UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Guidelines for the Assessment of Black-box Interpretability Methods

Gabriel Gazetta de Araujo

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Gabriel Gazetta de Araujo

Guidelines for the Assessment of Black-box Interpretability Methods

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
October 2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

G289g

Gazetta de Araujo, Gabriel
Guidelines for the assessment of black-box interpretability methods / Gabriel Gazetta de Araujo; orientador Luis Gustavo Nonato. -- São Carlos, 2022.
74 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2022.

1. Machine Learning. 2. Interpretabilidade. I. Nonato, Luis Gustavo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Gabriel Gazetta de Araujo

Diretrizes para avaliação de técnicas de Interpretabilidade de modelos Caixa-Preta

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *EDIÇÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
Outubro de 2022

Dedico este trabalho para a minha mãe que sempre acreditou em mim e me apoia em todas as decisões que tomo na vida; e ao meu falecido pai, em que uma das últimas coisas que lhe contei foi que eu havia sido aceito no programa de mestrado na USP, e que igualmente me apoiou durante toda minha vida e processo educativo.

ACKNOWLEDGEMENTS

Gostaria de agradecer primeiramente a Deus por abençoar a minha vida e me permitir alcançar meus objetivos e oportunidades. Gostaria de agradecer especialmente meu orientador, Prof. Dr. Luis Gustavo Nonato for me aceitar como orientando e por prover-me tamanho suporte e ajuda durante todo o processo. Agradeço meu ex-chefe Reinaldo de Bernardi por permitir com que eu deixasse o trabalho toda semana para frequentar às aulas em São Carlos e principalmente por me encorajar a iniciar o programa de mestrado e incentivar meu desenvolvimento profissional. Gostaria de agradecer à USP, aos funcionários do ICMC, aos meus professores pelo conhecimento e apoio, à banca examinadora, aos meus colegas e amigos, em especial Luiz Hiroshi Horita por sempre me ajudar e me apoiar durante o meu Mestrado. Agradecimento especial para a minha mãe, Silvana Cassia Girotto Gazetta pelo incentivo ao estudo e apoio incondicional.

RESUMO

ARAUJO, G. G. **Diretrizes para avaliação de técnicas de Interpretabilidade de modelos Caixa-Preta**. 2022. 74 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Com o surgimento de redes neurais profundas e algorítmos complexos de aprendizado de máquina, tem-se buscando cada vez mais maiores performances com o objetivo de alcançar melhores acurácias de classificação em uma variedade de aplicações. A busca por maior acurácia leva a modelos preditivos complexos conhecidos como caixas-pretas, que não oferecem acesso ao processo interno de decisão: estes modelos providenciam pouca ou nenhuma explicação no motivo pelo qual um determinado resultado foi obtido ou o que influenciou este resultado. Infelizmente, estas desvantagens podem ser impactantes especialmente em aplicações sensíveis como em cenários legais, sociais, médicos ou financeiros em que uma classificação errada ou uma classificação gerada por motivos errados pode causar impactos significativos. Motivados por esta preocupação, técnicas de interpretabilidade começam a surgir com o objetivo de trazer, por uma variedade de métodos, explicações para resultados de modelos caixa-preta, ou então propondo algorítmos preditivos originalmente interpretáveis. Porém, tais técnicas ainda não são maduras e estão em constante desenvolvimento; da mesma forma, a avaliação de tais técnicas também carecem de amadurecimento. Atualmente, não há um consenso em como elas podem ser avaliadas ou comparadas, ou então quais propriedades elas devem garantir. Este trabalho, partindo desta lacuna, propõe um conjunto de métricas avaliativas capazes de calcular três propriedades de técnicas de interpretabilidade. Tais métricas podem ser usadas para avaliar parâmetros ou determinar a melhor ferramenta de interpretabilidade para determinados experimentos.

Palavras-chave: aprendizado de máquina, redes neurais, interpretabilidade, aprendizado profundo, avaliação, modelos caixa-preta.

ABSTRACT

ARAUJO, G. G. **Guidelines for the Assessment of Black-box Interpretability Methods**. 2022. 74 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

With the rise of deep learning and complex machine learning algorithms, higher performance has been sought to reach equally high accuracy in a variety of environments and applications. The search for high accuracy has led to complex predictive models known as black-boxes that do not offer access to their decision-making processes: these models provide little to no explanations on why a certain outcome has resulted or what influenced that outcome. Unfortunately, these drawbacks can be utterly significant especially with sensitive scenarios such as legal, social, medical or financial applications that a misclassified outcome or even an outcome classified for the wrong reason might cause tremendous impacts. Driven by this consternation, interpretability techniques have come into play in an effort to bring, through a variety of methods, explanations to the outcome of a black-box model or even the reasoning behind that model, or sometimes proposing an interpretable predicting algorithm altogether. However, these techniques are not well established yet, which means that they are in constant development; similarly, the assessment of these techniques is also lacking. Currently, there is not a consensus on how they can be evaluated or even what properties interpretability methods are supposed to meet. Driven by that gap, this work proposes a set of evaluation metrics that are capable of calculating three desired properties obtained from interpretability techniques. These metrics can be used to assess and determine the best parameters or the best interpretability technique for determined experiments.

Keywords: machine learning, neural networks, interpretability, assessment, deep learning, black-box.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artifical Intelligence
AUC	Area Under the Curve
DNN	Deep Neural Networks
EU	European Union
GAM	Global Attribution Mapping
GDPR	General Data Protection Regulation
IAT	Implicit Association Test
ILSVRC	ImageNet Large Scale Visual Recognition Challenge

CONTENTS

CHAPTER

1

INTRODUCTION

Black-box classifiers have an off-balance between interpretability and accuracy, solely focusing on performance instead of being clear on why specific decisions were made or why the output ended up resulting in a certain value or class (FONG; VEDALDI, 2017). The lack of interpretability can affect essentially any application of Machine Learning that is deployed with a black-box model, the impact depends on how sensitive the subject is. Uninterpretable models may have major social and legal impacts once they may inherit bias and prejudice from human data (CALISKAN; BRYSON; NARAYANAN, 2017) (GREENWALD; MCGHEE; SCHWARTZ, 1998), up until the point of being sexist (DATTA; TSCHANTZ; DATTA, 2014) and even racist (ANGWIN *et al.*, 2016). Since black-box models cannot reason why a certain prediction was made, they are vulnerable to adversarial effects and perturbations (GOODFELLOW; SHLENS; SZEGEDY, 2015) (LIANG *et al.*, 2017) independently of the type of data being used, whether they are images, text, or tabular data; this is especially concerning since this vulnerability can open doors for malicious attacks that can harm sensitive systems, becoming even more concerning (PAPERNOT *et al.*, 2016).

Addressing the lack of interpretability is urgent, regulations such as the European General Data Protection Regulation that requires the right of citizens to understand the decision process on automated decision-making systems are becoming the rule (GOODMAN; FLAXMAN, 2017); meanwhile, these systems become less reliable as they cannot provide explanations on the decision process. To address that, interpretable machine learning methods start to emerge.

Interpretability methods are relatively new as the demand for explanations has recently risen, with several methods being published in the latest years. A variety of techniques, a lack of definition, and most of all, a lack of well-established methods on how to evaluate these techniques pose challenges to explainable machine learning and AI systems. These challenges are manifold: different interpretability techniques can provide explanations differently, and comparing them can be impracticable. As more techniques become available each year, it is crucial to have methods and frameworks to evaluate them. Some authors propose taxonomies to

group these techniques in a way that the fashion they deliver explanations can be differentiated (DU; LIU; HU, 2018) (LIPTON, 2016). But ultimately, interpretability methods do not have a well-established taxonomy or a consensus on how they can be evaluated (DOSHI-VELEZ; KIM, 2017). This gap in definition and frameworks allow research on how interpretability methods can be evaluated and assessed. Furthermore, with the advancement of such techniques, it is not clear which technique should be chosen by the user, nor how they can assert if they made the right choice. In sum, how can interpretability methods be compared given the increasing variety of methods available?

This work has three goals: first, through a literature review, it aims to help solidify a unified taxonomy that can be helpful to measure the influence of different explainability methods for black-box models. Secondly, this work aims to survey current relevant interpretability techniques, describing how they work and summarizing their features, and categorizing them into the proposed taxonomy. Thirdly, this work's main goal is to propose a first step in the effort of comparing explainability methods: qualitative functionally grounded evaluation metrics that can be adopted to assess different interpretability techniques in tabular data, more specifically, it proposes three metrics that can be used and adapted to different techniques and different applications so the best interpretability method for specific problems can be chosen.

This work is divided into six chapters: chapter 1 is limited to the introduction; chapter 2 is a contextualization of machine learning, deep learning, and the problem with black-box models, it also presents Interpretability and reviews its taxonomy. Chapter 3 contains an assessment of Interpretability properties and metrics based on the taxonomy presented in the previous chapter. Chapter 4 is a survey of current interpretability techniques which segregates them into different categories according to chapter 2's taxonomy. Besides, chapter 4 also selects three interpretability metrics to be tested according to the metrics selected. Finally, chapter 5 develops and proposes algebraic formulas to calculate the selected metrics. Moreover, Chapter 5 presents a toy dataset and demonstrates how the proposed metrics can be used to assess different interpretability techniques in other studies.

CHAPTER

2

CONTEXTUALIZATION

2.1 Machine Learning and Deep Learning

2.1.1 Machine Learning

Often, computer algorithms are necessary to solve real-world problems, and usually, there are certain known guidelines that these algorithms must follow to deliver the desired result. However, sometimes these guidelines are not known or cannot be written in a rule-based format, but can be extracted by a large quantity of data examples. The act of using historical data or datasets with multiple examples with the goal of extracting patterns and creating models to predict or describe data is known as Machine Learning (ALPAYDIN, 2010). Explaining data through models is not an easy task and not every existing pattern might be explainable or understood; because of that, Machine Learning aims to construct a good approximation and not a completely accurate mimic of real events (ALPAYDIN, 2010). The act of constructing a good approximation and performing well on new, unobserved data is known as generalization, according to Goodfellow, Bengio and Courville (2016).

Mello and Ponti (2018) agree that machine learning focuses on how computer algorithms can learn and recognize characteristics and patterns to be able to handle ordinary daily problems as well as support specialists in decision making by approximating a conditional probability function between the input space of examples and the desired output. Similarly, Goodfellow, Bengio and Courville (2016) state that machine learning has the capability of acquiring knowledge by extracting patterns from raw data.

2.1.2 Deep Learning

Traditional Machine Learning techniques have the limitation of not being capable of generalizing on more complex problems such as image classification, speech recognition, or

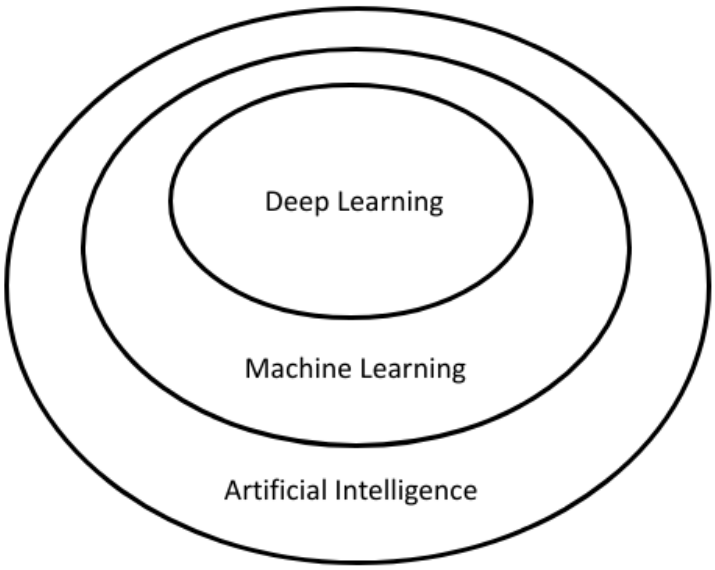
problems with high dimensionality due to their lack of ability to create simpler representations that can be understood by algorithms Ponti *et al.* (2017). Deep Learning, according to Goodfellow, Bengio and Courville (2016) is a branch of Machine Learning capable of successfully solving complex systems and problems that could not be processed or would require extensive processing of regular Machine Learning models due to high dimensionality and complexity. Deep Learning techniques are capable of doing so by expressing and representing complex concepts out of simpler concepts; or, as explained by LeCun, Bengio and Hinton (2015), Deep Learning computational models are able, through dense processing, convolutional or recurrent layers, to learn data features and representations with multiple levels of abstraction; they exemplify with image classification to explain how convolutional deep learning models simplify data in order to be processed: the image is fed into the algorithm as arrays of pixel values, and each layer is responsible for identifying a set of features in those arrays. Correspondingly, Goodfellow, Bengio and Courville (2016) deliver an example in which an image of a person can be represented and understood by Deep Learning methods by converting it through dense and deep layers into simpler concepts such as edges and corners, allowing for the image to be processed and classified, depending on the goal of the task.

2.1.3 Brief History of Machine Learning and Deep Learning

Machine Learning and Deep Learning have, according to Goodfellow, Bengio and Courville (2016) three waves of development, the first one is called *cybernetics* which involves the creation of neural networks. During this period, which dated from the 1940s to 1960s, McCulloch and Pitts (1943) developed studies to understand brain function through a single neuron linear model that was able to recognize between active or inactive neurons or two categories; however, the weights for the model had to be set by an operator, in that sense, the model created by the authors could not learn. Later, Samuel (1959) created in 1952 the first computer program that had the capacity of learning: an algorithm that could play the game of checkers based on previous moves performed by players. A few years later, in 1958, Rosenblatt (1958) proposed the first Neural Network, the Perceptron, based on a single representation of a neuron, which received an input along with random attributed weights, and through error calculation between the obtained output and the desired output, the Perceptron adapted these weights until it learned how to represent the data it was trained with. However, a single neuron cannot learn non-linear functions such as the *or-exclusive* function, as was shown by Minsky and Papert (1969) and ended up resulting in a backlash and weakened the studies in biologically inspired learning according to Goodfellow, Bengio and Courville (2016).

The second wave in Deep Learning, according to Goodfellow, Bengio and Courville (2016) known as *connectionism*, emerged in the 1980s due to several studies but was mostly thrust by the creation of Multilayer Perceptrons and the back-propagation algorithm by LeCun *et al.* (1989), Rumelhart, Hinton and Williams (1986) and many other researchers that helped

Figure 1 – Relationship between Deep Learning, Machine Learning and Artificial Intelligence



Source: Goodfellow, Bengio and Courville (2016).

and contributed to the advances in the area. According to Goodfellow, Bengio and Courville (2016) the third wave started in the mid-2000s mostly due to the modeling of the first deep network by Hinton, Osindero and Teh (2006). Some years later, Krizhevsky, Sutskever and Hinton (2012) developed the first Deep Convolutional Neural that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. The ILSVRC, according to Russakovsky *et al.* (2015) was an annual challenge that evaluated algorithms for image classification in large datasets. Krizhevsky, Sutskever and Hinton (2012)'s Deep CNN was the first artificial intelligence model that won the ILSVRC and has been responsible for popularizing and showing the computational power of Deep Learning models. Since then, many other models have been created, each time more advanced and deeper than before.

