PDP and ICE approaches basically make use of the utility values associated with each feature, changing them and visualizing the changes in the output prediction.

The visualization of ICE plots is limited to one feature. Functions of a single real-valued variable, i.e., the ICE function for one numerical feature, can be plotted as a graph of the multiple values of $f_i(x)$ against each corresponding value of i (example in Figure 4.6). Functions of a single categorical variable, i.e., the ICE function for one categorical feature, can be represented by a box plot, instead of a bar plot, because the predictions are not averaged (example in Figure 4.8). Other types of ICE plots (centered and derivative) are proposed in the original paper [49], which are not covered here.

Permutation Feature Importance (PFI)

Permutation Feature Importance (PFI) method was originally proposed by Breiman in 2001 for a random forest (RF) model [51]. The author, when introduced random forests, stated that these models are not easy to understand (they are in fact seen as black-boxes) and that it is necessary to at least understand how the input variables are providing the elevated predictive accuracy. Then, PFI is a global feature importance method, and RF-specific, being used for both classification and regression tasks.

PFI works by measuring the increase in the prediction error of the model after permuting the feature's values, which breaks the relationship between the feature and the true outcome [6]. Breiman, in his paper provided how this is computed for a RF classifier: "Suppose there are p input variables. After each tree is constructed, the values of the ith variable in the out-of-bag examples are randomly permuted and the out-of-bag data is run down the corresponding tree. The classification given for each $\mathbf{x}_n$ that is out of bag is saved. This is repeated for $i = 1, 2, ..., p$. At the end of the run, the plurality of out-of-bag class votes for $\mathbf{x}_n$ with the ith variable noised up is compared with the true class label of $\mathbf{x}_n$ to give a misclassification rate" [51].

The output of this approach, i.e., the global feature importance measure (FS value), is the percent increase in misclassification error rate, for each feature, as compared to the out-of-bag rate (with all feature's values intact). This works for classification. For regression, instead of the error rate, the mean squared error (MSE) is computed. In both cases, a feature is considered important if permuting its values increases the prediction error, because the model relied on the feature for the prediction. On the contrary, a feature is considered unimportant if permuting its values does not change the prediction error, because the model ignored the feature for the prediction [6].

In 2010, Altmann et al. [52] proposed an heuristic method called permutation importance (PIMP) that provides p-values for the importances. Furthermore, also based on PFI, Fisher et al. [53] recently proposed a model-agnostic version of this method called model reliance [6].

Local interpretable model-agnostic explanations (LIME)

Linear models like the one described in equation 2.1 are transparent (white-boxes) and considered easily understandable to humans. These type of models are often used to explain more complex non-linear models (black-boxes), being for this reason called surrogate models. Local interpretable model-agnostic explanations (LIME) proposed by Ribeiro et al. [40] in 2016 is an XAI method in which the authors locally train surrogate models to approximate and explain individual predictions of any type of

black box model. LIME is then a local model-agnostic method, and can be used with any type of data input. It creates a feature summary visualization for tabular data and a saliency map for images and text.

The primary intuition behind LIME is presented in Figure 2.8, where the goal is to locally explain a complex black-box model i.e., explain a single instance of interest x (bold red cross) with two explanatory features (two axes). The decision function f for a binary classifier is represented by the blue/pink background, each colour indicating the combinations of values of the two variables where the complex model classifies the observation with that class. LIME samples instances around x, generating an artificial dataset (dots and crosses), to which a surrogate model (in this case a simple linear model indicated by the dashed line) is fitted to construct a local approximation of the underlying model. The size of the dots and crosses in Figure 2.8 represent the proximity to the instance being explained [40, 44].

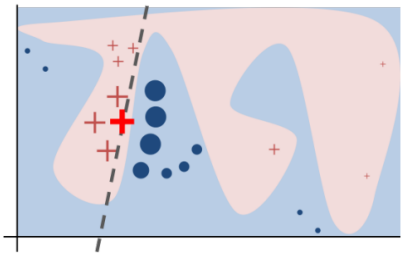


Figure 2.8: Intuition behind LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanatory function, which is here a simple linear model fitted to the sampled instances. Taken from [40].

Mathematically, LIME minimizes the following expression:

$$\xi(x) = \underset{g \in G}{argmin} L(f, g, \pi_x) + \Omega(g), \tag{2.6}$$

where f is the model being explained; g is the surrogate model which can be any model belonging to a class of potentially interpretable models G, such as linear models or decision trees; $\pi_(x)$ is a proximity measure between a sampled instance to x, so as to define a neighborhood around x; $L(f, g, \pi_x)$ is a loss function that measures the discrepancy between g and f in the locality defined by π_x; and $\Omega(g)$ is a measure of complexity for model g (e.g., for decision trees can be the depth of the tree, and for linear models the number of non-zero weights). In practice, the goal is to minimize the lost function $L(f, g, \pi_x)$ to ensure local fidelity between f and g around x, while keeping $\Omega(g)$ low enough to be easily understandable by humans.

In LIME paper, the authors explain the computation of $\xi(x)$ by focusing on sparse linear models (default model used in the method's implementation) with a limited number K of non-zero coefficients. The algorithm shown in Figure 2.9, taken from the paper, describes how LIME finds its solution. There are 3 relevant steps:

1. Interpretable data representation

   Models f and g can operate on different data spaces [44], as "explanations need to use a representation that is understandable to humans, regardless of the actual features used by the (underlying) model" [40]. In this sense, there is some function in LIME implementation that transforms x into its

interpretable version x'. Interpretable representations frequently used for image, text, and tabular data, are, respectively: superpixels, based on image segmentation; groups of words; discretization of continuous features and combination of categorical features. For all data types, a binary vector $x' \in (0,1)^{d'}$ is usually defined indicating the presence or absence of a (group of) pixel(s), word(s), or discretization group.

2. Sampling around the instance of interest

   In order to train the local-approximation model g, it is necessary to create N new data instances z'_i in the interpretable data space around the instance of interest x'. This is done by using perturbations, i.e., "by drawing nonzero elements of x' uniformly at random" [40]. From the perturbed sample z', the sample z in the original data space is recovered to obtain $f(z)$ (used as a label for the explanation model g) and $\pi_x(z)$ (to define the local neighborhood around x). By default, an exponential smoothing kernel[8] is used for the latter. The kernel width σ determines how local the artificial dataset is: A small σ means that an instance must be very close, a larger σ means that instances that are farther away are also considered for the local model g [6]. The big challenge is finding the best kernel or width. Looking at the original python code for tabular data, the kernel width σ is 0.75 times the square root of the number of columns of the training data. It is not clear how the authors arrived at this result, as they do not provide any reason(s) for it. The same happens for text and image data, where σ is equal to 25 and 0.25, respectively. Although it seems just a simple line of code, it is a big issue, as the explanation can drastically change by changing the kernel width [6].

3. Fitting a weighted surrogate model

   Given the dataset Z of perturbed samples (close to the the instance of interest) with the associated labels $f(z)$ and $\pi_x(z)$, a weighted surrogate model g can be fitted. The most common choices for class G are generalized linear models. To get sparse models, i.e., models with a limited number of features, LASSO (least absolute shrinkage and selection operator) or similar regularization-modelling techniques are used, which are useful to explain models with a very large number of explanatory (input) variables [44]. By default, in the algorithm presented above, the K-LASSO method with K non-zero coefficients is used, and weights w are learned via least squares, where equation 2.6 is optimized to get an explanation $\xi(x)$. Ideally, the features with high contribution in LIME (with height weights) are the features that are most important for that specific data point.

Anchors

The authors of LIME, 2 years later, introduced a novel local model-agnostic XAI method based on if-then rules, which they called Anchors [41]. An anchor explanation is a rule that sufficiently "anchors" the prediction locally, meaning that changes to the rest of the feature values of the instance do not change the prediction. The authors guarantee that anchors "are intuitive, easy to comprehend, and

[8] $\pi_x(z) = exp(-D(x,z)^2/\sigma^2)$, where D is a distance function (by default cosine distance for text and euclidean distance L2 for images and tabular data) with width σ.

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ **do**
    $z'_i \leftarrow sample\_around(x')$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  end for
  $w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
  return w

Figure 2.9: Limiting class G to sparse linear models with a limited number K of non-zero coefficients, this algorithm may be used to find an explanation $\xi(x)$ that includes K most important features. Taken from [40].

have extremely clear coverage – they only apply when all the conditions in the rule are met, and if they apply the precision is high (by design)". It can be applied to any type of data.

The primary intuition behind anchors is presented in Figure 2.10, that depicts both LIME and anchors locally explaining a complex binary classifier that predicts either positive or negative (blue or red background) using two instances of interest (+ and -). LIME explanations work by learning a linear decision boundary that bests approximate the underlying black-box model under a perturbation space D, with some local weighting. Anchors approach also deploys a perturbation-based strategy for instance x, using the same perturbation space D, but the coverage is adapted to the model's behavior and making their boundaries clear. The adaption to the model's behaviour can be clearly seen in the anchor on the right of Figure 2.10, as the coverage adapts and gets broader. The same does not happen with LIME, which can be verified by the fact that the explanation on the right is a much better local approximation of the black box model than the one on the left (that did not adapt).

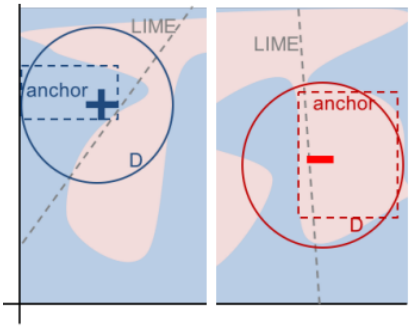


Figure 2.10: Intuition behind anchors vs LIME. There are two instances of interest (+ and -). Contrary to LIME, anchors adapts their coverage to the model's behavior (the anchor on the right is broader), making their boundaries clear. Taken from [41].

Mathematically, an anchor A is a local explanation of instance x if:

$$\mathbb{E}_{D(z|A)}[1_{f(x)=f(z)}] \geq \tau, A(x) = 1 \tag{2.7}$$

where f is the model being explained, being used to predict a label for x and its perturbations (like instance z); A is a set of predicates, i.e., the resulting decision rule, such that $A(x) = 1$ if all its feature predicates are true for instance x[9]; $D(.|A)$ denotes the conditional distribution when the rule A applies,

[9]For example, for the text input instance x = "This movie is not bad.", $f(x)$ = Positive (sentiment), $A(x) = 1$ where A = {"not", "bad"}.

i.e., it indicates the distribution of neighbors of x that include all the predicates included in A; and $0 \leq \tau \leq 1$ specifies the desired level of precision, i.e. only rules that achieve a local fidelity of at least τ are considered (default value of τ is 0.9) [41].

In practice, a rule is constructed until there is statistical confidence concerning their precision. Moreover, the approach consists in choosing a rule that has the highest coverage among all eligible rules (all those that satisfy the precision threshold). There is a trade-off between precision and coverage [6].

Since the artificial generation of the dataset D may lead to a huge number of samples, anchors exploits a multi-armed bandit algorithm [54]. Furthermore, since the number of all possible anchors is exponential, the method uses a bottom-up approach and a beam search [33, 55]. These components are not described here, as they are very complex and out of the scope of this work.

It is important to mention that LIME and Anchors original code implementations (in Python) use an approach to provide a global perspective of the underlying model, by explaining several represented individual predictions of that model, which the authors call submodular pick [40, 41].

Shapley values

The Shapley value is a solution concept from coalition game theory that assigns a payoff to each player according to their contribution to the total payout [56]. This approach was adapted to ML by Štrumbelj and Kononenko [57, 58], by assuming each feature value is a player in a game where the prediction is the payout. The goal is to compute each feature's payoff, i.e., to know how to fairly distribute the payout among the features, which is done by calculating the average marginal contribution of that feature value across all possible coalitions [6]. Shapley values is local a model-agnostic, working for both classification and regression tasks in the tabular domain.

Taking equation 2.1, the average marginal contribution of the ith feature's value for some instance x is given by equation 2.2, where $E(X_i)$ is the mean effect estimate for feature i. So, for this simple additive model, calculating the marginal contribution of a feature i for $f(x)$ it is calculating the feature influence having as baseline the average effect of that feature (the latter is given by formula 2.3), i.e. the contribution is the difference between the feature effect minus the average effect. Summing all the feature influences for instance x, the result is the predicted value for that instance minus the average predicted value:

$$\sum_{i=1}^{p} \phi_i(x) = \sum_{i=1}^{p} (w_i x_i - wiE(X_i)) = f(x) - E[f(x)] \tag{2.8}$$

This result is straightforward, in this case, due to the fact that the linear model is additive (that is, the features do not interact). With the help of coalitional game theory, it is possible to estimate these feature influences for any type of (black-box) model. So, the shapley value of a feature value is its marginal contribution to the payout (model prediction), weighted and summed over all possible feature value combinations. In this sense, the shapley value for feature (utility) value x_i of instance x is:

$$\phi_i(x) = \sum_{S \subseteq \{1,\dots,p\} \setminus i} \frac{|S|!(p-|S|-1)!}{p!}(\triangle(S \cup j) - \triangle(S)) \tag{2.9}$$

where S is a subset of features (and p is the total number of features) and $\triangle(S)$ is the change in prediction caused by observing the values of a subset S of features for instance x: $\triangle(S) = f_S(x) - E[f]$. "The Shapley value is the only attribution method that satisfies the properties Efficiency, Symmetry, Dummy and Additivity, which together can be considered a definition of a fair payout" [6].

Computing Shapley values is computationally expensive so most model-agnostic implementations only estimate approximate Shapley values, such as the approach proposed by Štrumbelj and Kononenko [47], which consists in an approximation with Monte-Carlo sampling:

$$\hat{\phi}_i = \frac{1}{M}\sum_{n=1}^{M}(f(x^m_{+i}) - f(x^m_{-i})) \tag{2.10}$$

where M is the total number of samples drawn at random and with replacement (from the training data); $f(x^m_{+i})$ is the prediction for x, but with a random number m of feature values replaced by feature values from a random data point in M, except for the respective value of feature i; and $f(x^m_{+i})$ is the same, but also replacing the feature value i. The authors guarantee that this approach is an "unbiased and consistent estimator of $\phi_i(x)$" and the approximation algorithm is provided in detail in the original paper [47]. In the case of an additive model, Equation 2.8 holds for instance x and also for each feature value individually.

SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) by Lundberg and Lee [59] is another alternative approach to approximate shapley values. Together with LIME, SHAP is currently the most used method within the category of (local) model-agnostic post-hoc XAI. The authors proposed SHAP values as a unified measure of feature influence, based on shapley values. As stated above, shapley values are used to fairly distribute the payout (i.e., the prediction of the model f being explained) among the features. "A player can be an individual feature value, e.g. for tabular data. A player can also be a group of feature values. For example to explain an image, pixels can be grouped to superpixels and the prediction distributed among them" [6]. This way, like in LIME, SHAP can be used with any type of data, and it also involves the creation of an interpretable data representation. SHAP authors, in their paper [59], introduced the concept of additive feature attribution (AFA) methods, where the (local) explanation of model f, i.e., the explanation of an individual instance x is given using an explanation (surrogate) model g that is a linear function of binary variables:

$$\xi(x) = g(z') = \phi_0 + \sum_{i=1}^{P}\phi_i z'_i \tag{2.11}$$

where $z' \in \{0,1\}^P$; P is the number of simplified input features; and ϕ_i are shapley values. This concept of simplified input features is the same as the concept of interpretable data representation

introduced in LIME, and, similarly, to compute shapley values, a binary vector is defined that indicates the presence or absence of a feature value "in the game". The representation as a linear model of simplified input features is a trick for the computation of the shapley values. When all feature values are present, for x, the instance of interest, formula 2.11 simplifies to [6]:

$$\xi(x) = g(x') = \phi_0 + \sum_{i=1}^{P} \phi_i \tag{2.12}$$

This view connects shapley values and LIME, as both belong to AFA methods. Moreover, according to Lundberg et al. [59], shapley values represent the only possible method in the class of AFA methods that will simultaneously satisfy three important properties: local accuracy, consistency, and missingness. Local accuracy (same as additivity property) states that when approximating the original model f for a specific input x, the explanation's influence values (here calculated as shapley values) sum up to the output $f(x)$, i.e., $f(x) = g(x')$.

Specifically, the SHAP authors proposed KernelSHAP, an alternative, kernel-based estimation approach, which is essentially an adaptation of another AFA method, the LIME method by Ribeiro et al. [40], to estimate shapley values. KernelSHAP estimates for an instance x the influence of each feature value x_i to the prediction, following a similar algorithm as the one presented in Figure 2.9 for LIME. The difference is on the definition of the local neighborhood around x, i.e., in how π_x is defined, which leads to the returning of shapley values as w's of the fitted linear model. So, to obtain weighted shapley values, SHAP authors proposed the SHAP kernel[10]:

$$\pi_x(z) = \frac{M-1}{\binom{M}{z}|z|(M-|z|)} \tag{2.13}$$

In practice, if this kernel was used with LIME algorithm 2.9, LIME would also estimate shapley values! So, kernelSHAP is basically a combination of the AFA methods LIME and shapley values. The authors state that "jointly estimating all SHAP values using (linear) regression provides better sample efficiency than the direct use of classical Shapley equations"[59]. To conclude, the biggest distinction to LIME is the weighting of the (sampled) instances in the surrogate model. LIME weights the instances according to how close they are to the original instance. The more 0's in the binary vector $z' \in \{0,1\}^P$, the smaller the weight in LIME. SHAP weights the sampled instances according to the weight z' would get in the shapley value estimation. "Empty" vectors (i.e.few 1's) and "full" vectors (i.e. many 1's) get the largest weights. "The intuition behind is: We learn most about individual features if we can study their effects in isolation. If a coalition consists of a single feature, we can learn about this feature's isolated main effect on the prediction. If a coalition consists of all but one feature, we can learn about this feature's total effect (main effect plus feature interactions). If a coalition consists of half the features, we learn little about an individual feature's contribution, as there are many possible coalitions with half of the features" [6].

The SHAP authors [59] also proposed other model-specific methods for estimating shapley values, such as DeepSHAP and TreeSHAP. They are more efficient approaches for deep learning and tree-

[10] $\binom{M}{z}$ is read as "M choose z" and is the number of ways to select (a subset of) z features from a set of M features.

based models, respectively. They are not further described here.

Lundberg et al. [60] suggested using $mean(|\phi_i|)$ as a global feature importance estimate, when averaged over all instances and where influence values ϕ_i are shapley values. SHAP global feature importance is an alternative to PFI. However, PFI is based on the increase in model error, while SHAP is based on magnitude of feature influence values, being questioned whether the use of the latter for estimating importance values is reasonable.

Counterfactual explanations (CFEs)

Counterfactuals were firstly introduced as an XAI method by Watcher et al. [61]. The authors created a local model-agnostic method that gives CFEs under the intuition that model explanations should take a similar form to the following one: *"You were denied a loan because your annual income was £30,000. If your income had been £45,000, you would have been offered a loan"*. The decision is followed by a counterfactual, indicating a causal relationship between the input feature "annual income" and the outcome "loan denied".

An agnostic counterfactual explanation can be defined as "the smallest change to the world that can be made to obtain a desirable outcome, or to arrive at the closest possible world, without needing to explain the internal logic of the system" [61]. The smallest change to the world corresponds to the smallest change to the feature(s) values that change the prediction into a desirable predefined outcome. "Multiple counterfactuals are possible, as multiple desirable outcomes can exist" [61].

A naive approach to generate CFEs is searching by trial and error, which involves randomly changing feature values of the instance of interest x and stopping when the desirable outcome is predicted. A better approach is to define a loss function based on different criteria, such as "the smallest change to the world". This loss takes as input x, a counterfactual x' and the desired (counterfactual) outcome y'. Then, an optimization algorithm that minimizes this loss is used. Many CFE-based methods use this approach, differing in their definition of the loss function and optimization method. The original paper [61] minimizes a loss function defined by equation 2.14 that measures how far the predicted outcome of the counterfactual $f(x')$ is from the predefined outcome y' (first term) and how far the counterfactual x' is from the instance of interest x (second term, where d is a distance function[11]), adding a parameter λ that balances the first term with the second term[12] [6]:

$$L(x, x', y', \lambda) = \lambda * (f(x') - y')^2 + d(x, x') \tag{2.14}$$

CFEs authors state that "the choice of optimizer for these problems is relatively unimportant", using the ADAM optimization algorithm [62] (a gradient-based approach) to minimize the loss function 2.14 for all their experiments. The initialization of each run is done with different random values for x' and final counterfactual is the best minimizer of equation 2.14.

[11] The authors suggest using as distance function the Manhattan distance weighted by the inverse median absolute deviation [61].

[12] A higher value of λ favors the first term, meaning the generated counterfactuals have predictions close to the desired outcome; a lower value favors the second term, meaning the generated counterfactuals are close to the instance of interest in terms of feature values.

The goal of this approach is not to provide insights on the inner workings of a black-box model or on its decision-making, but rather to identify and reveal which external factors would require changing in order for the desirable output to be achieved [63]. The authors highlight that "as a minimal form of explanation, counterfactuals are not appropriate in all scenarios. In particular, where it is important to understand system functionality, or the rationale of an automated decision, counterfactuals may be insufficient in themselves. Further, counterfactuals do not provide the statistical evidence needed to assess algorithms for fairness or racial bias" [61]. However, CFEs, such as the previously described XAI methods, represent a first step that balances the achievement of the requirements for trustworthy AI, such as transparency, explainability, and accountability, while potentially increasing public acceptance of automatic decisions. There are other CFE-based model-agnostic (and also model-specific) such as Guided Proto [64], which uses class prototypes to find counterfactual explanations of classifier predictions, and Multi-Objective Counterfactuals (MOC) method [65], which translates the counterfactual search into a multi-objective optimization problem. MOC simultaneously minimizes a four-objective loss function, in which the objectives are: 1)the prediction of a counterfactual x' should be as close as possible to the desired prediction y'; 2) the counterfactual x' should be as similar as possible to the instance of interest x ; 3) change as few features as possible; 4) a counterfactual instance should have feature values that are likely (a training data or another dataset is required to guarantee this) [6]. Comparing with the originally described approach, MOC adds the latter two objectives.

Contextual Importance and Utility (CIU)

Contextual Importance and Utility (CIU) [42] is inspired from MAUT and comprises ideas from all of the methods introduced before (note that it was introduced before the former, in 1996). It is a data and model-agnostic method that provides a unified definition of global and local feature importance that is applicable also for post-hoc explanations, where the *value utility* concept provides instance level assessment of how favorable or not a feature value is for the outcome.

CIU uses core MAUT concepts of feature $importance$, $influence$, and value $utility$ and specifies how they can be estimated for any model $f(x)$ for a specific instance or context x. Similar to LIME and SHAP, simplified input features are also implemented, but CIU specifically uses utility functions $u_i(x_i)$ to transform each feature value $x_1, ..., x_p$ into final utility values, which are constrained to the range $[0, 1]$[13]. Joining all the utility functions, and considering a simple linear model, equation 2.1 for an instance or context x takes the form of a p-attribute utility function (concept from MAUT):

$$u(x) = u(x_1, ..., x_p) = w_1 u_1(x_1) + ... + w_p u_p(x_p) \tag{2.15}$$

CIU estimates the values w_i and $u_i(x_i)$ in Equation 2.15 for one or more input features $\{i\}$ in a specific context x and any black-box model f, where the context is defined by the instance or another set of inputs $x_{\{I\}}$ where $\{i\}$ and $\{I\}$ are index sets and $\{i\} \subseteq \{I\} \subseteq 1, ..., p$. This idea of structuring the input domain was introduced by Främling in 1996 [42], for defining what was called intermediate

[13]In the example of the calculation of Paul's weighted average grade, the final utility value of each course is the obtained grade for that course in percent (a utility function was used to transform the grades from a range $[10, 20]$ to $[0, 1]$).

concepts that make it possible to group features together and introduce higher levels of abstraction with several levels of detail[14]. As most model-agnostic post-hoc XAI methods only attempt to estimate the importance/influence of one feature i for the global output, CIU's intermediate concepts are not further considered here, for simplification purposes. So, from now on, only the case when $\{i\}$ has one single index i and the case when $\{I\} = 1, ..., p$ (the "entire" instance x) are considered. However, it should be noted that the possibility of creating abstraction level of explanations, which is not present in none of the previously described methods, is an advantage that makes it possible to use user-adapted vocabularies and condense the amount of information shown.

Since $u_i(x_i)$ are in the range $[0, 1]$ and $u(x)$ is also constrained to the range $[0, 1]$, the importance of feature i is w_i by definition. However, as stated before, When studying a complex (non-linear) model f, w_i are not known. Contextual Importance (CI) takes the range of variation $[0, w_i]$ as the importance value to estimate, which can be done using equation 2.5 introduced for the ICE (and PDP) method, but considering the utility function, instead of $f(x)$ directly. This gives us an estimation of the range $[umin_i(x), umax_i(x)]$, and CI $in[0, 1]$, which corresponds to the factor w_i in equation 2.15. Then, CI is defined as:

$$CI_i(x) = \frac{umax_i(x) - umin_i(x)}{umax - umin} \tag{2.16}$$

If the model $f(x)$ is linear, then $CI(x)$ should be the same for all/any instance x, like in equation 2.1. In this sense, CI is conceptually identical with global feature importance. If the model is non-linear, then $w_i(x)$ depends on the instance x. The utility values $umin_i$ and $umax_i$ have to be mapped to actual output values $y = f(x)$. If f is a classification model, then the outputs y are typically estimated probabilities for the corresponding class, so $u(x)$ is simply equal to y. Then, in this case, $umax$ and $umnin$, are set to the minimal (MIN) and maximal (MAX) y values present in the training set, i.e. 1 and 0 for classification tasks. Moreover, $umax_i$/$umin_i$ is simply the maximum/minimum value $f(x)$ takes when when feature i is varied over its range while the values of other input features remain fixed.

Contextual Utility (CU) corresponds to the factor $u_i(x_i)$ in equation 2.15, expressing to what extent the current feature value x_i contributes to obtaining a high output utility $u_i(x)$. Considering again a classification model, it expresses how favorable that current feature value $u_i(x)$ is to obtain a high prediction probability for the corresponding class. CU $in[0, 1]$ is defined as:

$$CU_i(x) = \frac{u_i(x) - umin_i(x)}{umax_i(x) - umin_i(x)} \tag{2.17}$$

Besides CI and CU, Främling et al. [66] introduced the term contextual influence (CInfl), which defines feature influence in a similar way as in Equation 2.2. However, it uses $w_i u_i(x_i)$ instead of $w_i x_i$, where w_i and $u_i(x_i)$ correspond, in the CIU context, to $CI_i(x)$ and $CU_i(x)$, respectively. This is depicted in Equation 2.18, where $E(U(x_i)$ is the expected utility value for feature i, which is represented by ϕ_0 for consistency purposes. Since utility $u \in [0, 1]$ for all features, it intuitively makes sense to use the average

[14]Intermediate concepts correspond to the different levels of the inference tree of a rule-based expert system. In the case of selecting the best car, for example, a typical intermediate concept would be "performances", which groups together basic concepts like "power", "weight","top speed" and "acceleration" [42]. This is close to the "coalition" notion for Shapley values.

utility value 0.5 as a constant baseline for all features. This is the default approach, however, any value can be considered for the baseline.

$$\phi_i = w_i u_i(x_i) - wiE(U(x_i)) \overset{\text{CIU}}{=} CI_i(x)CU_i(x) - CI_i(x)\phi_0 \tag{2.18}$$

Contextual influence (CInfl) makes it possible to produce influence-based explanations like for AFA methods (LIME and kernelSHAP), in addition to the original CIU explanations, CI and CU. Another difference is in how the sampling of instances around x is performed, used to estimate $umin$ and $umax$. The approach proposed in the CIU package in R[15] [67] uses all possible values when a feature i is categorical (which can be provided or retrieved from the training data); when a feature i is numerical, the model is sampled using a set of instances consisting of 1) the instance of interest x, 2) x with feature value x_i replaced by the smallest possible value for that feature min_i, 3) x with x_i replaced by the greatest possible value for feature that feature max_i, and 4) a set of instances where x_i is replaced with a random value from the interval $[min_i, max_i]$ (which can also be provided or retrieved from the training data). This approach guarantees exact values for min_i and max_i if $f(x)$ is monotonous.

Concluding, the CIU method provides a unified definition of (global) feature importance, (local) feature utility and (local) feature influence, applicable in different contexts. It provides explanation flexibility based on solid theory, without creating any surrogate model g (like in LIME) or making any linearity assumptions (like in kernelSHAP). Moreover, the underlying idea of CU is also counterfactual by the PDP/ICE approach, i.e. keeping everything else unchanged, what happens if modifying this/these values (also known as “Ceteris paribus” (CP) principle). Moreover, CU values provide information on how to change the feature values (or the world, expression used above when describing the method CFEs) so that a desirable outcome can be obtained, which in CIU corresponds to the highest output utility $u_i(x)$.

2.4 XAI: Evaluation

Evaluation of XAI systems is another important factor in the design process of AI systems, bearing in mind that different properties are needed to assess explanation validity and quality for a specific context [68]. There are two main ways of evaluating XAI systems: objective evaluations (i.e., without user-study), usually using quantitative measures, and human-centered evaluations, usually involving user-studies with either domain experts or lay persons [12]. In the literature, they are widely referred as functionally-grounded and application and human-grounded evaluation approaches, where the latter two belong to the human-centered level with respect to domain experts and lay persons, respectively [35].

