Explaining Explainable AI

Núria Camí Cervelló, Ana de Garay Seldas and Claudia Herron Mulet
University of Barcelona
MSc in Fundamental Principles of Data Science
{ncamicer7, adegarse24, clherrom12}@alumnes.ub.edu

Abstract

In the last years, Explainable Artificial Intelligence (XAI) has gained popularity and importance in the Machine Learning field. Explainability offers us a remedy for the distrust of opaque models. Model agnostic methods such as LIME, SHAP, or iBreakDown provide us instance-level interpretability for complex machine learning models. The main goal of this paper is to explain the fundamental concepts of XAI, as well as to present three of the highest impact methods. Furthermore, we experiment with publicly available implementations to benchmark these techniques on a common dataset.

1 Introduction

Year by year, the amount of digital data increases. According to the International Data Cooperation, in 2025 we will have generated 163ZB of data [1]. Nowadays, we already live in a data-driven world where society and technology complement each other. In this scenario where there is a need to extract knowledge from huge datasets and automatize processes, Machine Learning (ML) plays a critical role. However, ML algorithms have been sometimes denoted as "black boxes" [2][3] when the relationship between the inputs and outputs is unclear, i.e., it cannot be explained meaningfully in common language. This opacity can be intentional, as a way of corporate self-protection, or unintentional, due to the lack of technical expertise of the majority of the population. Besides, human-scale reasoning in large-scale scenarios is limited [4]. This lack of transparency can lead to discrimination, fraud, financial risks, etc. As a response, a new tendency has arisen: Explainable Artificial Intelligence.

XAI methods provide users with human-understandable explanations. In the cases where personal data is involved, these techniques are especially important. Recently, there have been many advances in the General Data Protection Regulation (GDPR) that provide European citizens with a set of rights related to the processing of their data. In specific, Art. 13 states that the data controller[1] is obliged to provide to the data subject[2] with "meaningful information about the logic involved, as well as the significance and the envisaged consequences" when there is automated decision making, including profiling[3] (Art. 22). Goodman et al. [5] discuss this right to an explanation and point out the challenges about what is considered to be a "meaningful" explanation of a decision made by an algorithm. There is no consensus among experts on this topic, and a lot of research is being carried out, both on the legal and technical sides.

In addition, explainability brings value to many stakeholders [6]. By boosting transparency, developers can assess the quality and the pitfalls of a model to build more robust systems. Moreover, ML users will gain trust in these models and employ them as a tool to orient their decisions. As a result, it

[1] **Data controller-** the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data.

[2] **Data subject-** an identified or identifiable natural person.

[3] **Profiling-** any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

becomes of special importance to introduce explainability techniques for those results generated by means of ML.

In this paper, we present a review of the state-of-the-art explainability concepts and methods. Our contribution can be summarized in the following points:

- Present a summary of the most relevant concepts in the explainability field.
- Review in detail the theoretical background three of the most popular techniques
- Experiment with explainability software and libraries to gain insights on a toy dataset.

2 Background

As ML models increase in complexity, the ability to introspect and understand why a model made a particular prediction has become more and more difficult, especially since the boom of Deep Learning in 2012. It has also become more relevant, as ML models make predictions that increasingly influence important aspects of our lives, from the outcome of home loan applications to job interviews, medical treatment, or even incarceration decisions. To address this issue and make models more explainable and interpretable, many new methods have been developed since 2016. In this section, we will discuss some of the preliminary aspects of XAI: basic definitions, taxonomy and scope, and evaluation methods.

2.1 Basic concepts in XAI: interpretability, explainability, and black box models

One of the most popular definitions of **interpretability** is the one of Doshi-Velez and Kim [7], who regards it as "the ability to explain or to present in understandable terms to a human". Another popular characterization came from Miller in his work [8], where he defines interpretability as "the degree to which a human can understand the cause of a decision" [9]. Based on the above, interpretability is mostly connected with the intuition behind the outputs of a model, with the idea being that the more interpretable a machine learning system is, the easier it is to identify cause-and-effect relationships within the system's inputs and outputs. For example, in image recognition tasks, part of the reason that led a system to decide that a specific object is part of an image (output) could be certain dominant patterns in the image (input).