2.1.4 Relationship Between Machine Learning and Deep Learning

According to Goodfellow, Bengio and Courville (2016), Deep Learning is a particular type of Machine Learning that can explain complex concepts in simpler manners, having powerful mechanisms to represent data and learn from it. The authors state that both Machine Learning and Deep Learning are approaches to Artificial Intelligence, which aims to tackle problems and solve them similar to the way humans would do, being also capable of doing it even better and faster. Artificial Intelligence is a broad field that looks for automation, and, as its name suggests, implements artificial systems that are intelligent enough to perform determined tasks. Figure 1 shows an adaptation of the relationship between Deep Learning, Machine Learning, and Artificial Intelligence proposed by Goodfellow, Bengio and Courville (2016).

Throughout this section, it has been shown how Machine Learning and Deep Learning

have conquered great achievements and have been in the spotlight for being able to explain complex data with high accuracy; however, increasingly higher demands for complex and deep decision systems that are capable of processing and recognizing equally complex data faster and easier, end up resulting in decision systems that are hard to interpret and to understand the reasoning behind the decision making.

2.2 Machine Learning Bias and the problem with Black-Box Classifiers

A Gartner survey summarized by Costello (2019) has shown that when 3000 CIOs of different segments and industries were asked about Artifical Intelligence (AI), an increase of 270% of the CIOs responded that their companies have already deployed some kind of AI when compared to a 2015 survey. Driven by the capability of storing and processing large quantities of data inexpensively, along with progress in learning algorithms, machine learning, and artificial intelligence have been included in most aspects of life, such as medicine, forensics, business, technology, manufacturing, education, etc; the usage includes computer vision, image classification, speech recognition, natural language processing, and every supervised learning application. Due to the high diversity in applications, machine learning has been on the rise, allowing faster decision-making and simpler processes (JORDAN; MITCHELL, 2015).

To provide the best result or the most *accurate* result, some algorithms can sacrifice simplicity and clarity for higher accuracy, generating black-box classifiers. "A black box is a map $f : X \to Y$ from an input space X to an output space Y, typically obtained from an opaque learning process." (FONG; VEDALDI, 2017). This includes any classifier in which the learning process is not known by the observer or known but uninterpretable. The concern about black-box classifiers will become clear through the next paragraphs.

A study conducted by Caliskan, Bryson and Narayanan (2017) showed that machine learning models can inherit bias from human day-to-day language. As a source of human bias, the authors used the Implicit Association Test (IAT) developed by Greenwald, McGhee and Schwartz (1998) which documents responses when subjects associate terms with pleasant and unpleasant words. Amongst the findings, the study shows that names generally associated with black people were linked to more unpleasant words than names most associated with white people. Moreover, female names were associated with family over a career than male names. The models inherit stereotypical and even prejudicial behavior from datasets they were trained with; this is especially concerning since machine learning models can be used to score credit for loans or pre-selecting applicants for job interviews, for instance.

Datta, Tschantz and Datta (2014) presented a tool that analyses the interaction between users and online advertisement providers such as Google Ads. Amongst the findings, the authors observed that by tweaking the settings and changing the user's gender to female, fewer ads

related to high-paying jobs were shown than if the user was set to male. On a more concerning note, an experiment conducted by Angwin *et al.* (2016) observed that COMPAS, a machine learning model for risk assessment that predicts the incidence of new crimes was strongly biased against black people, classifying black subjects with a history of misdemeanors with higher risks than white subjects with a history of burglary or armed robbery.

Classification models use historical data to be trained; however, this data can be biased or contain social discrimination, the result is a classification model that behaves accordingly, creating discriminatory rules that can be embedded and hidden within layers of this model. It should be noted that discrimination can be direct (explicit) or indirect (implicit but with the same effect), and models can inherit both (PEDRESCHI; RUGGIERI; TURINI, 2008).

Taking into account the work presented in the previous paragraphs, it becomes clear that machine learning models must be unbiased and nondiscriminatory, including religion, gender, race, sexual orientation, and so on. Related to that, in 2016, the European Union (EU) has adopted the General Data Protection Regulation (GDPR), which is a series of regulations concerning Privacy and Data Protection for all citizens of every European Union's countries, the GDPR went into effect on April 2018. Amongst other topics, it restricts automated decision-making recommendation systems, leaving the citizen the right not to be subject to any decision based solely on such systems; also, it requires recommendation systems to have measures against any discriminatory effect and gives the citizen the right to obtain information on the logic involved in the result of the recommendation given by those systems (GOODMAN; FLAXMAN, 2017).

However, not only in social and legal aspects bias creates problems with black-box recommendation systems. Goodfellow, Shlens and Szegedy (2015) conducted research adding small perturbations, also known as adversarial effects in GoogLeNet, a deep convolutional neural network architecture developed by Szegedy *et al.* (2014). These small perturbations allowed us to observe a completely different classification of images. An example is an image of a panda bear, correctly classified, that is subject to a linear perturbation unperceptive to the human eye, resulting in the neural network classifying it as a gibbon. Furthermore, Su, Vargas and Sakurai (2017) performed an experiment on Deep Neural Networks (DNN) classified on popular datasets. The authors concluded that for a specific dataset almost a third of the images perturbed with the modification of a single-pixel changed the classification result altogether with confidence above 70. Other studies, such as research conducted by Nazaré *et al.* (2018), showed that CNNs were capable of learning noise levels and types while ignoring their visual category. Those alarming findings raise awareness of the vulnerability of neural networks.

Similarly, text-classification DNNs are also affected by adversarial effects, including absolute different class attribution when a single word was added to the text, according to Liang *et al.* (2017). Another study by Papernot *et al.* (2016) showed that by attacking a model with adversarial examples unnoticed by humans, DNN models misclassified more than 80% of them, reaching up to 96% of wrongly classified instances due to perturbations depending on the model.

The authors are particularly concerned with malicious content such as malware attacks that could harm systems such as vehicle control.

As could be noticed, black-box predictors can deliver biased or misclassified results, and since these models are not clear in terms of their classifying process, the users cannot expect to understand how the results came about. This lack of comprehensibility opens doors for a recently discussed feature in Machine Learning: interpretability.

2.3 Interpretability

Availability of large datasets, along with methodology improvements and the capability of processing large quantities of data have risen the performance of Artificial Intelligence systems that excel in uncountable complex applications and can even outperform humans in some tasks. Nevertheless, these systems often adopt black-box approaches which can deliver higher accuracy despite high complexity and non-linear structures. These opaque systems have several drawbacks as could be observed in the previous section; consequently, in order for these systems to be trusted, they should be *interpretable*. (SAMEK; WIEGAND; MÜLLER, 2017)

2.3.1 Basic Concepts on Interpretability

Doshi-Velez and Kim (2017) define interpretability in terms of Machine Learning as the "ability to explain or to present in understandable terms to a human"; thus, it means that predictions as in why the classifier returns certain classification or how a model classification work as a whole must be explainable and understandable to an individual. For Lou, Caruana and Gehrke (2012), interpretability is defined simply as the ability of users to understand the contribution of individual features in the model; in other words, being able to quantify the impact of each predictor. Lipton (2016); on the other hand, believes that establishing meanings and definitions of interpretability and related terms must be made by understanding real-world goals on interpretability and states that interpretability does not have a single meaning but rather several distinct ideas.

In addition, Doshi-Velez and Kim (2017), believe that there is not a consensus on what interpretability is and how it can be evaluated; along with that, the authors also assert that current interpretable approaches rely on assuming that somehow all models that deliver a good result can be interpretable or use known interpretable models to be optimized to deliver better results. Concerned with that, the authors propose desiderata on interpretability and related terms to assess a lack of definition and evaluation. The authors believe that interpretability helps meet other important desiderata and elucidate:

> Notions of *fairness* or *unbiasedness* imply that protected groups (explicit or implicit) are not somehow discriminated against. *Privacy* means the method protects sensitive information in the data. Properties such as *reli-*

> *ability* and *robustness* ascertain whether algorithms reach certain levels of performance in the face of parameter or input variation. *Casuality* implies that the predicted change in output due to a perturbation will occur in the real system. Usable methods provide information that assists users to accomplish a task [...], while *trusted* systems have the confidence of human users[...](DOSHI-VELEZ; KIM, 2017)

Adding to that, Guidotti *et al.* (2018b) mention *accuracy* as an important definition and describe it as an evaluation on how correct model's predictions are when exposed to unseen data. Accuracy is important because it's one of the most sought aspects of models in real-life problems. The authors also elucidate the concept of interpretable data: data such as images and texts are known for being interpretable, in a way that no metadata is needed to comprehend, as they are more related to the manner humans actually communicate and understand. Tabular data as in spreadsheets and matrices are not easily interpretable by humans, as metadata might be needed to allow meaningful insights; at the same time, this type of data can be easily managed by algorithms since little to no treatment is necessary; the same can't be said about text or images since they do need more complex preprocessing techniques such as feature extractions to identify important patterns and characteristics to be later processed in Machine Learning algorithms.

With such terms being known, further concepts of interpretability can be discussed.

2.3.2 Importance of Interpretability

Samek, Wiegand and Müller (2017) believe that models or their outputs being interpretable is important for several reasons. The first of them is the verification of the system, which might be crucial depending on the domain (health-care for instance) since it has to be understandable so it can be trusted and deployed. Furthermore, in cases where system outputs are reasonable and explainable, it is possible to find patterns in data that were still unknown or unnoticed by specialists. Further, interpretability can facilitate the performance of a predictive system, helping identify bias in datasets that could influence results and bring biased outputs. Lastly, the authors state that interpretability has been becoming legally mandatory, especially for sensitive systems (GDPR).

Lipton (2016) points out that interpretability is needed when in some scenarios, predictions and their metrics such as accuracy are not sufficient to portray a model; sometimes, other aspects rather than validation error must be optimized by the model, in the author's words: "in all cases, interpretations serve those objectives that we deem important but struggle to model formally".

Similarly, Ribeiro, Singh and Guestrin (2016) believe that an algorithm classification cannot always be trusted despite its high accuracy: interpretability and explaining the results need to occur. They also believe that *trust* is intertwined with interpretability, once that if users don't trust a model, it's unlikely that it will be used, and interpretability can help increase trust.

The authors separate *trust* into two related definitions: trusting a model as a whole, and trusting a single prediction. Correspondingly, Guidotti *et al.* (2018b) separate interpretability into two categories: global interpretability, when an individual is able to understand the whole login and process of a model; and local interpretability, when an individual can only identify the reasons behind a specific decision of a model. The authors agree that global interpretability, or the whole decision process of a model being interpretable, generates more trust to the user.

According to the passages above, it can be noticed that interpretability is a rising key term in machine learning, and in some cases, it must be guaranteed; but how can models be interpretable? Can it be assumed that black-box models might become interpretable?

2.4 Interpretability Taxonomy

Doshi-Velez and Kim (2017) claim that there is no consensus on interpretability taxonomy, especially regarding how it is evaluated. However, there are also several methods and ways a model can be interpretable, knowing what they are and learning how to differentiate and assess each one is fundamental to narrow down and group different techniques, and ultimately, know what to expect of them.

Du, Liu and Hu (2018) group interpretability into intrinsic and post-hoc interpretability. Similarly, Lipton (2016) states that interpretability can be divided (but not limited) into two main categories: model transparency and post-hoc interpretability, which is then divided into sub-categories: visualization, text explanations, local explanations, and explanation by example. In contrast, Guidotti *et al.* (2018b) explore more alternatives separating interpretability into four categories: model explanation, outcome explanation, Black-box inspection and Transparent Design. In this work, an adaptation of those will be adopted.

Du, Liu and Hu (2018) state that interpretability can be further divided into two types: global and local interpretability. In accordance with the authors, global interpretability is met when an individual is able to fully understand the behavior of a model as a whole, while local interpretability is related to understanding a set of predictions and making sense of the reasons behind the model's prediction.

2.4.1 Intrinsic Transparency

Transparency is related to the capability of fully understanding how a model works as a whole or at least its components. If a certain model is transparent, it can be understood that the model can be fully comprehended by humans; because of that, models with high complexity such as deep neural networks or random forests cannot be categorized as transparent. It can be said that simple algorithms such as linear regressions and decision trees generate transparent models once their mechanism can be understood; however, it might not always be the case, since even these algorithms can generate complex models when faced with complex datasets that require a

huge amount of nodes of a decision tree for instance. With that being said, model transparency depends mostly on the human cognitive capacity rather than depth or other model parameters (LIPTON, 2016).

Du, Liu and Hu (2018) argue that transparent models can be obtained by either adding constraints on black-box models or by using simpler algorithms to create less complex models. The first adds constraints to simplify models, having the downside of being likely to reduce model performance; the second considers simplified methods to mimic a black-box behavior. These simplified algorithms have to be simple and have the capacity of being interpretable, Freitas (2013) analyses comprehensible models, between them, he describes as being interpretable models such as decision trees, classification rules, and bayesian network classifiers.

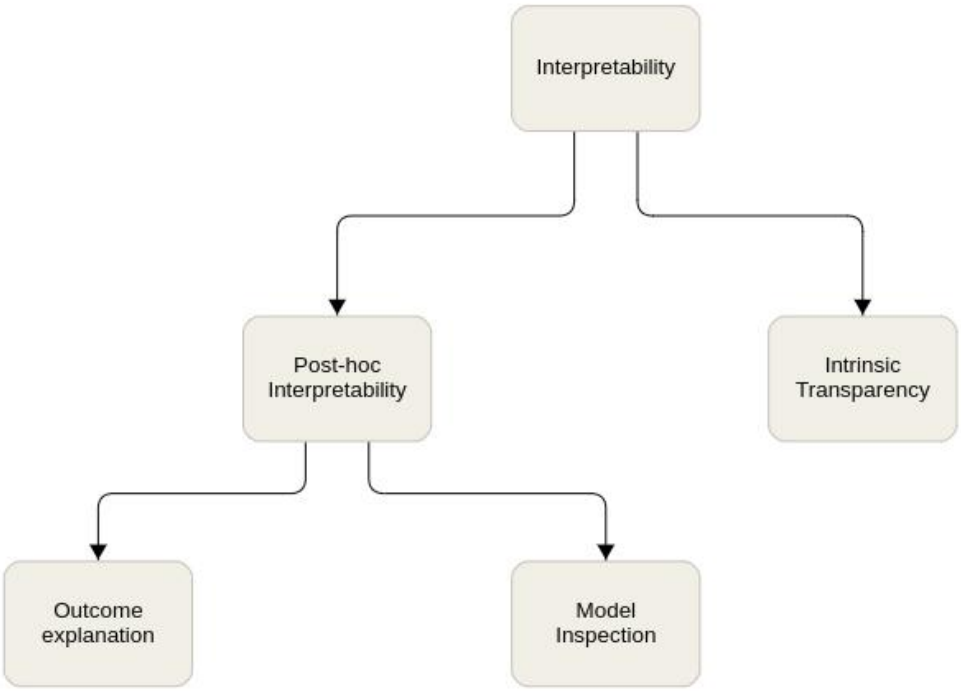
Lastly, Guidotti *et al.* (2018b) point out that a transparent box is a model that is locally or globally interpretable on its own; however, if a model is locally interpretable and transparent, consequently the same model might not be transparent at some point. For that reason, in this work, a model will only be considered transparent if it is indeed globally transparent.

2.4.2 Post-hoc Explanations

For Lipton (2016) Post-hoc explanations do not elucidate how black-box models work but instead help to provide useful information after the black-box model has already been deployed. These types of explanations can be of different types such as outcome explanations or visualizations, but the main idea is that these methods are used *after* the model has been trained; hence its name: post-hoc. Guidotti *et al.* (2018b) describe that outcome explanation, one of the main post-hoc explanation methods consists of a method that according to a certain sample that was fed into a predictive model, is capable of explaining the reason behind that particular prediction. Another method of post-hoc explanation also described by the authors consists of inspecting black-box models to provide representations of their behavior such as sensitivity analysis that can provide feature importance, for instance.