Some XAI benchmark studies (see table A.1) have been conducted. However, most of them focus on human-grounded (i.e., user studies are performed) and empirical properties, and the ones that use quantifiable and objective measures are limited and should be improved, as, besides being specific to particular types of methods, they are not consistent and do not follow any specific criteria. This

[15] This approach is applicable to any feature set i and I, including 1, . . . ,N.

shows that the task of evaluating methods for explainability is another big challenge. Moreover, this task becomes even more challenging due to the fact that there is still a lack of a common agreement between the research community about the definition for the term explainability and other similar concepts [11, 12]. A definition of unifying properties and metrics for evaluating and benchmarking explanation strategies is difficult, particularly when human-grounded evaluations are addressed, being necessary to focus on objective evaluations that both validate the efficiency of the method and focus on the human side, aligning the generation of the explanation with the cognitive model of the end-user [33].

Objective evaluations make use of close-ended questions that can be functionally analysed for different XAI methods [69]. According to [70], there is missing in the literature "a standard procedure to measure, quantify, and compare the explainability of enhancing approaches that allows scientists to compare these different approaches". This way, a crucial point within XAI research is to have objective metrics that can assess different properties that together assess the overall explainability and to be able to compare different methods regarding different levels of explainability. This does not mean that human-centered evaluations should not be considered, in fact, they can constitute an additional evaluation approach, where open-ended questions can be used to verify the previously assessed objective levels of explainability and achieve deeper insights [69].

Concluding, the existing papers in academia show that the research activity in XAI field has been growing fast, but in different directions, demonstrating two main challenges, which are the lack of common formalism for defining XAI related concepts and identifying the essential properties scholars should consider in order to make explainability methods understandable and easily accessible for end-users, and, most importantly, non-experts [8–10]. Section 2.2.3 provides succinct and unambiguous definitions of transparency, interpretability, and explainability, with the intent of addressing the first challenge. The second challenge becomes even more important to overcame due to the fact that a large number of different XAI methods exist in the literature. Then, it is important to define a set of evaluation criteria that allow researchers to benchmark them and select the best method to use (considering different contexts and audiences). There is the need to build a comprehensive and consensual benchmark framework for XAI methods that can integrate machine learning workflows and pipelines. This is the main objective of this thesis, which focuses on trying to solve the second mentioned challenge.

2.4.1 Property Identification

As mentioned above, there has been a shift in XAI publications towards organization of the field and methods evaluation. Moreover, there has been an increasing interest in the evaluation and comparison of the existent XAI methods, which can be effectively performed if the relevant properties all the methods are meant to cover are correctly identified. Accordingly, a lot of properties for XAI evaluation have been proposed, most of them overlapping. From the SoTa analysis, 60 properties were identified. Having this huge number of proposed properties in the literature raises a big misunderstanding regarding this topic. Figure 2.11 shows the extent to which these properties have been introduced or mentioned in state-of-the-art surveys. Moreover, the number of surveys that propose objective metrics (i.e., without

user-studies) to assess the respective property is shown with a darker color, next to the total number of surveys that introduce or mention the property. This histogram highlights the lack of a systematic organization of the properties devoted to XAI evaluation, and the lack of quantifiable and objective metrics.

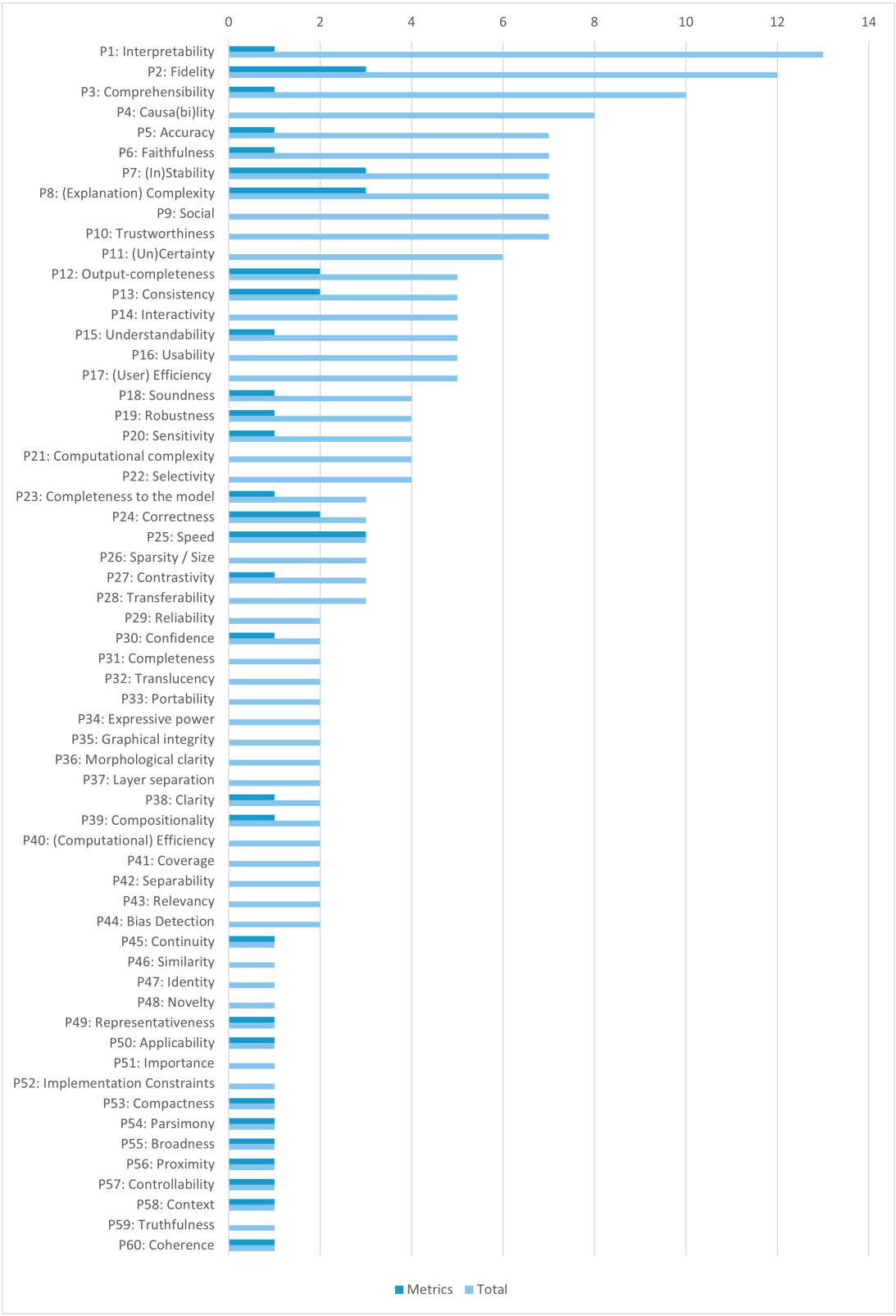


Figure 2.11: SoTa Property Identification: number of surveys that introduce or mention XAI evaluation properties. The number of surveys that propose objective metrics to assess the respective property is shown with a darker blue color.

Chapter 3

XAI Benchmark Framework Formalization

This chapter is dedicated to the building process of the comprehensive and consensual benchmark framework for XAI methods. Section 3.1. comprises the selection and description of evaluation properties. Section 3.2 provides an (R) implementation of the developed framework for tabular data.

3.1 Property Selection

As a large number of different XAI methods exist in the literature, it is important to define a set of evaluation criteria that allow researchers to benchmark them. Since knowledge about this topic is scattered, in this section, property selection and respective formalization is completed, presenting an aggregated view of what to evaluate by arriving to 10 concrete properties on explanation quality and validation. This selection was achieved by reviewing all the properties found in the literature, presented in the histogram of figure 2.11, and "merging" them together in a non-overlapping, clear and consensual way. Each property is introduced in each of the subsections, where both quantitative and qualitative metrics are suggested. It is important to underline that only objective measures (i.e., without user-studies) are used here, which have been mentioned among the XAI community as important to adopt and not sufficiently studied [5, 36]. Table 3.1 summarizes the 10 selected properties, including a brief description, the included properties from Figure 2.11, and the target group(s). The target groups include the (X)AI developers (dev), deployers (dep), and end-users, who are all somehow affected by all of the properties. However, in table 3.1, the bold check indicates the prominent group.

3.1.1 Representativeness

Representativeness has not been commonly used as a property in the literature to evaluate XAI techniques. However, it comprises the scope and level of dependency, which are widely mentioned in the SoTa surveys as a taxonomy to separate the explainability methods. Thus, the representativeness

Table 3.1: 10 selected properties for evaluation and benchmark of XAI methods. The coverage of these properties helps in the achievement of 2 XAI goals: understandability and usability of (X)AI systems.

Property	Description	All properties	Dev	Dep	End-user
Representativeness	Describes the extent to which the method "looks" inside the black-box model	P31, P32, P33, P49, P50	**X**	**X**	X
Structure & Speed	Describes how and how fast the explanation is provided	P8, P34, P35, P36, P37, P38, P39, P51 & P21, P25, P40, P52	X	X	**X**
Selectivity	Considers the size of the explanation	P22, P26, P41, P53, P54, P55, P56	X	X	**X**
Contrastivity	Assesses how contrastive the explanation is w.r.t. some reference target	P27, P42	X	X	**X**
Interactivity	Assesses the extent to which a user can control or explore the explanations	P9, P14, P43, P57, P58	X	**X**	**X**
Fidelity	Assesses if the method creates a surrogate model or makes linearity assumptions	P2, P5, P23	**X**	X	X
Faithfulness	Describes how reliable the explanation is w.r.t. the black box	P6, P12, P18, P24	**X**	X	X
Truthfulness	Describes how reliable the explanation is w.r.t. the true world	P4, P10, P28, P44, P59, P60	X	X	**X**
Stability	Assesses how stable and consistent the method is	P7, P13, P19, P20, P29, P45, P46, P47	**X**	**X**	**X**
(Un)Certainty	Assesses if the method provides (un)certainty measurements together with the explanation	P11, P30, P48	**X**	**X**	**X**
Goals: Understandability and Usability	The 10 properties above lead to these 2 main goals: understandability and usability of (X)AI systems	P1, P3, P15 & P16, P17	**X**	**X**	**X**

property assesses the extent to which the generated explanation addresses the entire model behavior considering its scope and translucency/portability. The former, which corresponds to the 6th column in table A.3, indicates if the method aims to explain the entire model behavior (global explanation) or a single prediction (local explanation) [8], while the latter indicates the level of dependency from the black box model f, i.e., the extent to which the explanation relies on looking into the internal dynamic of the model, such as the model's parameters [71, 72]. Thinking of a white-box model, it reveals all the mathematical operations and parameters, and therefore it is fully translucent. This is one extreme. At an intermediate level are model-specific methods, which rely on the inner workings of the underlying model f and therefore are very translucent (e.g. GadCAM [38]). On the other extreme, there are model-agnostic methods, that do not consider any internal parameters of f and have zero translucency (e.g. LIME [40]). Portability, which describes the range of ML models which the explanation method can be applied to, is inversely proportional to translucency [71] and corresponds to the 5th column in table A.3. In this sense, model-specific methods are highly translucent but have low portability, and model-agnostic methods have low translucency but are highly portable. The advantage of high translucency is that the method can rely on more information to generate explanations. The advantage of low translucency is that the explanation method is more portable [6]. The portability of a method can also be assessed by considering if it needs access to the training data to compute an (new) explanation. Furthermore, XAI methods can be data-specific or data-agnostic. Besides scope and portability, this criterion, i.e., the applicability of the method, is also used here has a metric to evaluate representativeness, but in terms of type of input data the explanation can be applied to. A design choice needs to be made by developers regarding the representativeness of the method by selecting an explanation type suited for a specific context (considering both the data and the model from which the explanation is being generated). For

this reason, this property is only evaluated qualitatively to compare and categorize different XAI methods [72]. The metrics to (qualitatively) assess are: scope, portability, and applicability, and can be directly formalized to assess any type of XAI method. Nonetheless, there seems to be a preference between scholars for methods that are highly portable (model-agnostic) and that can be used for all type of inputs (data-agnostic).

Representativeness is a useful property for AI developers when employing an XAI method in their model design, as it gives information about the extent to which the method "looks" inside the black-box. It also concerns AI deployers, specifically at the applicability level, as it is relevant for this target group to know if the method can be used for any type of input data. For example, if the AI deployer is an hospital, it may be more useful to employ a data-agnostic method, as in this domain there is often image (e.g. MRI or CT images), tabular (e.g., clinical data) and text (e.g., doctor annotations) data. However, the hospital may only want to employ an XAI system regarding a specific problem, for example, brain tumor detection trough MRI images, and, in this case, a data-specific method for image data is suitable.

3.1.2 Structure & Speed

The structure property assesses the composition (i.e., structure) of the explanation, considering it should be presented in a way that increases its clarity to the user [72]. It has not been widely used as a property in the literature to evaluate XAI methods, although other properties that are referred to as important to consider at this stage can be included in the structure of the explanation, as illustrated in the second column of table 3.1. Then, the structure includes four main properties, here used as qualitative evaluation metrics: expressive power, graphical integrity, morphological clarity, and layer separation [69, 71]. Speed of the explanation is also included together with this property, as it concerns how much time the explanation takes to be generated, bearing in mind that this should be fast enough to be employable in real-world applications [11].

Expressive power describes the language of the explanation (if-then rules, histograms, textual explanations, etc.). It can be used to assess and make a comparison between different methods, as some representation formats are usually considered to be more easily understandable than others [72, 73]. For example, rules and counterfactuals, by providing a logic structure, are often seen as more suitable for the lay end-user [33]. Another preferred format is textual explanations [74]. Furthermore, the "language" can include the usage of higher-level information, abstractions, or suitable terminology, which are seen as approaches that increase the explanation clarity [72]. As an example, if the end-user is a person trying to understand why his/her loan was declined, suitable, simple, and clear terminology should be displayed so that it is easier for that person to easily understand the reasons and change his/her behaviour. Of course, this suitable terminology, or the use of abstractions, should be an aspect to be discussed and agreed between the AI developer and the AI deployer, where in this example the deployer is a bank that it is using an XAI system to predict if someone should or not be accepted for a loan. This could mean that the terms used in the explanation are different from the features given as input to model [72]. Graphical integrity assesses how well the explanation reflects the relevance of fea-

tures or of parts of the explanation. The final explanations should highlight the most important features and differentiate those with positive and negative attributions. Morphological clarity assesses the extent to which the relevant features are clearly highlighted and not noisy or confusing to the user. Lastly, layer separation is used to check if the explanation visualization omits or occludes the original input instance (it can be a row of a tabular data set, an image, or a sentence in a text) which should be visible for user inspection [69, 71]. Morphological clarity and layer separation are particularly relevant to consider when dealing with image data.

In this sense, the structure and speed property concerns how and how fast the explanation is displayed, rather than what is explained or how it is obtained. For this reason, the structure is usually only qualitatively evaluated [72]. The speed is evaluated by performing a runtime analysis that can consist in measuring the time, in seconds, for a method to generate a single explanation or the number of explanations per second [75]. Some formats and representations should be more easily understandable than others, and it is an important aspect to be considered during an AI system design so that the explanation is presented to the end-user with maximum clarity and minimum noise and ambiguity. Structure can also, and should, be assessed by conducting a human-grounded evaluation. A good structure leads to user efficiency and good understandability of the method. A fast method leads to computational efficiency and practical usability of the method. This is why structure and speed is a property that prominently concerns the end-users of an AI system.

3.1.3 Selectivity

Selectivity has been widely mentioned in the literature to evaluate XAI techniques, belonging to the group of properties of the so-called human-friendly explanations. In this sense, this property mostly concerns the end-users of an AI system, as it assesses the size of the explanation, bearing in mind the human cognitive capacity limitations. It has also been referred to as compactness [72] or sparsity [1, 70]. It is a common view between scholars that XAI methods should be able to provide selective explanations, making the explanation very short, displaying only 1 to 3 reasons (this number may vary from person to person), even if the world is more complex [6, 69]. In this sense, there is a preference towards methods that can focus on only a few causes deemed to be necessary and sufficient to explain a particular instance, and not all of them [76].

The selectivity of a method is often evaluated by directly measuring the explanation (absolute or relative) size [72]. This metric depends on both the type of explanation (expressive power) and the type of data. Examples include: the number of features in an explanation, average path length in a decision tree, reduction w.r.t. complete data sample, the number of decision rules in a set [72], number of features that can be ignored (effective complexity), mutual information between original samples and corresponding features extracted for explanations [36], or image entropy or the file size of the compressed heatmap image [76]. When dealing with counterfactual explanations, the explanation size can be assessed by the number of generated counterfactuals and for the counterfactual itself by measuring how similar the explanation is to the original instance, which is also called proximity [11]. Usually, small changes are

desired to go from the original instance to the counterfactual one - selective counterfactual explanation [72]. This can be measured using distance metrics, or through the number of changed features.

Closely related to the stability property is requiring that a counterfactual explanation is in the proximity of the actual instance to prevent that the explanation is an outlier, which is specifically undesirable for the property truthfulness [72]. Finally, a qualitative metric should be added, which consists in assessing whether XAI methods have a parameter to tune the explanation size. This is relevant because the end-user can be an expert or a lay-user that may want access to the complete set of reasons for a particular decision or just part of it. This is particularly relevant when dealing with tabular data with a big number of input variables.

3.1.4 Contrastivity

Contrastivity has been mentioned in the literature to evaluate XAI techniques, also seen as a human-friednly property. For this reason, it prominently concerns the end-users of an AI system. It studies the discriminativeness of an explanation in relation to a ground-truth event or target, aiming to facilitate comparisons between them [72]. Another property named separability is closely related to contrastivity which implies that data instances from different populations must have dissimilar explanations [77].

Humans tend to think in counterfactual cases, i.e., "How would the prediction have been if input X had been different?" [6, 8]. In this sense, explanations that present some contrast between the instance to explain and a point of reference are preferable. However, this makes the explanation application-dependent because it requires some ground-truth point for comparison [77]. And this often depends on the type of method, on the instance to be explained, but also on the user receiving the explanation [6]. A way of presenting contrastive explanations is to use a standard reference point. Methods that present counterfactual explanations are gaining a lot of attention because they are contrastive to the current instance [12], being this the predefined reference point. Another way is to compare to a predefined output, like the average prediction. In this sense, a qualitative metric should be included here, which consists in assessing whether the generated explanation provides some contrastivity, considering the mentioned criteria.

Nauta et al. [72] suggest using a quantitative metric, Target Sensitivity, which assesses the contrastivity relative to another class, bearing in mind that class-specific features highlighted by an explanation should differ between classes. This is particularly relevant when an adversarial attack happens, which fools the underlying model f such that it makes a different prediction for a slightly perturbed input. In that case, a different prediction should also lead to a different explanation. Target Sensitivity can be measured using different distance metrics between the explanations, histogram intersection or the structural similarity index measure (SSIM) between two heatmaps (depending on the type of data).

3.1.5 Interactivity

Interactivity has been widely mentioned in the literature to evaluate XAI techniques, also belonging to the group of human-friendly explanations. It assesses if the explanation is displayed in an interactive

form, bearing in mind the user social context [70, 74]. If so, it is relevant to assess the extent to which the user can control the explanation [72], which can be done with and without an user-study. This property is linked to the idea that explanations are social. They should be seen as a conversation (interaction) between the explainer (XAI system) and the "explainee" (end-user), "implying that the explainer must be able to leverage the mental model of the 'explainee' while engaging in the explanation process" [8].

When exploring the interactivity of the explanation, one should take into consideration the social context of the AI system and the target audience [71]. This means that providing a meaningful explanation varies according to the application domain and the specific audience. In this sense, this property is application-dependent, and the way to build meaningful and controllable explanations should be discussed and agreed between the AI developer and the AI deployer, where the final goal is the creation of an interactive tool with the specific XAI method and dataset. If possible, it is helpful to include experts from the humanities (e.g., psychologists, sociologists and anthropologists) [6].

There are already interactive tools for XAI, however they require a considerable level of expertise (e.g. Manifold and ActiVis designed for deployers Uber and Facebook, respectively). There is one interactive tool that is easily controllable by the user – LRP tool[1]. It provides four LRP demos [79]: a simple explanation demo based on NNs that predict handwritten digits and were trained using the MNIST data set; a more complex demo based on a neural network implemented using Caffe that predicts the contents (classes) of the image; a demo that explains classification on natural language where a neural network predicts the type of document; and a visual question answering demo based on recurrent attention units using the VQA data set. All of the demos are very intuitive to use, being a great example of how an interactive tool should look like (for a lay audience and a simple application).

After an interactive system is developed, its evaluation of interactivity can be done by discussing how controllable it is and why the controllable format improves the quality of the explanations, showing examples, or quantified, by measuring the improvement of explanation quality after human feedback, where the user is seen as a system component [72]. The majority of the methods does not provide any interactive (demo) tool. So, firstly, it is important to (qualitatively) assess whether the XAI method provides any possibility of interaction, and how favorable it is for the creation of one.

This property mostly targets the end-users and deployers of an AI system. For the latter, it is of their interest to easily assess how the XAI system works and how the visualization of the explanations is provided. For example, a deployer can be a hospital that is looking for an XAI system to predict and explain the presence of a brain tumor in a MRI image (for the doctors, or even patients, as end-users, to use). In this case, the LRP tool presented above would be very helpful for the deployer to check the explanations. Then, an interactive tool would be built together with the AI developer to that specific medical application. This is crucial because the effectiveness of interactivity is highly coupled between the algorithm and the user [68].

[1] Layer-wise Relevance Propagation (LRP) is a local model-specific XAI method that explains a NN classifier's prediction specific to a given data instance by attributing relevance scores to important components of the input by using the topology (back-propagation approach) of the learned model itself [78].

3.1.6 Fidelity

Fidelity has been widely used in the literature to evaluate XAI techniques, sometimes referred to as model or output completeness. It assesses if the explanation is created by a surrogate model or system g or if any linearity assumptions regarding the model are made. It is important to consider this property because methods that use a surrogate model (also known as proxy model [80]), just by using it, are decreasing the fidelity, and therefore degrading the accuracy of the explanation provided [81]. When this happens, the extent to which g can accurately cover the black box decisions should be evaluated [9, 33]. If linearity is assumed, it may happen that if the underlying model is highly non-linear, the explanation is not correct, as fidelity to the model does not exist. High fidelity is one of the most desired properties of an explanation because an explanation with low fidelity is not in agreement with the original predictive model, and therefore it becomes useless [71]. XAI methods that do not create any proxy task should then be preferred over the ones that do, as the first have 100% fidelity. This is a validation property that is crucial for AI developers to consider when employing an XAI method in their model design.

Fidelity can be directly evaluated by quantifying how closely the surrogate g approximates or agrees with the black box model f predictions [70]. Surrogate Agreement (SA) metric is used for this aim, by comparing the prediction of black-box model f and surrogate model g when applied to the same input samples [33]. Preservation Check (PC) metric has also been used between scholars to evaluate fidelity, which consists in comparing the prediction of f when applied to data based on the explanation as input and to the original input sample [72].

3.1.7 Faithfulness

Faithfulness has been extensively used in the literature to evaluate XAI techniques, sometimes referred to as correctness, truthfulness, or soundness. It assesses the capacity of an explainability method to faithfully represent the black-box behaviour (globally or locally), i.e., to reliably describe the underlying decision structure of the analyzed model [63, 75]. Here, model-specific methods are preferred, as they rely on the internals of the model. It is important to emphasize that fidelity and faithfulness are not the same although sometimes presented as such; a developer can always build another model that gives the same predictions as the original one for all instances (high fidelity) but has arbitrarily manipulated explanation maps (low faithfulness) [72]. Even when explanations are of high fidelity to the underlying models, they may fail to represent the model behaviour under normal conditions [74]. Therefore, both properties should be evaluated separately.

Faithfulness can be evaluated regarding different model tasks. A widely used metric is Incremental Deletion (ID), which strategy is to incrementally remove each of the input features[2] deemed relevant by the explainability method, in either descending or ascending order, and measuring the change in the output of the predictive model f [82]. A locally faithful explanation should result in a wrong decision by model f when the k% most important features in the explanation are removed from the input sample.

[2]Here, the input features can be different variables, according to the type of data input. For example, for images the input features are pixel values; for text, each word can correspond to one feature; for tabular data, each feature usually corresponds to each column.

The criticism on metrics like ID is that removal of features, such as setting them to zero, can lead to out-of-distribution samples, violating one of the core assumptions in ML: the training and evaluation data come from the same distribution [72, 83]. One solution is to replace "removed" features with values from the original data distribution. Another solution is to retrain the model on the perturbed data, which leads to the next metric, RemOve And Retrain (ROAR). The ROAR metric [83] consists, again, in incrementally removing the fraction of input features considered to be most important according to the XAI method but this time measuring the change to the original model accuracy upon retraining. A globally faithful method explanation will identify features as important whose removal causes the most damage to model test accuracy. This approach guarantees that the training and evaluating data come from the same distribution. There are different ways of assessing the change in the model output/performance, including reporting the average change in log-odds score, AUC, steepness of curve, number of features needed for a different decision [69], or the correlation between the importance assigned by the XAI method to each feature and the respective effect on the decision/accuracy of the predictive model [82]. ID and ROAR should be compared with other baselines, such as a random ranking. This serves as a control, representing a lower bound in performance that all explainability methods are expected to outperform [83]. It is important to confirm that the meaningful features predicted by the model are really meaningful [70]. Moreover, the deletion of a single feature also allows to check for specific properties, such as the "null attribute" indicating that omitting a feature that has no effect on the output of the model, should have an importance score of zero [72]. Note that these metrics can actually be seen as XAI methods themselves, using a similar idea to PFI-based methods: they measure "the increase in the prediction error of the model after we permuted the feature's values, which breaks the relationship between the feature and the true outcome" [6]. In this sense, this analysis should be taken carefully.

Another way to evaluate the faithfulness of a XAI method for explainability is by training a white-box model as the black-box model - White-Box Check (WBC). A white-box model is fully transparent and therefore explainable by itself, so the explanation can be compared with the true reasoning of the predictive model to evaluate how similar they are [72]. This comparison can be performed globally or locally, depending on both the white-box model and the XAI method. The similarity between explanations should be qualitatively assessed. When possible, it can also be quantitatively assessed by comparing directly the feature contribution values. Instead of a fully transparent model, [84] used a RF and compared the explanations with feature importance scores given by the black-box model. However, these feature importance scores are calculated through the PFI method introduced in section 2.3.3 which does not show the true reasoning behind the original model, but it is rather another explainability method that is adopted by the library where the RF model is being used.

An explanation that looks reasonable to a user is not guaranteed to also be correctly reflecting the behaviour of the model. Hence, it is important to guarantee that an explanation is both truthful (next property) and correct, i.e., it should also cover the fidelity and faithfulness properties. For example, an explanation highlighting snow in the background to distinguish between a husky and a wolf [40] (see figure 3.1) is not true in the real word, but it is true to the model, as it is showing the reasoning of a bad classifier [72]. The opposite can also happen; an explanation may be true but incorrect. Thus, it is

essential to evaluate all properties, specifically truthfulness, fidelity, and faithfulness in an independent way. Evaluating the faithfulness is therefore different from the reasonableness to the user [72], and it should be guaranteed by AI developers when employing an XAI method. It does not concern the deployers or the end-users.

3.1.8 Truthfulness

Truthfulness has been widely mentioned in the literature to evaluate XAI techniques, often referred to as trustworthiness as seen in Figure 2.11 (truthfulness was the chosen word to not confuse with the higher concept of trustworthy AI). It is a property that prominently concerns the end-users of an AI system, as it indicates whether the explanation is in concordance with the user's true world. This includes being accordant with prior relevant domain knowledge and beliefs of the "explainee" [69, 72] and suitable in other (unfamiliar) situations (this is human-friendly) [6, 71], but also be able to detect models with bias [80] and discover new insights because the true world can also be unexpected. When the explanation is consistent with prior domain knowledge and beliefs of the end-user, it motivates a positive attitude towards the AI system [68]. This usually also increases the confidence of whether the system will act as intended when facing other dynamic real-world situations [5, 77]. These factors help increasing the explanations plausibility and reasonableness [72]. However, model explanations may sometimes help to discover new and relevant insights in a certain field, especially in areas where there is not much domain knowledge yet [44]. Moreover, explanations can help to detect bias in the underlying model, which helps in model improving and debugging so that the best AI system is displayed to the user. This is the case of the example "Husky vs Wolf" classifier given above; bias was detected, the model can be improved.

(a) Husky classified as wolf (b) Explanation

Figure 3.1: Original image data and explanation of a wrong model's prediction in the "Husky vs Wolf" task. Taken from [40].

To evaluate the agreement with domain knowledge, explanations can be compared with an (domain expert) annotated data set that are seen as ground-truth explanations [72]. However, this is often not available and human-grounded evaluation (i.e., with user studies) is usually more suitable in this case. For example, one metric could be the user's accuracy in finding the better model by only looking at explanations [68]. Here, two objective metrics are suggested, which consist in comparing different XAI methods across the models (Methods Agreement) and comparing different XAI models across the methods (Models Agreement). By comparing methods and evaluating how consistent they are and how similar are their results, it is possible to create a measurement of confidence regarding their use. Furthermore, combining various techniques can provide more additional insights [44]. By comparing different models, it is possible to understand how they differ from each other, even when they offer similar performance measures, probably because their outputs is based on different features and relations extracted from the same training data (this is known as "Rashomon effect"). This is useful to reveal

the capacity of an XAI method to detect bias or missed relationships and discover insights about the black-box model. For example, a rigid model may miss relationships or interactions that might be found by a more flexible one, which consequently provides suggestions for improvements of the former [44].

3.1.9 Stability

Stability has been widely used in the literature to evaluate XAI methods. It assesses how stable and consistent the method is. Two levels of stability need to be considered: stability for identical inputs, and stability for similar inputs. Identical data instances must produce identical explanations [77]. Similar data instances (input samples with the same label and slightly different feature) must produce similar explanations [10, 71]. These axioms together ensure the coherency of explanations [70]. If this does not happen, then the XAI method is unstable, which can be the result of a high variance or the non-deterministic components of the method [11]. A deterministic method will always give the same explanation given the same instance. Conversely, a non-deterministic method may give different explanations for the same instance. For example, random perturbation and feature selection methods that LIME uses result in unstable generated interpretations. Consequently, different explanations can be generated for identical/similar data instances, which is problematic for deployment [63].