Interpretability goes hand in hand with explainability. Although both are usually used by researchers interchangeably, some works identify their differences and distinguish these two concepts. **Explainability** is associated with the internal logic and mechanics that are inside a machine learning system. The more explainable a model, the deeper the understanding that humans achieve in terms of the internal procedures that take place while the model is training or making decisions. It's worth noting, however, that explainability does not describe how the model works — rather, it offers a rationale to interpret human-understandable responses. Otherwise, an interpretable model does not necessarily translate to one that humans are able to understand the internal logic of or its underlying processes. Therefore, regarding machine learning systems, interpretability does not imply explainability or vice versa [9]. In fact, there is not a concrete mathematical definition for interpretability or explainability, nor have they been measured by some metric. For this reason, we will use both terms throughout this paper.

Both for lack of interpretability and explainability, ML models are often nicknamed as "**black boxes**". The opaque nature of their internal workings and the qualitative understanding between the input variables and the response are still obstacles to get over nowadays. Here is where ML interpretability methods come into play.

2.2 Taxonomy and scope of XAI methods

As a discipline, XAI can be split according to different criteria. It should be pointed out that the classification of ML interpretability techniques should not be one-sided, and further divisions to the ones presented here could be considered.

Firstly, when it comes to how insights are extracted, we can distinguish between model-intrinsic (or pre-hoc) and model-agnostic (post-hoc) methodologies. **Model intrinsic explainability** focuses on

those models that can explicitly describe its behavior based on its components, due to its simple structure (decision trees or sparse linear models, for instance). In contrast, **model-agnostic explainability** focuses on characterizing model outputs through external interpretation methods, without knowing the internal mechanism of the model itself (e.g. permutation feature importance - decrease/increase in model performance when a single feature value is randomly shuffled) [10]. Very closely related with the previous division, we can also consider model-specific explanation techniques, only restricted to a specific family of models, as opposed to model agnostic techniques, applicable to any kind of algorithm [9].

XAI implies understanding an automated model, supporting two variations according to its scope: understanding the entire model behavior (global explainability) or understanding a single prediction (local explainability). **Global explainability** facilitates the interpretation of the whole logic of a model and follows the entire reasoning leading to all the different possible outcomes. On the other hand, **local explainability** is used to generate an individual explanation, usually to justify why the model made a specific decision for an instance [11]. In general, global explanations have a higher recall view of the entire model prediction space, but lower precision due to aggregations.

Down below a diagram of the different classifications described is shown (Figure 1) and several state-of-the-art examples of XAI methods are given (Table 1). It is worth noting that XAI methods are very heterogeneous in the way that they present and compute explanations. For example, Partial Dependence plots [12] or Individual Conditional Expectation (ICE) plots [13] are initially designed as fixed visual tests, white other methods such as SHAP [14] or LIME [15] extract feature importance weights that can fit into different representation strategies. Furthermore, there are many ML models that are highly interpretable, such as linear regression, logistic regression, decision trees, Naive Bayes or k-nearest neighbors. On the other hand, lots of model specific methods focus on Deep Learning, such as Integrated Gradients [16], SmoothGrad [17] or DeepLIFT [18].There are two reasons for considering interpretation methods developed specifically for neural networks: NNs learn features and concepts in their hidden layers and we need special tools to uncover them; moreover, the gradient can be utilized to implement interpretation methods that are more computationally efficient than model-agnostic methods that look at the model "from the outside" [19].