As reasoned by Du, Liu and Hu (2018), the differences between post-hoc explanations and transparent interpretability "lies in the trade-off between model accuracy and explanation fidelity". To clarify, the authors state that interpretable models can provide an accurate explanation of how models intrinsically work; however, accuracy might have to be sacrificed as complexity decreases to ensure explainability; on the other hand, post-hoc interpretability does not sacrifice performance since the black-box model is not tampered with, however, once they work on model approximations or explanations of individual instances, they end up with limitations as of how close they can mimic or explain the predictions of a model.

Figure 2 – Structure of Interpretability Methods



Source: Elaborated by the author.

2.4.3 Adopted Structure

The concepts and methods of interpretability research help structure the following chapter of this work, which consists of a survey of relevant interpretability techniques recently published. Based on the references aforementioned, this work features the interpretability method structure as demonstrated in Figure 2.

As it can be noticed, the adopted structure has two main interpretability categories: Intrinsic Transparency and Post-hoc Interpretability, the latter can be sub-categorized in Model Inspection and Outcome Explanation. With that being said; in the following survey, the techniques reviewed will be then classified into Intrinsic Transparency, Outcome Explanation, or Model Inspection; furthermore, the techniques can also be labeled as global or local explanators, as well as being model-agnostic (meaning that it can be applied to any classifier) or being specific to a certain kind of predictor. This taxonomy is helpful to narrow down and separate between methods of black-box interpretability in the next chapter.

CHAPTER

3

INTERPRETABILITY PROPERTIES AND METRICS

A particularly important aspect of interpretability is its assessment; that is, how can interpretable methods be compared and assessed, or how to choose between different techniques according to specific problems? According to Doshi-Velez and Kim (2017), interpretability assessment remains a challenge that has not yet been fully addressed in current research works in the area.

Doshi-Velez and Kim (2017) claim that there are three ways interpretability can be measured: application-grounded, human-grounded, and functionally-grounded. Application-grounded evaluation is related to complex and real experiments that require a specialized person to assess whether predictions were helpful to achieve a certain goal; these experiments are often time-consuming and expensive since they usually require a specialized evaluation. Also, even though a specialist might state that an explanation has a relation to the real reasoning behind the outcome, measuring how explanations are provided by different techniques might not be straightforward and very structured. For that reason, application-grounded evaluation can be subjective if specific metrics are not set.

Human-grounded evaluations, on the other hand, are simpler evaluations that measure the general quality of an explanation. There are several ways to apply these evaluations, but essentially, non-experts are asked to choose better explanations in terms of quality; some examples are choosing a model output based on the explanation and the input, choosing a true model output vs. a corrupted output to understand whether the human could understand the true output and the explanation. Generally, these experiments are less expensive than application-grounded evaluations since they do not require specialists in a certain domain; however, in order for these evaluations to provide meaningful results, a reasonable amount of testers should participate and metrics should be well-established.

Lastly, functionally-grounded evaluations do not require humans, but instead, use a

formal definition of interpretability as a proxy for quality. Because of that, these experiments are both cheaper and faster to perform, but require that the models have been validated before; that is, they have been already proven to be interpretable. The main challenge to this type of evaluation is determining a proxy for explanation quality that is capable of translating the properties and characteristics of the methods.

Considering the evaluation methods described by Doshi-Velez and Kim (2017), this work is focused on functionally-grounded evaluation as it allows a cheaper and generally faster approach to assessing interpretability methods. Moreover, it consists of a more tangible way to measure interpretability since the techniques to be demonstrated in the following chapters are applied to previously validated models and datasets, as the evaluation method requires.

As aforementioned, a consensus on how interpretability can be measured and assessed remains unmet. Also, the majority of publications and research development in Machine Learning Interpretability is focused on creating newer and more advanced methods but not much focus has been given to finding metrics to compare these techniques according to Adadi and Berrada (2018). In this section, previously studied evaluation desiderata will be reviewed.

As mentioned in section 2.4, there's a trade-off between interpretability and complexity where complex models are not easily interpretable and vice-versa. Accordingly, Herman (2017) describes a trade-off between interpretability and completeness, where interpretability aims for explanations that are simple to comprehend whereas completeness is concerned with the fidelity of the explanation to the model. Adding to that, Gilpin *et al.* (2018) claim that researchers should aim for explanations that are both interpretable and complete, which might be a challenge. While reviewing current evaluation desiderata, the notion of both interpretability and completeness will be considered to understand how metrics measure both aspects of explainability.

Alvarez-Melis and Jaakkola (2018) describe three metrics on interpretability evaluation: Explicitness/Intelligibility, Faithfulness, and Stability. Explicitness and intelligibility measure how interpretable the explanations are, for the authors, the key point is whether an explanation is immediately understandable. Evidently, determining if an explanation is understandable is not trivial whatsoever, and a complete problem understanding should be done before. Faithfulness, on the other hand, can be somewhat easier to measure, it is related to understanding whether relevant features (for techniques that use that approach) are indeed relevant to predicting an object with respect to the original model. Lastly, stability measures how coherent explanations are for similar inputs; in other words, it measures if local perturbations significantly alter the explanations. The concept is simple but the actual measuring can be tricky, the authors elaborate that measuring stability depends on the input, suggesting different approaches for image vs. tabular data and also, in tabular data between discrete and continuous inputs.

Doshi-Velez and Kim (2017), emphasize the importance of establishing a consensus on what interpretability is and how it can be evaluated; the authors also assert that current explainable approaches rely on assuming that somehow all models that deliver a good result

can be interpretable or use known interpretable models to be optimized to deliver better results. Concerned with that, the authors propose desiderata on explainability and related terms to assess a lack of definition and evaluation. The authors believe that interpretability helps meet other important desiderata and elucidate:

> Notions of *fairness* or *unbiasedness* imply that protected groups (explicit or implicit) are not somehow discriminated against. *Privacy* means the method protects sensitive information in the data. Properties such as *reliability* and *robustness* ascertain whether algorithms reach certain levels of performance in the face of parameter or input variation. *Casualty* implies that the predicted change in output due to a perturbation will occur in the real system. Usable methods provide information that assists users to accomplish a task [...], while *trusted* systems have the confidence of human users[...](DOSHI-VELEZ; KIM, 2017)

In a similar fashion, Lipton (2016) describes desiderata that can be employed in order to assess interpretability properties and techniques. The author states that *trust* is an important metric in interpretability, providing examples of why whether or not an individual trusts a prediction or explanation is subjective and depends on subjects and scenarios. A second desideratum described by the author is *casualty*; for him, it has a different meaning than described by Doshi-Velez and Kim (2017), arguing that casualty is more related to casual relationships between patterns or rules found in models applied to a general notion about real-world data. Another desideratum described by Lipton (2016) is *transferability*, which measures whether explanations and findings can be used in related settings. The fourth desideratum is described as *Informativeness* is also related to the latter, measuring the ability to provide useful information to support decision-making, such as described by Doshi-Velez and Kim (2017) in *usability*. Lastly, the fifth desideratum described is *Fairness and Ethical Decision-Making*, the author's definition concerns whether explanations yield ways of certifying or contesting them; this is linked to increased problems with black-box models and their issues such as described in Chapter 2. Note that the desiderata described by Lipton (2016) are qualitative rather than quantitative, differing from the functionally-grounded evaluations described by Doshi-Velez and Kim (2017) which are chosen to be adopted in this work.

Murdoch *et al.* (2019) developed a framework to evaluate interpretability methods called PDR (Predictive, Descriptive, Relevant). Their framework is comprised of three desiderata to assess interpretability: predictive accuracy, descriptive accuracy, and relevancy. According to the authors, predictive accuracy is associated with the black-box model approximation of data relationships and patterns, in other words, it measures how well the black-box model generalizes unseen data - e.g.traditional accuracy or the coefficient of determination. On the other hand, descriptive accuracy measures how well an explanation or an interpretation method represents the patterns learned by the model, it has a close meaning to the concept of *faithfulness* described by Alvarez-Melis and Jaakkola (2018) or *completeness* by Herman (2017) and Gilpin *et al.* (2018). Relevancy, on the other hand; is a qualitative desideratum that is concerned if

Table 1 – Overview of Interpretability Desiderata according to different authors and how they are related

Doshi-Velez and Kim (2017)	Lipton (2016)	Alvarez-Melis and Jaakkola (2018)	Gilpin *et al.* (2018)	Murdoch *et al.* (2019)	Honegger (2018)
Fairness-Unbiasedness	Ethical decision-making	-	-	-	-
Reliability-Robustness	-	Stability	-	-	Stability
Casualty	Casualty	-	-	-	Identity and Separability
Usability	Transferability and Informativeness	-	-	Relevancy	-
Trust	Trust	-	-	-	-
Privacy	-	-	-	-	-
-	-	Faithfulness	Completeness	Descriptive Accuracy	-
-	-	Explicitness-Intelligibility	-	-	-

Source: Research data.

explanations are useful, or *relevant* for a specific audience, for that reason, it can be subjective and change according to different topics and domains. The authors' definition of relevancy and the examples provided demonstrate that their concept of relevancy is closely related to the concept of *informativeness* and *transferability* as described by Lipton (2016), and *usability* by Doshi-Velez and Kim (2017); although it can be also linked to *trust*, cited by authors such as Lipton (2016) and Doshi-Velez and Kim (2017). Although the authors provide some examples of how these desiderata can be applied according to other works in the area, standardized methods of measuring these desiderata (except predictive accuracy) remain unmet. Honegger (2018) developed a simple framework comprised of three desiderata: identity, which measures if identical objects have identical explanations; separability, which measures if non-identical objects also have different explanations; and stability, that measure if similar objects have similar explanations. Again, these desiderata are only applied to techniques that provide individual explanations.

According to the information presented, even though desiderata differ from work to work, common ground and similar points of view can be sensed in some of them. For that reason, Table 1 presents an overview of every interpretability desiderata cited in the paragraphs above, while trying to associate them according to the authors' descriptions.

When observing Table 2, it is possible to visualize the similarities between each desider-

atum even though they are named differently according to each author; it is also possible to note that some aspects are repeated in most of the frameworks, such as the concept of stability and faithfulness. On the other hand, some aspects are not widespread across the frameworks proposed and can be suitable for a specific framework such as *privacy* and *explicitness/intelligibility*. One can note that even though some of the aspects can be somewhat simple to measure, such as stability; straightforward methods to measure most desiderata remain a challenge. That is, calculating *trust* or *fairness* can be subject to interpretation, and no consensus on how to measure such aspects has been met yet, with many authors using different approaches to compare methods and interpretability. In order to better understand the metrics suggested by each author while clarifying their proper meaning, Table 2 presents a summary of accumulated metrics proposed by the authors.

Table 2 – Summary of interpretability desiderata with their meaning

Metric	What it does
Ethics	Measures ethical decision making, avoiding biased and prejudicial models.
Stability	Measures whether explanations are stable with relation to output: similar inputs must provide similar explanations.
Identity and Separability	Measure if identical objects have identical explanations and vice-versa
Usability	Identifies the capability of providing useful information to support decision-making
Trust	Identifies if explanations are trusted by users to be used in decision-making
Privacy	Measures the ability of protecting sensitive information
Faithfulness	Identifies whether features deemed important to the explanation method are indeed relevant to predicting an object within the original model
Intelligibility	Measures how intelligible, or clear are the explanations to the user

Source: Research data.

By observing these metrics, it can be noted that most of them are human-grounded metrics, being qualitative rather than quantitative, focusing on the perception of the user rather than on calculations. Since these metrics usually require human subjects to evaluate them, it is not always possible due to factors such as time, budget, and other restrictions. Considering this work focuses on functionally-grounded evaluations, it will solely focus on quantitative metrics that can be evaluated without the intervention of several human subjects. With that said, Table 2 allows observing three functionally-grounded quantitative metrics: stability, identity, and faithfulness, which are the metrics that are being proposed along with mathematical approach to achieve them.

CHAPTER

4

SURVEY ON INTERPRETABILITY TECHNIQUES

This section consists of a survey reviewing and comparing a few techniques that allow to explain and interpret black-box models to assess the problems described in section 2.2; clearly, there are hundreds of tools available with numbers rising each day; however, this survey aims to map the main and most relevant or novel ones. The tools were selected based on a few criteria such as the number of citations and type of approach adopted to give explanations to bring a balanced set of tools; also, this survey aimed for techniques that preferably have the code available. The survey is structured according to an adaption of literature as described in subsection 2.4.3.

4.1 Overview of Existing Methods

A summary of techniques and methods for interpreting black-box models along with key features that are important to note for a better understanding of their mechanism for future research can be analyzed in Table 3.

Table 3 is segregated into three types of interpretability approaches, being: Model Inspection, Outcome Explanation, and Intrinsic Transparency. Also, techniques can be differentiated between explainers, which demonstrates which models the technique can be used to explain (Agnostic, Deep Neural Networks, Decision Forests, etc), and data type, which describes what types of data the technique can explain (tabular, image, text or agnostic). Also, Table 3 discloses whether the approach used to explain is local or global and whether there is any code available to test the technique.

Table 3 – Overview of Surveyed Tools

Tool	Explainer	Data Type	Approach	Method	Code
Lime	Agnostic	Agnostic	Local	Outcome Explanator	Yes
Lore	Agnostic	Tabular	Local	Outcome Explanator	Yes
Anchors	Agnostic	Agnostic	Local	Outcome Explanator	Yes
Deconv Nets	DCNN	Images	Local	Model Inspection	Yes
SHAP	Agnostic	Agnostic	Local/Global	Outcome Explanator / Model Inspection	Yes
G-SHAP	Agnostic	Agnostic	Local/Global	Outcome Explanator / Model Inspection	Yes
DeepLift	DNN	Agnostic	Local	Outcome Explanator	Yes
Grad-CAM	DCNN	Images	Local	Outcome Explanator	Yes
Interpretable Decision Sets	-	Tabular	Global	Intrinsic Interpretability	No
GAM	DNN	Tabular	Global	Model Inspection	Yes
TabNet	DNN	Tabular	Local/Global	Intrinsic Interpretability	Yes
Generalized SHAP	Agnostic	Agnostic	Local/Global	Outcome Explanator / Model Inspection	Yes

Source: Research data.

4.2 Review of Most Relevant Methods in our context

4.2.1 LIME: Local Interpretable Model-Agnostic Explanations

LIME, proposed by Ribeiro, Singh and Guestrin (2016) is an interpretability technique, which according to the authors, is capable of explaining instances of any classifier, as it is described as being model-agnostic. It works by learning a simple linear model with the predictions of the black-box model. The authors believe that an *interpretable explanation* should use representations that humans can understand regardless of how complex the features analyzed by the actual model are; because of that, the authors use a local surrogate approach to explain the outcomes.

Aiming for local fidelity, the authors claim that the explainer works by minimizing the function $\xi(x) = \underset{g \in G}{\operatorname{argmin}}\ \zeta(f,g,\pi_x) + \Omega(g)$ where the loss function is given by $\zeta(f,g,\pi_x) = \sum_{z,z' \in Z} \pi_x(z)(f(z) - g(z'))^2$. In these functions, let x be the instance to be explained, g an inter-

pretable model from a class of potential models G, $f(x)$ the probability that x belongs to a certain class, π a proximity measure that locally explores x, and Ω a measure of complexity, which has to be low enough to make the model understandable. As for the loss function, let z be the samples on the original distribution, and z' a set of perturbed new samples. In other words, LIME uses a kernel size to randomly draw new samples from the neighborhood of x with a kernel size, predicts them using f and weights (π) these sampled instances according to their proximity to x.