Several metrics have been proposed to evaluate stability, particularly to measure the similarity between neighbouring input samples (different criterion can be used to select the neighbours), including Location Instability, Local Spearman Rank Correlation coefficients, Top-kintersection [69], Lipschitz Continuity [33] Cosine Similarity, Rule Match, Normalized Distance [72], or average feature-variability (shapash python library). The selected metric usually depends on the type of explanation and/or data input in use. Consistency has been commonly referred in the literature to assess explanations between two functionally equivalent models, i.e., between two models that give the same outputs for all inputs despite having different implementations and architectures [69]. A quantitative metric is Implementation Invariance which assesses whether the explanation method is invariant to specific implementations of the predictive model by validating whether two implementations that give the same output for an input, also get the same explanation [69, 71]. For methods that do not consider the internals of the black box but rather observe input and output, this will happen with the same error as for the Identity metric defined above, as explanations do not change across different model implementations or architectures. Therefore, this metric can be applied to model-specific methods that consider specific internal parameters of the black box model.

Concluding, the stability property contributes to the reliability of the method, meaning it should have the ability to maintain certain levels of performance independently from the parameters or from the input data [9]. As mentioned above, it desirable that identical inputs have identical explanations. In practice, this can address to what extent the explanation method is deterministic, which is usually a design choice [72]. This property is also very important to guarantee the truthfulness property. A stable system is desired in a way that it guarantees generalization beyond a particular input or generalization to new contexts [72]. As an example, “if we say that a second balcony increases the price of a house, then

that also should apply to other houses (or at least to similar houses)" [6]. Moreover, it should also be ensured that data instances with change in all but one feature must generate explanations which magnify the change [77], or that if a feature increases or stays the same, the impact of this feature should not decrease [70]. These are all situations related with the stability of an XAI method. Stability is a validation property for AI developers to consider when employing an XAI method in their model design. It also concerns deployers and end-users. The former need to guarantee that an explanation does not change when checked twice by the latter (for example in two different days). In the same way, users do not want a completely different explanation for two similar cases.

3.1.10 (Un)Certainty

Besides explanations, providing prediction uncertainty regarding both the black-box model and the XAI method has been identified as an important factor for both developers, deployers and end-users [68]. On a first level, the explanations should reflect the certainty of the ML model, i.e., the confidence measure of the black box prediction. An explanation that includes the model's prediction probability, as well as its certainty increases its usability [6, 10, 71]. One way of providing a confidence measure about the underlying model is to obtain and show its accuracy, so that the target user has an idea of how good the performance of the black box predictor is [9]. On a second level, and most importantly, the explanations should reflect their own certainty. Not only the ML itself, but also its explanations, are computed from data and, hence, are subject to uncertainty [12, 36]. Moreover, it is important to consider how the explanation was generated, such as the presence of random generation or sampling [33].

In this sense, providing explanations together with a measure of its uncertainty is a desired property for XAI. One way of providing this information is to measure the fidelity, faithfulness, or likelihood of the explanation [69]. Another property that is closely related with certainty is the novelty of the explanation [71], which assesses if the explanation reflects the instance of interest, of which the prediction is to be explained, comes from a "new" region in the feature space that is far away from the distribution of the training data. The higher the novelty, the more likely it is that the model will have low certainty due to lack of data. In such cases, the model may be inaccurate, and the explanation may be useless [6]. One way of providing this information is to locate the instance in the distribution of the training data [71]. These aspects can be qualitatively evaluated, by assessing if the method provides any (un)certainty measurements together with the explanations, i.e., the level of transparency.

3.1.11 XAI Goals and Summary

The 10 properties included here can help achieving two main goals: understandability and usability of XAI methods. Understandability is often referred to as interpretability, which was considered as an XAI level in subsection 2.2.3. If a method is able to support user understanding and comprehension of the black-box model decision strategy and predictions [68, 75], this goal is successively achieved [71]. The actual usability of explanations from the point of view of the end-user [33] depends on the target audience and context [71].

These desired properties of XAI approaches that collectively give an aggregated perspective of what to evaluate were chosen based on an extensive systematic literature review, with particular emphasis to include all of the examined properties in Figure 2.11, reducing semantic overlap, and grouping various terms that describe the same property. In summary, explainability is a multi-faceted concept [72] and this is shown by the 10 selected properties. Several aspects of explainability can be evaluated, regarding explanation quality or method validation and its target group. It is important to get insight into all properties so that a fair trade-off can be achieved. Some properties might be more relevant than others in different contexts, and it is on the AI developer and deployer to choose the best technique(s), bearing in mind the extent to which each property can contribute to each of the requirements for trustworthy AI. For example, if it is important to specifically favour human agency and oversight, then the properties structure and interactivity should be carefully evaluated. In order to objectively assess and compare new and existing XAI approaches, this collection of evaluation properties with respective metric(s) formalization can help in a more thorough and inclusive evaluation and comparison. The benchmark framework presented in this section gives AI developers, deployers, and end-users practical resources to assess each of the 10 properties while utilizing a common formalism and taxonomy, which promotes the uniformity that is lacking in XAI field.

3.2 Metrics Formalization for Tabular Data

The proposed framework is application-agnostic (in terms of application domains) and can be applied to any type of method and data, specifically the properties should be covered to the maximum extent by any method. However, some metrics depend on the type of data (or method). For that reason, Table 3.2 formalizes the metrics specifically for tabular data, which, when necessary, can be accordingly adapted to other data types, as suggested above. The first column refers to the property, the second column introduces the metric (Q - qualitative, q - quantitative), and the third does the respective metric formalization for tabular data (when a metric is specific for a type of method, it is stated in italic). Note that some properties (or metrics) formalization is independent from the type of data or method. The code developed in R to implement the quantitative metrics is available as opensource on Github and it is ready to be used for tabular datasets and both classification and regression problems - see file “Benchmark.R”. Only the explanation of the quantitative metrics is given here, and the relevant functions for each are mentioned. For the other metrics, the description given in Table 3.2 is intuitive. All metrics, whether qualitative or quantitative, should be accompanied by careful and relevant discussion. It is notable that FS methods are more easily compared, as these provide specif attribution values for each feature.

3.2.1 Quantitative metrics

Selectivity

In shapash, a compacity metric is introduced to measure the explanation size of FS explanations. Here, this metric is adapted for R, where the function *min_nb_features()* is implemented with the goal

Table 3.2: 10 selected properties for evaluation and benchmark of XAI methods and respective metrics formalization for tabular data.

Property	Metric	Formalization for Tabular Data
Representativeness	Scope (Q)	Local (L) vs Global (G)
	Portability 1 (Q)	Model-Specific (S) vs Model-Agnostic (A)
	Portability 2 (Q)	Bool: Does the method needs access to the training data to give a (new) explanation?
	Applicability (Q)	Data-Specific (S) vs Data-Agnostic (A)
Structure & Speed	Expressive Power (Q)	Provide the language of the explanation (type of output)
	Graphical Integrity (Q)	Bool: Does the explanation differentiates bewteen features with positive and negative attributions?
	Morphological Clarity (Q)	Bool: Does the explanation displayed highlights the most relevant features in a clear way?
	Layer Separation (Q)	Bool: Does the explanation includes the original input instance? (only for local methods)
	Runtime Analysis (q)	Time per explanation (in seconds)
Selectivity	Explanation Size (q)	*FS* - minimum number of features needed for the explanation to be close enough to the one obtained with all features
		Rules - Number of conditions in a decision rule
		Data points - Number of changed features
	Size Parameter (Q)	Bool: Is it possible to adjust the explanation size?
Contrastivity	Level of Contrastivity (Q)	Bool: Is the explanation contrastive?
	Target Sensitivity (q)	*FS* - L2 norm between explanations (original and "new")
		Rules - Percentage of features in the original conditions that are in the "new" conditions
Interactivity	Possibility of Interaction (Q)	Bool: Is it provided a (demo) interactive tool?
Fidelity	Surrogate Agreement (q)	Ratio between the prediction of f and g when applied to the same input samples
	Preservation Check (q)	Ratio between the prediction of f when applied to data based on the explanation as input and to the original input sample
Truthfulness	XAI Methods Agreement (Q)	Compare explanations between methods for the same model
	Models Agreement (Q)	Compare explanations between models for the same method
Faithfulness	Incremental Deletion (q)	*FS* - Incrementally remove each of the input features deemed relevant by the local explanation and measure the change in the output of the predictive model f
	ROAR (q)	*FS* - Incrementally remove each of the input features deemed relevant by the global explanation and measure the change to the original model accuracy upon retraining
	White-Box Check (Q)	Compare the explanation with the true reasoning of the white-box model
	White-Box Check (q)	*FS* - and quantitatively compare their similarity
Stability	Identity (q)	*FS* - Calculate feature variability for the same instance
	Similarity (q)	*FS* - Calculate feature variability for similar instances
(Un)Certainty	Level of Transparency (Q)	Bool: Does the explanation provide any measure of (un)certainty?

of determining the minimum number of features needed for the output of a FS explanation to be close enough to the one obtained with all features. The closeness is defined via the following distances:

- For regression:

$$distance = \frac{|output_{allFeat} - output_{currentFeat}|}{output_{allFeat}} \tag{3.1}$$

- For classification:

$$distance = |output_{allFeat} - output_{currentFeat}| \tag{3.2}$$

The *distance* is an optional parameter of the function, which default value is 0.1, i.e., when the the explanation with the current number of features ($output_{currentFeat}$) is 10% close to the explanation provided using all features ($output_{allFeat}$), the algorithm stops and returns the current number of features as the output of the Explanation Size metric. In other words, the selected number of features explain 90% of the underlying model. The function *min_nb_features()* can be applied to any FS method, as long as the (global or local) summary values for each feature is provided (parameter *phi*).

For rule-based methods, the size of an explanation can be measured through the number of conditions in a decision rule. For methods that return data points as explanations (such as CFEs) the number

of changed features of the closest data point (counterfactual) can be used for as a proximity metric [3]. A function *mean_size()* is added to compute the mean size of local explanations over n instances of the training data (default is 200, randomly selected). This function is specific for each of the selected methods in Section 2.3.3.

Contrastivity

Nauta et al. [72] reported that Target Sensitivty metric has only been used for heatmaps. Here, it is extended for tabular data, particularly for classification problems. For FS methods, the L2 norm (euclidian distance) between explanations before and after the adversarial attack can be computed using "wavethresh" R package (function *l2norm()*). The higher the score, the lower the similarity, and therefore the better the target sensitivity - a large difference between the explanations is desired. For rule-based methods, this metric can be formalized by checking the percentage of features in the original conditions that also appear in the "new" conditions, i.e., in the obtained rules after the adversarial attack. In this case, the lower the score, the better. For tabular data, the adversarial attack can be simulated by finding the closest instance x' to the instance of interest x that changes the target output to another (predefined) class. This is the output of CFEs, so the first counterfactual returned by this method may be used for this purpose (see Section 4.2.1).

Fidelity

As stated above, the Surrogate Agreement (SA) metric compares the prediction of the underlying black-box model f ($output_f$) with the prediction of the surrogate model g ($output_g$) when applied to the same input instance. Preservation Check (PC) compares the prediction of f when the input is an instance based on the explanation ($output_{exp}$) with the original input instance ($output_{orig}$). The mathematical formalization of these metrics is the following (both are accuracy ratios):

$$SA = \begin{cases} \frac{output_f}{output_g} & \text{if } output_f < output_g \\ \frac{output_g}{output_f} & \text{otherwise} \end{cases} \tag{3.3}$$

$$PC = \begin{cases} \frac{output_{orig}}{output_{exp}} & \text{if } output_{orig} < output_{exp} \\ \frac{output_{exp}}{output_{orig}} & \text{otherwise} \end{cases} \tag{3.4}$$

Both metrics can be applied to any method that uses a surrogate approach, for any type of data. The only requirement is that the surrogate model output is known for SA, and that it is possible to create the same type of input data from the explanations for PC. From the selected methods in Section 2.3.3, only LIME and Anchors respect these requirements. In this sense, the function *fidelity()* was implemented to evaluate their local fidelity, which computes the mean fidelity of the local explanations over n instances of the training data (default is 200, randomly selected), together with an histogram showing the distri-

[3] The distance between the instance of interest and the returned counterfactual (or other type of returned data point) could also be computed as a proximity metric.

bution over the data (*plot=TRUE* as default). This function is specific to these methods. Note that, for Anchors, PC score is known; it corresponds to the precision and coverage metrics outputted by the explanation itself, which result from applying the rule to the instance neighbors, getting their predictions using the underlying model and comparing it with the original predictions (ratio in equation 3.4 refers to the precision).

Faithfulness

The Incremental Deletion (ID) metric is implemented for tabular data in function $incrDel()$. Receiving the FS values of any local explanation (parameter *loc*), this function incrementally removes each of the input features by decreasing order of FS value (i.e., it removes first the most relevant features), and measures the respective change in the effect on the predictive model prediction. So that out-of-distribution samples are not created, the adopted solution is to replace "removed" features with values from the original data distribution. A *base* instance needs to be defined considering a predefined output (with optimal values that lead to the "opposite" prediction of the instance of interest), so that there is a feature value to be replaced with. *incrDel()* returns a data frame with the FS values (in percent) of the provided local explanation and correspondent model prediction after "removal" of each feature. The default *percentage* of input features to remove is 0.9, i.e., 90% of the features are removed by decreasing order of contribution.

The ROAR metric is implemented in a similar way using function *roar_accs*. Receiving the FS values of any global explanation (parameter *globExp*), this function incrementally removes each of the input features by decreasing order of FS value (i.e., it removes first the most important features), and measures the respective change in the original model test accuracy upon retraining, guaranteeing that the training and evaluating data come from the same distribution. Here, the removal of the features is done by shuffling the training data (which breaks the relationship between a feature and the true outcome) instead of replacing them with a constant mean value, as performed in the original paper [83] (for images; here extended for tabular domain). *roar_accs()* returns a data frame with the FS values (in percent) of the provided global explanation and correspondent model test accuracy upon retraining (predictive accuracy for classification tasks, and RMSE for regression tasks) after "removal" of each feature. The default *percentage* of input features to remove is again 0.9.

To access the change in the model output (for ID) or performance (for ROAR), the function *compare_auc* is implemented to assess how much each of the methods providing (local or global) FS explanations is better than a random explanation. If parameter *plot=TRUE*, it also plots the functions which show the decrease in model prediction/accuracy for all methods, including the random guess for comparison. This function uses the output of the previous functions as one of the inputs (parameter *accs*), and computes the AUC for all values, returning how much each explainer is better than the random explainer in percent.

The WBC metric can be quantitatively assessed when the WB is a simple linear regression model and the XAI methods are FS methods. Then, it is possible to compare each of the feature importance/influence values. No implementation function is needed for this.

Stability

For the similarity metric, the neighbourhood criterion was adapted for R from the shapash python library. Accordingly, function *find_neighbours* is defined considering 3 steps:

1. Pick top N (parameter *n.neighbors*; default is 10) closest neighbors (from training *data*), where the closeness is computed as the Gower distance (available in "gower" package from CRAN), which deals with numerical and categorical variables.

2. Filter neighbors whose *model* output is too different from instance of interest x:

   - For classification:

   $$|output_{neighbors} - output_x| < 0.1 \tag{3.5}$$

   - For regression:

   $$|output_{neighbors} - output_x| < |output_x| * 0.1 \tag{3.6}$$

3. Filter neighbors whose distance is too big compared to a certain threshold which is by default the 95th percentile, i.e., only neighbors that capture 95% of all distances are selected. Function $radius()$ returns this threshold which is the maximum allowed distance between points to be considered as neighbors.

Then, the feature variability calculated across the instances' neighborhood can be calculated and plotted (with function *sens_plot()*) and the average value is computed as the final result for the Similarity metric (with function *sens_result()*). The same is done to obtain the Identity metric but across different iterations for the same instance, i.e., the variance in feature values across n (parameter *n.sens*; default is 50) iterations of the XAI method for the same input sample is computed. The feature attribution values are calculated for similar instances using function *similarity()*. The feature attribution values are calculated for the same instance n times using function *identity()*. Both these functions are specific for (FS) selected methods in Section 2.3.3.

Chapter 4

XAI Benchmark Framework Application in the Medical Domain

This Chapter provides an example of how the developed benchmark framework can be used to compare XAI methods, showing the extent to which the selected properties and respective evaluation metrics can assist in a more comprehensive, inclusive, and consensual benchmark study. Firstly, in Section 4.1, general experimental results are provided, considering the used dataset, ML models, and XAI methods. Then, in Section 4.2., framework application results are provided and the comparison and discussion of the results obtained for each property across different XAI methods is performed. Finally, in Section 4.3, an enhanced solution for the framework and method selection is provided.

4.1 Experimental Results

All used software was written in R and is available as opensource on Github, including the source code for producing the results shown here. The experiments were run using R JupyterLab 5.0.11 (with R version 4.2.1), available for general use in JupyterHub for anyone at Aalto University.

4.1.1 Heart Failure Prediction Dataset

Various datasets have been widely used between scholars to compare ML models performances and to illustrate the application of different XAI methods. There are datasets for any type of data and different application domains. In this work, a tabular dataset from the medical domain is considered. The heart failure prediction dataset (from now on referred to as heart dataset) is publicly available in Kaggle, and will be described below. Other famous publicly available examples include: iris data set (tabular, flower recognition), titanic dataset (tabular, probability survival on the titanic disaster), boston housing dataset (tabular, housing price prediction), MNIST dataset (images, digit recognition), etc..

Cardiovascular diseases (CVDs) are the number 1 cause of death worldwide, taking around 17.9 million lives each year. Heart failure (or heart disease) is a common event caused by CVDs and the

heart dataset from Kaggle contains 11 features that can be used to predict a possible heart disease. The attribute information of this dataset consists of 11 (clinical) features and the output label (this is a classification problem) [85]:

1. **Age**: age of the patient [year]
2. **Sex**: sex of the patient [M: Male, F: Female]
3. **ChestPainType**[1]: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. **RestingBP**[2]: (systolic) resting blood pressure [mm Hg]
5. **Cholesterol**[3]: serum cholesterol [mg/dl]
6. **FastingBS**[4]: fasting blood sugar (1: if FastingBS $>$ 120 mg/dl, 0: otherwise)
7. **RestingECG**[5]: resting ECG results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. **MaxHR**[6]: maximum heart rate achieved [Numeric value between 60 and 202]
9. **ExerciseAngina**: exercise-induced angina [Y: Yes, N: No]
10. **Oldpeak**[7]: oldpeak = ST depression induced by exercise relative to rest [Numeric value measured in depression]
11. **ST_Slope**[8]: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. **HeartDisease**: output class [1: heart disease, 0: Normal]

[1] Angina is a type of chest pain caused by reduced blood flow to the heart. Typical Agina (TA) consists of (1) substernal chest pain or discomfort that is (2) provoked by exertion or emotional stress and (3) relieved by rest or nitroglycerine (or both). Atypical Angina (ATA) applies when 2 out of 3 criteria of TA are present. Non-Anginal Pain (NAP) applies when 1 or none of the criteria is present [86]. NAP excludes heart disease [87].

[2] Systolic pressure refers to the blood pressure (BP) in the arteries that occurs when the heart contracts or beats, pushing blood out. When the heart relaxes between beats , blood pressure in the arteries falls, as the heart is being filled with blood. This is the diastolic pressure. Normal blood pressure is less than 120 mmHg systolic and 80 mmHg diastolic [88].

[3] Serum cholesterol represents the amount of total cholesterol in the blood, comprising the amount of high-density lipoprotein (HDL), low-density lipoprotein (LDL), and (20% of) triglycerides in the blood. For people aged 20 years and older, the optimal serum cholesterol level is between 125 and 200 mg/dL [89].

[4] Fasting blood sugar (BS) is the blood sugar after an overnight fast (not eating). A fasting blood sugar level less than 100 is normal, 100 to 125 mg/dL indicates prediabetes, and higher indicates diabetes [90].

[5] The ST segment corresponds to the plateau phase of the action potential of an electrocardiogram (ECG). The ST segment and the T-wave are electrophysiologically related, so it is common to study changes that occur to both of them together - ST-T changes [91]. These changes lead to the so-called ST segment deviations (elevation or depression), which may occur in different conditions, including heart disease. They can also lead to Left ventricular hypertrophy (LVH), which occurs when the heart's left pumping chamber thickens and stops pumping efficiently [92].

[6] It is possible to calculate the (normal) maximum heart rate (HR) by subtracting the age from 220. For example, if a person is 45 years old, subtract 45 from 220 to get a maximum heart rate of 175 beats per minute (bpm) [93].

[7] Oldpeak value is equal to the ST depression induced by exercise relative to rest. ST depression is a trace in the ST segment of an ECG that is abnormally low below the baseline. ST depression equal or higher than 0.5mV is considered pathological [91].

[8] Physiological ST depressions are induced by physical exercise, normally displaying an upsloping segment. Heart failure has been associated with flat and downsloping depressions [91].

People with CVD or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension or diabetes) need early detection and management wherein an explainable ML model can be of great help. As the purpose of this chapter is to apply the developed benchmark framework for XAI methods, problem and dataset description will not be further discussed here. The variable of interest is **HeartDisease**, which is a factor; there are 5 numerical features (Age, RestingBP, Cholesterol, MaxHR, and Oldpeak); and 6 categorical features (Sex, ChestPainType, FastingBS, RestingECG, Exercise Angina, and ST_Slope) which were converted to factors (with respective levels mentioned above). After cleaning the data, i.e., removing outliers and null values, the heart dataset, whose first six rows of the data frame are presented in Figure 4.1, was divided into training and testing datasets, for further machine learning modelling. The final size of the training and testing datasets, is 527 and 175 observations, respectively, and in both of them the binary target attribute (HeartDisease) is balanced. Data preprocessing results and the main conclusions drawn after performing an exploratory data analysis (EDA) on the training data can be assessed (and visualized) in the R notebook "01_Data.ipynb".

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
1	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
2	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
3	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
4	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
5	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0
6	39	M	NAP	120	339	0	Normal	170	N	0.0	Up	0

Figure 4.1: First 6 rows of the heart dataset.

4.1.2 Machine Learning Models

The following ML models were trained for the binary classification problem of predicting heart disease:

- **Logistic Regression (LR)**

  A generalized linear model (GLM) is a generalization of the ordinary linear model 2.1, where the linear model is related to the dependent variable, via a link function [94]. The GLMs cover widely used statistical models, such as logistic regression (LR), where the link function is the logistic function (logit), forcing the output of a linear equation to be between 0 and 1:

  $$logit(P) = \frac{1}{1 + exp(-P)} \tag{4.1}$$

  Equation 2.1 works for regression problems; for classification problems, the output probabilities are between 0 and 1, so wrapping the right side of the equation into the logit function the output is forced to be in that interval [6]:

  $$P(y = 1) = \frac{1}{1 + exp(-(w_0 + w_1 x_1 + ... + w_p x_p))} \tag{4.2}$$

Reformulating this equation, it is shown in equation 4.3 that LR is a linear model for the log-odds, meaning that a change in feature i by one unit changes the odds ratio (multiplicative) by a factor of $exp(w_i)$ [6]. So, the interpretation of the weights in LR differs from the interpretation of the weights in linear regression 2.1, but both models are seen as white-box models, meaning they are considered transparent and understandable by themselves.

$$log\frac{P(y=1)}{1-P(y=1)} = log(odds) = w_0 + w_1x_1 + ... + w_px_p \tag{4.3}$$

- **Random Forest (RF)**

  Linear regression and logistic regression models fail in cases when the dependent variable has a non-normal distribution or when variables interact with each other. Decision trees, also seen as white-box models, appear in such cases, and can be used for classification and regression tasks. Tree based models split the input data several times according to certain cutoff values in the features. During splitting, different subsets of the dataset are created, with each input instance belonging to one subset. The final subsets are called leaf (or terminal) nodes and the intermediate subsets are called split (or internal) nodes. To predict the output in each leaf node, the average output of the training data in this node is used. There are different ML algorithms based on this approach, that may differ in the structure of the decision tree (e.g., number of splits per node), the criteria on how to find the splits, or when to stop splitting [6].

  Random forests (RFs) [51] are a combination of predictive decision trees such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. A RF consists of hundreds of decision trees that "vote" for predictions, according to this random vector. To understand how the decision was made, it would be necessary to look into the votes and structures of each of the hundreds of trees individually [6]. This is impossible, so, RF is considered a black-box model, meaning it lacks interpretable/explainable tools for humans to understand the model working logic and outputs. Nevertheless, RF models prove to yield good predictive performance, ability to grasp low-order feature interactions, and robustness [44, 51].

- **Support Vector Machine (SVM)**

  A support vector machine (SVM, or support vector network) [95] is another learning machine approach for classification and regression tasks, being typically used for binary classification. "The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear decision surface is constructed [95]." In other words, the objective is, from the input data, to find a hyperplane in p-dimension feature space (where p is the number of features) that distinctly classifies the input instances through the created decision boundary. For the binary classification problem, it is necessary to separate the two classes of data instances. For that, SVM approach is to find a hyperplane that has the maximum margin, i.e., the maximum distance between instances of both classes. Support vectors

link data points in the hyperplane to those that are the closest to the separating hyperplane. They act as a "machine", moving the position and orientation of the hyperplane in order to maximize the margin. The loss function that helps maximize the margin is the hinge loss [96].

Special properties of the decision surface, that depend on different parameters such as the used kernel function, ensures high generalization ability of the learning machine, producing great performance results [95]. However, SVM is a powerful ML model that has complex mathematical formulations, being particularly difficult to understand when the number of features exceeds 3 (a high-dimension feature space is created). Therefore, it is considered a black-box, just like the RF model.

The first is a simple (white-box) model, the second and the last are more complex (black-box) models and were chosen because they use different approaches (tree and statistical-based respectively). The "stats" (*glm()* function with parameter family = binomial(link = logit)), "randomForest" (*randomForest()* function with default parameters), and "e1071" (*svm()* function with parameter type = C-classification and probability=TRUE) packages were used for the LR, RF, and SVM models, respectively.

For a classification model, the predictive performance is usually computed by calculating the accuracy, i.e., out of all the predictions, what percentage is correctly made. For the LR, RF, and models, the accuracy is equal to 82%, 83%, and 85%, respectively, computed using the testing data (175 observations). White-box model LR and black-box model RF both have similar performances, being the black-box SVM model the one who performs better. The implementation of these 3 models, together with other evaluation metrics for both training and testing datasets can be assessed in the "02_Models.ipynb" R notebook. Other evaluation metrics include precision, recall, and F1-measure, which are also important to consider, together with the predictive accuracy. In the medical domain it is especially important to have a high recall, as it is crucial to develop a ML model that has the minimum number of false negatives. In the considered problem, a false negative is when a patient is misclassified as not having heart disease. All models report similar recall scores (around 80%).

4.1.3 Explanations

The experiments obtained from the application of CIU and other 8 well-known XAI methods on the heart dataset is reported next. The IML package is used for PDP, ICE, and Shapley values; the 'lime' package for LIME; the 'anchorsOnR' package for Anchors; the 'shapper' package for kernelSHAP[9]; the 'counterfactuals' package for CFEs; and the 'ciu' package for CIU. The default parameters are used for all packages unless stated otherwise. The "randomForest" package computes permutation feature importance (PFI). This is a small selection of the 131 XAI methods displayed in Table A.3, which were introduced in Section 2.3. The "03_Explanations.ipynb" notebook provides all the code to produce the explanations that are going to be displayed next (and others) - for which the reader is referred for more

[9]Currently, this package only works with a lower version of R than the one provided by JupyerLab from Aalto University. Therefore, all the source code for producing the results for kernelSHAP was written using Anaconda 2.3.2 (with R version 3.6.1 and python version 3.7.13, which is also needed because "shapper" is an R wrapper of SHAP python library). Because of this, the running time is much slower than when using JupyterLab.

accurate visualizations. The installation (elapsed) time of all packages is less or around 1/2 minutes, expect for the CFEs package, that takes 6/7 minutes. The time required to output each explanation is given by $Elapsed$ time, placed in the figure captions. "Global.R" script contains the code to compute the global feature importance values for Shapley values, kernelSHAP, and CIU. All the global feature importance values (also PFI for RF) were converted into percent values (between 0 and 1) for consistency (and comparison) and are presented in Table B.2.

To show the results of (local) explanations, from now on the instance of interest x is patient A (observation number 16 of the testing data), with the following clinical feature values:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
16	57	M	ATA	140	265	0	ST	145	Y	1	Flat

'true label: 1 - with heart disease'

Figure 4.2: Instance of interest is patient A (testing instance number 16), with the clinical values presented here. The true label is 1, i.e., class 1, meaning the patient has heart disease.

The predictions obtained were correctly made (output = class 1) by all implemented ML models:

Table 4.1: Obtained predictions for patient A. All models correctly classified patient A as having heart disease (class 1).

	LR	RF	SVM
Class 0	0.10	0.15	0.12
Class 1	0.90	0.85	0.88

The clinical feature values for a healthy patient (patient B - prediction is class 0 with high probability) and for a patient wrongly diagnosed with no heart disease (patient C - prediction is class 0, but with probability close to 0.5, which shows uncertainty), together with respective model predictions and explanations, can be further assessed in the R notebook "03_Explanations.ipynb". This is relevant for comparison, and to check if the explanations give insight about why the model is making an uncertain prediction for patient C.