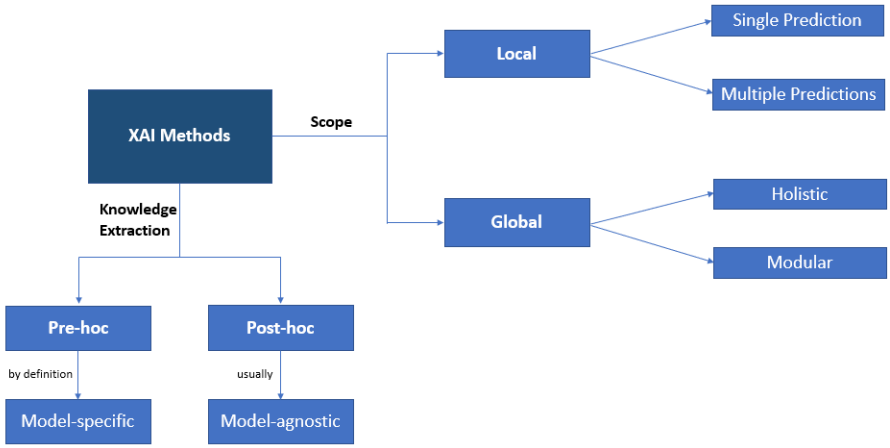

Figure 1: XAI Taxonomy diagram [19]

Another possible categorization in XAI refers to the various alternatives for presenting the desired explanation [19], such as feature summary statistics, visualizations, interpretation of model internals (e.g. weights in linear models or the learned tree structure on DTs), or intrinsically interpretable model approximations. There are more plausible ways of classifying these methods, such as those related to the input's different data types or the ones that deal with the different purposes of interpretability [9].

In section 3 we will focus on describing three state-of-art model-agnostic methods for local explainability: LIME, SHAP, and iBreakDown.

Explainability models	Model specific	Model agnostic
Local	Integrated gradients Saabas Occlusion Maps SmoothGrad DeepLIFT	SHAP LIME iBreakdown ICE
Global	TCAV	Partial Dependent Plot

Table 1: Examples of XAI methods

2.3 Evaluation of XAI methods

A crucial property that interpretability methods should satisfy to generate meaningful explanations is **robustness** to local perturbations of the input. In its most intuitive form, such a requirement states that similar inputs should not lead to substantially different explanations. Robustness is a crucial property interpretability methods should strive for due to two main reasons. First, for an explanation to be valid around a point, it should remain roughly constant in its vicinity, regardless of how it is expressed. On the other hand, if we seek an explanation that can be applied in a predictive sense around the point of interest as described above, the robustness of the simplified model implies that it can be approximately used instead of the true complex model, at least in a small neighborhood.

To conclusively evaluate this robustness, we need objective tools to quantify it. Since we are interested in a local notion of stability, i.e., for neighboring inputs, the neighborhood-based local Lipschitz continuity is evaluated in the explainability method (in Appendix for completeness). This tool works as a parametric notion of stability that measures relative changes in the output with respect to the input [20].

3 Methodology

In this Methodology section, we dive into three State-of-the-art explainability techniques: SHAP, LIME, and iBreakdown.

3.1 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations is a framework proposed by Lundberg et al. that measures the importance of each of the features given the output of a ML model. Similar to LIME and iBreakdown, SHAP focuses on local explanations. Lundberg et al. defend that the best explanation of a simple model is the model itself. For this reason, SHAP is especially targeted to complex models such as neural networks or ensembles.

Before entering into details on how to compute the SHAP values it is worth explaining what an *additive feature attribution method* is. Our goal is to find an *explanation model*, g, that is an interpretable approximation of the original function, f, computed employing ML. Often, explanation models take as input simplified versions of the original input, namely $x = h_x(x')$. Additive feature attribution methods propose an explanation model that is a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

where $z' \in \{0,1\}^M$, M is the number of simplified input features, and $\phi_i \in \mathbb{R}$. Each feature has attributed an effect on the output, and ideally, $g(z') \approx f(h_x(z'))$, i.e., the output of the explanation model coincides with the output of the "black-box" ML model. Many SOTA models fall inside this category such as DeepLIFT, LIME, Layer-Wise Relevance Propagation or Quantitative input influence (QII).