Furthermore, since LIME is a local interpreter on specific instances and not on the black-box model as a whole, the authors try to compensate for this by proposing a submodular pick based on feature importance on a set of representative samples on the dataset to bring more *trust* to the explainer. Also, to give evidence of the effectiveness of LIME, the authors provide a set of examples that involve demonstrations of text and image datasets processed with algorithms such as Support Vector Machines and Google's Inception Neural Network as well as user experiments comparing the explanations with those obtained with different techniques. Overall, LIME is considered by the authors a successful approach to provide interpretable insights on black-box models. Limitations pointed out by the authors go as far as not being able to apply the proposed submodular pick on images and interpretability not guaranteed when the black-box being used is highly non-linear even in the neighborhood of the samples, this may happen since LIME uses a linear approach to explain the instances.

4.2.2 Anchors: High-Precision Model-Agnostic Explanations

Anchors is a model-agnostic interpretability technique proposed by Ribeiro, Singh and Guestrin (2018), similar to LIME, also proposed by the same authors. Anchors works locally to explain instances and specific examples of data, and not the model as a whole. In other orders, given a certain instance x, Anchors tries to determine the explanation for the prediction $f(x)$. This technique was built to be applied to any kind of data, whether it may be tabular, text, or image. Anchors works essentially by finding local if-then rules that, as its name suggests, locally *anchors* its prediction to a certain value or class, guaranteeing that the prediction isn't likely to change in the presence of a determined anchor disregarding any changes in other features.

To achieve the anchors, there are two different approaches: *Bottom-up*, and *Beam Search*. The bottom-up approach is more time-consuming since it explores a space of potential anchors, because of that, the authors consider Beam Search the usual approach: it finds the best anchor candidates by evaluating a batch of perturbed data in concern with the precision of this set of data. With a set of predicates found, the anchor chosen is the one with higher precision and coverage. Precision can be understood as the ratio of the number of times when the label has not changed after newer perturbations to the number of all samples related to the anchor, whereas coverage is the probability that a determined anchor for a sample to also be present in other samples.

The authors performed experiments on two different settings: with simulated users and

real users, both comparing Anchors with LIME in different datasets. The experiment with real users tested if the candidates could predict model behavior using LIME and Anchors explanations, the authors observed that Anchors achieved higher precision and comprehensibility and it was also noted that even though LIME explanations made the users more confident to infer about model behavior and predictions, their precision was lower than with Anchors. These results match with the nature of both LIME and Anchors: one is a linear model and the other shows that in the presence of a determined feature (predicate) the prediction isn't likely to change, allowing less room for errors in human interpretation than a linear model.

As for Anchors' limitations, the authors emphasize that when an anchor is close to the boundary of a black-box's decision function, its rule ends up being overly specific, once otherwise, minimal changes might result in a different classification.

4.2.3 LORE: Local Rule-Based Explanations

LORE, proposed by Guidotti *et al.* (2018a) is a model-agnostic explanator that provides post-hoc interpretability by outcome explanations on binary outcomes. It works by locally exploring the neighborhood of an instance being explained and then fitting an interpretable model to explain the results, such as decision rules.

To address the problem of explaining a result obtained by a black-box model given by b, LORE explores a single instance x that has to be explained by generating a neighborhood of instances around x; this neighborhood is generated through genetic algorithms that optimize a population until it covers both outcomes $Z_{=}$ and $Z_{\neq}$ which $Z_{=}$ are the instances (z) created by the genetic algorithm that b classifies with the same label as x, while $Z_{\neq}$ is the population that b provides a different prediction for x. The fitness function used by the generic algorithms measures the distance between x and z aiming to find the limit in perturbations that causes b to change the prediction of x; in other words, the population generated by the genetic algorithm can be used to find the decision boundary of the black-box classifier b.

After generating the population Z, LORE fits a transparent predictor in the same population in order to locally mimic the behavior of the black-box classifier. The transparent predictor adopted is a decision tree due to its simplicity to extract decision rules. Following the path that instance x takes on the decision tree, a decision rule r can be found by $r = (p \rightarrow y)$ where p is a Boolean condition. Furthermore, since LORE explores the decision boundary, it is capable of explaining the changes in the feature space that would cause x to change the label of its respective predicted output $\hat{y}$, it does that by looking for paths in the decision trees that lead to a prediction $y \neq \hat{y}$, by finding $\Phi = (p[\delta] \rightarrow \hat{y})$, where δ is a counterfactual split condition. Finally, LORE delivers an explanation e such as $e = \langle r, \Phi \rangle$.

Guidotti *et al.* (2018a) performed experiments in a variety of settings, including the neighborhood generation, distance functions, and local vs. global approaches, the results sup-

ported the authors' choice in the presented algorithm architecture. Interestingly, the authors consider the generation of a single decision tree in all test instances to be a global approach, which might not be the case, once considering the calibration instances can be determinant for reaching a global explanator. In summary, the authors concluded that a decision tree used to explain single instances in the test set reaches better *hit* and *fidelity* scores, and most importantly, shorter tree depths, which consequently reach less complex rules.

The authors also considered comparing LORE with other tools such as Lime (RIBEIRO; SINGH; GUESTRIN, 2016) and Anchors (RIBEIRO; SINGH; GUESTRIN, 2018). Concerning Lime, the authors state that the rule explanation format of LORE, along with its counterfactual rules qualitatively outperforms Lime with its weights. Quantitative-wise, the results show higher *hit* scores taking turns for both Lime and Lore, depending on the dataset and black-box classifier used. As for a comparison with Anchors, qualitatively, LORE has the advantages of showing counterfactual rules; on the other hand, Anchor's rules can be simpler to understand since it only shows the features that solidify the output to be a certain state, whereas LORE shows every parameter involved in the prediction, which can become harder to understand as the rules become more complex. Quantitatively, Anchor's explanations also outperform LORE, showing more precision than LORE independently of the dataset or black-box algorithm used.

As for the restrictions, LORE can only be used in tabular data with binary independent variables, the authors cite, as future research to develop LORE to be able to perform with image and text data; moreover, the comprehensibility of the rules extracted by LORE should be tested in more complex settings (i.e. large tree depths that consequently delivers more complex rules) to understand and evaluate human comprehensibility.

4.2.4 SHAP - Shapley Additive Explanations

SHAP, developed by Lundberg and Lee (2017), is based on Shapley values proposed by Shapley (1953) which essentially is a solution concept in game theory that consists of finding the contribution of each actor or player in a coalition to achieve an outcome. The authors related this concept to model predictions to find out the contribution of each input feature to black-box model predictions. SHAP combines Shapley values with insights from current approaches of feature attribution methods such as LIME to provide a unified measure of feature importance on model predictions and behavior. The authors provide different options for model approximation, they developed a model-agnostic approximator along with four model-specific approaches differing from linear models to deep learning models, the advantage of using model-specific explanators, according to the authors, is mostly based on faster results. The model-agnostic explanator, called by the authors *Kernel SHAP* works similarly to LIME: it builds a weighted linear regression based on the input variables and the predictions of these variables given by the black-box model.

SHAP is considered both an outcome explanator and a model inspection since it can provide a global approach to how the model behaves concerning the whole training dataset as

well as explain the reasoning behind an individual prediction. It generates a variable importance ranking based on Shapley values and coefficients based on local linear regression as in LIME. These coefficients are able to explain the contribution of each variable to achieve the final prediction. It starts with a base value $E[f(z)]$ which is essentially the mean of all predicted values $y'$for the whole dataset; afterward, it calculates the next $\phi(n)$ SHAP value, which shows how a certain value of a specific attribute affects the final prediction by pushing the base value to values higher or lower than it; additively n SHAP values are calculated, resulting in the final prediction along with how much each feature contributed to it. This feature will be explored further in the next section.

Lundberg and Lee (2017) performed experiments such as computational efficiency and comparing explaining methods with human intuition. The latter consisted of two settings, one of them being sickness scores based on symptoms and the other money won by individuals; the authors claim that SHAP performed similar and consistent with human intuition on those two datasets, which was not the case with other methods tested by the authors (LIME and DeepLift). However, human intuition on feature contribution on simple datasets might not be the ideal test to be made to evaluate the performance of a model explaining method, other aspects must be addressed since simply concerning two datasets might be subjective to the complexity of the data being used.

4.2.5 DeepLIFT: Learning Important Features Through Propagating Activation Differences

DeepLIFT, which stands for Deep Learning Important Features, is a deep neural network model explanator created by Shrikumar, Greenside and Kundaje (2017) that is capable of assigning importance scores to the inputs for a given output by backpropagating the contributions in the network.

DeepLIFT aims to explain the output by the difference between a real input and a neutral reference input. To do so, first let x_n represent neurons in the layers that compute the target output neuron, given by t. Then, t_0 represents the reference from t. In that sense, DeepLIFT calculates the difference from reference $\Delta t = t - t_0$. Subsequently, DeepLIFT assigns contribution scores to Δx_i; that is, how differences from the reference in Δt is affected by the difference from reference Δt through the summation of $C_{\Delta x_i, \Delta t}$ so that the result is equal to Δt.

DeepLIFT reaches an explanation by backpropagating the signal from an output layer back to the input layer. It does so by using an analogy to the concept of partial derivatives through *multipliers* given by $m_{\Delta x \Delta t}$. Multipliers are used to calculate the chain rule to backpropagate the signal and identify the contribution to reach an output. Now let x_n be input neurons, y_n be neurons in hidden layers and t be the target output neuron, DeepLIFT uses the concept of difference from reference cited in the paragraph above to calculate $\sum m_{\Delta x_i \Delta y_i} m_{\Delta y_i \Delta t}$ DeepLIFT

can also measure positive and negative contributions by grouping positive and negative terms. The reference used to calculate the contributions comes from a reference input that depends on each application. Generally, reference inputs are chosen based on what the results are compared against, an example would be a background image of all zeros in the MNIST dataset.

To evaluate DeepLIFT, Shrikumar, Greenside and Kundaje (2017) used a convolutional neural network on digits dataset and obtained importance scores given by DeepLIFT and other methods, and analyzed how pixel changes affect the output from a determined class so that it alters to another class. DeepLIFT outperformed other methods such as *Guided Backpropagation* by being able to identify what pictures should change to alter the outcome number *8* to *3* or *6*. DeepLIFT's limitations are mostly based on the determination of reference inputs that have to be manually done and tested, an empirical selection of reference inputs solely based on the dataset being used would be the ideal scenario.

4.2.6 Interpretable Decision Sets Framework

Driven by the desire of building highly accurate, close to state-of-the-art machine learning models, while being highly interpretable at the same time, Lakkaraju, Bach and Leskovec (2016) developed a framework based on if-then rules that are claimed to provide both high accuracies while being interpretable. The authors give attention to the fact already discussed in this work that even though if-then rules are clear and explainable on their own, as rules are added and decision boundaries become *narrower*, these rules cease to be interpretable and become too complex to be understood. Concerning that, the proposed framework structures decision rules differently: instead of using hierarchical rules with if-then and else-ifs, Lakkaraju, Bach and Leskovec (2016) propose decision sets that can be applied independently, in any order; for the authors, reading rules that don't have a connection to other rules is key to interpretability, for them, each rule is an independent explanator.

In order to find the optimal decision sets, the framework balances rule complexity while maintaining both high accuracy and interpretability; to do so, it maximizes an objective function that contains terms for each metric it aims to optimize. It includes precision and recall to optimize accuracy and size (of the decision set), length, (how long are the rules) coverage (how many points in the dataset are covered by the rule), and overlap (of rules) to optimize interpretability. To maximize the objective function a smooth local search algorithm is applied to find a decision set that is a smoothed local optimum that satisfies and balances each term of the objective function.

The authors evaluate the framework with a set of different experiments; first, using different datasets for multi-class classification, they tested accuracy with parameters such as Area Under the Curve (AUC) and observed that decision sets framework performed almost as well as black-box decision tree algorithms (i.e. 75 vs. 77) such as Random Forest and Gradient Boosting Trees. To assess interpretability, Lakkaraju, Bach and Leskovec (2016) measured the same indicators used to optimize interpretability such as overlap, coverage, length, and the

number of rules, and compared the results with other rule-based methods; results showed that decision sets had more overlap than other methods but had more coverage and more importantly, the rules were shorter and fewer. Finally, the authors performed a user study with humans to understand how interpretable the framework is; more than forty university students which were familiar with concepts of logical structures and if-then rules tested the Decision Set Framework along with other rule-based classification methods. The students were asked some questions to evaluate their comprehension of the dataset and the classification decision-making, the results showed significant (81% vs 17%) more correct questions using the decision-set framework rather than other decision rule methods, they also took less time to answer the questions.

As for the limitations of the framework proposed, one can ask what happens when an attribute value x is not satisfied by any rule or satisfied by more than one. Lakkaraju, Bach and Leskovec (2016) claim that this occurrence was infrequently in their experiments (with 14% in the first case and 22% in the second); but when it happens, x is assigned to the most frequent class within the training dataset in the first case, and when satisfied by two rules, commonly the one assigned to x is the one with highest F1 score; however, the authors do not set this condition as mandatory and leave up to the users to choose other approaches. This lack of rigor might show a significant limitation of the framework since even though an attribute was not assigned to a rule in 14% of the cases in the experiments, it does not mean that this value will be the same in every scenario, especially in data with complex decision boundaries; and even if it does, that might be a high percentage depending on the sensitivity of the data being interpreted.

4.2.7 GAM: Global Attribution Mapping

Observing that most explanators are locally-based, explaining only single outputs, Ibrahim *et al.* (2019) were driven by that gap and developed Global Attribution Mapping (GAM), a global interpretability method that is capable of explaining representations of black-box models by grouping similar local feature importance to compose global explanations for different subpopulations.

To generate the attributions, GAM first starts with a set of local feature importance generated by other techniques; then, these are normalized to eliminate positive or negative influences and instead focus on similarities between the vectors. These vectors are then weighted and ranked by distance metrics such as Kendall's Tau or Spearman's Rho, measuring the squared rank distances between feature important information on each subpopulation. After that, the local attributions are clustered by similarity by detecting global patterns in them. The clustering algorithm adopted is the *k-medoids* which results in subpopulations that are similarly explained by the same feature importance levels.

To evaluate the technique, Ibrahim *et al.* (2019) conducted a few experiments. One of them consisted of finding global attributions on a real-world dataset. Since GAM required a prior calculation of local attributions based on existing methods, the authors adopted techniques such

as LIME and DeepLIFT, already reviewed in this survey. The results were feature importance on different populations, and the global importance levels achieved by GAM remained the same independently of the prior local explanator adopted. The feature presented by GAM also matched the features given by state-of-the-art global feature-relevance methods; however, the levels of those features, although proportionate, did not match the ones given by GAM. The authors also applied a user experiment asking for 55 people to remove the least important features that contributed to achieving a prediction, the fact that the majority of them were able to do so is hardly a relevant metric for evaluating user experience or the relevance of the method presented.