PDP

There are short lines on the x-axis of both PDP and ICE plots, which are the quartiles of the specific feature being studied, meaning 10% of that feature values are less than the first line and 90% are less than the last line. This is known as rug plot, which shows the distribution of the feature. This makes it possible to check if that feature values are fairly evenly distributed across its range [6]. That being said, PDP visualizes the relationship a model has learned, which can be called model inspection [9]. The influence of the age feature on the predicted probabilities for class 1 is visualized in Figure 4.3 for all models. For the LR model, PDP shows a linear relationship between the target and feature age, which in fact happens for all features as, as seen above, LR is a linear model for the log-odds. For the SVM model, PDP seems to show an exponential relationship between the target and feature age. For all models, the PDP shows age has a positive influence on the target output. However, interestingly, for the RF model, the predicted probability falls when age increases from 65 to 77, but there is not much training data, so the ML model may not have learnt a meaningful prediction for this range.

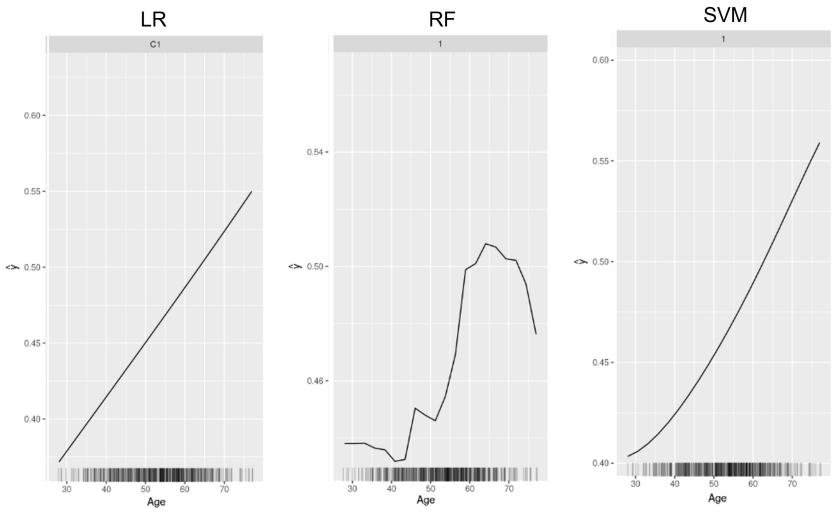


Figure 4.3: PDPs for the used heart disease prediction models (class=1, i.e., with heart disease) and Age. Marks on the x-axis indicate the data distribution. $Elapsed :< 1s$.

To illustrate a PDP with a categorical feature, the effect of the sex feature on the predicted probabilities is shown in Figure 4.4. Feature Sex seems to have a similar effect on the predictions of all models, being clear that males have a higher probability of developing heart disease. It is also possible to visualize the PDP of two features (maximum) at once. PDPs in Figure 4.5 show an interaction between Age and MaxHR features for RF and SVM, which is not detectable using the LR model. For ages below 60, patients who have maximum heart rate above 150 bpm have a lower predicted heart disease risk. This makes sense, as for ages below 60, healthy patients should have a MaxHR above 160 bpm (220-60=160).

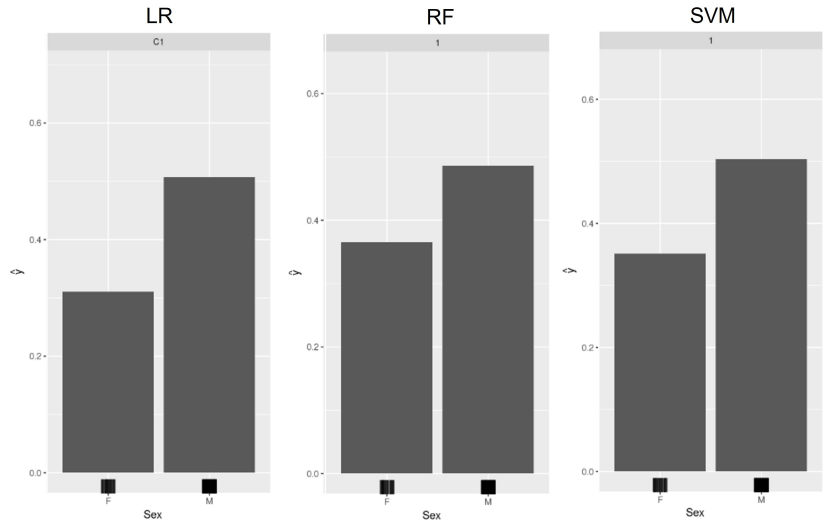


Figure 4.4: PDPs for the used heart disease prediction models (class=1, i.e., with heart disease) and Sex. Marks on the x-axis indicate the data distribution. $Elapsed :< 1s$.

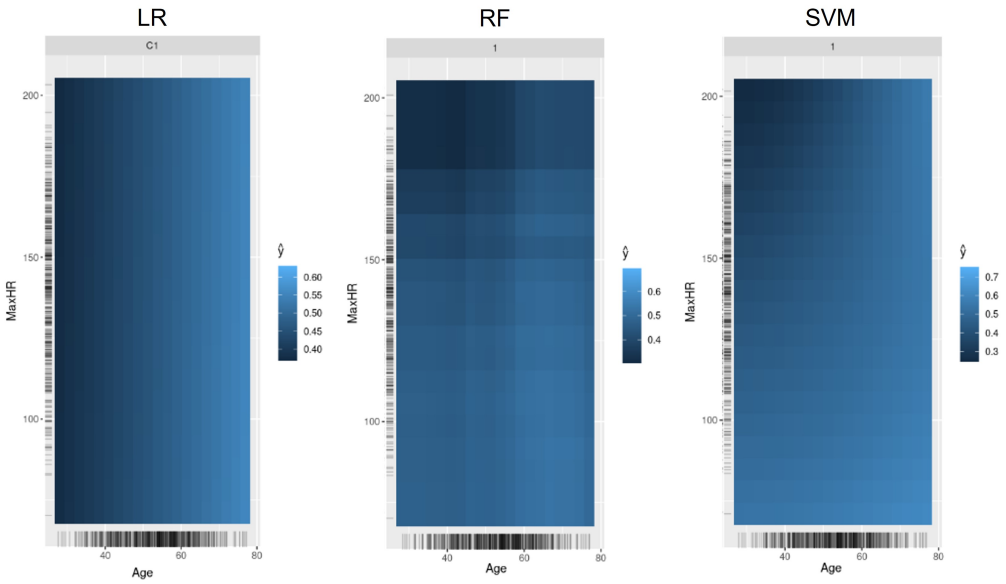


Figure 4.5: PDPs for the used heart disease prediction models (class=1) and the interaction of Age and MaxHR. Marks on the x-axis indicate the data distribution. $Elapsed :< 1s$.

ICE

The ICE plots reveal that for most patients the age effect follows the average pattern revealed in PDPs (the yellow curve corresponds to the PDP in Figure 4.3, but there are some exceptions, for example: For the patients that have a high predicted probability at the age of 50, the predicted heart disease probability (for LR and SVM models) does not change much if increasing the age. This also happens for the RF model, but after 70 years old, the predicted probability decreases a lot, which is unexpected. Nonetheless, in general, all curves seem to follow the same course as the yellow one, so there are no obvious interactions. That means that the PDP is already a good summary of the relationships between the age feature and the target prediction. The same does not happen for the RestingBP feature, whose ICE (+PDP) plots are depicted in Figure 4.7. If only considering PDP explanations, it would be concluded that feature RestingBP is not important to the output the SVM model, as the yellow curve is almost flat (no variation). But, when considering ICE explanation curves, it is possible to conclude that the PDP is not a good summary of the relationship between this feature and the model predictions, as an heterogeneous relationship is uncovered. For example, for the patients that have a high predicted probability at low RestingBP value, the SVM predicted heart disease probability tends to increase if this feature value is increased until 150 mm Hg.

To illustrate an ICE plot for a categorical feature, the effect of the ST_Slope feature on the predicted probabilities is shown in Figure 4.8, where the bold lines in the middle of the boxes give the average prediction, i.e., the bold line is located at the height of the bar plot outputted by PDPs. Feature ST_Slope seems to have a similar effect on the predictions of all models, with ST_Slope = Flat being the category that contributes to the highest extent to high output probabilities, i.e., patients with ST_Slope = Flat have a high probability of having heart disease. If ST_Slope = Up (healthy situation), the probability is low, although the RF model seems to (abnormally) show also high probabilities for this feature category.

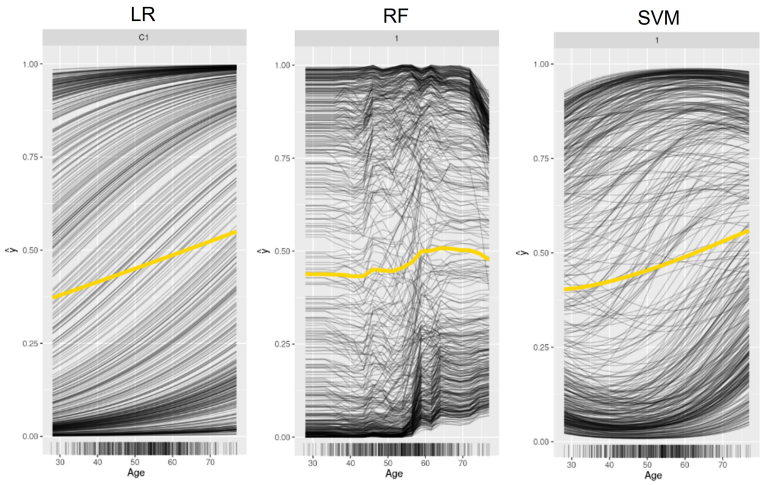


Figure 4.6: ICE plots for the used heart disease prediction models (class=1) and Age. Yellow curve is the PDP. Marks on the x-axis indicate the data distribution. $Elapsed :< 1s$.

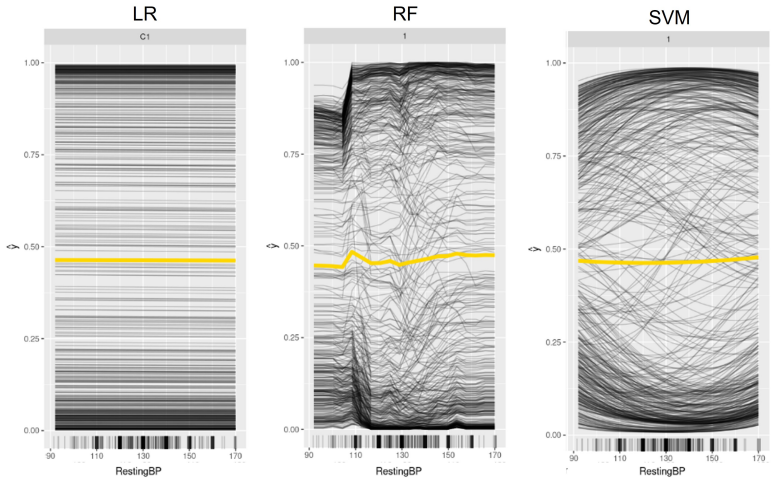


Figure 4.7: ICE plots for the used heart disease prediction models (class=1) and RestingBP. Yellow curve is the PDP. Marks on the x-axis indicate the data distribution. $Elapsed :< 1s$.

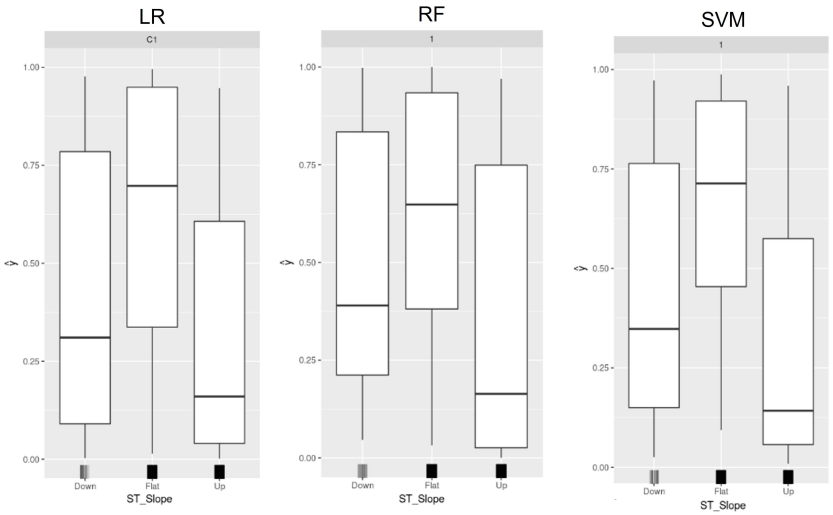


Figure 4.8: ICE plots for the heart disease prediction models (class=1) and ST_Slope. The bold lines in the middle of the boxes give the average prediction. Marks on the x-axis indicate the data distribution. $Elapsed :< 1s$.

PFI

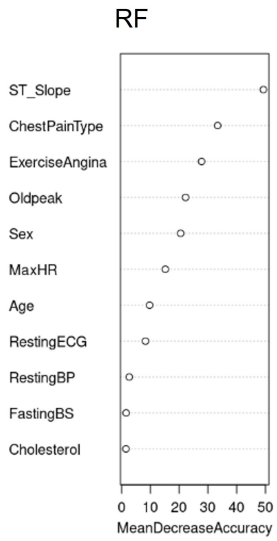


Figure 4.9: PFI plot: The importance of each of the features for predicting heart disease with a random forest. $Elapsed :< 1s$.

The *randomForest()* function computes a global feature importance method as originally described by Breiman [51] and introduced in Section 2.3.3. Figure 4.9 shows the mean decrease accuracy between the 2 classes after permuting each feature. Features associated with an error rate decrease by a factor of 0 (= no change) are not considered important by the PFI method for the prediction. Note that, when looking for the class specific measures, the error rate is higher for the (mis)classification of class 0 (normal condition). Neverthless, the feature with the highest importance by the PFI method is ST_Slope. associated with a mean decrease accuracy of 49.3% after permutation. Next is ChestPainType feature, with a mean decrease accuracy of 33.4%. RestingBP, FastingBS, and Cholesterol are not considered important for heart disease prediction by this XAI method. PFI values, specific for the RF model, are displayed in table B.2 (in percent).

LIME

The resulting explanation plot for the predictions given by each model for patient A (see Table 4.1) is shown in Figure 4.10. The length of the bar indicates the magnitude (absolute value), while the color indicates the sign (red for negative, blue for positive) of the estimated coefficient. The output of the explanation (data frame) includes a column *feature_weight*, which provides the estimated coefficients for the features by the k-LASSO method for the explanation. The *model_intercept* column provides the value of the intercept. Thus, the linear combination of the transformed explanatory variables used in the surrogate model approximating the underlying model around the instance of interest, patient A, corresponds to the number given in $model\_prediction$ (it is the prediction given by the linear surrogate model). This is exemplified in Figure 4.11 for the RF model. The feature weights calculated by LIME are provided in detail in Table B.1.

The used LIME R package also allows for the creation of a heatmap showing how the different features selected across different instances influence each case. This plot is useful to find common features that influence all observations or if a big number of instances is analyzed at the same time (it makes the previous bar plots difficult to discern). An example is given in Figure B.1 for the LR model.

Anchors

To create the explainer (function) for Anchors, the parameters *bins* and *coverage_perturbation_count* were modified. The first was used to create discretization values for the numerical features (so that the generalization to other instances is bigger). The second was used to change the number of perturbations

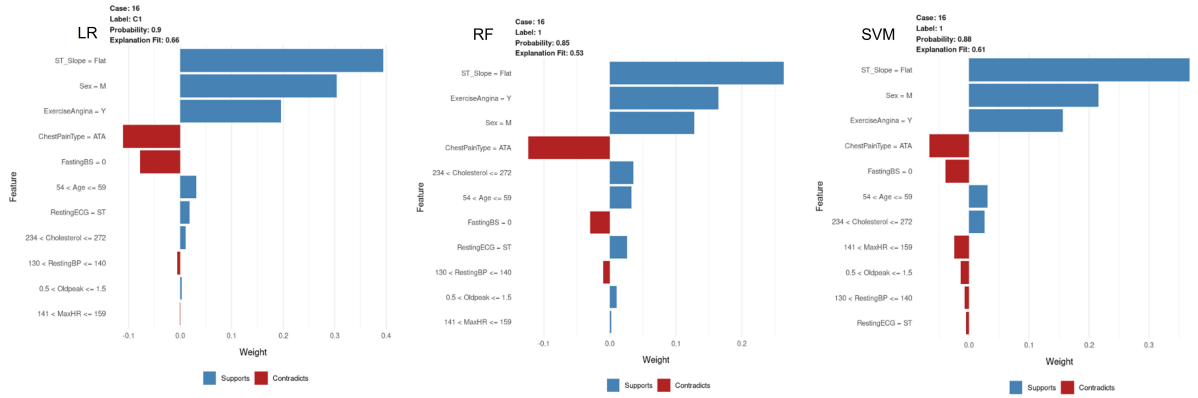


Figure 4.10: Illustration of the LIME method results for heart disease prediction (class label=C1/1) for patient A. The probability value corresponds to the class 1 probabilities given by each model. The Explanation Fit value corresponds to the r-squared measure of the fitted surrogate model. The numerical features are discretized into quartiles. $Elapsed :< 1s$.

```
intercept <- explanation_rf$model_intercept[1]
weights <- explanation_rf$feature_weight
sum <- sum(c(intercept,weights))
paste0("Linear model equation: ", sum, ", confirmation: ", explanation_rf$model_prediction[1])
```

```
'Linear model equation: 0.704390971259514, confirmation: 0.704390971259514'
```

Figure 4.11: LIME prediction: the linear combination of the transformed explanatory variables used in the surrogate model g approximating the RF model around the instance of interest x (patient A) corresponds to the $g(x)$.

(samples) to be taken for coverage from default 1000 to 500, which reduces the computational time considerably (it is still slow, but not as much; a trade-off between accuracy and running time needs to be made). That being said, the textual explanations given by Anchors for patient A are depicted in Figure 4.12 (left) for all models. The results provide a logic structure and are very easy to interpret. It is possible to understand which features are most important for the model's prediction (class 1). A visualization tool is also provided; an example is shown in Figure 4.12 on the right for the rule given for the LR model. Taking a bigger insight at the explanations, the rules show that the most important feature is clearly ST_Slope. When anchors are based on a few features, they additionally have high coverage and hence apply to other instances in the (neighboring) perturbation space (this is the case for patients A and B). However, other observations may not be as distinctly classified by the model as more features are of importance. An example is the case of patient C, where is notable that anchors get more specific, comprise more features, and apply to only a few instances from the perturbation space (low coverage) - see Figure Notebook "03_Explanations.ipynb" Section Anchors for patient C. This can be a sign that the instance is near the decision boundary, as the instance is located in a volatile neighborhood [6], which is in fact what it is happening for patient C.

Shapley values

Bar plots in Figure 4.13 present the distribution of the contributions to the prediction of patient A compared to the average prediction for the dataset, as described in Section 2.3.3. Bars to the right and to the left represent, respectively, the positive and negative shapley values across the coalitions. It is clear that the flat ST depression of patient A results in a positive influence for the output, when

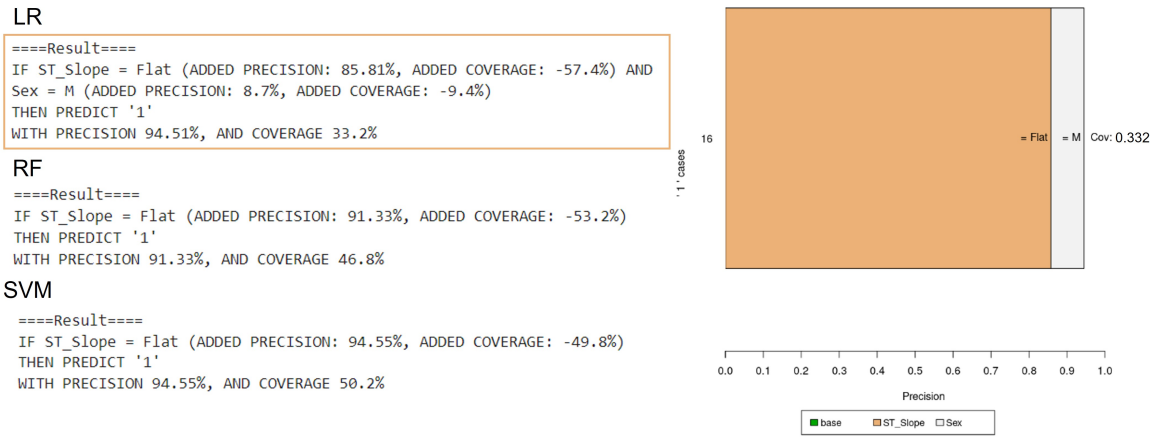


Figure 4.12: Anchors explaining patient A prediction with rules. Left: Textual explanations for all models. Right: Visualization for LR. Each bar depicts the feature predicates contained by the anchor. The x-axis displays the rule's precision, and the bar's thickness corresponds to its coverage. The "base" rule contains no predicates. $Elapsed : 19s, 12s$, and $14s$, respectively for LR, RF, and SVM models.

compared to the average prediction (baseline). On the other hand, the effect of ChestPainType=ATA is in all cases negative, with the magnitude of Shapley values varying across models. The shapley values for the prediction for Patient A for the different models are depicted in Table B.1. The shapley values are provided by this method in its output, together with the associated variance across all coalitions.

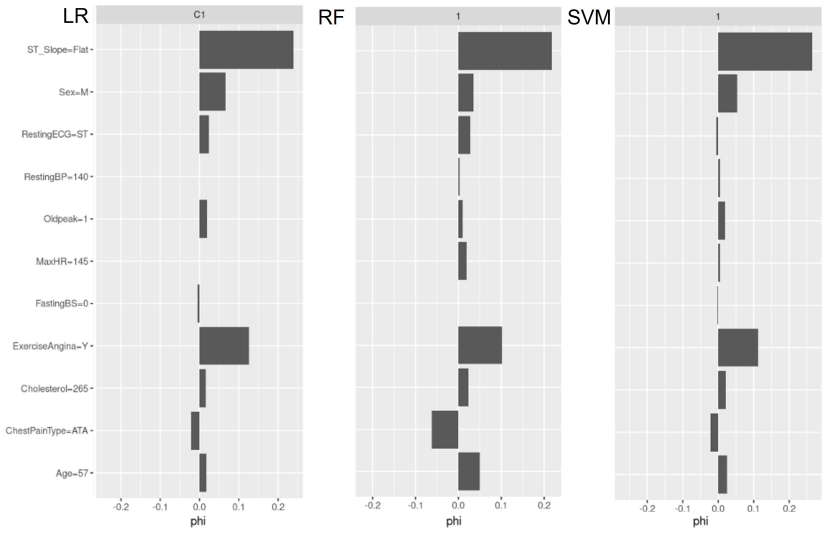


Figure 4.13: Shapley values for patient A in the heart dataset. $Elapsed :< 1s$.

kernelSHAP

As stated in Section 2.3.3, kernelSHAP is a different approach to compute shapley values. When using the function to create the explanations, the parameter *nsamples* was changed to 100 (meaning instances are sampled to generate the explanation values), to be consistent with Shapley and CIU (it also decreased the computational time considerably). Bar plots in Figure 4.14 present again the distribution of the contributions to the prediction of patient A compared to the average prediction for the dataset, as described in Section. Green and red bars represent, respectively, the positive and negative shapley

values across the coalitions. The results are very similar to the previous method, but this time is possible to verify the efficiency (or local accuracy) property, meaning $f(x) = g(x')$ and $g(x')$ is given by equation 2.12. So, model linearity is assumed, which results in the sum of shapley values yielding the difference of actual and average prediction (the linear model is "forced" to pass through the baseline which is the average of all predictions and the actual prediction). This can be easily visualized in the plots, where each shapley value is a bar that "pushes" to increase (positive sign) or decrease (negative sign) the average prediction (baseline), balancing each other out at the actual prediction of the data instance. The kernelSHAP values for the prediction for Patient A for the different models are depicted in Table B.1. Moreover, both shapley and SHAP global feature importance values were computed, using Lundberg et al. [59] approach, and are displayed in table B.2.

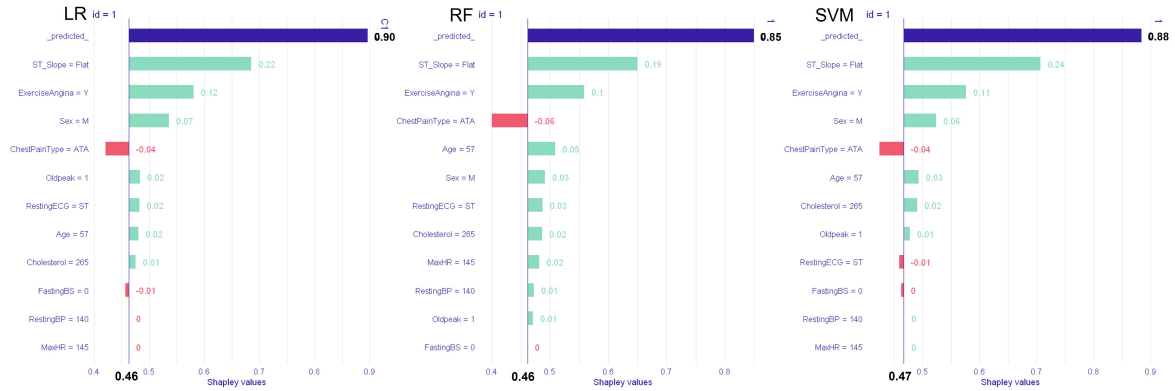


Figure 4.14: kernelSHAP values to explain the predicted heart disease probabilities of patient A. $Elapsed$: $1s, 2s$, and $2s$, respectively for LR, RF, and SVM models.

CFEs

The MOC method proposed by Dandl et al. [65] was used to obtain CFEs as explanations in R. The counterfactuals should answer, in this case, the following question: "how the input features need to be changed to get a predicted probability lower than 50%"? In other words, what is the smallest change that should be done to some feature(s) so that the model(s) predicts class 0, instead of class 1. Figure 4.15 shows the five best counterfactuals (parameter *n_counterfactuals* was changed when implementing the MOC explainers). The results are given also as a data frame (similarly to 4.2), but where each column contains the respective feature change (when it happens). It is also possible to access the number of changed features and the distance to the instance of interest, which is not being shown here. From the resulting data frames, it is clear that the models have different behaviours, giving different prediction probabilities, as the way features change their input values is different across the models. Some of the features are not susceptible to change in the real world, such as the patient sex, or (lower) age. However, the relative frequency of changed features (that can also be visualized as a plot) gives an idea of which factors are the most important to the obtained prediction. FastingBS seems to be the least important feature, as its value never changes for any of the generated counterfactuals.

LR

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>
59	M	ATA	140	221	0	Normal	164	Y	0.0	Up
58	M	NAP	130	213	0	ST	140	N	0.0	Flat
46	M	ATA	140	275	0	Normal	165	Y	0.0	Up
54	M	ATA	132	182	0	ST	141	N	0.1	Up
44	M	ATA	120	184	0	Normal	142	N	1.0	Flat

RF

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>
57	M	ASY	130	207	0	ST	96	Y	1	Flat
59	M	ASY	140	274	0	Normal	154	Y	2	Flat
61	F	ASY	130	294	0	ST	120	Y	1	Flat
59	M	NAP	130	318	0	Normal	120	Y	1	Flat
53	F	ATA	140	216	0	Normal	142	Y	2	Flat

SVM

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>
59	M	ATA	140	221	0	Normal	164	Y	0.0	Up
46	M	ATA	140	275	0	Normal	165	Y	0.0	Up
54	M	ATA	132	182	0	ST	141	N	0.1	Up
56	M	ATA	126	166	0	ST	140	N	0.0	Up
52	M	ATA	140	100	0	Normal	138	Y	0.0	Up

Figure 4.15: Counterfactual explanations: the 5 top generated counterfactuals that are "close" to the instance of interest (patient A) and that change the output to class 0. $Elapsed :< 1s$.

CIU

Figure 4.16 shows the output value as a function of one feature (Age) value for the three models. These can be thought as a line from the ICE plots in Figure 4.6[10]. PDP and ICE simply plot these "ceteris paribus" (CP) functions, however CIU transparently uses them for the calculation of CI, CU, and CInfl values like illustrated in Figure 4.16.

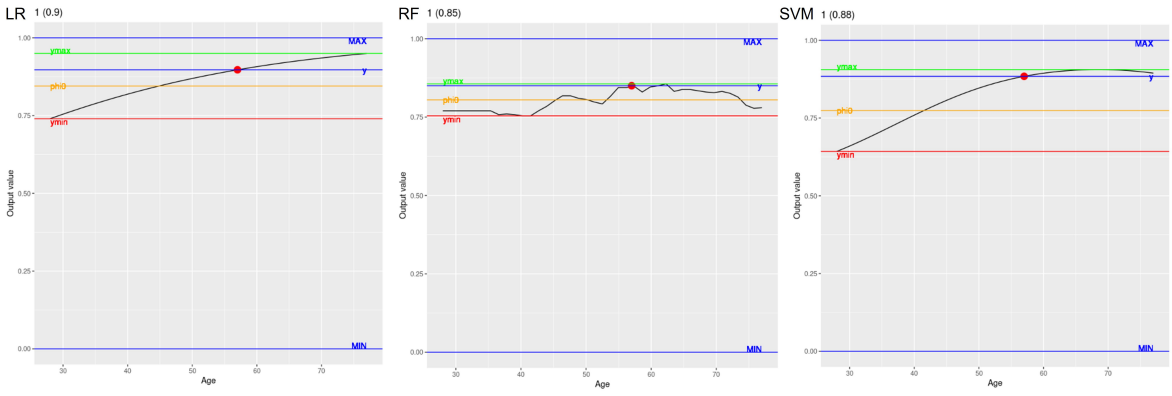


Figure 4.16: Output value (class=1) as a function of one feature value for the three models, with illustration of CIU calculations. The red dot shows the x_i value of the instance to be explained for the feature i (Age=57). The output range is [0, 1] as it is a classification problem. The used labels in the Figure are $MIN = umin$, $MAX = umax$, $ymin = umin_i(x)$, $ymax = umax_i(x)$, $y = u_i(x)$ and $phi0 = ymin + \phi_0(ymax - ymin)$, the same used to describe the CIU method in Section 2.3.3.