Lundberg et al. prove that Shapley values, from theoretical game theory, are the unique solution of the problem 1 when the following properties are satisfied: local accuracy, missingness, and consistency (in Appendix for completeness). In short, Shapley values tell us how to fairly distribute the payout

among the players in a game. In the context of ML, players are features and the payout is the model prediction. So, Shapley values tell us how we can distribute feature importance. Moreover, we are interested in preserving the beforementioned properties because they are a mathematical formulation of human intuition of what an explanation should be. As a result, the only possible explanation model, g, is constructed by the following effects ϕ_i:

$$\phi_i = \sum_{z''} \frac{|z'|!(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z' \setminus i)] \tag{2}$$

where $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x'.

To compute the SHAP values for measuring feature importance, the authors propose Shapley values of a conditional expectation function of the original model, i.e., $f(z') = f(h_x(z')) = E[f(z)|z_S]$, to define simplified inputs. Usually, these weights are approximated rather than computed, due to the high computational cost. There are model-agnostic approximation methods (*Shapley sampling values* and *Kernel SHAP*) and model-type-specific methods (*Max SHAP* and *Deep SHAP*). In this paper, we experiment in section IV with Kernel SHAP.

3.2 Local Interpretable Model-agnostic Explanations (LIME)

Ribeiro et al. [15] proposed LIME for Local Interpretable Model-Agnostic Explanation which goal is to identify an interpretable model that is locally faithful to the original ML model (for example, linear). Fig. 2. shows a toy example to present intuition for LIME.

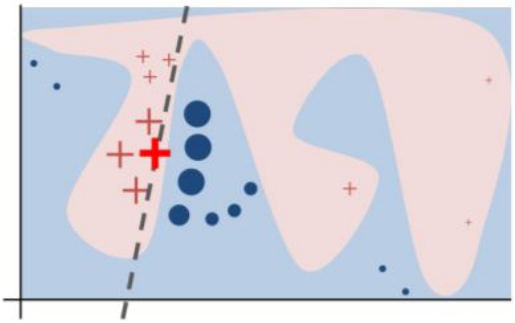

Figure 2: The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful [15].

The local linear explanation model that LIME uses adheres to (1) exactly and is thus an additive feature attribution method. LIME refers to simplified inputs x' as *interpretable inputs*, and the mapping $x = h_x(x')$ converts a binary vector of interpretable inputs into the original input space. Different types of h_x mappings are used for different input spaces. For bag of words text features, h_x converts a vector of 1's or 0's (present or not) into the original word count if the simplified input is one, or zero if the simplified input is zero. For images, h_x treats the image as a set of superpixels; it then maps 1 to leaving the superpixel as its original value and 0 to replacing the superpixel with an average of neighboring pixels (this is meant to represent being missing).

To find ϕ in (1), LIME minimizes the following objective function:

$$\xi = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \, \mathcal{L}(f, g, \pi_{x'}) + \Omega(g) \tag{3}$$

The loss function $\mathcal{L}$ in the previous equation is a measure of how unfaithful g is in approximating f in the locality defined by $\pi_{x'}$, which is a proximity measure between an instance z' to x', to define

locality around x'. Besides, Ω penalizes the complexity of g. Since in LIME g follows (1) and $\mathcal{L}$ is a squared loss, (3) can be solved using penalized linear regression [14].

3.3 iBreakDown

iBreakDown is a model-agnostic tool for locally explaining ML model predictions. It is the successor of BreakDown [21], mainly based on decomposing model predictions into parts that can be attributed to particular features. When dealing with additive models BreakDown is a good explainability approach, however, for non-additive models, it fails to capture relevant variable interaction information. As stated by Staniak and Biecek on *Explanations of model predictions with Live and BreakDown packages* [21], "presented methods decompose final prediction into additive components attributed to particular features; this approach will not work well for models with heavy components related to interactions between features".