4.2.8 G-SHAP: Generalized Shapley Additive Explanations

Based on SHAP, Bowen and Ungar (2020) created a Generalized version of Shapley values explanator. According to the authors, it aims to answer additional questions than SHAP, such as what differs a prediction between groups of observations, what causes a sample to belong to a class rather than another, and why a model can perform poorly on a specific sample and not others.

G-SHAP calculation is similar to the original SHAP, with the exception of a set of Ω additional arguments, and extra calculations to achieve the three extra features mentioned above. In the first case, general classification explanations, G-SHAP's goal is to distinguish the feature characteristics between different classes, in other words, to answer which feature characteristics are typical of each class. G-SHAP does that by calculating the probability of every observation in a set of test samples belonging to one specific class and comparing it to background data with multiple classes, by doing that, G-SHAP is able to segregate different classes and observe the differences between them. The second extra feature is called intergroup difference, which shows the reason behind certain samples belonging to a certain class rather than the other; G-SHAP calculates that by having two observation groups (belonging to different target classes, the results are compared to determine which features are determinant for each group. The third feature is called model failure and it tries to identify why the model may fail on real-world data or on specific samples. G-SHAP compares the contribution of each feature j to the model performance on the training data to the same j feature to the test data. If j's feature contribution reduces between training and test, G-SHAP concludes that it is responsible for the poor test performance.

4.2.9 TabNet: Attentitive Interpretable Tabular Learning

Developed by Arik and Pfister (2019), TabNet is a high-performance neural network that focuses on tabular data and is capable of providing interpretable feature attributions at the same time. Driven by the lack of usage of neural networks for tabular data, which is mainly treated with ensemble decision trees, the authors argue that DNN for tabular data is relevant for several reasons, including data streaming, generative modeling, and integration with more complex data types such as images. TabNet learns with backpropagation similarly to any neural network and

uses gradient descent as an optimizer. Unlike most algorithms for tabular data, TabNet has a sequential attention mechanism that is capable of choosing meaningful features at each decision step separately from each input, this characteristic ensures that each instance is fed with the right features at each decision step, increasing interpretability and efficiency; at each step, the selected features contribute to a portion of the decision related to these features. TabNet's architecture is briefly described in the succeeding paragraph.

Initially, raw tabular data comprised of both numeric and categorical features are batch normalized and split in two different directions, one aims to provide the explanations and the other to reach the output value. The explanation process goes straight to feature selection by a sparse mask. This mask is calculated through sparsemax normalization that promotes sparsity. The other process also goes through the sparse mask but only after it has gone through a feature transformer block. This block has two steps, both containing a fully connected layer followed by batch normalization and a gated linear unit, at the end of the second block, the residual values from the first one are combined with the values obtained by the second step; after that, the data is split into a ReLU function, to be combined with each decision step to generate a final prediction, or into an attentive transformer block to generate predictions, which essentially consults prior scale information to consider how much of each feature has been used in previous decision steps. Since this network has N decision steps, each step also has the same set of blocks. As aforementioned, the final prediction is a combination of each decision step after the ReLU function while the explanations are generated by the aggregation of sparse masks. The explanations are both local and global as the masks show an image similar to the input data size, with rows representing instances and columns representing features, when a feature is significant, it is filled with a brighter shade of white, and when it's not significant it is filled with black; therefore, the authors claim that it can be visible and easy to understand which features were important to the whole dataset and specific to the prediction of each instance.

Arik and Pfister (2019) performed a series of experiments to evaluate interpretability and classification/regression performance on TabNet. Initially, the experiments consisted of a comparison between TabNet and other models that present explicit feature selections in 6 datasets, TabNet obtained satisfactory results, reaching the best performance in half of the datasets tested. However, the main advantage shown by this study is that even though TabNet accomplished performance scores similar to other networks, it did so with a notably smaller number of hyper-parameters (43k vs. 26k, depending on the dataset and networks compared); in fact, the authors claim that differences in hyperparameters did not significantly influence the results, showing what they also claim to be one of the most remarkable characteristics of TabNet. Preeminently, TabNet outperforms high-performance algorithms for tabular data such as ensemble decision trees and other neural networks tested on four different datasets. When evaluating interpretability, one of the experiments done by the authors was testing the global feature importance given a certain dataset that a single feature was enough to correctly predict more than 98% of the samples; the feature importance attributed to this feature by TabNet was

higher than by other methods such as Lime and Deeplift. Overall, interpretability evaluation appears deficient since no experiment was able to efficiently evaluate this attribute; besides, a user study could show significant relevance when evaluating interpretability once human interpretation is the main goal of interpretable models.

4.3 Analysis of Chosen Methods

Through the brief reviews of interpretability techniques that are shown in the previous section, it is practicable to narrow down the most relevant for a more in-depth analysis. As it could be noticed, how each explainability method delivers the explanations differs according to how it was designed. For a more inclusive approach, the selected techniques should work as post-hoc outcome explanators, along with providing their explanations locally and most importantly, being feature attribution methods. This last property allows us to understand how each feature contributes to the prediction, feature attribution methods are comprehensible and widely adopted, and the interpretation of feature contribution is independent of the technique in consideration.

For those reasons, a chosen method for closer analysis and assessment is also the most popular one, with the most referenced papers, LIME. Another relevant method is SHAP, which uses Shapley values to measure feature contribution for both local and global explanations, allowing it to be considered both an outcome explanator and model inspector; these characteristics make SHAP stand out from most of the other tools, for that reason, that was also a chosen technique.

Primarily, most techniques presented provide interpretability by post-hoc explanations, amongst the post-hoc methods, most of them are outcome explanators with local explainability. A closer inspection allows noticing the majority of the work by perturbing the inputs in order to perceive the decision boundary allowing to explain how changes in the input would affect the output. Another important filter between techniques is feature attribution, which presents the explanation in form of feature importance to the model. For those reasons, a chosen method for a closer analysis is the most popular one, with the most referenced papers, LIME. Another relevant method is SHAP, which uses Shapley values to measure feature contribution for both local and global explanations, allowing it to be considered both an outcome explanator and model inspector; these characteristics make SHAP stand out from most of the other tools, for that reason, that was also a chosen technique.

The analysis consists of a qualitative evaluation applying the chosen techniques on tabular data and comparing how they present the explanations, the consistency between each explanation, and usability. Concerning the data used to compare the techniques, choosing a particular dataset calls for the right balance between complexity while still being able to be interpretable and evaluated by humans; in that sense, the data selected was the Wine dataset,

Figure 3 – Sample Explanation with LIME

(a) Prediction Probabilities (b) Explanations (c) Sample Features

Source: Elaborated by the author.

originally published by Cortez *et al.* (2009). This dataset summarizes the result of a chemical analysis of wines made with grapes grown in the same region in Italy but derived from three different cultivars. The analysis determines the quantities of 13 constituents found in each of the three types of wine; in other words, it is a classification problem.

To perform a qualitative comparison between tools, it is important to set the same threshold in every test; if possible (in the first two techniques), the same classifier should be used and the tools should locally explain the same example. Considering that, an Extreme Gradient Boosting Tree classifier with a hundred estimators was created, these estimators represent the number of trees used to classify Wine's three possible classes. In that sense, an *XGBoost* model is a complex system that has (in this example) 100 trees to discern from, being impossible for humans to reason about its decision-making process.

4.3.1 LIME

LIME, as described in the survey, searches the neighborhood of samples to fit a linear model and extracts information on how samples are classified. Figure 3b shows LIME's reasoning on why the selected sample belongs to the predicted class along with counterfactual information on the reasons why it might not belong to that class. As can be seen, several feature values are shown and some of the bars are higher on the left side (that represents the reasons why the sample does not belong to class 0). In that sense, it can be difficult to draw significant conclusions. Moreover, Figure 3c shows in green, actual feature values present in the sample that contributed to predicting the class given; and in teal, feature values that counterfactually contributed to the prediction.

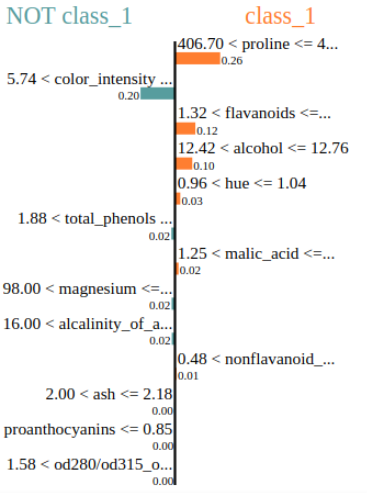
Overall, to positively interpret the results given by LIME, one has to look at both Figure 3b and Figure 3c. and compare each actual feature value from the table to the explanation given. For instance, the first feature *proline* has a value of 450, the explanation shows that if *proline* is less than 505.5, it is a feature value typical of class 1. Predominantly, the four main features selected by LIME to explain the output value were proline, color intensity (counter explanation), flavonoids, and malic acid. As it can be seen; LIME, while being simple to use and its explanations though are thoroughly provided, might not be considered intuitive and simple to understand, a good balance between what was most relevant to obtain the final prediction instead of all features would be an interesting improvement that could show clearer and less confusing explanations.

Furthermore, by analyzing LIME's available source code, a few interesting points can be noted. For tabular data with continuous variables such as the one adopted to generate the examples shown, LIME discretizes the data into quartiles by default, this results in LIME not being able to differentiate patterns within the same quartile. A solution for this, already provided by LIME, would be to discretize the data into more bins or simply use continuous variables; however, these do not solve the issue: LIME uses local linear models to fit the data, in that way, many of them would fit the data with different intercepts which would result in LIME choosing an aleatory model to explain the data; on the other hand, when discretizing into more bins, the same effect would happen, but with a less extent, note in Figure 4 how the explanations for the same sample change when simply running LIME twice.

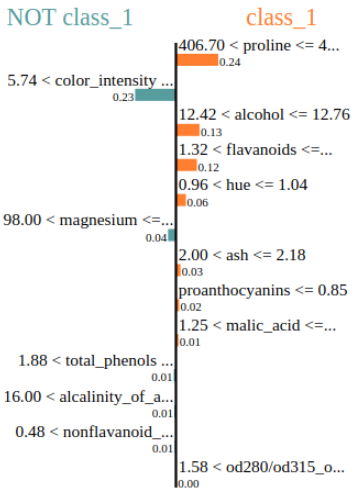
Figure 4 shows that even though the top two explanations remained the same, other feature values can even change from being an explanator of a class to not being associated with that class as is the case of *ash*, that appears on the left side of the explanation (not class 1) on the first run, and in the right side (class 1) on the second run. Note that the actual explanation values of the top features also change when compared to the explanation provided when discretizing into quartiles as shown in Figure 3b. These issues can be considered a substantial drawback of LIME; nevertheless, since LIME works as a local surrogate model, it can be expected that the explanations are an approximation of the real values, providing a general idea of the black-box model. In this context, it turns out to be up to the user to determine whether the model can be *trusted* or not.

Furthermore, as mentioned in subsection 4.2.1, LIME uses a determined kernel size to search the neighborhood of a sample, according to the source code, unless defined, kernel size is set to default: the square root of the number of columns times 0.75. It is unclear how the authors came up with that kernel size, it could make sense that kernel size should be learned from the data itself rather than being a parameter set by the user. However, once LIME is a post-hoc explanator that, for instance, can explain a single observation, learning from a set of only one array of values might not be the ideal solution to set a kernel size.

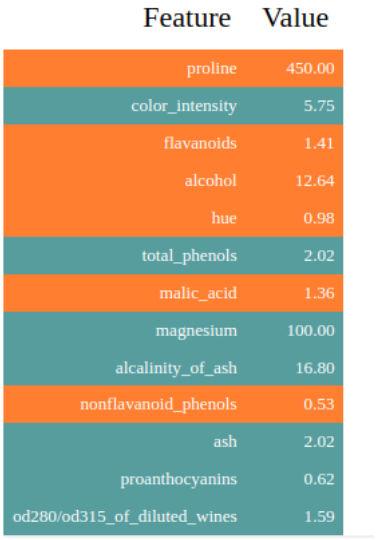
Figure 4 – Divergences in LIME's explanation



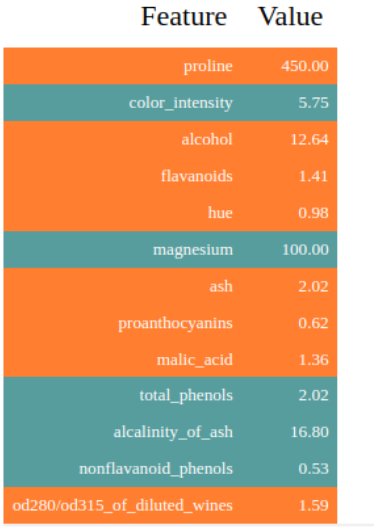
(a) Explanation - 1st Run



(b) Explanation - 2nd Run



(c) Sample Features - 1st Run



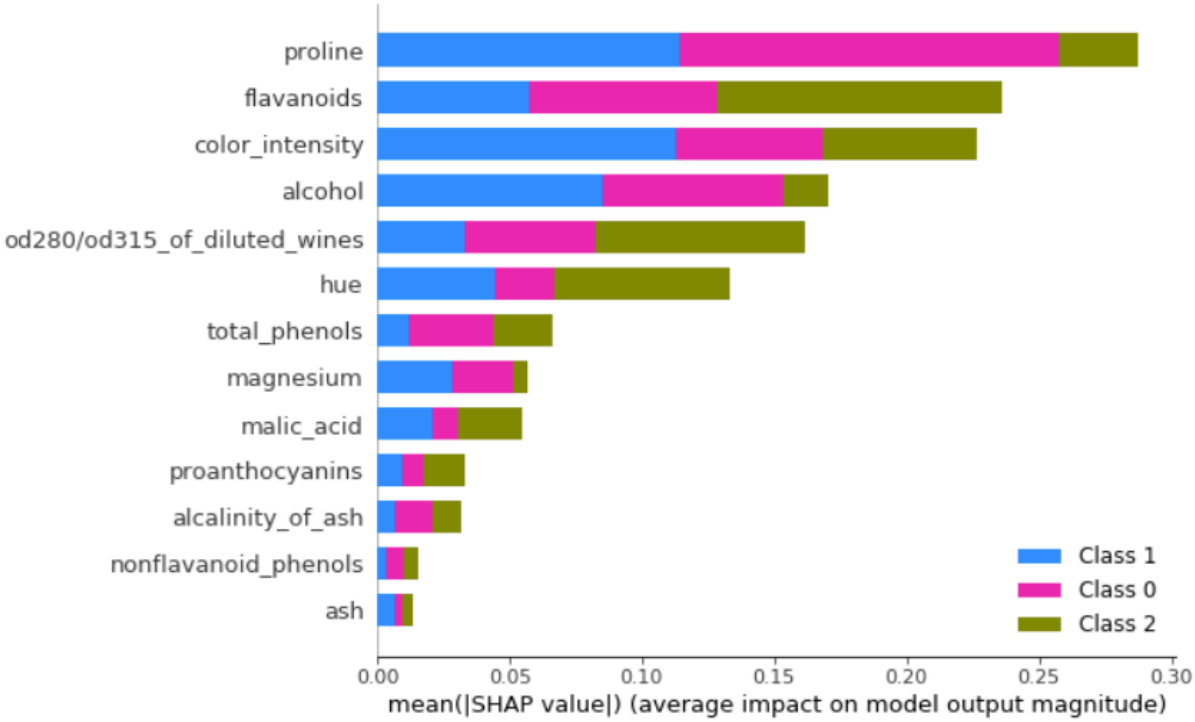
(d) Sample Features - 2nd Run

Source: Elaborated by the author.