The results of CI and CU calculations for each feature can be visualized in each bar of Figure 4.17. The length of the bar indicates the importance of each feature (CI value), while the color filling indicates

[10]The difference is in the approach used to derive the perturbation space. ICE/PDP uses a (fixed) grid and CIU uses the sampling approach suggested in [67] and described in Section 2.3.3.

how favorable is the current feature value for the obtained prediction (relative CU value). The CI together with relative CU values are also plotted for class 0 with RF model, due to the fact that patients are probably more interested in knowing why aren't they healthy. It is clear that, contrary to class 1, in class 0 bars, almost none of them is filled. This shows that the current clinical results of patient A are very favorable to the diagnosis of heart disease. CI and CU values, due to the fact that are computed using utility functions, can also be given in the form of textual explanations. This is illustrated in Figure 4.18 for the RF model and considering that class 0 is the desired output. This clearly demonstrates that CIU is also counterfactual, meaning it is clear which features should be changed (and to what extent) to achieve a desirable output.

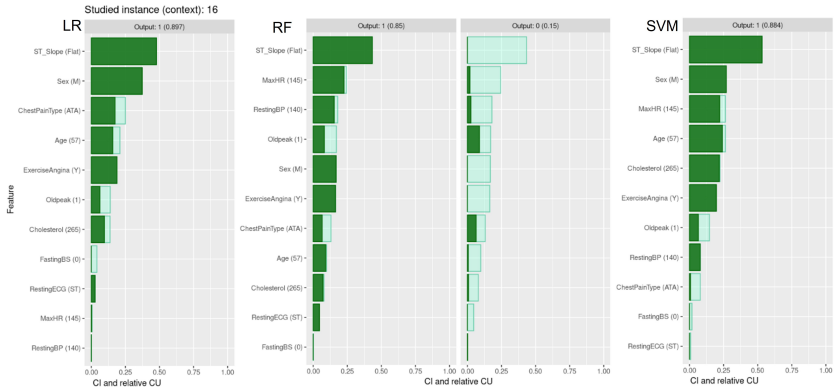


Figure 4.17: CIU bar plot explanations of heart disease prediction (output:1) of patient A for the three models. For the RF model, the bar plot explanation considering healthy prediction (output:0) is also illustrated. $Elapsed :< 1s$.

RF

```
The value of output '0' for instance '16' is 0.15, which is very bad (CU=0.15).
Feature 'ST_Slope' is important (CI=0.434) and value 'Flat' is very bad (CU=0).
Feature 'MaxHR' is slightly important (CI=0.244) and value '145' is very bad (CU=0.074).
Feature 'RestingBP' is not important (CI=0.178) and value '140' is very bad (CU=0.135).
Feature 'Sex' is not important (CI=0.168) and value 'M' is very bad (CU=0).
Feature 'ExerciseAngina' is not important (CI=0.164) and value 'Y' is very bad (CU=0).
Feature 'Oldpeak' is not important (CI=0.164) and value '1' is average (CU=0.537).
Feature 'ChestPainType' is not important (CI=0.13) and value 'ATA' is average (CU=0.492).
Feature 'Age' is not important (CI=0.102) and value '57' is very bad (CU=0.059).
Feature 'Cholesterol' is not important (CI=0.082) and value '265' is very bad (CU=0.098).
Feature 'RestingECG' is not important (CI=0.046) and value 'ST' is very bad (CU=0).
Feature 'FastingBS' is not important (CI=0.002) and value '0' is very good (CU=1).
```

Figure 4.18: CI and CU values can be translated into textual explanations, seen as more easily understandable for lay humans. $Elapsed :< 1s$.

Finally, similarly to LIME and SHAP(ley), feature influence (CInfl) values computed by CIU are plotted in Figure 4.19, having as baseline ϕ_0, i.e., the average utility value of 0.5 (orange line in Figure 4.16). An influence-based explanation might not give any indication of how to favorably change a certain feature if the current value is average, which might result in a close-to-zero influence value that would give an impression that that feature does not have any importance. The CI, CU, and CInfl values for patient A are depicted in Table B.1. Moreover, CIU global feature importance values were computed, under the idea that contextual importance is mathematically similar to global feature importance, and are displayed in table B.2.

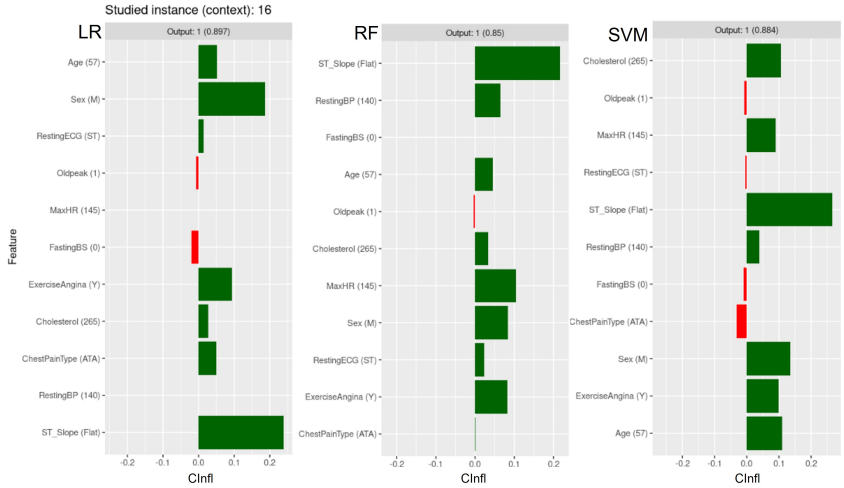


Figure 4.19: Contextual influence bar plot explanations of heart disease prediction (output:1) of patient A for the three models. $Elapsed :< 1s$.

4.2 XAI Benchmark Framework Results

4.2.1 Baseline Comparison

Representativeness

Table 4.2 contains the benchmark results for this property, where three levels of representativeness are compared: scope, portability, and applicability of the explainability methods. As mentioned above, this property is only evaluated qualitatively, being useful for developers to access in a quick way the usefulness of the method for their specific AI system.

Table 4.2: Results of representativeness property. These metrics are not specific for tabular data, but rather a general overview of each XAI method. Please check Table 3.2 for labels clarification. In bold are the best results.

Metric	PDP	PFI	ICE	LIME	Anchors	Shapley	SHAP	CFEs	CIU
Scope	G	G	L/G	L	L	L/G	L/G	L	**L/G**
Portability (1)	A	S	A	A	A	A	A	**A**	**A**
Portability (2)	T	T	T	T	T	T	T	**F**	**F**
Applicability	S	S	S	A	A	S	A	S	A

The results of the scope metric are depicted in the second row of Table 4.2. ICE has L/G, as, although being considered a local method, it provides a plot where the PD function for all instances is present. Shapley, SHAP, and CIU methods have L/G, as it is possible to compute global importance measures as described in Section 2.3.3. Table B.2 shows results for the global PFI method (only for RF), and using $mean(CI)$ and $mean(|\phi_i|)$ for each feature. All methods agree to some extent, even though CIU seems to better distribute the importance between features. Shapley and kernelSHAP give identical results, and are also closer to the values given by PFI for the RF model. It is not possible to conclude which one is the most "correct" one. Lundberg et al. [59] claim that Shapley values give the best indication of the global importance. However, as shown by the theory and results (see section 4.2.1), CI is a "true" importance measure, rather than the influence values given by Shapley values. This is why L/G is in bold for CIU method in Table 4.2.

Regarding the portability metric, the first and second row illustrates whether a method is model-agnostic (A) or model-specific (S), and if the method needs access to generate a (new) explanation, respectively. The method may need access to the data to create the respective explainer, but also when computing a new explanation. CFEs and CIU methods are in bold, in both rows, due to the fact that they do not require access to the data or the model itself. Both methods only require access to the model's prediction function, which is possible to provide via a web API, for example. This is attractive for companies that are interested in protecting model and data, due to data protection reasons or interests of the model owner, for example [6]. The opposite happens for PDP, ICE, Shapley, and kernelSHAP methods, which is a disadvantage, as they all need access to the data when calculating new values for a new data instance/feature. kernelSHAP also needs access to the model when computing a new explanation, making it the worst method at this level. Moreover, while almost all the SoTa XAI methods are limited to machine learning based models, CIU explanations and CFEs are applicable to any system f, making them truly model-agnostic. Finally, regarding the applicability metric, agnostic-data (A) methods are usually preferred. Although it is not being studied here, other types of data and ML models, including DL models, that do not learn features directly from the data and do not need manual feature extraction, are also possible to be explained by XAI methods. Specifically, the ones that are data-agnostic (A) (LIME, SHAP, Anchors and CIU) also focus on this type of models and (raw) data.

Structure & Speed

Table 4.3 contains the evaluation results for this property, where both the structure and the speed of an explanation are analyzed. Four levels of structure are compared: expressive power, graphical integrity, morphological clarity, and layer separation. As mentioned above, this property is only evaluated qualitatively, being useful for developers to access in a quick way and conclude about the XAI method that is more suitable for a specific end-user. The speed is evaluated quantitatively; the table makes a conclusion whether each method is slow or fast based on the runtime analysis that was made during the computation of the explanations (see Figures captions in Section 4.1.3 and Appendix B).

Table 4.3: Results of structure & speed property. These metrics provide a general overview of the type of output provided by each method. Please check Table 3.2 for labels clarification. In bold are the best results.

Metric	PDP	PFI	ICE	LIME	Anchors	Shapley	kernelSHAP	CFEs	CIU
Expressive Power	FS	FS	FS	FS	Rules	FS	FS	Data points	**FS**
Graphical integrity	F	F	F	**T**	F	T	**T**	F	**T**
Morphological clarity	T	T	T	T	T	T	T	T	T
Layer separation	N/A	N/A	F	T	F	**T**	**T**	T	**T**
Runtime Analysys	Fast	Fast*	Fast	Fast	Slow	Fast	Fast	Fast	Fast

Regarding the expressive power, anchors uses if-then rules (textual explanations, that can also be plotted as a bar chart) and CFEs provides data points (see Figures 4.12 and 4.15, respectively), which are both seen as more suitable for lay-users, by providing a logic structure. All the other methods give feature summary (FS) results as explanations, although providing distinct summary values, as well as different types of visualizations. PDP and ICE provide PD functions, which values can be assessed in a data frame or visualized as PD (or CP) profiles, as shown in Figures 4.3 and 4.6. PFI outputs a plot that

shows the decrease in model accuracy when removing each feature (see Figure 4.9). LIME, Shapley, and Kernel SHAP provide feature influence values as explanations that can be visualized as bar plots, as depicted in Figures 4.10, 4.13, and 4.14. Finally, CIU can provide contextual influence bar plot, which are comparable with those provided by LIME and SHAP(ley). Furthermore, it provides explanations using CI and CU in different ways (bar plots and pie charts) and prioritizing CI or CU depending on the purpose of the explanation (see Figures 4.17 and 4.19), which is not possible for the other methods. As it can be seen, CIU can also plot PD/CP profiles, which consist in input-output values from where CIU values can be "read" and validated directly (see Figure 4.16). Moreover, CI and CU values can be translated into textual explanations, as shown in Figure 4.18, which are usually seen as more easily understandable by lay-users. In the case of heart disease prediction, it is clear for patients, through these textual explanations, to understand how each clinical feature value contributed to the prediction and how changing that value would change the prediction to a better one (counterfactual approach). As CIU is the most diverse method, covering approaches from all previous methods, it is considered to have the best expressive power result, being marked in bold.

Regarding the graphical integrity of the explanation, only the methods that output bar plots achieve it completely, as they show clearly a distinction between features with positive (bars to the right) and negative (bars to the left) attributions. LIME, kernelSHAP, and CIU are chosen as the better ones (in bold) for this metric, due to the fact that they fill each bar with a color clearly associated to positive (green for SHAP and CIU, blue for LIME) and negative (red for all) values. As for the morphological clarity, almost all methods highlight the most important features in a clear way. If only provided using a number between 0 and 1, like in Table B.2, it is clear in the way that it can be seen as how much (in percent) each feature affects heart disease prediction. For bar plots, the most important/influential features are made clear by the extension of each bar. For anchors (rules), the importance is made clear by the features present in the decision set. For PFI, by the features with the higher decrease in model accuracy. For CFEs, by the visualization of the frequency of feature changes across all counterfactuals. For ICE and PDP, the importance is seen in the steepness/flatness of the curves (a horizontal line means that feature does not change the prediction of the model and therefore it is not "important"), however, this requires some understanding of these terms. Moreover, the ICE plots can become overcrowded, contributing to the decreasing of its clearness to the end-user. Finally, layer separation is evaluated only for local methods, as it accesses if the original input sample is included in the explanation. This is true for CFEs, in the way that the method provides a parallel plot that connects the (scaled) feature values of each counterfactual and highlights the instance of interest in blue (see R Notebook "03_Explanations.ipynb"). This is also true for the methods that provide bar plots, as the feature values are included on the left side of each bar. For LIME, however, the numerical continuous variables are discretized to obtain interpretable categorical data. So, in bold are the methods that clearly provide the feature values of the instance of interest. Nonetheless, layer separation is particularly important for images, and not when dealing with tabular data instances.

Finally, the last row evaluates if the explanation runtime is fast or slow, based on the elapsed times (provided together with the explanations in subsection 4.1.3). PFI has an asterisk because this method

is included in the RF R package, so its results are immediate. Overall, feature importance/influence explanation algorithms are faster than rule-based ones. In particular, for these methods, CIU is the most efficient (visible particularly when looking at the global explanation runtimes in Table B.2), followed by Shapley and then LIME. Note that the computation time, particularly in SHAP, depends considerably on the number of samples to generate when exploiting the algorithm. Furthermore, in all methods the number of features to which an explanation is being computed affects the speed.

Concluding, CIU seems to be the preferable method for this property, as the levels of structure are covered to the maximum extent, and it is also the most efficient in terms of speed. Nonetheless, all of the methods, except for Anchors (and kernelSHAP), provide an explanation in a reasonable amount of time. Moreover, methods that provide influence/importance values seem to cover more the levels of structure included here, which contradicts the fact that anchors and CFEs are preferable over these. For all methods, a good structure should be included, leading to end-user efficiency and good understandability of the method. It is important to mention, that the aspects related with the structure of each method can always be further developed, when necessary, considering specific necessities or desires of a specific AI deployer (in fact, there are also other implementations of SHAP and LIME, that provide different visualizations, not assessed here). A fast method leads to computational efficiency and practical usability of the method.

Selectivity

Table 4.4 contains the evaluation results for this property, where three different approaches are used to compute the size of the explanations, depending on their type of output (expressive power). Note that for this particular data set, the maximum number of features is 11, so, even when all features are present in the explanation, it is already quite selective. PDP and ICE methods are not included here because both are always illustrated for one feature at a time (they are selective by default).

Table 4.4: Results of selectivity property. Please check Table 3.2 for labels clarification. In bold are the best results.

Model	Metric	Anchors	CFEs	LIME	Shapley	SHAP	CIU	PFI
LR	Explanation size	2	6	7	6	6	7	N/A
RF	Explanation size	2	5	6	8	7	9	6
SVM	Explanation size	2	6	7	7	7	8	N/A
	Size parameter	T	T	**T**	F	F	**T**	F

For Shapley, kernelSHAP, CIU, and PFI, the global FS values (in Table B.2) were used to select the number of features that explain 90% of the underlying model behaviour. For LIME, *mean_size()* function was used to compute the mean over 100 instances of the training data, as this model only provides local FS values. This function can also be used for the other FS methods, to have an idea of how selective are the explanations over the data, as a histogram is plotted together with the result (if parameter $plot = TRUE$). Comparing the FS methods, all of them behave in a similar way, showing the most selective explanation is given by the LR model. The mean number of changed features in the generated (closest) counterfactuals is similar to the number of features that explain 90% of the underlying model for FS methods. Anchors are the most selective method, always considering an average of two

conditions (for all models), which include two features. The distribution over the training data for both methods is illustrated in R notebook “04_Benchmark.ipynb”, in Selectivity Section.

All the results for the explanation size metric show that giving a selective explanation mostly depends on the model being explained, on the data (i.e., the feature values), and also on the type of explanation output. Anchors are usually selective by default. The other methods are not, which does not mean a selective explanation can not be provided, depending on the end-user. It is better for an explanation not to be selective, than not showing a truthful or faithful explanation (other properties). A trade-off should be made, bearing in mind that sometimes it might not be possible to give a selective explanation without seriously compromising the truthfulness property due to the complexity of the underlying model or data. In this sense, the most relevant metric to consider is the Size parameter one. The methods which allow the possibility of changing the explanation size are preferable. For CFEs, it is not possible to set a number for the maximum number of changed features, but it is possible to choose the number of counterfactuals to generate. For a lay person, it is possible to generate just one counterfactual as an explanation. Considering CFEs, it would be possible to show to patient A just one counterfactual (the first line of the data frame in Figure 4.15, the one that has the clinical values more similar to patient A). For LIME and CIU, it is possible to select the number of features to display in the bar plot. For this reason, the T is in bold for these two methods, being the preferred ones for this property. For LIME and Anchors, it is also possible to choose multiple (or just one, of course) instances to compute an explanation, which is an advantage.

For FS methods, the function *mean_size()* can actually be employed in a more useful way, by using it to select the number of features that are crucial to explain the underlying model to the highest extent (default is 90%, but this parameter can be changed). This is exemplified in Figures 4.20 e 4.21 for LIME and CIU, respectively, showing that it is possible to give a (more) selective without compromising other properties. These bar plots are the selective version of the explanations illustrated in Figures 4.10 and 4.17, respectively. As CIU provides distinct FS values, it is possible to select which are the crucial features considering different CIU variables, as shown on the left of Figure 4.21. On the right, the 7 most important features (CI as the CIU variable to consider) are used to display the prediction of patient A using CI, as they should be the ones to display. However, if a person is interested in comparing his/her prediction with a baseline one, it may be more relevant to use the features selected using CInfl values.

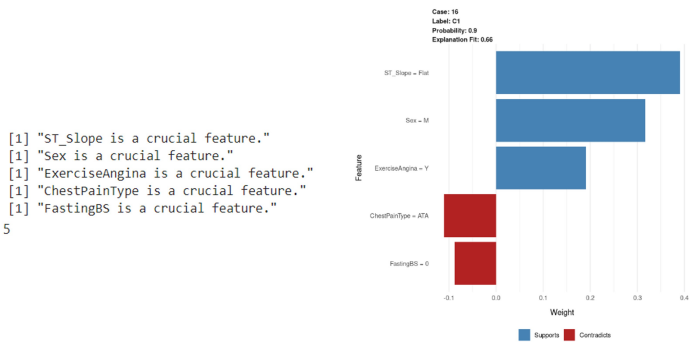


Figure 4.20: LIME selective explanation. Left: Selected features (5 in total) that explain 90% of the LR model’s output. Right: LIME giving an explanation for patient A prediction only with the crucial (selected) features.

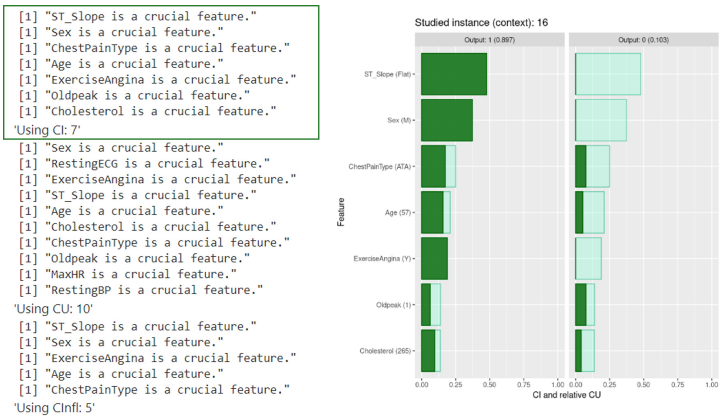


Figure 4.21: CIU selective explanation. Left: Selected features, considering different CIU variables, that explain 90% of the LR model's output. Right: CIU giving an explanation for patient A prediction only with the most important features (7 features were selected considering CI as the CIU variable).

Contrastivity

Table 4.5 contains the evaluation results for this property, where two different approaches were used to compute the target sensitivity, depending on the type of output (expressive power). The level of contrastivity is a qualitative metric that should always be considered, and that assesses how contrastive the explanations are to a predefined output or/and to the current instance. Global methods (PDP and PFI) are not included here, as they are not contrastive.

Table 4.5: Results of contrastivity property. Please check Table 3.2 for labels clarification. In bold are the best results.

Model	Metric	Anchors	LIME	Shapley	SHAP	CIU	CFEs	ICE
	Level of contrastivity	F	T	T	T	**T**	T	T
LR	Target Sensitivity	0	**0.41**	0.27	0.23	0.34	N/A	N/A
RF	Target Sensitivity	0	0.33	0.25	0.23	**0.62**	N/A	N/A
SVM	Target Sensitivity	0	0.32	0.21	0.21	**0.40**	N/A	N/A

Regarding the first metric, Anchors does not reveal any level of contrastivity in the explanations showed in the previous section. CFEs are always contrastive to the current instance, which is clear from the changes in the feature values. ICE plots, when including the PDP (using a yellow curve, like in Fig 4.6), are contrastive to the average prediction. FS methods that provide influence values (SHAP, Shapley, LIME, and CIU) are contrastive to the predefined baseline, which for the current problem is the average prediction. CIU is in bold (best result) due to the fact that the provided explanations, besides being contrastive to a predefined output by using CInfl values, are also contrastive to the current instance by using relative CU values, which show how to improve the respective feature values (CIU is also counterfactual).

To compute the second metric, an adversarial attack to patient A data was simulated. The closest counterfactual (first line of each data frame in Figure 4.15) found by CFEs method for each model was used as the slightly perturbed data instances to fool the respective models into changing the prediction to class 0. The data with the original values and the changed ones for patient A are depicted in Figure 4.22, together with the predicted probability for class 1.

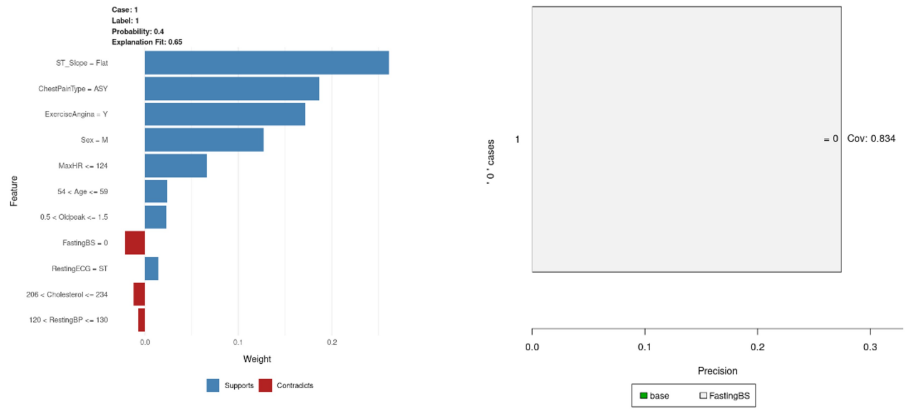
For Anchors, the percentage of features in the original conditions that are in the "new" conditions

A data.frame: 4 × 12

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	C1_pred_prob
	<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<dbl>
patient A	57	M	ATA	140	265	0	ST	145	Y	1.0	Flat	0.85
fooled LR	58	M	NAP	130	213	0	ST	140	N	0.0	Flat	0.46
folled RF	57	M	ASY	130	207	0	ST	96	Y	1.0	Flat	0.40
fooled SVM	50	M	NAP	140	233	0	Normal	163	N	0.6	Flat	0.46

Figure 4.22: Adversarial attack simulation for patient A. The first row corresponds to the original clinical feature values for patient A. The others correspond, respectively, to the first rows of the data frames in Figure 4.15, showing the values that fool the respective models into the prediction depicted in the last column.

is 0 for all models, which means the method is always target-sensitive, i.e., class-specific. For the methods that provide FS values, the L2 metric between the explanations before and after the adversarial attack was computed. The results are illustrated in Table 4.5, which shows the best results for CIU (RF and SVM model) and LIME (LR model). However, these results should be considered with caution, together with the visualization and summary tools that each method provides. The reader is referred to R notebook "04_Benchmark.ipynb" Section Contrastivity. All the methods prove to be class-specific, showing good scores for the target sensitivity metric. However, in the case of the RF, not only the model is fooled, but also the XAI methods, especially the ones that use a surrogate model/system as explanations, such as LIME and Anchors (see Figure 4.23). This is due to the unexpected behaviour of the model around the fooled instance of interest. The other methods also show unexpected FS values (see Figures 4.24 and 4.25), which may actually lead to the perception that something wrong is happening (the adversarial attack). However, only by visualizing CIU plots, illustrated in Figure 4.25, it is possible to in fact understand what is happening - it seems that an outlier was found. In fact, with these CP plots, it is even possible to understand the reason for the explanations provided by the other methods.



Linear model prediction = 0.90 (class 1) Anchors prediction = 0.27 (class 0)

Figure 4.23: LIME (left) and Acnhors (right) RF explanations after adversarial attack simulation for patient A. Although the real predicted probability for class 1 is 0.4 (class 0 is predicted), the displayed explanations use the feature weights obtained using LIME and Anchors surrogate methods, which in fact do not predict class 0, but class 1 with a high probability.

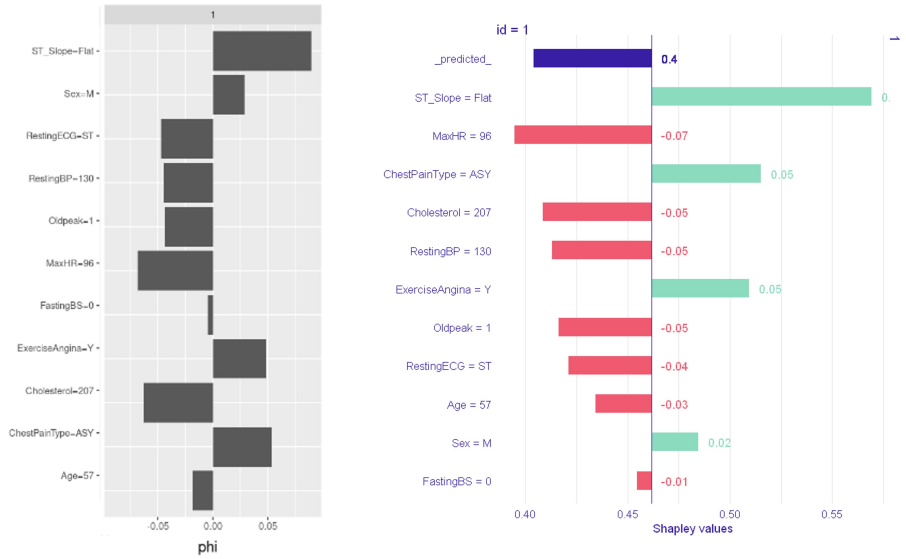


Figure 4.24: Shapley (left) and kernelSHAP (right) explanations after adversarial attack simulation for patient A. The influence values displayed in the bar plots are very close to 0, which is unexpected.

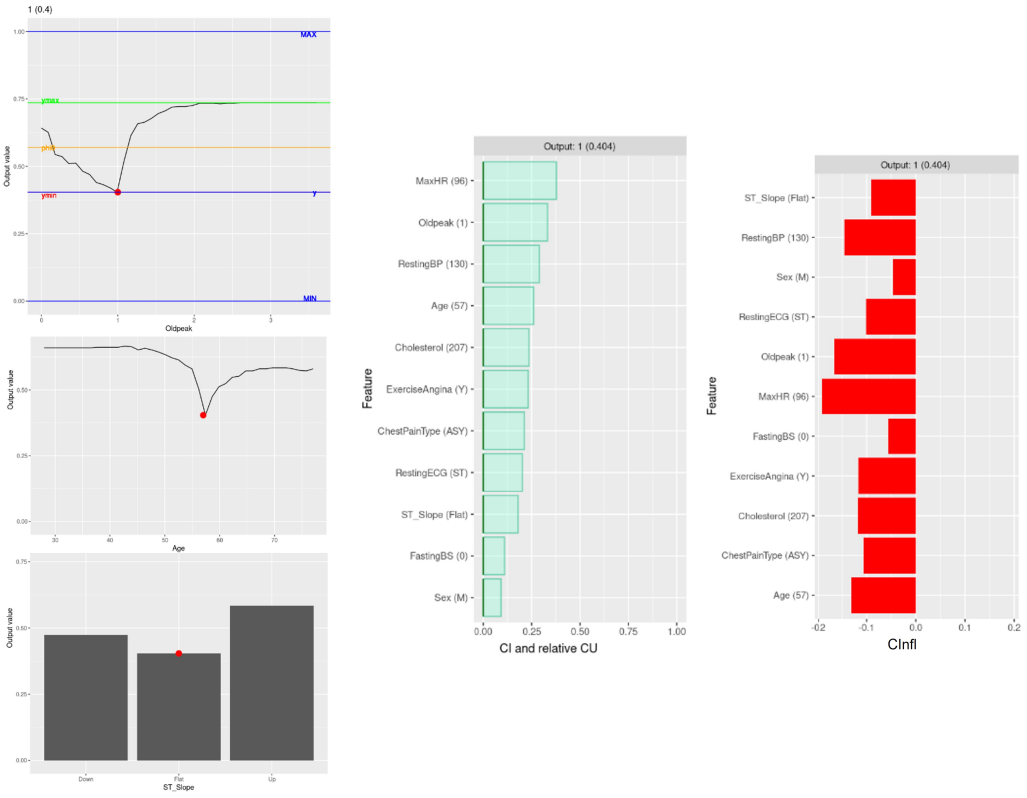


Figure 4.25: CIU explanations after adversarial attack simulation for patient A. The CIU values (middle) and CInfl values (right) that explain the output prediction for class 1 (0.4) seem very weird, as CU values are always 0 (and consequently the CInfl values are all negative). This can be explained by the input-output plots on the left, that show patient A is a local minimum of the RF model.