The key issue of local methods such as SHAP, LIME or BreakDown, is that they show additive local representations, while complex models are usually non-additive. Therefore, these methods often do not include all nuances of a model, such as relations between features, and therefore sometimes turn out to be imprecise. Thus, we need to find approaches that are more accurate to explain the underlying model. iBreakDown attempts to solve this problem taking into account these interactions. Moreover, its implemented algorithm is faster (complexity $O(p)$ instead of $O(p^2)$ in plain Breakdown) [22].

Let $f : \mathbb{X} \subset \mathbb{R}^p \to \mathbb{R}$ be a predictive model and $x^* \in \mathbb{X}$ be an observation to explain. The scheme of iBreakDown algorithm is the following [23]:

1. Calculate a single-step additive contribution for each feature:

$$\Delta_i = E[f(x)|x_i = x_i^*] - E[f(x)] \tag{4}$$

   where the first term represents the average prediction of model f if feature x_i is fixed to x_i^*, and the second term is the expected model prediction (baseline Δ_0).

2. Calculate a single-step contribution for every pair of features.

$$\Delta_{i,j} = E[f(x)|x_i = x_i^*, x_j = x_j^*] - E[f(x)] \tag{5}$$

   Then, subtract additive contribution to assess the interaction specific contribution:

$$\Delta_{i,j}^I = \Delta_{i,j} - \Delta_i - \Delta_j \tag{6}$$

3. Once the order of single-step importance is determined based on Δ_i and $\Delta_{i,j}^I$ scores, the final explanation is the attribution to the sequence of

$$\Delta_{i,j|J} = \Delta_{J\cup\{i,j\}} - \Delta_J \tag{7}$$

   scores, where J is a set of indexes of features. Here contributions of features are calculated sequentially. The effects of consecutive variables depend on the change of expected model prediction while all previous variables are fixed.

In an experiment carried out over 28 binary classification datasets from OpenML100, it was found after applying iBreakDown that about 71% of local explanations consisted of interactions [23]. Therefore, the usage of additive methods would strongly simplify the explanations, making them less accurate and more uncertain.

4 Results

In this section, we experiment with the previously described explainability techniques. SHAP[4], LIME[5] and iBreakDown [6] have publicly available python packages in GitHub. To benchfmark them, we use the Titanic Kaggle dataset [24]. It is often used to solve the binary classification task of predicting if a passenger on board in the Titanic was likely to survive or not. Each row in the dataset

[4]**SHAP-** https://github.com/slundberg/shap

[5]**LIME-** https://github.com/marcotcr/lime

[6]**iBreakdown (Python)-** https://github.com/ModelOriented/piBreakDown

is a passenger and each column is one of the 10 socioeconomic features. For simplicity purposes, we decided to keep only 3 features: age, sex, and Pclass (a categorical variable that can be 1st, 2nd, or 3rd cruise class). Our target variable is the boolean "Survived". Also, we performed data cleaning to remove nan values, which lead us to a final dataset of 714 passengers.

We intend to explain one sample prediction made with the ensemble model RandomForest, with each of the above-described methods. This ML model is classically denoted as a black-box model because it consists of a large number of deep trees, where each tree is trained on bagged data using random selection of features. As a result, drawing conclusions on how the output is generated is a challenging task. After training the RandomForest, we obtained 80.42% accuracy on the testing dataset (20% of the initial data). The observation chosen for our explainability analysis corresponds to a 65-year-old male passenger, who traveled on 1st class. The baseline (or expected model prediction) for survival is 0.416 and, in particular, our selected passenger had a 0.023 probability of surviving.