4.3.2 SHAP

As shown in the survey, SHAP uses Shapley values to measure the contribution of each feature to predictions, also it is capable of providing both local and global explanations. SHAP's global explanation of the whole Wine dataset can be seen in Figure 5, the goal is to provide a *Model Inspection* approach that permits comprehension of feature contributions to a model. The graph presents horizontal bars for each feature, showing SHAP values that contribute to the model outcome; the bars are also stacked on the contribution for each class. Regarding that, it can be said that SHAP provides a good overall global explanation, demonstrating what is contributing more or less to the model outcome, and which classes are more impactful within that feature.

Figure 5 – SHAP Global Explanations for Wine Dataset



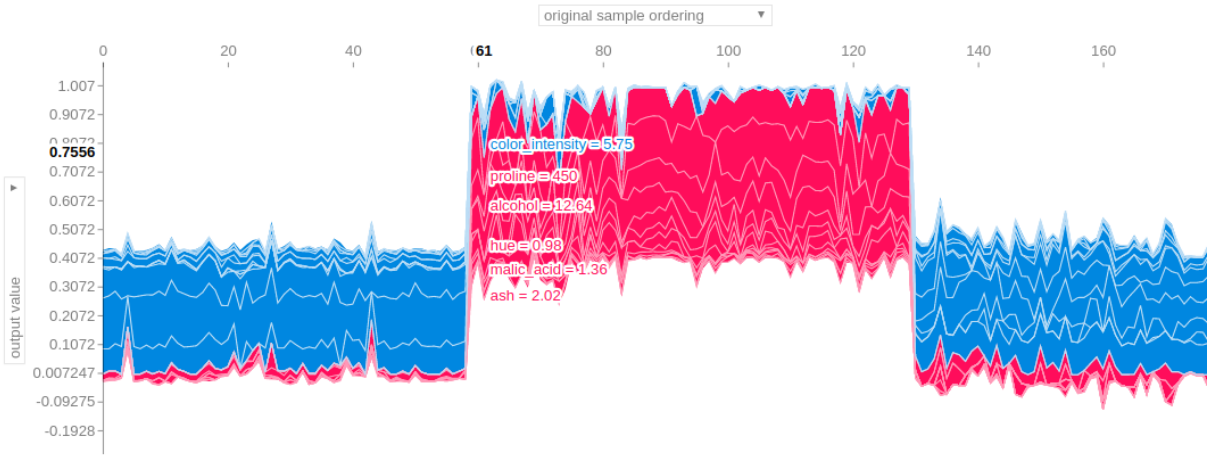
Source: Elaborated by the author.

However, stacking more than two categories might not be the ideal visualization for perceiving the right proportion, let's take the example of alcohol and flavonoids in Figure 5, comparing the contribution for class 0 (in pink) is barely possible for those features since they seem almost the same length. At the same time, even though Figure 5 provides a good explanation of features that does not help in understanding model behavior or how features push towards a certain outcome. To do so, SHAP provides a global/local explanation with positive and negative feature contributions for each sample. In regression problems, there is only one visualization that provides full information; on the other hand, for classification problems, SHAP is able to show different visualizations for each class. This is necessary because lower or higher feature values can contribute differently (i.e. positively or negatively) for each class. Figure 6 depicts global explanations for the class that belongs to the same example used when analyzing LIME's local explanation.

The plot displays different values and colors in red and blue, red push the output value up and blue push it down; note that this visualization shows information for class 1 only, which means that any samples belonging to classes 0 and 2 will have an output value close to 0 (not class 1). From this perspective, it can be noticed how specific classes respond to each feature as well as observing how each feature will affect the result and the proportion it will do so by how thick the feature block is.

SHAP also has a different approach to showing global explanations with contributions

Figure 6 – SHAP Global Explanations with Positive and Negative Feature Contribution



Source: Elaborated by the author.

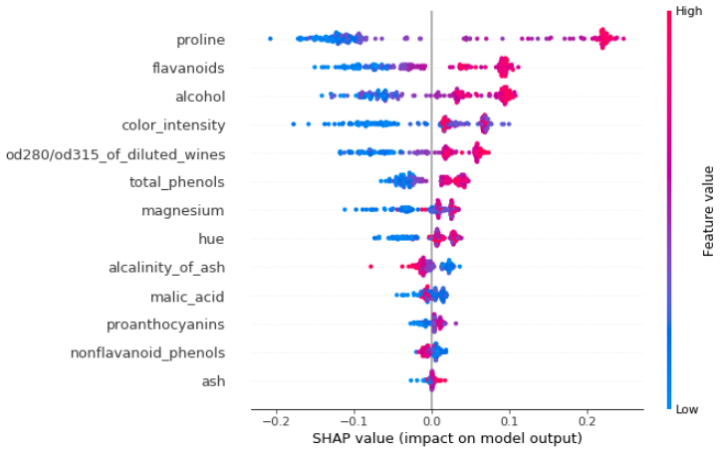
for each feature and class as shown in Figure 7. Essentially, what it allows us to observe is how a continuous feature value will impact the outcome when it is high or low; for instance, high values of proline are contributors for the model to predict class 0, whereas lower values are typical of classes 1 or 2. Because of the clear relation and understanding of classes and feature values, this is actually a major contribution to understanding model behavior and helps to increase *trust*.

As for SHAP's individual outcome explanation, Figure 8 shows a plot on one example of the dataset where the outcome is 0.76, belonging to class 1, the feature values in red are typical of class 1, so they push the prediction higher from a determined baseline; on the other hand, the feature values in blue are not typical of class 1, so they tend the prediction downwards. In summary, each feature contributes positively (red) or negatively (blue) to changing the outcome from a certain baseline to the obtained output.

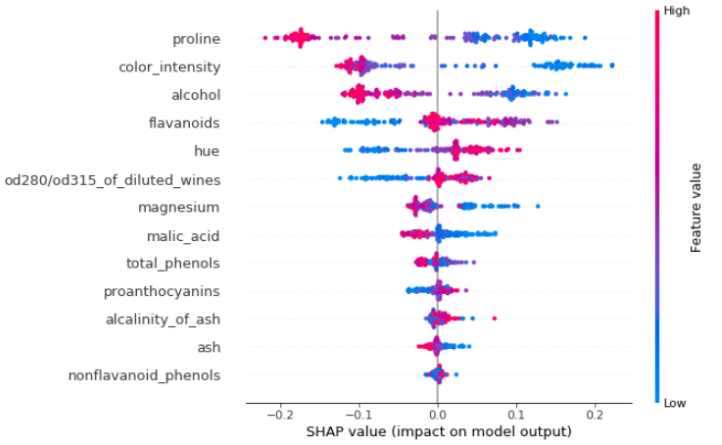
It is important to note how Figure 8 matches the feature contribution to each class as in Figure 5. For instance, Figure 5 portrays that proline, color intensity, and alcohol are claimed the main impactful features for class 1 (in blue), correspondingly, the local individual explanation depicted in Figure 8 shows those same features when explaining that the sample belongs to class 1.

Overall, SHAP provides several ways of visualizing explanations and understanding contributions to the output. By analyzing all or most of the explanations provided, one can thoroughly grasp both global and local explanations and make sense of the behavior of a black-box model.

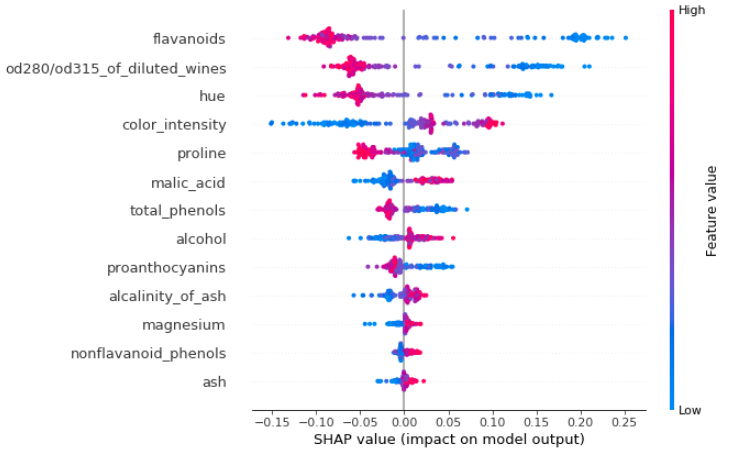
Figure 7 – Class Positive/Negative Influence



(a) Class 0



(b) Class 1



(c) Class2

Source: Elaborated by the author.

Figure 8 – SHAP Individual Explanation Example

Source: Elaborated by the author.

CHAPTER

5

FORMULATION AND EXPERIMENT

Traditionally, systems can be measured qualitatively and quantitatively, and interpretability techniques do not differ. However, qualitative metrics can be subjective to each user/person's own perceptions and usually require case studies and crowd-sourcing user experiments. Consequently, the emphasis of this study is to focus on existing quantitative metrics and propose a comprehensive guideline to evaluate post-hoc interpretability methods concerning structured tabular data.

5.1 Guidelines - Formulation

This work aims to establish common quantitative metrics to measure explanations provided by different interpretability techniques. Based on Table 2, the metrics that can be considered quantifiable are Stability, Identity, and Faithfulness. In the following subsections, formulations of these metrics are presented, which can be adopted into different datasets, serving as a guideline on how to measure different outcomes of feature attribution interpretability methods.

5.1.1 Faithfulness

Faithfulness identifies if features that are important to the explanation method are also important to predicting an instance using the original model. Quantifying faithfulness can be done in different ways, once it can be prone to interpretation. However, it should always be capable of answering the following question: are the features that are truly important to define the result also important to the original model? As is the case of this work, if the true relevant features are known, it becomes simpler to compare the feature importance provided by the explanator and the true relevant features. In other settings, calculating faithfulness can be done using a proxy notion of faithfulness, as suggested by (ALVAREZ-MELIS; JAAKKOLA, 2018).

To calculate faithfulness, an adaptation of Jaccard's similarity index was used. The formulation can be described as follows.

Suppose there are ten different continuous features, the first two were designed to represent the classes, the next four features are composed of a linear combination of the two relevant features, and the remaining four features are irrelevant noise. Denote by $A_1 = \{1,2\}$, $A_2 = \{3,4,5,6\}$ and $A_3 = \{7,8,9,10\}$, the index set of the three group of features and let $B_1 = \{b_1, b_2\}$ be the indices of the two most relevant features provided by an interpretability method X, $B_2 = \{b_3, b_4, b_5, b_6\}$ the next four relevant features, and $B_3 = \{b_7, b_8, b_9, b_{10}\}$ the indexes of the four least relevant features. The faithfulness measure of X method can be defined by Equation 5.1.

$$F_(X) = \alpha_1 \frac{|A_1 \cap B_1|}{|A_1 \cup B_1|} + \alpha_2 \frac{|A_2 \cap B_2|}{|A_2 \cup B_2|} + \alpha_3 \frac{|A_3 \cap B_3|}{|A_3 \cup B_3|}, \sum_i \alpha = 1 \tag{5.1}$$

Where $F_{(X)} \in [0,1]$, 0 represents no match and 1 represents a perfect match and faithfulness to the true relevant features. The weights α_i can be tweaked to alter the importance of each feature group.

5.1.2 Stability

As previously reviewed, stability can be comprehended as the effect caused by the explanation for similar inputs. In other words, it measures how greatly perturbations affect the explanation given by the interpretability method. Nogueira, Sechidis and Brown (2018) claim that stability is linked to robustness, as a stable method is more robust and consequently more trusted by the user. The authors state that measuring stability can be challenging due to a vast amount of metrics proposed by different authors throughout the years. In this work, instead of proposing a new one or an adaptation of existing metrics, a robust similarity metric is suggested: the cosine similarity.

Suppose the interpretability method X provides feature attribution values for each of the features available for each input sample, given by $O_n = \{1,2,3,...,10\}$. If the input samples are somehow perturbed, X will result in a also perturbed vector of feature attributions given by $P_n = \{1,2,3,...,10\}$. Quantifying this perturbation can be performed as described in Equation 5.2, with the dot product of feature attribution vectors by the multiplication of the two norms. $St_{(X)} \in [0,1]$.

$$St_(X) = \frac{\overrightarrow{O_n} \cdot \overrightarrow{P_n}}{\|\overrightarrow{P_n}\| \times \|\overrightarrow{O_n}\|} \tag{5.2}$$

5.1.3 Identity

Identity is proposed by Honegger (2018) in an axiom to assess explainability. For them, identity assesses if identical objects also have identical explanations; that is, if two samples are precisely the same, the feature attribution values given by the interpretability method should also be the same. Calculating this property is relevant since when an explanation is consistent and shows little to no sign of randomization, it promotes trust in the user. In sum, the goal of identity is to identify undesired randomization in the resulting explanations.

Honegger (2018) suggest that identity can be calculated by exposing the same sample more than twice to the explanation method and comparing the results. It can be calculated using the same metric as in Stability: the cosine similarity, as described in Equation 5.3.

Let χ_a be the original sample vector drawn to be tested and χ_b the duplicated sample. If the cosine similarity between these two vectors is absolute, the cosine similarity between the feature attribution values ε_a resulted from the original sample vector and the feature attribution values from the duplicated vector ε_b, ideally, should also be one. Observe that $Id_{(X)} \in [0,1]$.

$$\frac{\overrightarrow{\chi_a} \cdot \overrightarrow{\chi_b}}{\|\overrightarrow{\chi_a}\| \times \|\overrightarrow{\chi_b}\|} = 1 \Longrightarrow Id_(X) = \frac{\overrightarrow{\varepsilon_a} \cdot \overrightarrow{\varepsilon_b}}{\|\overrightarrow{\varepsilon_a}\| \times \|\overrightarrow{\varepsilon_b}\|} \tag{5.3}$$

5.2 Experimentation

5.2.1 Experiment Setup and Methodology

With the three explainability metrics established, they can be applied to identify the best interpretability method for a specific problem or find the best parameters within that method to reach explanations that pose more trust to users.

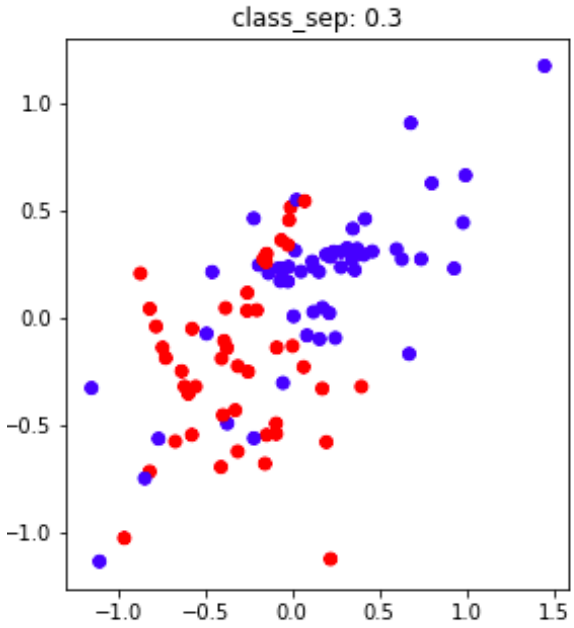
The experiment aims to find the best parameters for a specific synthetic classification problem that can be adapted to other real-world data. This experiment consists of three parts.