Interactivity

None of the XAI methods being compared includes a demo interactive tool that allows to easily access the explanations, i.e., without actually going through the implementation code. It is highly recommended that an interactive tool is added to the authors GitHub page, which can simply be a demo

example for a common and easy application like performed for the LRP tool presented in Section 3.1.5. For the tabular domain, a widely known and simple example like the housing price prediction problem, could be implemented together with an interactive tool (similar to shapash demo - housing price) in which users would only have to control the feature values (number of floors, year sold, etc.) to obtain a prediction. Then, it would be possible for AI deployers and end-users to easily access the explanations, without having to implement any code or functions (this is for AI developers). Considering the present problem, the heart disease prediction, the deployer would be a hospital and the end-user the doctors (and possibly patients). Having this type of demo, even if in another application domain, it is possible for them to conclude if the explanations provided are detailed enough, easily understandable, and also how controllable and easy to interact they are. The last part is important, as is has been shown that people tend to prefer interactive explanation results, being more usable and useful than fixed explanations [9]. Then, the specific social context of the explanation should be described by the AI deployer (in this case, a hospital), so that an effective interactive XAI tool for the target end-users can be "build".

It is important to mention that the LIME method used here, in R, provides an application for interactively exploring text models, but only when some code is implemented to derive the explanation. Nonetheless, it is an advantage over the other methods, showing that the integration of an interactive tool is easily achieved. When implementing the methods, AI researchers can also have an idea on how easy it is to obtain an explanation. If it is difficult to obtain an explanation, then it is probably also difficult to make a clear interactive and controllable explanation. For example, anchors and LIME suffer from a highly configurable setup, where the chosen perturbation space and the tuning hyperparameters have a great impact on the algorithm which can lead to non-meaningful results. For the end-user, it is good to have some configurable parameters, such as the explanation size or the type of output to display, but not complex ones that should be optimized by the methods themselves. For CIU, only the hyperparameter *sample.size* related with the configurations of the method itself is controllable (default is 100, meaning 100 instances are sampled for estimating CI and CU), and when tuned the results do not suffer a meaningful modification (related with the accuracy). It is in fact the only non-deterministic parameter in CIU, which makes it more stable than other perturbation-based methods - see Section 4.2.1.

Fidelity

From all the methods used to create model explanations, only LIME and Anchors implement proxy approaches and consequently compromise their fidelity. KernelSHAP uses a linear surrogate model g to estimate shapley values as an explanation. Moreover, both XAI methods that estimate shapley values (the "original" and SHAP) are AFA methods, meaning fidelity is compromised, linearity assumptions are made, but it is not possible to calculate a fidelity score, as the method does not provide any metric possible to use to estimate it. All the other explainability approaches (PDP, PFI, ICE, CIU, and CFEs) do not create any proxy model g or make any linearity assumption about the underlying descriptive model, and therefore the fidelity is 100%, being the preferred ones when considering this property.

The SA metric was used for LIME and PC metric for Anchors. For both methods, the 100 randomly selected data samples from the training set were used to get the mean score depicted in Table 4.6 and

Table 4.6: Results of fidelity property. Please check Table 3.2 for labels clarification. For Anchors: precision (coverage).

Model	Metric	LIME	Anchors
LR	SA / PC	0.90	0.94 (0.36)
RF	SA / PC	0.83	0.90 (0.35)
SVM	SA / PC	0.89	0.93 (0.39)

the histograms showing the fidelity distribution over the selected instances (illustrated in Figure B.2). Overall, LIME performs better than Anchors due to the fact that Anchors have low coverage. Although the latter are seen as easily understandable, rules can easily "trick" the end-users by having low coverage. Moreover, it is visible from the histograms that there are some cases where these methods (especially LIME) do not achieve local fidelity, and therefore show an incorrect explanation (all the data instances to the left of the red vertical line). This happens above, for the explanations computed after the adversarial attack on patient A data, for the RF model, and also for patient C. The fidelity score of the explanation obtained using LIME and Anchors is 0.44 and 0.27 (0.83), respectively. This happens due to the fact that the RF model is highly non-linear around that instance. LIME fails because it assumes linearity; Anchors also fails, although theoretically it can deal with non-linear and complex model predictions (note that coverage is high, which happens due to the fact that a "step" happens in the model function - visible in Figure 4.25). CIU, being 100% loyal to the model, in that same case, shows unexpected results due to the unexpected behaviour of the RF model around that specific interest, and clearly shows it through the input-output plots in Figure 4.25, with respective CIU values calculation (top left). This is preferable over giving a non-meaningful explanation like it happens for other methods.

Faithfulness

	Coefficients
	<dbl>
ChestPainTypeTA	-2.61
SexM	2.07
ChestPainTypeNAP	-1.70
ST_SlopeFlat	1.63
ChestPainTypeATA	-1.40
ExerciseAnginaY	1.28
ST_SlopeUp	-0.86
FastingBS1	0.55
Oldpeak	0.55

Figure 4.26: The results of fitting a LR model on the heart dataset. Shown are the features used in the model and their estimated coefficients (in terms of the odds).

To evaluate the faithfulness of each XAI method, the three metrics mentioned in Section 3.2 were used. Regarding the ID and ROAR metrics, only FS methods (except PDP and ICE, because they simply provide input-output plots - they are always faithful to the underlying model) can be assessed. The results obtained for these two metrics are very similar, and for that reason a table with the obtained scores is not provided. Overall, all the methods behave well, showing better performances than the random explainer. In other words, all the XAI methods are able to catch (globally and locally) the most relevant features used by the models. This can be visualized in Figure 4.27. Note that PFI, by being a more translucent method, is more faithful to the underlying model, as it "looks" at the inner workings of the model.

The metric that better evaluates the faithfulness to the underlying model is WBC. LR was used because it is a white-box model so the explanations can be compared to the true reasoning of the model. Figure 4.26 shows the estimated model coefficients (top 9). Despite being considered an interpretable model, these coefficients represent odds, which are not the most in-

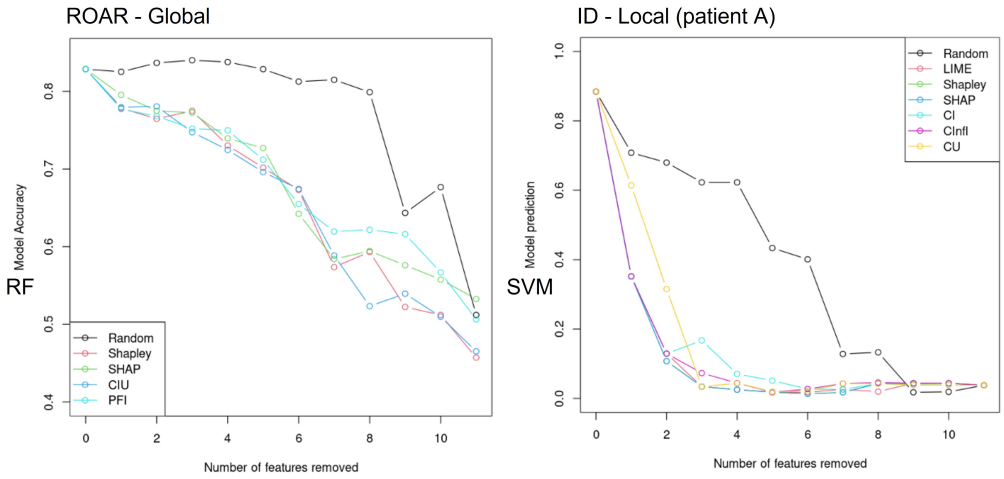


Figure 4.27: Plots obtained with ROAR (left) and ID (right) faithfulness metrics. ROAR assesses the change in RF model test accuracy after incrementally remove the features by decreasing order of (global) importance. ID assesses the change in SVM model prediction after incrementally remove the features by decreasing order of (local) attribution value. The baseline values used for ID were typical values for person with NO heart disease (healthy).

tuitive values to interpret; still, when the coefficient is positive, increasing that feature value will increase the heart disease probability. Looking at the data frame, it is visible that the most important features, in global terms, are ChestPainType, Sex, and ST_Slope. Looking at the global feature importance values presented in Table B.2 for the LR model, all the methods (Shapley, kernelSHAP, and CIU) consider these 3 as the most important (highest percent values), meaning they agree with the underlying model. Looking at the local explanations provided in Section 4.1.3 for each method, it is possible to assess the local faithfulness to the LR model. Focusing on the heatmap provided by LIME in Figure **??**, the feature weights provided as influence values agree to some extent with the LR model, however, although in terms of sign there is an agreement, in terms of magnitude there is not. For example, ChestPainType=TA should be redder (contributes less to heart disease prediction) than ST_Slope=Flat, according to the underlying model. Anchors explanation for LR prediction for patient A agrees with the underlying model in a way that the two most important features with a positive contribution for the output are present in the rule. CFEs is more difficult to evaluate, although it is possible to conclude, from Figure 4.15, that the changes on the feature values are made accordingly to the LR coefficients.

As mentioned, a logistic regression model is a known model, but it is still not super clear in terms of feature weights. So, a simple linear regression model was used to compare the explanations provided by LIME, SHAP(ley), (the most use methods in the literature) and CIU. The example of the calculation of the weighted average grade of a university student given in page 17 was used[11]. Paul grades were used as the instance of interest, and it is possible to check in Table 4.7 that CI, CU and CInfl (ϕ) values obtained correspond exactly to what was expected and calculated before. It is also confirmed that in fact LIME, Shapley, and SHAP give influence values (these values were converted using the same utility function as CIU for comparison), and not importance, meaning they are only relevant when compared to a predefined baseline. Table B.3 shows results from students with the average and maximum grades in all courses, showing the extent to which CI and CU values provided by CIU are crucial to explain

[11] CIU can use the function directly as the studied model, whereas the other methods require the availability of a training set and a trained model.

the predictions. For example, for a student with average grade in all courses (15), LIME and Shap(ley) values give 0, just like CInfl, but CI and CU values provide the actually importance (0.2, 0.3, and 0.5) of each course and the utility (0.5), respectively. Table B.3 also shows results for the global feature importance method in using $mean(CI)$ and $mean|Shapleyvalue|$. Only the CIU method retrieves the original weights of the linear model with zero variance, as CI values are identical for all instances in the case of linear models. Therefore, importance as defined by CI is conceptually identical to global feature importance. Even though Shapley values (standard approach and kernelSHAP) estimate instance-level influence influence they still give similar values as CI but with a high variance.

Table 4.7: Results of WBC metric for a linear regression model: $f(x) = 0.2x_1 + 0,3x_2 + 0,5x_3$ - example from page 17. Instance explained is Paul.

Paul	LIME	Shapley	SHAP	CI	CU	Cinfl
x1=10	-0.1	-0.1	-0.1	0.2	0	-0.1
x2=19	0.14	0.14	0.12	0.3	0.9	0.12
x3=15	-0.06	0.03	0	0.5	0.5	0

Stability

Table 4.8 contains the evaluation results for this property, where two computational metrics were used: stability for identical (Identity) and stability similar inputs (Similarity). PDP, ICE, PFI, and CFEs are not included here, as they are completely stable for identical inputs due to the adopted implementation approaches (deterministic). The Similarity metric does not apply. Regarding Anchors, this method also computes feature weights, that also vary, although usually it does not change the rule conditions. So, this property is usually only evaluated for FS methods, specifically the ones that give feature importance/influence values for all input features (tabular domain).

Table 4.8: Results of stability property. Please check Table 3.2 for labels clarification. The CIU variable used here was CInfl, so it also applies for CI and CU. In bold are the best results.

Model	Metric	LIME	Shapley	SHAP	CIU
LR	Identity	0.08	0.23	0.06	**0.00**
RF	Identity	0.12	0.28	0.06	**0.01**
SVM	Identity	0.11	0.28	0.05	**0.00**
LR	Similarity	0.28	0.33	0.25	**0.17**
RF	Similarity	0.37	0.40	**0.28**	**0.28**
SVM	Similarity	0.45	0.54	**0.38**	0.39

CIU is clearly the most stable method, and secondly is kernelSHAP. Although in the literature the most used metric is Similarity, the most important metric to be assessed is Identity. Of course, similar input instances should have similar results, including model predictions and explanations. But identical inputs should always have identical explanations; a patient (or a doctor) cannot have different explanations for his/her heart disease prediction result when checking twice (or more times). The method that provides the higher feature variability for the same input is Shapley, which is problematic for deployment. The scores presented in Table 4.8 can also be visualized in the form of violin plots, which show the distribution of ϕ values from 50 (default) runs with the studied instances/models and the studied methods - see notebook "04_Explanations.ipynb" Section Stability.

As stated above, it is important that similar inputs provide similar explanations, which usually happens. That being said, a better use of the function *find_neighbors* used to calculate the feature similarity can be made: it can be used to show the explanation for the instance of interest together with the explanations, for example, for the 2 most similar instances. In the heart problem, a doctor can see patients with similar clinical features and assess the explanations.

Certainty

All explanations given by the XAI techniques provide the ML model's prediction probability, on which they are based. Regarding (un)certainty of the methods, only LIME and Anchors (developed by the same authors) provide some confidence measure regarding the (local) explanations. Note that the only methods that provide a measure of explanation certainty are the ones that use surrogate models, and so they only provide measures associated with the fidelity of the explanation towards the black box model. LIME provides the model prediction (right prediction) together with the local model prediction (surrogate prediction). From here the user can have an idea about the fidelity of the explanation towards the black box model (this is calculated above – see section 4.2.1). It also provides a probability information measure called "Explanation Fit", which corresponds to the surrogate model r2 - which can be considered a PC metric. However, besides (and more importantly) the model r2, the model fidelity should be provided. For example, in Figure 4.23, the explanation fit is reasonable, but the fidelity is very low, which can be misleading to the end-user. Anchors gives explanations with two certainty measures, precision and coverage. It is similar to the r2 value provided by LIME, as it evaluates the proxy model itself - PC metric. Using the same example, Anchors behaves better than LIME because it actually shows the precision is very low.

PDP and ICE methods do not need to provide any (un)certainty because they focus in CP profiles, which simply show the model behaviour over the range of values of a feature i. PFI, CFEs, and kernelSHAP also do not provide any certainty measures regarding the approach they use. Shapley method summarizes the distributions of the variable-specific contributions for the selected random orderings. These variance values give an idea of coverage. Finally, CIU values can be "read" directly from input-output plots, showing exactly where the calculated values come from. This makes CIU quite transparent at least when compared to other methods, like LIME, Shap(ley), and Anchors. The latter might be considered black-boxes themselves, as they involve very complex approaches difficult to understand when the main idea of XAI is in fact to make the model (and of course the explanations) understandable for the end-users. One big problem with LIME, in particular, is the definition of the kernel settings, which are not clearly explained by the authors and can lead to big differences in the explanations (affecting mostly the faithfulness property, besides not providing certainty). Developers can make a design choice whether their XAI method will contain a certainty measure regarding the output of model f. As it has been argued that referring probabilities might not be so effective, since people have difficulties to correctly estimate probabilities [69], it should be decided by the end user, whether to see the certainty measures or not.

Truthfulness

The truthfulness property assesses if the explanations are in concordance with the user "true" world. This actually depends on the model itself, on which the explanations are given for. For an explanation to be truthful, the data provided to the ML models, on which they learn, also needs to be truthful. For example, it is known that men are more likely to develop heart disease than women; this information was present in the data; the models learnt from this data; the explanations show that when Sex=M, this has a positive influence on the heart disease probability. The same happens for ST_Slope=Flat.

The two suggested metrics for the evaluation of this property consisted in comparing different XAI methods across the models (Methods Agreement) and comparing different XAI models across the methods (Models Agreement). This property was left to the end because until now this type of comparisons have been made. If the methods prove to have high scores in all the previous properties, it is almost certain that it will also cover truthfulness. Overall, the methods seem to agree on the selected features provided in the explanations, specifically in terms of order of importance/influence. In terms of model agreement, LR and SVM seem to agree more with each other when comparing with RF model. LR and RF revealed similar predictive accuracy performances. The first, being a more flexible model, missed an interaction between Age and MaxHR suggested by RF (and SVM), which was shown by PDPs in Figure 2.4. This type of methods that perform model inspection is very useful for detection of model bias. Although this interaction was spotted by RF, explanations revealed that this model has associated bias, specifically for the case presented in Figure 4.25, in which CIU is the most helpful method in terms of model improvement. Concluding, comparing the models shows that the best model for this problem is SVM; comparing the methods shows that some methods are more useful than others for bias detection. In terms of truthfulness, Anchors seems to be the method with the worst performance, as being very selective, it may miss some important features (note that sometimes selectivity is preferred). Moreover, with LIME and SHAP, it is possible to hide biases [97], which is a big disadvantage, as the end-users cannot be sure about the truthfulness of the explanation they are receiving. This is a problem associated with perturbation-based XAI methods, where generated instances can potentially be out-of-distribution (OOD), and be used to fool the explanations [97]. CIU is also a perturbation-based method, but, as it transparently shows how the CIU calculations are performed, it is not possible to hide such biases.

All of the methods introduced here analyze each input features independently from the other features. However, more often than not, features are correlated and jointly present some aspects of observations [44], which compromises the truthfulness of the methods. For example, FastingBS and Cholesterol in the heart data are correlated, and both are related to the blood glucose levels of the patient. Many of the methods can be generalized to allow a joint analysis of groups of two or more features [44]. CIU is the only method here that provides that possibility through the use of intermediate concepts (these were not studied here, but were introduced in Section 2.3.3): permutations are done for a group of concepts (explanatory variables), allowing for evaluation of the importance, utility and influence of the entire group. That being said, CIU seems to be the most relevant of the compared methods for the successful achievement of this property.

4.2.2 Enhanced Solution

The application of the framework showed that explainability is a multi-faceted concept and that the 10 properties are all connected to some extent and contribute to the complete evaluation of XAI methods regarding its explanation quality or validation and its target group. Validation properties are fidelity, faithfulness, stability, (un)certainty, contrastivity, and truthfulness. It is important that all of them are covered by an XAI method to the maximum extent. It was made clear from the framework application that fidelity should not be compromised and when this happens, the most important metric to assess is SA, that should be provided together with the explanations. As the methods usually rely on perturbations of the input space and observations in changes of the model output, faithfulness is usually present. A white-box check like the one made with a simple linear regression model is relevant to clearly assess the values that the methods are giving and see how reliable they are. Stability for identical inputs is not commonly evaluated in the literature and it should definitely be, as a method that give FS values that change randomly from one explanation to the next cannot be used in real word scenarios, especially in high-risk domains, like medicine. Regarding the (un)certainty, more than providing probabilistic measures of confidence, explanations should be to the most extent transparent in the implemented approaches, and not themselves black-boxes which are very difficult to actually interpret. Finally, contrastivity in terms of target sensitivity revealed to be an important aspect to consider, as it is possible to detect, through explanations (this is the goal), if an adversarial attack happened. Truthfulness is a validation property in a way that it can detect the capability of a method to detect bias (and to hide), which can then lead to an improved and unbiased ML model for deployment. Contrastivity and truthfulness also assess the quality of the explanations, together with the remaining 5 properties. These are relevant, but can usually depend on the application, and sometimes can actually be improved, like mentioned for the interactivity, structure, and selectivity properties.

In summary, these observations contribute to an enhanced overview of the developed application-agnostic framework. It is important to get insight into all properties so that a fair trade-off can be achieved. Some properties might be more relevant than others in different contexts, and it is on the AI developer and deployer to choose the best technique(s), bearing in mind that validation properties increase the usability, and quality properties increase the understandability, which together contribute, at different levels, to each of the requirements for trustworthy AI. Finally, it was also possible to conclude about the relevancy of the CIU method, which showed to outperform the others.

Chapter 5

Conclusions

This chapter highlights the main conclusions of the present dissertation. Additionally, section 5.1 points out the encountered limitations and suggestions for future work.

The main goal of the present dissertation was to build a benchmark framework for XAI methods, which evaluation is still a novel and inconclusive field in the literature, and compare different XAI methods using the developed framework. A number of properties was first identified in SoTa literature. In particular, it was verified the lack of a systematic organization of the properties devoted to XAI evaluation, and the lack of quantifiable and objective metrics. Furthermore, the lack of common agreement regarding XAI-related concepts makes this task even more difficult. For that reason, terminology clarification was made, and, most importantly, a selection of 10 properties was made, based on the former identified properties so that a comprehensive and consensual benchmark framework for XAI methods could be developed. Then, this collection of evaluation properties was extensively described and objective metrics were proposed for different types of methods and data. As it is highly dependent on the type of data, the respective metric(s) formalization for tabular data was made, as a concrete example of the (computational) metrics that can be used. Finally, the framework, which can be applied to any type of application domain, was validated in a real-word scenario in the medical domain: heart disease prediction. The comparison of different XAI methods showed the relevancy of the CIU method, which covers to a better extent the selected properties of explainability, when compared to other methods. Nevertheless, suggestions regarding each of the methods considering different properties was made, and it was concluded that explainability is a multi-faceted concept; the 10 properties are all connected to some extent and contribute to the complete evaluation of XAI methods.

Objective evaluation can provide quantitative (computational) metrics without requiring user-studies [36]. The latter was the focus of the proposed framework. However, it should be noted that this is not to replace human-centered evaluations, but can in fact guide in the selection of the best techniques to present to participants in a user-study. Then, properties, specifically properties that evaluate the quality of explanations and that users can help improving (structure, interactivity, and truthfulness) may be evaluated in a human-ground scenario, improving the efficiency of the assessment of XAI methods [36]. CIU, as one promising XAI method to deploy, can be evaluated in a human-grounded way, so that

its visualization and controllable tools can be improved and made more efficient. The explanations also can, and should be checked with domain-experts.

As mentioned above, it might be unreasonable to expect an XAI method to cover completely the 10 selected properties. In practice, trade-offs between desired explanation properties will have to be made when developing or choosing an XAI method. As examples, faithfulness might contradict with truthfulness; and selectivity and truthfulness have impact on each other. The application domain, practical usability, or nature of the prediction task, can determine which properties should be underlined [8, 72]. In the light of this, it is proposed to firstly evaluate (and consequently improve) explanations for validation-related properties (fidelity, faithfulness, and stability in particular), without considering the simplification or "embellishment" of the given explanation. Secondly, a further analysis can consist on the evaluation (and consequent improvement) of explanations for quality-related properties, where the user social context, preferences, and cognitive capacity properties should be incorporated. At this step, the human-grounded evaluation can be integrated together with the proposed objective metrics. Moreover, insights from experts in the humanities fields (e.g. psychologists, sociologists, and anthropologists) can also provide a multi-disciplinary view on XAI and contribute to more innovative and relevant results.

To sum up, the present dissertation has as main contribution a benchmark framework, which concretely addresses how to evaluate different aspects of explainability methods, and therefore provides guidance to AI developers, useful information to AI deployers, and recommendations on how to make the explanations more accessible to end-users. The benchmark framework provides all the target-groups resources to assess each of the 10 properties while utilizing a common formalism and taxonomy, which promotes the uniformity that is lacking in XAI field.

5.1 Limitations and Future Work

The evaluation and comparison of the selected XAI methods considering the 10 properties was based in a simple case for the tabular domain, where only ML models that learn from structured features were used; so the explanations are based on feature assessment. As future work, it is relevant that other more complex ML models, in particular DL models that do not rely on feature engineering, are used for the comparison of different XAI methods. Future work should assess the relevancy of the CIU method with NN-based XAI methods (such as the known LRP method), that adopt different strategies, like gradient-based ones, using the suggested benchmark framework. Further extensions to address concern the application of the presented work to other models, data types, applications, and contexts.

Regarding the implementation of the XAI methods, kernelSHAP and Anchors R implementation faced some difficulties; the first did not work in JupyterLab, the second took way more time to run than the Python implementation. So, in the future, especially when explaining DL models, it might be more pertinent to perform these experiments in Python.

Finally, despite an objective assessment of explanations is necessary for comparison of XAI methods, as future work they can be complemented with human-grounded evaluation metrics, putting forward dynamic design cycles under a human in the loop paradigm.

Bibliography

[1] E. Tjoa and C. Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021. doi: 10.1109/TNNLS.2020.3027314.

[2] S. Knapič, A. Malhi, R. Saluja, and K. Främling. Explainable artificial intelligence for human decision support system in the medical domain. *Machine Learning and Knowledge Extraction*, 3(3):740–770, 2021. ISSN 2504-4990. doi: 10.3390/make3030037. URL https://www.mdpi.com/2504-4990/3/3/37.

[3] M. Esmaeili, R. Vettukattil, H. Banitalebi, N. R. Krogh, and J. T. Geitung. Explainable artificial intelligence for human-machine interaction in brain tumor localization. *Journal of Personalized Medicine*, 11(11), 2021. ISSN 2075-4426. doi: 10.3390/jpm11111213. URL https://www.mdpi.com/2075-4426/11/11/1213.

[4] J.-B. Lamy, B. Sekar, G. Guezennec, J. Bouaud, and B. Séroussi. Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach. *Artificial Intelligence in Medicine*, 94:42–53, 2019. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2019.01.001. URL https://www.sciencedirect.com/science/article/pii/S0933365718304846.

[5] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2019.12.012. URL https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[6] C. Molnar. *Interpretable Machine Learning*. 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book.

[7] High-Level Expert Group on AI. Ethics guidelines for trustworthy ai. Report, European Commission, Brussels, Apr. 2019. URL https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.

[8] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

[9] R. Guidotti et al. A survey of methods for explaining black box models. 2018. doi: 10.48550/ARXIV.1802.01933. URL https://arxiv.org/abs/1802.01933.

[10] G. Vilone and L. Longo. Explainable artificial intelligence: a systematic review. 2020. doi: 10.48550/ARXIV.2006.00093. URL https://arxiv.org/abs/2006.00093.

[11] R. Moraffah et al. Causal interpretability for machine learning - problems, methods and evaluation. *SIGKDD Explor. Newsl.*, 22(1):18–33, may 2020. ISSN 1931-0145. doi: 10.1145/3400051.3400058. URL https://doi.org/10.1145/3400051.3400058.

[12] C. Molnar et al. Interpretable machine learning – a brief history, state-of-the-art and challenges. In *ECML PKDD 2020 Workshops*, pages 417–431, Cham, 2020. Springer International Publishing. ISBN 978-3-030-65965-3.

[13] K. Främling. Explaining results of neural networks by contextual importance and utility. In R. Andrews and J. Diederich, editors, *Rules and networks: Proceedings of the Rule Extraction from Trained Artificial Neural Networks Workshop, AISB'96 conference*. Brighton, UK, 1996.

[14] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL https://doi.org/10.1093/mind/LIX.236.433.

[15] B. Copeland. Alan turing: British mathematician and logician. https://www.britannica.com/biography/Alan-Turing, note = "Accessed: 2022-09-16", August 2022.

[16] C. Manning. Artificial intelligence definitions. urlhttps://hai.stanford.edu/sites/default/files/2020-09/AI-Definitions-HAI.pdf, note = "Accessed: 2022-09-16", September 2020.

[17] J. McCarthy. What is artificial intelligence? http://jmc.stanford.edu/articles/whatisai.html, note = "Accessed: 2022-09-16", November 2007.

[18] A. Kaplan and M. Haenlein. Siri, siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, 62(1):15–25, 2019. ISSN 0007-6813. doi: https://doi.org/10.1016/j.bushor.2018.08.004. URL https://www.sciencedirect.com/science/article/pii/S0007681318301393.

[19] I. C. Education. Artificial intelligence (ai). https://www.ibm.com/cloud/learn/what-is-artificial-intelligence, note = "Accessed: 2022-09-16", June 2020.

[20] M. Turek. Explainable artificial intelligence (xai). https://www.darpa.mil/program/explainable-artificial-intelligence, note = "Accessed: 2022-09-16", 2018.

[21] A. Bibal, M. Lognoul, A. de Streel, and B. Frénay. Legal requirements on explainability in machine learning. *Artif. Intell. Law*, 29(2):149–169, June 2021.

[22] A. J. London. Artificial intelligence and black-box medical decisions: Accuracy versus explainability. *Hastings Cent. Rep.*, 49(1):15–21, Jan. 2019.

[23] A. Vellido. Societal issues concerning the application of artificial intelligence in medicine. *Kidney Dis. (Basel)*, 5(1):11–17, Feb. 2019.

[24] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King. Key challenges for delivering clinical impact with artificial intelligence. *BMC Med.*, 17(1):195, Oct. 2019.

[25] S. Lo Piano. Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. *Humanit Soc Sci Commun*, 7(1), Dec. 2020.

[26] H. K. Dam, T. Tran, and A. Ghose. Explainable software analytics, 2018. URL https://arxiv.org/abs/1802.00603.

[27] C. B. Azodi, J. Tang, and S.-H. Shiu. Opening the black box: Interpretable machine learning for geneticists. *Trends in Genetics*, 36(6):442–455, 2020. ISSN 0168-9525. doi: https://doi.org/10.1016/j.tig.2020.03.005. URL https://www.sciencedirect.com/science/article/pii/S016895252030069X.

[28] M. Yap et al. Verifying explainability of a deep learning tissue classifier trained on RNA-seq data. *Sci. Rep.*, 11(1):2641, Jan. 2021.