4.1 SHAP

In Figure 3 we observe one of the explanatory plots proposed by the SHAP[4] library. These kinds of visualizations are called *Force plots* and they are especially useful to depict the influence of each feature on the model's prediction for a specific instance of the data. Starting from the base value, we observe how our features produce an increase (in red) or decrease (in blue) of the probability of surviving. So, in this specific case, we see that the fact that the passenger is a 65-years-old man lowers the survival probability while the first-class ticket rises it. As this is an additive feature attribution method, we obtain the predicted probability applying equation 1, i.e., adding up the base value and the feature effects. Furthermore, we empirically confirm with real data that our explanation model, g, is an interpretable approximation of the original ML function, f, and therefore $g(z') \approx f(h_x(z')) \approx 0.02$.

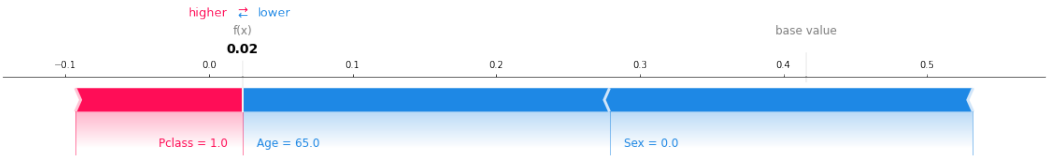

Figure 3: SHAP explanation

4.2 LIME

The LIME[5] package for python offers the visualization shown in Figure 4. The left side part reveals that the underlying model predicts a 98% probability of non-survival for the selected passenger. On the right side, the weights corresponding to each variable are shown in blue bars whether they contribute negatively to its survival, or in orange, otherwise.

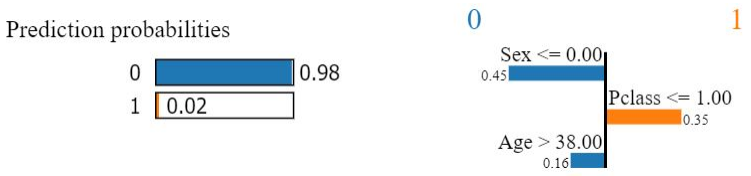

Figure 4: LIME explanation

In our example, the variables *Sex* and *Age* are the most weighted features explaining his very low chance of survival.

4.3 iBreakDown

In Figure 5 we observe the explanatory waterfall plot provided by piBreakDown package[7] for python (iBreakDown implementation for python, the original package was developed for R[7]). The first row of our plot shows the average model prediction, while the last one shows the model's prediction for our particular observation. Intermediate bars present the contributions of each feature, after taking into account possible interactions between them. Red means a negative effect on the survival probability, while green means a positive effect. The order of features on the y-axis corresponds to their sequence.

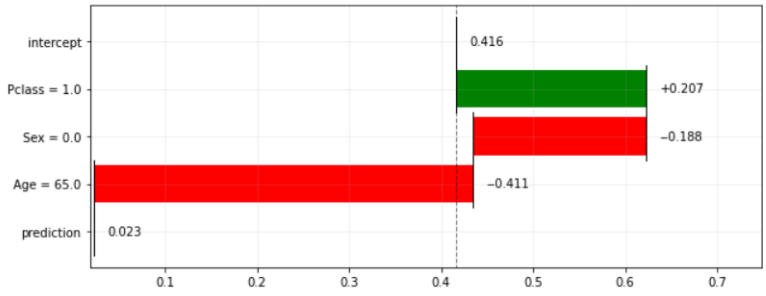

Figure 5: iBreakDown explanation

4.4 Comparison

Finally, we compare how each feature contributes to the output in each of the presented explanation methods. As we can see in Table 2, the three techniques are coherent in the sense of which features contribute positively/negatively to survival. LIME's explanation attributes the greatest positive impact to passenger's class, as well as the greatest negative impact to sex, whereas iBreakDown gives the greatest negative influence on age.

Feature	SHAP	LIME	iBreakDown
Pclass = 1	0.1163	0.3488	0.2073
Sex = Male	-0.2530	-0.4450	-0.1883
Age = 65	-0.2555	-0.1604	-0.4113

Table 2: Feature contributions per explainability model for the first passenger

In order to check if the employed methods are robust, we intuitively prove that they are locally-Lipschitz. To do so, we perform a new explanation over a similar observation. In Table 3 we present the results obtained for a 54-year-old man also in first class. As we can see by the low variation of the feature importance, we verify that the obtained contributions per feature are as well similar, hence, all three models are robust.