- Part 1: Data preparation and models
- Part 2: Explaining outcomes and calculating metrics
- Part 3: Identifying best parameters and comparing results

In order to apply and evaluate interpretability metrics, a sample dataset is crucial. For that, an artificial dataset is generated using Python's NumPy and scikit-learn libraries. The usage of synthetic data is two-fold: it allows one to grasp the metric efficiency while evaluating the interpretability technique at the same time. The synthetic data is generated for a binary classification problem, containing 10 numeric variables and 100 samples. Two out of ten features are set to be determinants for which class a sample belongs. Four other features are generated as

a linear combination of the two relevant features. The four remaining features are generated as noise, being considered redundant features, without any correlation to the output value. A visual data distribution and spatial separation between the synthetic samples Figure 9.

Figure 9 – Synthetic Data Distribution



Source: Elaborated by the author.

With the data generated, the next step is to create a black-box classification model. To do so, a 70/30 split into train and test data to train a Random Forest model. After that, The explanations were generated using Lime, Shap, and G-Shap. G-Shap was selected as a third contender due to its model-agnostic feature attribution method; however, this technique does not include visual techniques as the other due, returning only vectors with feature contributions of either a single instance or the overall dataset, for that reason it was included here rather than primarily in the survey section.

Each interpretability technique has its own parameters that can be tuned to find the best explanations. Note that the parameters selected here are limited to the explainer of each method, parameters that are related to calculating the explanation of each sample were ruled out of this experiment. Starting with Lime, this technique possesses specific modules for different approaches, such as tabular, image, or text. In the case of this work, the focus is on tabular data, because of that, LIME's tabular module is considered. The tabular module has a series of parameters that can be fine-tuned to reach explanations. In this work, three were considered: *kernel width, discretizer*, and *sample around distance*. *Kernel width* is related to the kernel size that Lime draws samples from, the default value is the $sqrt(numberof features) * 0.75$; so in this work, other kernel width values will be sampled arbitrarily around the default value. The second parameter, *sample around distance* is a binary parameter that sets if samples are drawn

around the instance being explained or on the mean of the feature data. Lastly, discretizer is a parameter that sets if continuous variables should be discretized into quartiles, deciles, or an entropy-based discretization. It is also possible to not discretize at all, so this approach will also be considered. SHAP on the other hand does not allow parameter tweaking other than the algorithm used to estimate Shapley values and a link function (in this case, only one of the two possible values is applicable); because of that, *algorithm* is the only parameter that is being altered in SHAP's experiment. This parameter allows eight different inputs. Lastly, G-SHAP has two tweakable parameters, one is the g function to calculate Shapley values, as default, Shapley values are calculated with the mean function, and other functions were not capable of running the experiment or did so with errors. The other parameter is the number of samples to draw from: the largest, the more replicable the experiment is, but also more time-consuming.

The list below presents an overview of all settable parameters and the values we adopted for each of them.

1. LIME
    a) Kernel width
        i. 0.7
        ii. 1.2
        iii. 1.8
        iv. 2.3
        v. 3.0
        vi. 3.7
    b) Discretizer
        i. Decile
        ii. Quartile
        iii. Entropy
        iv. No discretization
    c) Sample Around Distance
        i. True
        ii. False
2. SHAP
    a) Algorithm
        i. Permutation
        ii. Partition
        iii. Tree

iv. Kernel
v. Sampling
vi. Linear
vii. Deep
viii. Gradient

3. GSHAP

a) Samples

i. 10
ii. 25
iii. 50
iv. 75
v. 100
vi. 125
vii. 150

Using every possible parameter value for each interpretability technique, the subsequent step is to calculate faithfulness, stability, and identity for every sample in the dataset. The goal is to identify the best parameters for each interpretability technique and then point out the ideal explainability method for the specific problem. To grasp the overall result rather than comparing the result for each sample, central tendency measures are calculated for the dataset being tested. To present the results and visualize them according to the parameters set, an adaptation of Ludwig Wittgenstein's logic Truth Table is used. Afterward, the best parameters are visually compared with distribution visualization techniques such as box plots. Ultimately, this analysis can be performed to select an ideal interpretability technique according to a specific problem and consequently set the parameters to meet certain interpretability goals and properties or simply pose more trust to the user.

5.2.2 SHAP's Results

SHAP has a single tweakable parameter: *algorithm*; however, calculations in Table 4 show that choosing any of the available values does not alter the explanation values at all. Interestingly, SHAP reaches maximum identity values in every sample in the dataset, showing that explanations are highly consistent independently of how many times it is requested. Stability values are mostly on the higher end of the range as can be seen on Figure 10c. Faithfulness is where SHAP falls short, with a mean of 0.57, Figure 10a shows the faithfulness range, reaching either maximum values or values as low as 0.2.

Table 4 – Shap results

Algorithm	Faithfulness		Stability		Identity	
	Mean	Std	Mean	Std	Mean	Std
Permutation	0.57	0.25	0.85	0.11	1.00	0.00
Partition	0.57	0.25	0.85	0.11	1.00	0.00
Tree	0.57	0.25	0.85	0.11	1.00	0.00
Kernel	0.57	0.25	0.85	0.11	1.00	0.00
Sampling	0.57	0.25	0.85	0.11	1.00	0.00
Linear	0.57	0.25	0.85	0.11	1.00	0.00
Deep	0.57	0.25	0.85	0.11	1.00	0.00
Gradient	0.57	0.25	0.85	0.11	1.00	0.00

Source: Research data.

Figure 10 – SHAP's boxplot distribution of Faithfulness, Identity and Stability values

(a) Faithfulness (b) Identity (c) Stability

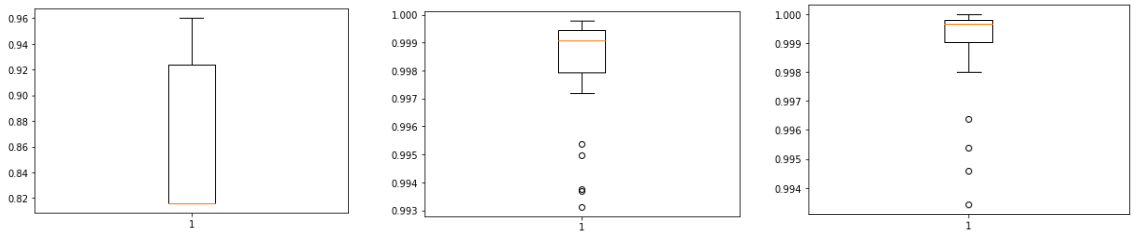
Source: Elaborated by the author.

Faithfulness calculates if true relevant features are also important to predicting the model, that is, the real important features are identified by the explainability method as the most relevant features. Calculating faithfulness is straightforward when the real important features are known to the user, but proves far more complex when these features are unknown. For simplicity, the calculations presented here are performed on a synthetic dataset, with known feature properties. The formulation requires α weight values for the most relevant features, the linear combined features, and the redundant features. These values are inputs to the function and can be altered by the user; here, the weights used were 0.7, 0.2, and 0.1 respectively, giving more importance to the most relevant features.

5.2.3 LIME Results

LIME's results are considerably more complex to interpret than SHAP's. Lime's Kernel size dictates the neighborhood size to draw samples. The default value proposed by the authors is as aforementioned, the $sqrt(nfeatures) * 0.75$; which is approximately 2.3. Inserting different kernel sizes above and below this standard value allows observing that indeed the default value seems the most appropriate, reaching higher values for every metric, independently of other parameters, with very few exceptions. Regarding the discretizer options, better results are often

Figure 11 – LIME's boxplot distribution of Faithfulness, Identity and Stability values



(a) Faithfulness (b) Identity (c) Stability

Source: Elaborated by the author.

reached with quartile discretization or no discretization at all. In terms of sampling, in most cases, it makes little to no difference whether the samples are drawn around the instance or on the mean of the feature data, with an exception when in combination with continuous discretization set to False. When combining sampling around the mean of the feature data with no discretization, the highest values for faithfulness, stability, and identity are reached, as can be seen with the bold option in Table 5. For that specific setting, the distribution can be seen in Figure 11, with low variation in Identity and Stability and more variation in Faithfulness.

5.2.4 G-SHAP Results

As visible in Table 6, G-SHAP returns stronger results as the amount of drawing samples increase; however, the rate as each metric improves decreases as the samples increase, following a logarithmic pattern. Most importantly, the processing cost of higher sample values can be limiting; for instance, with a determined hardware, calculating identity for the test set with 25 samples took less than 2 minutes, whereas the calculation with 150 samples took over 20 minutes. With that being said, as can be seen in Table 6 better results are reached with more samples for faithfulness and Identity. Interestingly, stability results were better with 75 samples. Note that the rate of how faithfulness values increased was smaller as the sample size went up. When considering the boxplot distributions of the test results with 75 samples (selected as the best balance between results and computational cost) shown in Figure 12 it can be noted that Identity is highly consistent and does not variate like faithfulness and stability. Note that although Faithfulness has a mean of 0.59, there are some samples that plummet to less than 0.2.

5.2.5 Comparisons and Considerations

Asserting which interpretability method reaches better explanations is complicated. Defining a "good" explanation is highly related to the application and what the user targets to reach. Metrics as proposed by the authors mentioned in this work aim to gauge specific interpretability properties that can be deemed less or more important depending on the application.

Table 5 – LIME results

			Faithfulness		Stability		Identity	
Kernel	Discretizer	Sampling	Mean	Std	Mean	Std	Mean	Std
0.7	Decile	True	0.55	0.25	0.74	0.32	0.95	0.17
0.7	Decile	False	0.55	0.26	0.72	0.34	0.95	0.17
0.7	Quartile	True	0.68	0.21	0.68	0.31	0.87	0.32
0.7	Quartile	False	0.67	0.21	0.68	0.31	0.87	0.32
0.7	Entropy	True	0.69	0.20	0.60	0.33	0.81	0.38
0.7	Entropy	False	0.65	0.21	0.52	0.33	0.81	0.38
0.7	None	True	0.59	0.26	0.76	0.25	0.91	0.15
0.7	None	False	0.57	0.29	0.72	0.27	0.72	0.39
1.2	Decile	True	0.69	0.20	0.59	0.28	0.91	0.12
1.2	Decile	False	0.69	0.20	0.56	0.33	0.91	0.12
1.2	Quartile	True	0.76	0.18	0.72	0.28	0.97	0.07
1.2	Quartile	False	0.76	0.18	0.62	0.29	0.97	0.07
1.2	Entropy	True	0.71	0.21	0.62	0.30	0.95	0.21
1.2	Entropy	False	0.71	0.21	0.62	0.30	0.94	0.18
1.2	None	True	0.62	0.26	0.82	0.21	0.98	0.05
1.2	None	False	0.79	0.17	0.95	0.12	0.98	0.06
1.8	Decile	True	0.70	0.19	0.67	0.27	0.89	0.32
1.8	Decile	False	0.70	0.19	0.63	0.26	0.87	0.33
1.8	Quartile	True	0.77	0.16	0.69	0.26	0.96	0.07
1.8	Quartile	False	0.77	0.16	0.69	0.26	0.96	0.07
1.8	Entropy	True	0.72	0.21	0.56	0.20	0.84	0.25
1.8	Entropy	False	0.72	0.21	0.51	0.34	0.84	0.25
1.8	None	True	0.66	0.25	0.82	0.24	0.91	0.29
1.8	None	False	0.84	0.05	0.99	0.01	0.99	0.01
2.3	Decile	True	0.70	0.19	0.67	0.26	0.87	0.30
2.3	Decile	False	0.70	0.19	0.63	0.26	0.87	0.30
2.3	Quartile	True	0.77	0.16	0.69	0.25	0.94	0.12
2.3	Quartile	False	0.77	0.16	0.59	0.28	0.92	0.14
2.3	Entropy	True	0.72	0.20	0.58	0.32	0.88	0.20
2.3	Entropy	False	0.72	0.20	0.58	0.32	0.84	0.26
2.3	None	True	0.67	0.24	0.83	0.24	0.95	0.15
2.3	**None**	**False**	**0.85**	**0.06**	**0.99**	**0.01**	**0.99**	**0.01**
3.0	Decile	True	0.70	0.19	0.64	0.25	0.87	0.30
3.0	Decile	False	0.70	0.19	0.64	0.25	0.87	0.30
3.0	Quartile	True	0.77	0.16	0.69	0.25	0.94	0.12
3.0	Quartile	False	0.77	0.16	0.59	0.28	0.94	0.12
3.0	Entropy	True	0.72	0.20	0.58	0.32	0.90	0.17
3.0	Entropy	False	0.72	0.20	0.58	0.32	0.90	0.17
3.0	None	True	0.67	0.24	0.82	0.26	0.94	0.17
3.0	None	False	0.85	0.06	0.99	0.01	0.99	0.01
3.7	Decile	True	0.70	0.20	0.64	0.25	0.87	0.30
3.7	Decile	False	0.70	0.20	0.69	0.27	0.87	0.30
3.7	Quartile	True	0.77	0.16	0.69	0.25	0.92	0.14
3.7	Quartile	False	0.77	0.16	0.69	0.25	0.94	0.12
3.7	Entropy	True	0.72	0.20	0.58	0.32	0.95	0.12
3.7	Entropy	False	0.72	0.20	0.52	0.34	0.92	0.16
3.7	None	True	0.66	0.25	0.84	0.21	0.96	0.10
3.7	None	False	0.84	0.05	0.99	0.01	0.99	0.01

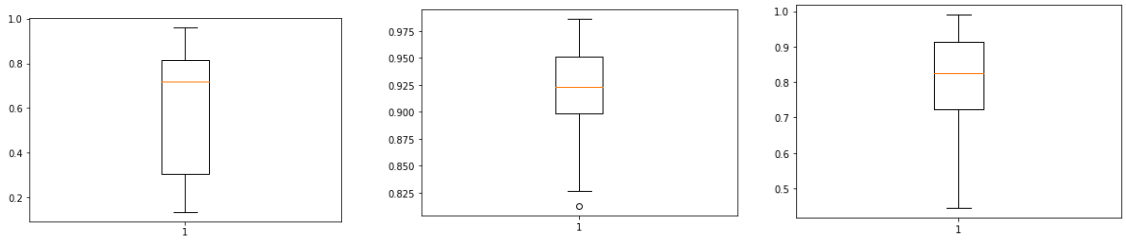
Source: Research data.

Table 6 – G-Shap results

Samples	Shapley	Faithfulness		Stability		Identity	
		Mean	Std	Mean	Std	Mean	Std
10	Mean	0.42	0.22	0.60	0.17	0.58	0.21
25	Mean	0.47	0.21	0.70	0.12	0.79	0.13
50	Mean	0.49	0.26	0.77	0.11	0.89	0.06
75	Mean	0.58	0.23	0.90	0.05	0.90	0.06
100	Mean	0.59	0.25	0.75	0.09	0.94	0.04
125	Mean	0.59	0.26	0.85	0.11	0.96	0.04
150	Mean	0.59	0.26	0.88	0.08	0.97	0.02

Source: Research data.

Figure 12 – G-SHAP's boxplot distribution of Faithfulness, Identity and Stability values



(a) Faithfulness (b) Identity (c) Stability

Source: Elaborated by the author.

With that being said, different experiments might want to focus on reaching higher levels of one metric over the other. Faithfulness reached higher values with Lime, but the scores varied according to the settings (Kernel size, discretizer, and sampling), ranging from 0.55 to 0.85. G-SHAP and SHAP had somewhat similar faithfulness results, ranging from 0.42 to 0.59 and stable at 0.57 respectively. Stability values ranged from 0.52 to 0.99 in LIME, 0.6 to 0.88 in G-Shap, and a stable 0.85 on Shap. Identity values reached maximum values in Shap, a range from 0.72 to 0.99 in Lime and 0.58 to 0.97 in G-Shap.