[29] L. Wu, R. Huang, I. V. Tetko, Z. Xia, J. Xu, and W. Tong. Trade-off predictivity and explainability for machine-learning powered predictive toxicology: An in-depth investigation with tox21 data sets. *Chem. Res. Toxicol.*, 34(2):541–549, Feb. 2021.

[30] S. Sarp, M. Kuzlu, U. Cali, O. Elma, and O. Guler. An interpretable solar photovoltaic power generation forecasting approach using an explainable artificial intelligence tool. In *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, 2021. doi: 10.1109/ISGT49243.2021.9372263.

[31] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. F. Moura, and P. Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 648–657, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3375624. URL https://doi.org/10.1145/3351095.3375624.

[32] A. Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural Comput. Appl.*, 32(24):18069–18083, Dec. 2020.

[33] F. Bodria et al. Benchmarking and survey of explanation methods for black box models. 2021. doi: 10.48550/ARXIV.2102.13076. URL https://arxiv.org/abs/2102.13076.

[34] D. Doran, S. Schulz, and T. R. Besold. What does explainable ai really mean? a new conceptualization of perspectives, 2017. URL https://arxiv.org/abs/1710.00794.

[35] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017. URL https://arxiv.org/abs/1702.08608.

[36] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5), 2021. ISSN 2079-9292. doi: 10.3390/electronics10050593. URL https://www.mdpi.com/2079-9292/10/5/593.

[37] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat Mach Intell*, 1(5):206–215, May 2019.

[38] R. R. Selvaraju et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019. doi: 10.1007/s11263-019-01228-7. URL https://doi.org/10.1007%2Fs11263-019-01228-7.

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization, 2015. URL https://arxiv.org/abs/1512.04150.

[40] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. URL https://arxiv.org/abs/1602.04938.

[41] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. *Proc. Conf. AAAI Artif. Intell.*, 32(1), Apr. 2018.

[42] K. Främling. Extracting explanations from neural networks. In *ICANN'95 proceedings*, volume 1, pages 163–168. Paris, France, 9, 1995.

[43] M. T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning, 2016. URL https://arxiv.org/abs/1606.05386.

[44] P. Biecek and T. Burzykowski. *Explanatory Model Analysis*. Chapman and Hall/CRC, New York, 2021. ISBN 9780367135591. URL https://pbiecek.github.io/ema/.

[45] K. Främling. Contextual importance and utility: atheoretical foundation, 2022. URL https://arxiv.org/abs/2202.07292.

[46] K. Främling. Explainable ai without interpretable model, 2020. URL https://arxiv.org/abs/2009.13996.

[47] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3):647–665, dec 2014. ISSN 0219-1377. doi: 10.1007/s10115-013-0679-x. URL https://doi.org/10.1007/s10115-013-0679-x.

[48] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Stat.*, 29(5):1189–1232, Oct. 2001.

[49] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.*, 24(1):44–65, Jan. 2015.

[50] B. M. Greenwell, B. C. Boehmke, and A. J. McCarthy. A simple and effective model-based variable importance measure, 2018. URL https://arxiv.org/abs/1805.04755.

[51] L. Breiman. *Mach. Learn.*, 45(1):5–32, 2001.

[52] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 04 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq134. URL https://doi.org/10.1093/bioinformatics/btq134.

[53] A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. 2018. doi: 10.48550/ARXIV.1801.01489. URL https://arxiv.org/abs/1801.01489.

[54] M. N. Katehakis and A. F. Veinott, Jr. The multi-armed bandit problem: Decomposition and computation. *Math. Oper. Res.*, 12(2):262–268, May 1987.

[55] C. Meske, E. Bunde, J. Schneider, and M. Gersch. Explainable artificial intelligence: Objectives, stakeholders, and future research opportunities. *Information Systems Management*, 39(1):53–63, 2022. doi: 10.1080/10580530.2020.1849465. URL https://doi.org/10.1080/10580530.2020.1849465.

[56] L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton, 2016. doi: doi:10.1515/9781400881970-018. URL https://doi.org/10.1515/9781400881970-018.

[57] E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, mar 2010. ISSN 1532-4435.

[58] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3):647–665, Dec. 2014.

[59] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

[60] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. Explainable ai for trees: From local explanations to global understanding, 2019. URL https://arxiv.org/abs/1905.04610.

[61] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. 2017. doi: 10.48550/ARXIV.1711.00399. URL https://arxiv.org/abs/1711.00399.

[62] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

[63] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021. ISSN 1099-4300. doi: 10.3390/e23010018. URL https://www.mdpi.com/1099-4300/23/1/18.

[64] A. Van Looveren and J. Klaise. Interpretable counterfactual explanations guided by prototypes, 2019. URL https://arxiv.org/abs/1907.02584.

[65] S. Dandl, C. Molnar, M. Binder, and B. Bischl. Multi-objective counterfactual explanations. In *Parallel Problem Solving from Nature – PPSN XVI*, pages 448–469. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58112-1_31. URL https://doi.org/10.1007%2F978-3-030-58112-1_31.

[66] K. Främling et al. Comparison of contextual importance and utility with lime and shapley values. In D. Calvaresi et al., editors, *Explainable and Transparent AI and Multi-Agent Systems - 3rd International Workshop, EXTRAAMAS 2021, Revised Selected Papers*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 39–54. Springer, 2021. ISBN 9783030820169. doi: 10.1007/978-3-030-82017-6_3.

[67] K. Främling. Contextual importance and utility in r: The 'ciu' package. In *AAA-21 Workshop, The Thirty-Fifth AAAI Conference on Artificial Intelligence, Virtual, February 8-9, 2021*, pages 110–114, 2021.

[68] S. Mohseni, N. Zarei, and E. D. Ragan. A multidisciplinary survey and framework for design and evaluation of explainable ai systems. 2018. doi: 10.48550/ARXIV.1811.11839. URL https://arxiv.org/abs/1811.11839.

[69] G. Vilone and L. Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.05.009. URL https://www.sciencedirect.com/science/article/pii/S1566253521001093.

[70] N. Burkart and M. F. Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, jan 2021. doi: 10.1613/jair.1.12228. URL https://doi.org/10.1613\%2Fjair.1.12228.

[71] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8), 2019. ISSN 2079-9292. doi: 10.3390/electronics8080832. URL https://www.mdpi.com/2079-9292/8/8/832.

[72] M. Nauta et al. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. 2022. doi: 10.48550/ARXIV.2201.08164. URL https://arxiv.org/abs/2201.08164.

[73] M. Robnik-Šikonja and M. Bohanec. Perturbation-based explanations of prediction models. In *Human and Machine Learning*, pages 159–175. Springer International Publishing, Cham, 2018.

[74] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *Commun. ACM*, 63(1):68–77, dec 2019. ISSN 0001-0782. doi: 10.1145/3359786. URL https://doi.org/10.1145/3359786.

[75] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Muller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, mar 2021. doi: 10.1109/jproc.2021.3060483. URL https://doi.org/10.1109%2Fjproc.2021.3060483.

[76] A. Binder, W. Samek, G. Montavon, S. Bach, and K.-R. Müller. Analyzing and validating neural networks predictions. 2016.

[77] A. Das and P. Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey. 2020. doi: 10.48550/ARXIV.2006.11371. URL https://arxiv.org/abs/2006.11371.

[78] S. Bach et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, 10(7):e0130140, July 2015.

[79] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek. The lrp toolbox for artificial neural networks. *Journal of Machine Learning Research*, 17(114):1–5, 2016. URL http://jmlr.org/papers/v17/15-618.html.

[80] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. 2018. doi: 10.48550/ARXIV.1806.00069. URL https://arxiv.org/abs/1806.00069.

[81] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps, 2018. URL https://arxiv.org/abs/1810.03292.

[82] V. Arya, R. K. E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, et al. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques, sept 2019. URL https://arxiv.org/abs/1909.03012.

[83] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks, 2018. URL https://arxiv.org/abs/1806.10758.

[84] K. N. Ramamurthy, B. Vinzamuri, Y. Zhang, and A. Dhurandhar. Model agnostic multilevel explanations. 2020. doi: 10.48550/ARXIV.2003.06005. URL https://arxiv.org/abs/2003.06005.

[85] fedesoriano. Heart failure prediction dataset. https://www.kaggle.com/fedesoriano/heart-failure-prediction., September 2021. Accessed: 2022-10-26.

[86] L. K. Hermann et al. Comparison of frequency of inducible myocardial ischemia in patients presenting to emergency department with typical versus atypical or nonanginal chest pain. *Am. J. Cardiol.*, 105(11):1561–1564, June 2010.

[87] R. Fass and S. R. Achem. Noncardiac chest pain: epidemiology, natural course and pathogenesis. *J. Neurogastroenterol. Motil.*, 17(2):110–123, Apr. 2011.

[88] J. H. medicine. High blood pressure/hypertension. https://www.hopkinsmedicine.org/health/conditions-and-diseases/high-blood-pressure-hypertension, 2022.

[89] M. N. Today. What is serum cholesterol? https://www.medicalnewstoday.com/articles/321519, September 2021.

[90] C. for Disease Control and Prevention. Diabetes tests. https://www.cdc.gov/diabetes/basics/getting-tested.html, August 2021.

[91] D. Rawshani. The st segment: phisiology, normal appearance, st depression & st elevation. https://ecgwaves.com/st-segment-normal-abnormal-depression-elevation-causes/, August 2021.

[92] A. H. Association. What is left ventricular hypertrophy (lvh)? https://www.heart.org/en/health-topics/heart-valve-problems-and-disease/heart-valve-problems-and-causes/what-is-left-ventricular-hypertrophy-lvh, August 2020.

[93] C. for Disease Control and Prevention. Target heart rate and estimated maximum heart rate. https://www.cdc.gov/physicalactivity/basics/measuring/heartrate.htm, June 2022.

[94] I. S. Statistics. Generalized linear models. https://www.ibm.com/docs/en/spss-statistics/27.0.0?topic=statistics-generalized-linear-models, February 2021.

[95] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995.

[96] R. Gandhi. Support vector machine — introduction to machine learning algorithms. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, June 2018. Accessed: 2022-09-26.

[97] D. Slack et al. Fooling lime and shap: Adversarial attacks on post hoc explanation methods, 2019. URL https://arxiv.org/abs/1911.02508.

[98] A. Bibal and B. Frénay. Interpretability of machine learning models and representations: an introduction. In M. Verleysen, editor, *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 77–82. CIACO, 2016. ISBN 978-2-87587-026-1. 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2016 ; Conference date: 27-04-2016 Through 29-05-2016.

[99] Z. C. Lipton. The mythos of model interpretability, 2016. URL https://arxiv.org/abs/1606.03490.

[100] W. Samek, T. Wiegand, and K.-R. Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models, 2017. URL https://arxiv.org/abs/1708.08296.

[101] F. K. Došilović, M. Brčić, and N. Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215, 2018. doi: 10.23919/MIPRO.2018.8400040.

[102] D. S. Weld and G. Bansal. The challenge of crafting intelligible intelligence, 2018. URL https://arxiv.org/abs/1803.04263.

[103] W. Samek and K.-R. Müller. *Towards Explainable Artificial Intelligence*, pages 5–22. Springer International Publishing, Cham, 2019. ISBN 978-3-030-28954-6. doi: 10.1007/978-3-030-28954-6_1. URL https://doi.org/10.1007/978-3-030-28954-6_1.

[104] S. R. Islam, W. Eberle, S. K. Ghafoor, and M. Ahmed. Explainable artificial intelligence approaches: A survey. 2021.

[105] T. Miller, P. Howe, and L. Sonenberg. Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences, 2017. URL https://arxiv.org/abs/1712.00547.

[106] F. Doshi-Velez and B. Kim. *Considerations for Evaluation and Generalization in Interpretable Machine Learning*, pages 3–17. Springer International Publishing, Cham, 2018. ISBN 978-3-319-98131-4. doi: 10.1007/978-3-319-98131-4_1. URL https://doi.org/10.1007/978-3-319-98131-4_1.

[107] M. Krishnan. Against interpretability: A critical examination of the interpretability problem in machine learning. *Philos. Technol.*, 33(3):487–502, Sept. 2020.

[108] F. Emmert-Streib, O. Yli-Harja, and M. Dehmer. Explainable artificial intelligence and machine learning: A reality rooted perspective, 2020. URL https://arxiv.org/abs/2001.09464.

[109] F. Hussain, R. Hussain, and E. Hossain. Explainable artificial intelligence (xai): An engineering perspective, 2021. URL https://arxiv.org/abs/2101.03613.

[110] A. G. F. Hoepner, D. McMillan, A. Vivian, and C. W. Simen. Significance, relevance and explainability in the machine learning age: an econometrics and financial data science perspective. *The European Journal of Finance*, 27(1-2):1–7, 2021. doi: 10.1080/1351847X.2020.1847725. URL https://doi.org/10.1080/1351847X.2020.1847725.

[111] P.-J. Kindermans et al. The (un)reliability of saliency methods, 2017. URL https://arxiv.org/abs/1711.00867.

[112] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks, 2017. URL https://arxiv.org/abs/1711.06104.

[113] J. Adebayo, J. Gilmer, I. Goodfellow, and B. Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values, 2018. URL https://arxiv.org/abs/1810.03307.

[114] D. Alvarez-Melis and T. S. Jaakkola. On the robustness of interpretability methods, 2018. URL https://arxiv.org/abs/1806.08049.

[115] A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile, 2017. URL https://arxiv.org/abs/1710.10547.

[116] W. Samek, A. Binder, G. Montavon, S. Bach, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned, 2015. URL https://arxiv.org/abs/1509.06321.

[117] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Interpretable & explorable approximations of black box models, 2017. URL https://arxiv.org/abs/1707.01154.

[118] A.-p. Nguyen and M. R. Martínez. On quantitative aspects of model interpretability, 2020. URL https://arxiv.org/abs/2007.07584.

[119] C.-K. Yeh, C.-Y. Hsieh, A. S. Suggala, D. I. Inouye, and P. Ravikumar. On the (in)fidelity and sensitivity for explanations, 2019. URL https://arxiv.org/abs/1901.09392.

[120] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks, 2017. URL https://arxiv.org/abs/1703.01365.

[121] P. Schmidt and F. Biessmann. Quantifying interpretability and trust in machine learning systems, 2019. URL https://arxiv.org/abs/1901.08558.

[122] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, feb 2018. doi: 10.1016/j.dsp.2017.10.011. URL https://doi.org/10.1016%2Fj.dsp.2017.10.011.

[123] R. ElShawi, Y. Sherif, M. Al-Mallah, and S. Sakr. Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. *Comput. Intell.*, 37(4):1633–1650, Nov. 2021.

[124] D. Alvarez-Melis and T. S. Jaakkola. Towards robust interpretability with self-explaining neural networks, 2018. URL https://arxiv.org/abs/1806.07538.

[125] S. Mohseni, J. E. Block, and E. D. Ragan. A human-grounded evaluation benchmark for local explanations of machine learning, 2018. URL https://arxiv.org/abs/1801.05075.

[126] X. Cui, J. M. Lee, and J. P.-A. Hsieh. An integrative 3C evaluation framework for explainable artificial intelligence. In *AMCIS 2019 Proceedings*, 2019.

[127] K. Fauvel et al. A performance-explainability framework to benchmark machine learning methods: Application to multivariate time series classifiers, 2020. URL https://arxiv.org/abs/2005.14501.

[128] R. Tomsett, D. Braines, D. Harborne, A. Preece, and S. Chakraborty. Interpretable to whom? a role-based model for analyzing interpretable machine learning systems, 2018. URL https://arxiv.org/abs/1806.07552.

[129] M. Langer et al. What do we want from explainable artificial intelligence (xai)? – a stakeholder perspective on xai and a conceptual model guiding interdisciplinary xai research. *Artificial Intelligence*, 296:103473, 2021. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2021.103473. URL https://www.sciencedirect.com/science/article/pii/S0004370221000242.

[130] C. Zednik. Solving the black box problem: A normative framework for explainable artificial intelligence. *Philos. Technol.*, 34(2):265–288, June 2021.

[131] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. doi: 10.1073/pnas.1900654116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1900654116.

[132] S. R. Islam, W. Eberle, and S. K. Ghafoor. Towards quantification of explainability in explainable artificial intelligence methods. Nov. 2019.

[133] A. Rosenfeld. Better metrics for evaluating explainable artificial intelligence. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, page 45–50, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

[134] M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation, 2018. URL https://arxiv.org/abs/1802.00682.

[135] I. Lage, E. Chen, J. He, M. Narayanan, B. Kim, S. Gershman, and F. Doshi-Velez. An evaluation of the human-interpretability of explanation, 2019. URL https://arxiv.org/abs/1902.00006.

[136] D. Slack, S. A. Friedler, C. Scheidegger, and C. D. Roy. Assessing the local interpretability of machine learning models, 2019. URL https://arxiv.org/abs/1902.03501.

[137] V. Putnam and C. Conati. Exploring the need for explainable artificial intelligence (xai) in intelligent tutoring systems (its). In *IUI Workshops*, 2019.

[138] W. K. Diprose, N. Buist, N. Hua, Q. Thurier, G. Shand, and R. Robinson. Physician understanding, explainability, and trust in a hypothetical machine learning risk calculator. *J. Am. Med. Inform. Assoc.*, 27(4):592–600, Apr. 2020.

[139] F. Poursabzi-Sangdeh et al. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. URL https://doi.org/10.1145/3411764.3445315.

[140] G. J. Katuwal and R. Chen. Machine learning model interpretability for precision medicine, 2016. URL https://arxiv.org/abs/1610.09045.

[141] S. G. Rizzo, G. Vantini, and S. Chawla. Reinforcement learning with explainability for traffic signal control. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3567–3572, 2019. doi: 10.1109/ITSC.2019.8917519.

[142] P. Bracke, A. Datta, C. Jung, and S. Sen. Machine learning explainability in finance: An application to default risk analysis. *SSRN Electron. J.*, 2019.

[143] R. Elshawi, M. H. Al-Mallah, and S. Sakr. On the interpretability of machine learning-based model for predicting hypertension. *BMC Med. Inform. Decis. Mak.*, 19(1):146, July 2019.

[144] S. M. Lauritsen et al. Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nat. Commun.*, 11(1):3852, July 2020.

[145] E. Zihni, V. I. Madai, M. Livne, I. Galinovic, A. A. Khalil, J. B. Fiebach, and D. Frey. Opening the black box of artificial intelligence for clinical decision support: A study predicting stroke outcome. *PLoS One*, 15(4): e0231166, Apr. 2020.

[146] M. J. Ariza-Garzón, J. Arroyo, A. Caparrini, and M.-J. Segovia-Vargas. Explainability of a machine learning granting scoring model in peer-to-peer lending. *IEEE Access*, 8:64873–64890, 2020. doi: 10.1109/ACCESS.2020.2984412.

[147] H. Amini and L. Kosseim. Towards explainability in using deep learning for the detection of anorexia in social media. In *Natural Language Processing and Information Systems*, Lecture notes in computer science, pages 225–235. Springer International Publishing, Cham, 2020.

[148] L. Kurasinski and R.-C. Mihailescu. Towards machine learning explainability in text classification for fake news detection. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 775–781, 2020. doi: 10.1109/ICMLA51294.2020.00127.

[149] D.-S. Kim and S. Shin. The economic explainability of machine learning and standard econometric models-an application to the u.s. mortgage default risk. *Int. J. Strateg. Prop. Manage.*, 25(5):396–412, July 2021.

[150] T. Hepp et al. Uncertainty estimation and explainability in deep learning-based age estimation of the human brain: Results from the german national cohort mri study. *Computerized Medical Imaging and Graphics*, 92: 101967, 2021. ISSN 0895-6111. doi: https://doi.org/10.1016/j.compmedimag.2021.101967. URL https://www.sciencedirect.com/science/article/pii/S0895611121001166.

[151] J. R. Rico-Juan and P. Taltavull de La Paz. Machine learning with explainability or spatial hedonics tools? an analysis of the asking prices in the housing market in alicante, spain. *Expert Systems with Applications*, 171:114590, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.114590. URL https://www.sciencedirect.com/science/article/pii/S0957417421000312.

[152] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013. URL https://arxiv.org/abs/1312.6034.

[153] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks, 2017. URL https://arxiv.org/abs/1703.01365.

[154] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. 2017. doi: 10.48550/ARXIV.1704.02685. URL https://arxiv.org/abs/1704.02685.

[155] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions, 2017. URL https://arxiv.org/abs/1703.04730.

[156] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise, 2017. URL https://arxiv.org/abs/1706.03825.

[157] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. 11:1803–1831, aug 2010. ISSN 1532-4435.

[158] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks, 2013. URL https://arxiv.org/abs/1311.2901.

[159] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne. Learning how to explain neural networks: Patternnet and patternattribution, 2017. URL https://arxiv.org/abs/1705.05598.

[160] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net, 2014. URL https://arxiv.org/abs/1412.6806.

[161] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017. doi: 10.1109/iccv.2017.371. URL https://doi.org/10.1109%2Ficcv.2017.371.

[162] G. Montavon et al. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, may 2017. doi: 10.1016/j.patcog.2016.11.008. URL https://doi.org/10.1016%2Fj.patcog.2016.11.008.

[163] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2015. URL https://arxiv.org/abs/1502.03044.

[164] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). 2017. doi: 10.48550/ARXIV.1711.11279. URL https://arxiv.org/abs/1711.11279.

[165] D. Erhan, A. C. Courville, and Y. Bengio. Understanding representations learned in deep architectures. 2010.

[166] P. Cortez and M. J. Embrechts. Opening black box data mining models using sensitivity analysis. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 341–348, 2011. doi: 10.1109/CIDM.2011.5949423.

[167] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, 2016. URL https://arxiv.org/abs/1605.09304.

[168] T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions, 2016. URL https://arxiv.org/abs/1606.04155.

[169] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis, 2017. URL https://arxiv.org/abs/1702.04595.

[170] S. Tan, M. Soloviev, G. Hooker, and M. T. Wells. Tree space prototypes: Another look at making tree ensembles interpretable, 2016. URL https://arxiv.org/abs/1611.07115.

[171] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized input sampling for explanation of black-box models, 2018. URL https://arxiv.org/abs/1806.07421.

[172] A. Van Looveren and J. Klaise. Interpretable counterfactual explanations guided by prototypes, 2019. URL https://arxiv.org/abs/1907.02584.

[173] S. Lapuschkin et al. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1), mar 2019. doi: 10.1038/s41467-019-08987-4. URL https://doi.org/10.1038%2Fs41467-019-08987-4.

[174] R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. Hinton. Neural additive models: Interpretable machine learning with neural nets, 2020. URL https://arxiv.org/abs/2004.13912.

[175] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. Local rule-based explanations of black box decision systems, 2018. URL https://arxiv.org/abs/1805.10820.

[176] Y. Ming, H. Qu, and E. Bertini. Rulematrix: Visualizing and understanding classifiers with rules, 2018. URL https://arxiv.org/abs/1807.06228.

[177] Y. Zhou and G. Hooker. Interpreting models via single tree approximation, 2016. URL https://arxiv.org/abs/1610.09036.

[178] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives, 2018. URL https://arxiv.org/abs/1802.07623.

[179] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM, jan 2020. doi: 10.1145/3351095.3372850. URL https://doi.org/10.1145%2F3351095.3372850.

[180] A. Chattopadhay, A. Sarkar, P. Howlader, and V. 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, 2018. doi: 10.1109/WACV.2018.00097.

[181] A. Ghorbani, J. Wexler, J. Zou, and B. Kim. Towards automatic concept-based explanations, 2019. URL https://arxiv.org/abs/1902.03129.

[182] Y. Goyal, A. Feder, U. Shalit, and B. Kim. Explaining classifiers with causal concept effect (cace), 2019. URL https://arxiv.org/abs/1907.07165.

[183] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation, 2018. URL https://arxiv.org/abs/1802.07814.

[184] G. Plumb, D. Molitor, and A. Talwalkar. Model agnostic supervised local explanations, 2018. URL https://arxiv.org/abs/1807.02910.

[185] C.-K. Yeh, B. Kim, S. O. Arik, C.-L. Li, T. Pfister, and P. Ravikumar. On completeness-aware concept-based explanations in deep neural networks, 2019. URL https://arxiv.org/abs/1910.07969.

[186] J. R. Zilke, E. Loza Mencía, and F. Janssen. Deepred – rule extraction from deep neural networks. In T. Calders, M. Ceci, and D. Malerba, editors, *Discovery Science*, pages 457–473, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46307-0.

[187] N. Frosst and G. Hinton. Distilling a neural network into a soft decision tree, 2017. URL https://arxiv.org/abs/1711.09784.

[188] G. Montavon et al. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham, 2019. ISBN 978-3-030-28954-6. doi: 10.1007/978-3-030-28954-6_10. URL https://doi.org/10.1007/978-3-030-28954-6_10.

[189] O. Eberle, J. Buttner, F. Krautli, K.-R. Muller, M. Valleriani, and G. Montavon. Building and interpreting deep similarity models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1149–1161, mar 2022. doi: 10.1109/tpami.2020.3020738. URL https://doi.org/10.1109%2Ftpami.2020.3020738.

[190] M. Staniak and P. Biecek. Explanations of model predictions with live and breakDown packages. *The R Journal*, 10(2):395, 2019. doi: 10.32614/rj-2018-072. URL https://doi.org/10.32614%2Frj-2018-072.

[191] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. D. Bie, and P. Flach. FACE. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. ACM, feb 2020. doi: 10.1145/3375627.3375850. URL https://doi.org/10.1145%2F3375627.3375850.

[192] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization, 2015. URL https://arxiv.org/abs/1506.06579.

[193] R. Guidotti, A. Monreale, S. Matwin, and D. Pedreschi. Explaining image classifiers generating exemplars and counter-exemplars from latent representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13665–13668, Apr. 2020. doi: 10.1609/aaai.v34i09.7116. URL https://ojs.aaai.org/index.php/AAAI/article/view/7116.

[194] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure, 2016. URL https://arxiv.org/abs/1612.08220.

[195] M. Du, N. Liu, Q. Song, and X. Hu. Towards explanation of dnn-based prediction with guided feature inversion, 2018. URL https://arxiv.org/abs/1804.00506.

[196] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 131–138, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450363242. doi: 10.1145/3306618.3314229. URL https://doi.org/10.1145/3306618.3314229.

[197] S. Krishnan and E. Wu. Palm: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350297. doi: 10.1145/3077257.3077271. URL https://doi.org/10.1145/3077257.3077271.

[198] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, and K. N. Ramamurthy. Treeview: Peeking into deep neural networks via feature-space partitioning, 2016. URL https://arxiv.org/abs/1611.07429.

[199] P. Adler et al. Auditing black-box models for indirect influence, 2016. URL https://arxiv.org/abs/1602.07043.

[200] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu. Interpreting cnns via decision trees, 2018. URL https://arxiv.org/abs/1802.00121.

[201] M. G. Augasta and T. Kathirvalavakumar. Reverse engineering the neural networks for rule extraction in classification problems. *Neural Process. Lett.*, 35(2):131–150, apr 2012. ISSN 1370-4621. doi: 10.1007/s11063-011-9207-8. URL https://doi.org/10.1007/s11063-011-9207-8.

[202] D. W. Apley and J. Zhu. Visualizing the effects of predictor variables in black box supervised learning models, 2016. URL https://arxiv.org/abs/1612.08468.

[203] A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou. A peek into the black box: exploring classifiers by randomization. *Data Min. Knowl. Discov.*, 28(5-6):1503–1529, Sept. 2014.

[204] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks, 2016. URL https://arxiv.org/abs/1602.03616.

[205] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Gan dissection: Visualizing and understanding generative adversarial networks, 2018. URL https://arxiv.org/abs/1811.10597.

[206] G. Casalicchio, C. Molnar, and B. Bischl. Visualizing the feature importance for black box models. In *Machine Learning and Knowledge Discovery in Databases*, pages 655–670. Springer International Publishing, 2019. doi: 10.1007/978-3-030-10925-7_40. URL https://doi.org/10.1007%2F978-3-030-10925-7_40.

[207] H. Li, Y. Tian, K. Mueller, and X. Chen. Beyond saliency: Understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation. *Image and Vision Computing*, 83-84:70–86, mar 2019. doi: 10.1016/j.imavis.2019.02.005. URL https://doi.org/10.1016%2Fj.imavis.2019.02.005.

[208] M. R. Zafar and N. M. Khan. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems, 2019. URL https://arxiv.org/abs/1906.10263.

[209] I. Mollas, N. Bassiliades, and G. Tsoumakas. LioNets: Local interpretation of neural networks through penultimate layer decoding. In *Machine Learning and Knowledge Discovery in Databases*, pages 265–276. Springer International Publishing, 2020. doi: 10.1007/978-3-030-43823-4_23. URL https://doi.org/10.1007%2F978-3-030-43823-4_23.

[210] M. Setzu, R. Guidotti, et al. Global explanations with local scoring. In *Machine Learning and Knowledge Discovery in Databases*, pages 159–171, Cham, 2020. Springer International Publishing. ISBN 978-3-030-43823-4.

[211] E. Albini, A. Rago, P. Baroni, and F. Toni. Relation-based counterfactual explanations for bayesian network classifiers. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 451–457. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/63. URL https://doi.org/10.24963/ijcai.2020/63. Main track.

[212] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry. Xrai: Better attributions through regions, 2019. URL https://arxiv.org/abs/1906.02825.

[213] G. Zhao, B. Zhou, K. Wang, R. Jiang, and M. Xu. Respond-cam: Analyzing deep models for 3d imaging data by visualizations. 2018. doi: 10.48550/ARXIV.1806.00102. URL https://arxiv.org/abs/1806.00102.

[214] O. Lampridis, R. Guidotti, and S. Ruggieri. Explaining sentiment classification with synthetic exemplars and counter-exemplars. In *Discovery Science*, Lecture notes in computer science, pages 357–373. Springer International Publishing, Cham, 2020.