Feature	SHAP	LIME	iBreakDown
Pclass = 1	0.1482	0.3364	0.2073
Sex = Male	-0.2453	-0.4566	-0.1883
Age = 54	-0.2445	-0.1479	-0.3606

Table 3: Feature contributions per explainability model for the second passenger

5 Conclusion

In this research paper, we have explained the foundations of XAI. By analyzing the theory behind SHAP, LIME and iBreakDown we arrived to the understanding that these local model agnostic explainability techniques provide a simple and robust framework to explain the feature contribution in

[7] **iBreakdown (R)-** https://github.com/ModelOriented/iBreakDown

the output produced by complex ML models. The visualizations presented in the Results section are human-interpretable tools that non-technical ML users can use to gain insights into how the model assigns feature importance. Furthermore, we compared these three techniques and observed that the feature effects provide coherent but diverse results, as each method satisfies different constraints in the explainability model definition.

A Appendix

Property 1 (Local accuracy):

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{8}$$

The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$.

Property 2 (Missingness):

$$x'_i = 0 \Rightarrow \phi_i = 0 \tag{9}$$

Missingness constrains features where $x'_i = 0$ to have no attributed impact.

Property 3 (Consistency): Let $f_x(z') = f(h_x(z'))$ and $z' \setminus i$ denote setting $z'_i = 0$. For any two models f and f', if

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)) \tag{10}$$

for all inputs $z' \in \{0,1\}^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$.

Definition 1 (Local Lipschitz continuity function):
$f \colon \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is locally Lipschitz if for every x_0 there exist $\delta > 0$ and $L \in \mathbb{R}$ such that $\|x - x_0\| < \delta$ implies $\|f(x) - f(x_0)\| \leq L\|x - x_0\|$.

References

[1] D. Reinsel, J. Gantz, and J. Rydning, "Data Age 2025: The Evolution of Data to Life-Critical," tech. rep., 2017.

[2] C. O'Neil, *Weapons of math destruction: How big data increases inequality and threatens democracy*. New York: Crown, first edition ed., 2016.

[3] F. Pasquale, *The Black Box Society: The Secret Algorithms That Control Money and Information*. USA: Harvard University Press, 2015.

[4] J. Burrell, "How the machine 'thinks': Understanding opacity in machine learning algorithms," *Big Data & Society*, vol. 3, no. 1, p. 2053951715622512, 2016.

[5] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI Magazine*, vol. 38, p. 50–57, Oct 2017.

[6] A. Preece, D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty, "Stakeholders in explainable ai," 2018.

[7] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[8] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[9] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, 2021.

[10] Q. Ai and L. Narayanan.R, "Model-agnostic vs. model-intrinsic interpretability for explainable product search," *Proceedings of the 30th ACM International Conference on Information Knowledge Management*, Oct 2021.

[11] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[12] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189 – 1232, 2001.

[13] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," 2014.

[14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," 2016.

[16] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," 03 2017.

[17] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *CoRR*, vol. abs/1706.03825, 2017.

[18] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," *CoRR*, vol. abs/1704.02685, 2017.

[19] C. Molnar, *Interpretable Machine Learning*. 2019.

[20] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *CoRR*, vol. abs/1806.08049, 2018.

[21] M. Staniak and P. Biecek, "Explanations of model predictions with live and breakdown packages," *The R Journal*, vol. 10, no. 2, p. 395, 2019.

[22] P. Biecek and T. Burzykowski, *Explanatory Model Analysis*. Chapman and Hall/CRC, New York, 2021.

[23] A. Gosiewska and P. Biecek, "Do not trust additive explanations," 2020.

[24] "Titanic - machine learning from disaster." `https://www.kaggle.com/c/titanic/data`.