Analyzing each interpretability method separately, SHAP reached high levels of stability and identity but it did not reach the best faithfulness values compared to some of LIME's results. Notwithstanding, LIME only reaches high faithfulness, stability, and identity values when disabling discretization and setting the perturbation samples to be drawn around the mean of feature values. G-SHAP, on the other hand, was able to reach slightly better faithfulness results than SHAP depending on the number of samples, but it fell short on stability and identity.

Note that differently than SHAP, LIME consistently hits higher faithfulness values. Conversely, stability and identity values are typically lower in LIME, independently of kernel size and sampling, only reaching values similar to SHAP's when not discretizing continuous attributes. This observation goes back to the argument that different experiments can emphasize

different interpretability metrics, that is, researchers might want to prioritize methods that are capable of reaching better stability even if it means that faithfulness is somewhat sacrificed and vice-versa.

Different interpretability methods may meet certain explainability properties in certain ways that differ according to each method. Also, their behavior can be altered by the data being analyzed. Calculating faithfulness, stability and identity also allow mapping the best parameters to set in the interpretability technique to reach the results that meet certain criteria. As a toy example, this work proposed a synthetic classification problem with continuous variables. It is not expected that the results shown here are consistent regardless of the data being analyzed. It is up to the user to apply such metrics in their specific experiments and observe how the interpretability techniques meet metrics and properties; certain sensitive experiments may allow sacrificing faithfulness to maximize stability for instance. This experiment presents a guideline on how to implement these three metrics to analyze and assess interpretability methods, allowing to determine the best one for a specific experiment or the best parameters to explain instances or dataset patterns.

CHAPTER

6

CONCLUSION

6.1 Final Considerations

Black-box algorithms are known for their great generalization and representation capabilities, being increasingly used in several areas of knowledge that demand the use of Machine Learning and Artificial Intelligence applications. However, as the demand for high-performance models rises, so does the demand for knowledge of how these models work. Throughout this work, it has been discussed that black-box models cannot explain the reason behind a certain prediction, which opens room for interpretability methods. These methods are capable, through different approaches, to reach a certain level of explanation on a specific result or how the model behaves. It has also been discussed that as the data of the publication of this work, there is no consensus on how these methods should be assessed or compared due to factors such as the novelty of the technology and the challenge that different methods and the different ways they deliver the explanations pose. As a result, this lack of specificity leaves room for research, allowing authors to propose interpretability assessment frameworks and metrics. Important to note however that this is a first step in the effort of explaining and comparing different interpretability techniques rather than a breakthrough framework, further research is needed to allow a deeper comprehension of these methods.

This work reviewed many concepts and intuition behind interpretability and how it is defined, presenting a taxonomy based on this review. This work also presented a survey on current interpretability methods, how they were developed and how they deliver explanations. This work focused on the most common and established one: post-hoc explanators that explain outcomes based on samples or whole datasets when the black-box model has already been deployed. Along with post-hoc explanators, this work concentrated on model-agnostic explanators and tabular classification data to identify and research interpretability properties and metrics.

Based on the literature, this work proposes a set of functionally grounded evaluations to

analyze and assess interpretability methods. As (DOSHI-VELEZ; KIM, 2017), the goal of that kind of evaluation is to perform cheap and fast evaluations using interpretability properties as a proxy for quality. This work presented three different interpretability metrics with a formulation that can be used to assess techniques or to determine the best parameters, they are faithfulness, stability, and identity. Faithfulness measures if truly important features are the most significant to the explanator, stability measures the impact that input perturbation pose on explanations and identity defines if explanations are consistent when the input remains the same. To demonstrate these calculations, this work used a toy synthetic dataset and presented how the assessment between different interpretability techniques can be made using those three metrics. These metrics can be used in specific experiments when the user has already deployed a black-box classifier to analyze and assess feature attribution interpretability methods, allowing to determine the best one for the determined experiment or the best parameters to explain instances or dataset patterns.

6.2 Future Improvements and Research

Stability and Identity are straightforward metrics that can be applied in any setting, regardless of the goal of the model (regression or classification). Identity can also be applied in any tabular dataset, whether continuous or categorical. Stability, on the other hand, can only be applied to continuous variables since adding perturbation to categorical variables is not as straightforward as it is on continuous ones.

On the other hand, faithfulness, as proposed in this work, has a major weakness. It is a specific metric that can only be applied to known datasets, that is when the user already knows which features are important to reach the output and want to test which interpretability technique is able to translate that information. As a future improvement, adaptations in this regard can be done to utilize this metric on datasets that are not previously studied or when the user does not know which features are important to achieve the correct output. As a future implementation, when important features are unknown, a proxy notion of faithfulness can be reached by omitting each attribute and observing its impact on the probability of the predicted class, for instance.

BIBLIOGRAPHY

ADADI, A.; BERRADA, M. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). **IEEE Access**, v. 6, p. 52138–52160, 2018. Citation on page 34.

ALPAYDIN, E. **Introduction to Machine Learning**. 2nd. ed. [S.l.]: The MIT Press, 2010. ISBN 026201243X, 9780262012430. Citation on page 23.

ALVAREZ-MELIS, D.; JAAKKOLA, T. S. Towards robust interpretability with self-explaining neural networks. **CoRR**, abs/1806.07538, 2018. Available: <http://arxiv.org/abs/1806.07538>. Citations on pages 34, 35, 36, and 57.

ANGWIN, J.; LARSON, J.; MATTU, S.; KIRCHNER, L. Machine bias. ProPublica, 2016. Available: <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>. Accessed: 07/15/2019. Citations on pages 21 and 27.

ARIK, S. Ö.; PFISTER, T. Tabnet: Attentive interpretable tabular learning. **CoRR**, abs/1908.07442, 2019. Available: <http://arxiv.org/abs/1908.07442>. Citations on pages 47 and 48.

BOWEN, D.; UNGAR, L. Generalized SHAP: generating multiple types of explanations in machine learning. **CoRR**, 2020. Citation on page 47.

CALISKAN, A.; BRYSON, J.; NARAYANAN, A. Semantics derived automatically from language corpora contain human-like biases. American Association for the Advancement of Science, v. 356, n. 6334, p. 183–186, 2017. Citations on pages 21 and 26.

CORTEZ, P.; CERDEIRA, A.; ALMEIDA, F.; MATOS, T.; REIS, J. **Modeling wine preferences by data mining from physicochemical properties**. [S.l.]: Elsevier, 2009. 547-553 p. Citation on page 50.

COSTELLO, K. Gartner survey shows 37 percent of organizations have implemented ai in some form. Gartner, 2019. Available: <https://www.gartner.com/en/newsroom/press-releases/2019-01-21-gartner-survey-shows-37-percent-of-organizations-have>. Citation on page 26.

DATTA, A.; TSCHANTZ, M. C.; DATTA, A. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. **CoRR**, abs/1408.6491, 2014. Available: <http://arxiv.org/abs/1408.6491>. Citations on pages 21 and 26.

DOSHI-VELEZ, F.; KIM, B. Towards a rigorous science of interpretable machine learning. 2017. Citations on pages 22, 28, 29, 30, 33, 34, 35, 36, and 70.

DU, M.; LIU, N.; HU, X. Techniques for interpretable machine learning. **CoRR**, abs/1808.00033, 2018. Available: <http://arxiv.org/abs/1808.00033>. Citations on pages 22, 30, and 31.

FONG, R.; VEDALDI, A. Interpretable explanations of black boxes by meaningful perturbation. **CoRR**, abs/1704.03296, 2017. Available: <http://arxiv.org/abs/1704.03296>. Citations on pages 21 and 26.

FREITAS, A. A. Comprehensible classification models - a position paper. ACM SIGKDD Explorations Newsletter, v. 15, 2013. Citation on page 31.

GILPIN, L. H.; BAU, D.; YUAN, B. Z.; BAJWA, A.; SPECTER, M.; KAGAL, L. Explaining explanations: An approach to evaluating interpretability of machine learning. **CoRR**, abs/1806.00069, 2018. Available: <http://arxiv.org/abs/1806.00069>. Citations on pages 34, 35, and 36.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citations on pages 23, 24, and 25.

GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. Explaining and harnessing adversarial effects. In: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS. [S.l.], 2015. Citations on pages 21 and 27.

GOODMAN, B.; FLAXMAN, S. European union regulations on algorithmic decision-making and a "right to explanation". **AI Magazine**, Association for the Advancement of Artificial Intelligence (AAAI), v. 38, n. 3, p. 50–57, Oct 2017. ISSN 0738-4602. Available: <http://dx.doi.org/10.1609/aimag.v38i3.2741>. Citations on pages 21 and 27.

GREENWALD, A. G.; MCGHEE, D. E.; SCHWARTZ, J. L. K. Measuring individual differences in implicit cognition: The implicit association test. 1998. Citations on pages 21 and 26.

GUIDOTTI, R.; MONREALE, A.; RUGGIERI, S.; PEDRESCHI, D.; TURINI, F.; GIANNOTTI, F. Local rule-based explanations of black box decision systems. **CoRR**, abs/1805.10820, 2018. Available: <http://arxiv.org/abs/1805.10820>. Citation on page 42.

GUIDOTTI, R.; MONREALE, A.; TURINI, F.; PEDRESCHI, D.; GIANNOTTI, F. A survey of methods for explaining black box models. **CoRR**, abs/1802.01933, 2018. Available: <http://arxiv.org/abs/1802.01933>. Citations on pages 29, 30, and 31.

HERMAN, B. The promise and peril of human evaluation for model interpretability. **CoRR**, abs/1711.07414, 2017. Available: <http://arxiv.org/abs/1711.07414>. Citations on pages 34 and 35.

HINTON, G.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. **Neural computation**, v. 18, p. 1527–1554, 2006. Citation on page 25.

HONEGGER, M. Shedding light on black box machine learning algorithms: Development of an axiomatic framework to assess the quality of methods that explain individual predictions. **CoRR**, abs/1808.05054, 2018. Available: <http://arxiv.org/abs/1808.05054>. Citations on pages 36 and 59.

IBRAHIM, M.; LOUIE, M.; MODARRES, C.; PAISLEY, J. W. Global explanations of neural networks: Mapping the landscape of predictions. **CoRR**, abs/1902.02384, 2019. Available: <http://arxiv.org/abs/1902.02384>. Citation on page 46.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. Science, v. 349, p. 255–260, 2015. Citation on page 26.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012. Citation on page 25.

LAKKARAJU, H.; BACH, S. H.; LESKOVEC, J. Interpretable decision sets: A joint framework for description and prediction. v. 22, p. 1675–1684, 2016. Citations on pages 45 and 46.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. Nature, v. 521, n. 7553, p. 436–444, 2015. Citation on page 24.

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural Computation**, v. 1, p. 541–551, 1989. Citation on page 24.

LIANG, B.; LI, H.; SU, M.; BIAN, P.; LI, X.; SHI, W. Deep text classification can be fooled. **CoRR**, abs/1704.08006, 2017. Available: <http://arxiv.org/abs/1704.08006>. Citations on pages 21 and 27.

LIPTON, Z. C. The mythos of model interpretability. **CoRR**, abs/1606.03490, 2016. Available: <http://arxiv.org/abs/1606.03490>. Citations on pages 22, 28, 29, 30, 31, 35, and 36.

LOU, Y.; CARUANA, R.; GEHRKE, J. Intelligible models for classification and regression. p. 150–158, 2012. Citation on page 28.

LUNDBERG, S.; LEE, S. A unified approach to interpreting model predictions. **CoRR**, abs/1705.07874, 2017. Available: <http://arxiv.org/abs/1705.07874>. Citations on pages 43 and 44.

MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous acitivity. Bulletin of Mathematical Biology, 1943. Citation on page 24.

MELLO, R. F. de; PONTI, M. A. **Machine Learning: A Practical Approach on the Statistical Learning Theory**. 1. ed. [S.l.]: Springer, 2018. Citation on page 23.

MINSKY, M.; PAPERT, S. **Perceptrons**. [S.l.]: MIT Press, 1969. Citation on page 24.

MURDOCH, W. J.; SINGH, C.; KUMBIER, K.; ABBASI-ASL, R.; YU, B. Definitions, methods, and applications in interpretable machine learning. **Proceedings of the National Academy of Sciences**, Proceedings of the National Academy of Sciences, v. 116, n. 44, p. 22071–22080, Oct 2019. ISSN 1091-6490. Available: <http://dx.doi.org/10.1073/pnas.1900654116>. Citations on pages 35 and 36.

NAZARÉ, T. S.; COSTA, G. B. P. da; CONTATO, W. A.; PONTI, M. Deep convolutional neural networks and noisy images. In: MENDOZA, M.; VELASTÍN, S. (Ed.). **Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications**. Cham: Springer International Publishing, 2018. p. 416–424. ISBN 978-3-319-75193-1. Citation on page 27.

NOGUEIRA, S.; SECHIDIS, K.; BROWN, G. On the stability of feature selection algorithms. **Journal of Machine Learning Research**, v. 18, p. 1–54, 2018. Citation on page 58.

PAPERNOT, N.; MCDANIEL, P. D.; GOODFELLOW, I. J.; JHA, S.; CELIK, Z. B.; SWAMI, A. Practical black-box attacks against deep learning systems using adversarial examples. **CoRR**, abs/1602.02697, 2016. Citations on pages 21 and 27.

PEDRESCHI, D.; RUGGIERI, S.; TURINI, F. Discrimination-aware data mining. In: . [S.l.: s.n.], 2008. p. 560–568. Citation on page 27.

PONTI, M. A.; RIBEIRO, L. S. F.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: . [S.l.: s.n.], 2017. v. 30. Citation on page 24.

RIBEIRO, M.; SINGH, S.; GUESTRIN, C. Anchors: High-precision model-agnostic explanations. v. 32, 2018. Citations on pages 41 and 43.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "why should I trust you?": Explaining the predictions of any classifier. **CoRR**, abs/1602.04938, 2016. Available: <http://arxiv.org/abs/1602.04938>. Citations on pages 29, 40, and 43.

ROSENBLATT, F. The perceptron: a probabilistic model for information store and organization. Psychological Review, v. 65, 1958. Citation on page 24.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. Nature, v. 323, n. 6088, p. 533–536, 1986. Citation on page 24.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015. Citation on page 25.

SAMEK, W.; WIEGAND, T.; MÜLLER, K. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. **CoRR**, abs/1708.08296, 2017. Available: <http://arxiv.org/abs/1708.08296>. Citations on pages 28 and 29.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. IBM, 1959. Citation on page 24.

SHAPLEY, L. S. Stochastic games. **Proceedings of the National Academy of Sciences**, National Academy of Sciences, v. 39, n. 10, p. 1095–1100, 1953. Citation on page 43.

SHRIKUMAR, A.; GREENSIDE, P.; KUNDAJE, A. Learning important features through propagating activation differences. **CoRR**, abs/1704.02685, 2017. Available: <http://arxiv.org/abs/1704.02685>. Citations on pages 44 and 45.

SU, J.; VARGAS, D. V.; SAKURAI, K. One pixel attack for fooling deep neural networks. **CoRR**, abs/1710.08864, 2017. Available: <http://arxiv.org/abs/1710.08864>. Citation on page 27.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. 2014. Citation on page 27.


ICMC
USP
SÃO CARLOS