[215] B. Hoover, H. Strobelt, and S. Gehrmann. exbert: A visual analysis tool to explore learned representations in transformers models, 2019. URL https://arxiv.org/abs/1910.05276.

[216] A. Jacovi, O. S. Shalom, and Y. Goldberg. Understanding convolutional neural networks for text classification, 2018. URL https://arxiv.org/abs/1809.08037.

[217] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao. Weakly supervised instance segmentation using class peak response, 2018. URL https://arxiv.org/abs/1804.00880.

[218] Y. Qin, K. Kamnitsas, S. Ancha, J. Nanavati, G. Cottrell, A. Criminisi, and A. Nori. Autofocus layer for semantic segmentation, 2018. URL https://arxiv.org/abs/1805.08403.

[219] D. Kumar, A. Wong, and G. W. Taylor. Explaining the unexplained: A class-enhanced attentive response (clear) approach to understanding deep neural networks, 2017. URL https://arxiv.org/abs/1704.04133.

[220] F. Mayr et al. Regular inference on artificial neural networks. In *Machine Learning and Knowledge Extraction*, pages 350–369, Cham, 2018. Springer International Publishing. ISBN 978-3-319-99740-7.

[221] A. Blanco-Justicia, J. Domingo-Ferrer, S. Martínez, and D. Sánchez. Machine learning explainability via microaggregation and shallow decision trees. *Knowledge-Based Systems*, 194:105532, 2020. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2020.105532. URL https://www.sciencedirect.com/science/article/pii/S0950705120300368.

[222] S. Hara and K. Hayashi. Making tree ensembles interpretable, 2016. URL https://arxiv.org/abs/1606.05390.

[223] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[224] S. Tan, R. Caruana, G. Hooker, and Y. Lou. Distill-and-compare. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, dec 2018. doi: 10.1145/3278721.3278725. URL https://doi.org/10.1145%2F3278721.3278725.

[225] O. Bastani, C. Kim, and H. Bastani. Interpretability via model extraction, 2017. URL https://arxiv.org/abs/1706.09773.

[226] Q. Zhang, R. Cao, F. Shi, Y. N. Wu, and S.-C. Zhu. Interpreting cnn knowledge via an explanatory graph, 2017. URL https://arxiv.org/abs/1708.01785.

[227] L. Chu, X. Hu, J. Hu, L. Wang, and J. Pei. Exact and consistent interpretation for piecewise linear neural networks: A closed form solution, 2018. URL https://arxiv.org/abs/1802.06259.

[228] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes, 2016. URL https://arxiv.org/abs/1610.01644.

[229] J. Oramas, K. Wang, and T. Tuytelaars. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks, 2017. URL https://arxiv.org/abs/1712.06302.

[230] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers, 2017. URL https://arxiv.org/abs/1705.07857.

[231] J. Li, X. Chen, E. Hovy, and D. Jurafsky. Visualizing and understanding neural models in nlp, 2015. URL https://arxiv.org/abs/1506.01066.

[232] J. Adebayo and L. Kagal. Iterative orthogonal feature projection for diagnosing bias in black-box models, 2016. URL https://arxiv.org/abs/1611.04967.

[233] A. Bondarenko, L. Aleksejeva, V. Jumutc, and A. Borisov. Classification tree extraction from trained artificial neural networks. *Procedia Computer Science*, 104:556–563, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.01.172. URL https://www.sciencedirect.com/science/article/pii/S1877050917301734. ICTE 2016, Riga Technical University, Latvia.

[234] W. J. Murdoch and A. Szlam. Automatic rule extraction from long short term memory networks, 2017. URL https://arxiv.org/abs/1702.02540.

[235] S. K. Biswas et al. Rule extraction from training data using neural network. *International Journal on Artificial Intelligence Tools*, 26(03):1750006, 2017. doi: 10.1142/S0218213017500063. URL https://doi.org/10.1142/S0218213017500063.

[236] H. Deng. Interpreting tree ensembles with intrees, 2014. URL https://arxiv.org/abs/1408.5456.

[237] R. Turner. A model explanation system. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2016. doi: 10.1109/MLSP.2016.7738872.

[238] C. Burns, J. Thomason, and W. Tansey. Interpreting black box models via hypothesis testing. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*. ACM, oct 2020. doi: 10.1145/3412815.3416889. URL https://doi.org/10.1145%2F3412815.3416889.

[239] M. Ibrahim, M. Louie, C. Modarres, and J. Paisley. Global explanations of neural networks: Mapping the landscape of predictions, 2019. URL https://arxiv.org/abs/1902.02384.

[240] A. Henelius, K. Puolamäki, and A. Ukkonen. Interpreting classifiers through attribute interactions in datasets, 2017. URL https://arxiv.org/abs/1707.07576.

[241] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3):647–665, Dec. 2014.

[242] J. T. Sliwinski, M. Strobel, and Y. Zick. A characterization of monotone influence measures for data classification. *ArXiv*, abs/1708.02153, 2017.

[243] A. Datta, S. Sen, and Y. 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, 2016. doi: 10.1109/SP.2016.42.

[244] P. Tamagnini, J. Krause, A. Dasgupta, and E. Bertini. Interpreting black-box classifiers using instance-level visual explanations. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350297. doi: 10.1145/3077257.3077260. URL https://doi.org/10.1145/3077257.3077260.

[245] L. Liu and L. Wang. What has my classifier learned? visualizing the classification rules of bag-of-feature model by support region detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3586–3593, 2012. doi: 10.1109/CVPR.2012.6248103.

[246] G. Tolomei, F. Silvestri, A. Haines, and M. Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2017. doi: 10.1145/3097983.3098039. URL https://doi.org/10.1145%2F3097983.3098039.

[247] R. Khanna, B. Kim, J. Ghosh, and O. Koyejo. Interpreting black box predictions using fisher kernels, 2018. URL https://arxiv.org/abs/1810.10118.

[248] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples, 2014. URL https://arxiv.org/abs/1412.6572.

[249] S. C. Yang. Explainable artificial intelligence via bayesian teaching. 2017.

[250] S. Liu, B. Kailkhura, D. Loveland, and Y. Han. Generative counterfactual introspection for explainable deep learning, 2019. URL https://arxiv.org/abs/1907.03077.

[251] J. Moore, N. Hammerla, and C. Watkins. Explaining deep learning models with constrained adversarial examples, 2019. URL https://arxiv.org/abs/1906.10671.

[252] S. Rathi. Generating counterfactual and contrastive explanations using shap, 2019. URL https://arxiv.org/abs/1906.09293.

[253] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata. Generating counterfactual explanations with natural language, 2018. URL https://arxiv.org/abs/1806.09809.

[254] S. Barratt. Interpnet: Neural introspection for interpretable deep learning, 2017. URL https://arxiv.org/abs/1710.09511.

[255] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations, 2016. URL https://arxiv.org/abs/1603.08507.

[256] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations, 2017. URL https://arxiv.org/abs/1704.05796.

[257] B. J. Lengerich, S. Konam, E. P. Xing, S. Rosenthal, and M. Veloso. Towards visual explanations for convolutional neural networks via input resampling, 2017. URL https://arxiv.org/abs/1707.09641.

[258] D. Alvarez-Melis and T. S. Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models, 2017. URL https://arxiv.org/abs/1707.01943.

[259] X. Liu, X. Wang, and S. Matwin. Interpretable deep convolutional neural networks via meta-learning, 2018. URL https://arxiv.org/abs/1802.00560.

[260] P. E. Rauber, S. G. Fadel, A. X. Falcão, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, 2017. doi: 10.1109/TVCG.2016.2598838.

[261] T. Narendra, A. Sankaran, D. Vijaykeerthy, and S. Mani. Explaining deep learning models using causal inference, 2018. URL https://arxiv.org/abs/1811.04376.

[262] M. Harradon, J. Druce, and B. Ruttenberg. Causal learning and explanation of deep neural networks via autoencoded activations, 2018. URL https://arxiv.org/abs/1802.00541.

[263] A. Chattopadhyay et al. Neural network attributions: A causal perspective. 2019. doi: 10.48550/ARXIV.1902.02302. URL https://arxiv.org/abs/1902.02302.

[264] R. Guidotti et al. Explaining any time series classifier. In *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*, pages 167–176, 2020. doi: 10.1109/CogMI50398.2020.00029.

[265] C. Panigutti, A. Perotti, and D. Pedreschi. Doctor xai: An ontology-based approach to black-box sequential data classification explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 629–639, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3372855. URL https://doi.org/10.1145/3351095.3372855.

[266] A. Kanehira, K. Takemoto, S. Inayoshi, and T. Harada. Multimodal explanations by predicting counterfactuality in videos, 2018. URL https://arxiv.org/abs/1812.01263.

Appendix A

Literature Review

Table A.1: XAI SoTa papers distribution.

Type of paper	References
Book	[6, 44]
Survey/Review	[1, 5, 8–12, 33, 36, 63, 68–72, 74, 75, 77, 80, 98–104]
Discussion/Theory	[21–27, 31, 32, 34, 37, 43, 55, 105–110]
Evaluation/Comparison study	[33, 75, 76, 81, 83, 100, 111–124]
Framework/Approach	[35, 68, 106, 125–133]
User study	[2, 134–139]
Case/Use study	[3, 4, 25, 28–30, 109, 140–151]

Table A.2: Examples of XAI case/use studies in specific application domains.

Application Domain	Case/Use studies
Medicine	[3, 4, 140, 143–145, 147, 150]
Transportation	[25, 109, 141]
Finance	[142, 146, 149, 151]
Security	[148]
Legal	[25]
Military	[20]

Table A.3: Post-hoc XAI methods identified in the SoTa literature.

Name	Reference	Year	% surveys	Portability	Scope	Data	Problem	Software
LIME	[40]	2016	80	A	L	ANY	Both	Y
Shapley values	[58]	2014	69	A	L	TAB	Both	Y
SHAP	[59]	2017	69	A	L	ANY	Both	Y
LRP	[78]	2015	53	S	L	IMG/TXT	C	Y
Saliency Maps	[152]	2013	53	S	L	IMG/TXT	C	Y
Anchors	[41]	2018	46	A	L	TAB/TXT	C	Y
IntGrad	[153]	2017	46	S	L	IMG/TXT	C	Y
Grad-CAM	[38]	2019	46	S	L	IMG	C	Y
DeepLIFT	[154]	2017	42	S	L	IMG/TXT	C	Y
Influence Functions	[155]	2017	42	A	L	IMG	C	Y
SmoothGrad	[156]	2017	38	S	L	IMG	C	Y
Local gradients	[157]	2010	38	A	L	IMG/TAB	C	N

Table A.3: Post-hoc XAI methods identified in the SoTa literature.

Name	Reference	Year	% surveys	Portability	Scope	Data	Problem	Software
ICE	[59]	2015	34	A	G	TAB	Both	Y
DeconvNet	[158]	2013	34	S	L	IMG	C	Y
PatternNet	[159]	2017	34	S	L	IMG	C	Y
PatternAttribution	[159]	2017	34	S	L	IMG	C	Y
PDP	[48]	2001	30	A	G	TAB	Both	Y
Guided BackProp	[160]	2014	30	S	L	IMG	C	Y
Meaningful Perturbation	[161]	2017	30	S	L	IMG	C	Y
DTD	[162]	2017	30	S	L	IMG	C	Y
Show, Attend and Tell	[163]	2015	30	S	L	IMG	C	Y
TCAV	[164]	2017	30	A	G	IMG	C	Y
Activation Maximization	[165]	2010	26	S	L	IMG	C	Y
GSA	[166]	2011	23	A	G	TAB	Both	N
DGN	[167]	2016	23	S	G	IMG	C	N
Rationales	[168]	2016	23	S	L	TXT	C	Y
PDA	[169]	2017	19	S	L	IMG	C	Y
CAM	[39]	2015	19	S	L	IMG	C	Y
TSP	[170]	2016	15	S	L	TAB	C	N
RISE	[171]	2018	15	S	L	IMG	C	Y
CIU	[42]	2020	11	A	L/G	IMG/TAB	Both	Y
Guided Proto	[172]	2019	11	A	L	IMG/TAB	C	Y
SPRAY	[173]	2019	11	S	G	IMG	C	N
NAM	[174]	2020	11	S	L	TAB	Both	Y
LORE	[175]	2018	11	A	L	TAB	C	Y
RuleMatrix	[176]	2018	11	A	G	TAB	C	Y
STA	[177]	2016	11	S	G	TAB	C	N
CEM	[178]	2018	11	S	L	ANY	C	Y
DICE	[179]	2020	11	A	L	TAB	C	Y
Grad-CAM++	[180]	2018	11	S	L	IMG	C	Y
ACE	[181]	2019	11	A	G	IMG	C	Y
CaCE	[182]	2019	11	A	G	IMG	C	N
L2X	[183]	2018	11	A	L	IMG/TXT	C	Y
CFEs (original)	[61]	2017	11	A	L	IMG/TAB	C	N
PIMP	[52]	2010	11	A	G	TAB	C	Y
MAPLE	[184]	2018	11	A	L	TAB	Both	Y
ConceptSHAP	[185]	2020	11	A	G	IMG	C	Y
DeepRED	[186]	2016	11	S	L	IMG	C	N
Soft DT	[187]	2017	11	S	G	IMG	C	Y
LRP*	[188]	2019	7	S	L	ANY	C	N
BiLRP	[189]	2020	7	S	L	ANY	C	N
LIVE	[190]	2018	7	A	L	TAB	Both	Y
BreakDown	[190]	2018	7	A	L	TAB	Both	Y
FACE	[191]	2019	7	A	L	ANY	C	Y
Regularisation	[192]	2015	7	S	L	IMG	C	Y
ABELE	[193]	2020	7	A	L	IMG	C	Y
Erasure	[194]	2016	7	S	L	TXT	C	N

Table A.3: Post-hoc XAI methods identified in the SoTa literature.

Name	Reference	Year	% surveys	Portability	Scope	Data	Problem	Software
GFI	[195]	2018	7	S	L	IMG	C	N
MUSE	[196]	2019	7	A	L	IMG	C	N
PALM	[197]	2017	7	A	G	ANY	Both	N
TreeView	[198]	2016	7	S	G	IMG	C	N
GFA	[199]	2016	7	A	G	TAB	Both	Y
DT Extraction	[200]	2018	7	S	G	IMG	C	N
RxREN	[201]	2012	7	S	G	TAB	Both	N
ALE	[202]	2016	7	A	G	TAB	Both	Y
GoldenEye	[203]	2014	7	A	G	TAB	C	N
Multifaceted feature visualization	[204]	2016	7	S	L	IMG	C	N
GAN Dissection	[205]	2018	7	S	L	IMG	C	Y
PI, ICI	[206]	2018	3	A	G	TAB	Both	Y
SR map	[207]	2019	3	S	L	IMG	C	Y
DLIME	[208]	2019	3	A	L	ANY	Both	Y
LioNets	[209]	2019	3	S	L	TXT	C	Y
SkopeRules	N/A	2020	3	A	L/G	TAB	C	Y
GLocalX	[210]	2020	3	A	L/G	TAB	C	Y
CFX	[211]	2020	3	S	L	TAB	C	N
XRAI	[212]	2019	3	S	L	IMG	C	Y
Respond-CAM	[213]	2018	3	S	L	IMG	C	N
XSPELLS	[214]	2020	3	A	L	TXT	C	Y
exBERT	[215]	2019	3	S	L	TXT	C	Y
Slot Activation Vectors	[216]	2018	3	S	L	TXT	C	Y
Peak Response	[217]	2018	3	S	L	IMG	C	Y
Autofocus-Layer	[218]	2018	3	S	L	IMG	C	Y
CLEAR	[219]	2017	3	S	L	IMG	C	N
DFA	[220]	2018	3	S	L	TAB	C	N
Privacy-Preserving Explanations	[221]	2020	3	S	L	TAB	C	Y
No name	[222]	2016	3	S	G	TAB	Both	N
Distillation	[223]	2015	3	A	G	IMG	C	N
Distill-and-Compare	[224]	2018	3	A	G	TAB	Both	N
Model Extraction	[225]	2017	3	A	G	TAB	Both	N
Explanatory Graph	[226]	2017	3	S	G	IMG	C	N
OpenBox	[227]	2018	3	S	G	IMG	C	N
Probes	[228]	2016	3	S	G	IMG	C	N
Relevant Features	[229]	2017	3	S	L	IMG	C	N
Saliency Detection	[230]	2017	3	A	L	IMG	C	N
Compositionality	[231]	2016	3	S	L	TXT	C	N
OPIA	[232]	2016	3	A	G	TAB	Both	N
NNKX	[233]	2017	3	S	G	TAB	C	Y
Automatic Rule Extraction	[234]	2017	3	S	G	TXT	C	N
RxNCM	[235]	2017	3	S	G	TAB	C	N

Table A.3: Post-hoc XAI methods identified in the SoTa literature.

Name	Reference	Year	% surveys	Portability	Scope	Data	Problem	Software
inTrees	[236]	2014	3	S	G	TAB	Both	N
MES	[237]	2016	3	A	L	ANY	C	N
Hypothesis Testing	[238]	2019	3	A	L	IMG/TXT	C	Y
GAM	[239]	2019	3	A	G	IMG/TAB	C	Y
ASTRID	[240]	2017	3	A	L	TAB	C	N
Shapley Values-based	[57]	2010	3	A	L	TAB	C	N
SA-based	[241]	2014	3	A	L	TAB	Both	N
Monotone Influence	[242]	2017	3	A	L	IMG	C	N
QII	[243]	2016	3	A	G	TAB	C	Y
Rivelo	[244]	2017	3	A	L	TXT	C	N
RSRS Detection	[245]	2012	3	S	L	TXT	C	N
Feature Tweaking	[246]	2017	3	S	L	TAB	C	N
SQB	[247]	2018	3	A	L	IMG/TAB	C	N
Worst-case perturbations	[248]	2015	3	A	L	IMG	C	N
Bayesian Teaching	[249]	2017	3	A	L	TAB	Both	N
Counterfactual Introspection	[250]	2019	3	S	L	IMG	C	N
CADEX	[251]	2019	3	S	L	TAB	C	N
Counterfactual SHAP	[252]	2019	3	A	L	ANY	Both	N
Textual CFEs	[253]	2018	3	A	L	IMG	C	N
InterpNET	[254]	2017	3	A	L	IMG	C	Y
Discriminative Loss	[255]	2016	3	S	L	IMG	C	N
Network dissection	[256]	2017	3	S	L	IMG	C	N
Important Neurons and Patches	[257]	2017	3	S	G	IMG	C	Y
Pertubation-based	[258]	2017	3	A	L	TXT	C	N
CNN-INTE	[259]	2018	3	S	G	IMG	C	N
Hidden Activity Visualization	[260]	2017	3	S	G	IMG	C	N
Causal Inference	[261]	2018	3	S	G	IMG	C	N
Autoencoded Activations	[262]	2018	3	S	G	IMG	C	N
Causal Effects	[263]	2019	3	S	G	ANY	C	Y
LASTS	[264]	2020	3	A	L	TS	C	No
DoctorXAI	[265]	2020	3	A	L	TS	C	Y
Multimodal Information	[266]	2018	3	S	G	VID	C	No

Appendix B

Results

Table B.1: Local (for patient A) importance, utility and influence values given by different XAI methods for all implemented models.

LR	LIME	Shapley	kernelSHAP	CI	CU	CInfl
Age	0.03	0.02	0.02	0.21	0.75	0.05
Sex	0.30	0.07	0.07	0.37	1.00	0.19
ChestPainType	-0.11	-0.02	-0.04	0.25	0.70	0.05
RestingBP	-0.01	0.00	0.00	0.00	0.39	0.00
Cholesterol	0.01	0.02	0.01	0.14	0.70	0.03
FastingBS	-0.08	-0.01	-0.01	0.04	0.00	-0.02
RestingECG	0.02	0.02	0.02	0.03	1.00	0.01
MaxHR	0.00	0.00	0.00	0.01	0.44	0.00
ExerciseAngina	0.20	0.13	0.12	0.19	1.00	0.09
Oldpeak	0.00	0.02	0.02	0.14	0.45	-0.01
ST_Slope	0.39	0.24	0.22	0.48	1.00	0.24
$Elapsed$	$< 1s$	$< 1s$	$1s$	$< 1s$	$< 1s$	$< 1s$
RF						
Age	0.03	0.05	0.05	0.10	0.94	0.04
Sex	0.13	0.04	0.03	0.17	1.00	0.08
ChestPainType	-0.12	-0.06	-0.06	0.13	0.51	0.00
RestingBP	-0.01	0.00	0.01	0.18	0.86	0.06
Cholesterol	0.04	0.02	0.02	0.08	0.90	0.03
FastingBS	-0.03	0.00	0.00	0.00	0.00	0.00
RestingECG	0.03	0.03	0.03	0.05	1.00	0.02
MaxHR	0.00	0.02	0.02	0.24	0.93	0.10
ExerciseAngina	0.16	0.10	0.10	0.16	1.00	0.08
Oldpeak	0.01	0.01	0.01	0.17	0.48	0.00
ST_Slope	0.26	0.22	0.19	0.43	1.00	0.22
$Elapsed$	$< 1s$	$< 1s$	$2s$	$< 1s$	$< 1s$	$< 1s$
SVM						
Age	0.03	0.02	0.03	0.26	0.92	0.11
Sex	0.22	0.05	0.06	0.27	1.00	0.14
ChestPainType	-0.07	-0.02	-0.04	0.08	0.09	-0.03
RestingBP	-0.01	0.01	0.00	0.08	0.98	0.04
Cholesterol	0.03	0.02	0.02	0.22	0.98	0.11
FastingBS	-0.04	0.00	0.00	0.02	0.00	-0.01
RestingECG	-0.01	-0.01	-0.01	0.01	0.00	0.00
MaxHR	-0.03	0.00	0.00	0.26	0.84	0.09
ExerciseAngina	0.16	0.11	0.11	0.20	1.00	0.10
Oldpeak	-0.01	0.02	0.01	0.15	0.44	-0.01
ST_Slope	0.37	0.26	0.24	0.53	1.00	0.27
$Elapsed$	$< 1s$	$< 1s$	$2s$	$< 1s$	$< 1s$	$< 1s$

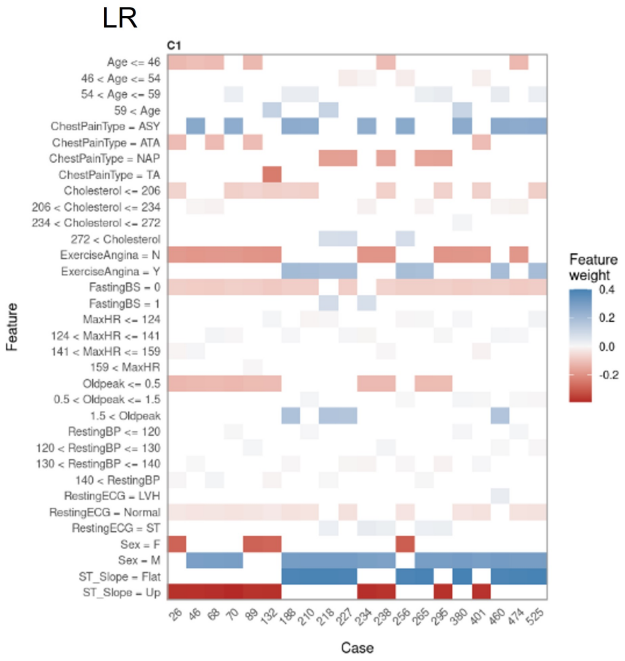


Figure B.1: Illustration of the LIME heatmap for heart disease prediction with LR (class label = C1) for 20 randomly selected instances from the training data.

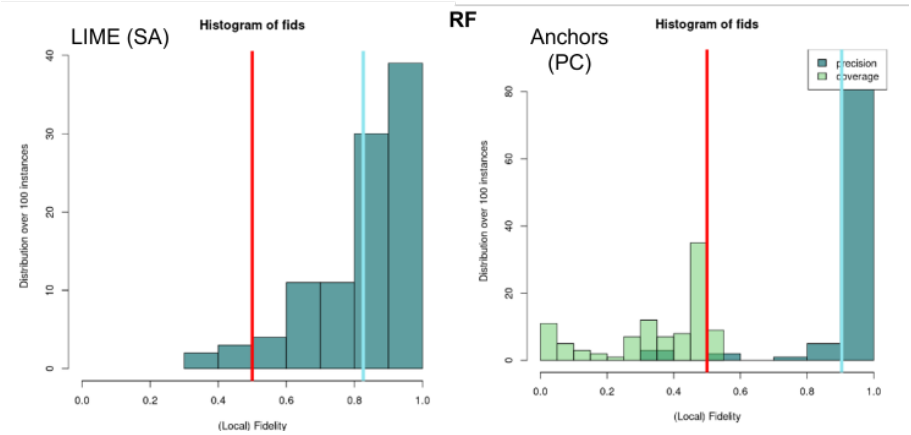


Figure B.2: Fidelity histograms for LIME (left) and Anchors (right) for RF model predictions explanations. Vertical red line represents 50% fidelity. Vertical blue line represents mean fidelity (mean precision for Anchors). It is important to assess both precision and coverage values on the right.

Table B.2: Global importance of input features in percent (range [0,1]) for heart data set. The values were computed using the entire training data: 527 instances (for Shapley, kernelSHAP and CI).

LR	Shapley	kernelSHAP	CI	PFI
Age	0.06 ± 0.04	0.06 ± 0.04	0.10 ± 0.03	
Sex	0.14 ± 0.09	0.14 ± 0.09	0.14 ± 0.06	
ChestPainType	0.18 ± 0.06	0.18 ± 0.06	0.20 ± 0.07	
RestingBP	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	
Cholesterol	0.03 ± 0.03	0.03 ± 0.03	0.07 ± 0.02	
FastingBS	0.02 ± 0.02	0.02 ± 0.02	0.03 ± 0.01	
RestingECG	0.02 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	
MaxHR	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	
ExerciseAngina	0.13 ± 0.04	0.13 ± 0.04	0.09 ± 0.02	
Oldpeak	0.10 ± 0.05	0.10 ± 0.05	0.13 ± 0.05	
ST_Slope	0.30 ± 0.08	0.30 ± 0.07	0.23 ± 0.07	
$Elapsed$	$334s$	$559s$	$73s$	
RF				
Age	0.06 ± 0.04	0.06 ± 0.04	0.10 ± 0.04	0.01
Sex	0.08 ± 0.06	0.08 ± 0.06	0.06 ± 0.06	0.13
ChestPainType	0.18 ± 0.06	0.18 ± 0.06	0.13 ± 0.07	0.20
RestingBP	0.03 ± 0.04	0.03 ± 0.03	0.08 ± 0.03	0.02
Cholesterol	0.04 ± 0.03	0.03 ± 0.03	0.06 ± 0.03	0.01
FastingBS	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.00
RestingECG	0.02 ± 0.02	0.02 ± 0.02	0.03 ± 0.02	0.04
MaxHR	0.06 ± 0.05	0.06 ± 0.04	0.10 ± 0.06	0.12
ExerciseAngina	0.14 ± 0.06	0.14 ± 0.06	0.09 ± 0.06	0.16
Oldpeak	0.10 ± 0.07	0.11 ± 0.06	0.16 ± 0.07	0.08
ST_Slope	0.27 ± 0.10	0.27 ± 0.09	0.18 ± 0.11	0.22
$Elapsed$	$366s$	$1407s$	$72s$	$< 1s$
SVM				
Age	0.06 ± 0.04	0.06 ± 0.04	0.10 ± 0.05	
Sex	0.11 ± 0.07	0.11 ± 0.07	0.08 ± 0.05	
ChestPainType	0.10 ± 0.03	0.10 ± 0.02	0.06 ± 0.03	
RestingBP	0.02 ± 0.02	0.02 ± 0.02	0.07 ± 0.03	
Cholesterol	0.04 ± 0.03	0.04 ± 0.03	0.08 ± 0.04	
FastingBS	0.01 ± 0.01	0.01 ± 0.02	0.01 ± 0.01	
RestingECG	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	
MaxHR	0.06 ± 0.04	0.06 ± 0.04	0.13 ± 0.06	
ExerciseAngina	0.11 ± 0.04	0.12 ± 0.04	0.06 ± 0.02	
Oldpeak	0.13 ± 0.09	0.13 ± 0.00	0.20 ± 0.12	
ST_Slope	0.35 ± 0.10	0.34 ± 0.09	0.20 ± 0.06	
$Elapsed$	$350s$	$1628s$	$42s$	

Table B.3: Extra results of WBC metric for a linear regression model: $f(x) = 0.2x_1 + 0,3x_2 + 0,5x_3$ - example from page 17. Instances explained are from a student with the average grade in all courses and from a student with the maximum grade in all courses. Global importance values are also shown, computed using 1000 randomly selected instances (total data size is 9261).

C2	LIME	Shapley	SHAP	CI	CU	Cinfl
x1=15	-0.03	0	0	0.2	0.5	0
x2=15	-0.04	-0.02	0	0.3	0.5	0
x3=15	-0.07	0	0	0.5	0.5	0
C3	LIME	Shapley	SHAP	CI	CU	Cinfl
x1=20	0.09	0.11	0.1	0.2	1	0.1
x2=20	0.14	0.16	0.15	0.3	1	0.15
x3=20	0.23	0.25	0.25	0.5	1	0.25
Global	Shapley	SHAP	CI			
x1	0.22 ± 0.15	0.21 ± 0.14	0.2 ± 0.0			
x2	0.31 ± 0.18	0.31 ± 0.18	0.3 ± 0.0			
x3	0.47 ± 0.20	0.48 ± 0.21	0.5 ± 0.0			
Elapsed	196s	3631s	6s			