Theoretical Behavior of XAI Methods in the Presence of Suppressor Variables

Rick Wilming [1] **Leo Kieslich** [1] **Benedict Clark** [2] **Stefan Haufe** [1 2 3]

Abstract

In recent years, the community of 'explainable artificial intelligence' (XAI) has created a vast body of methods to bridge a perceived gap between model 'complexity' and 'interpretability'. However, a concrete problem to be solved by XAI methods has not yet been formally stated. As a result, XAI methods are lacking theoretical and empirical evidence for the 'correctness' of their explanations, limiting their potential use for quality-control and transparency purposes. At the same time, Haufe et al. (2014) showed, using simple toy examples, that even standard interpretations of linear models can be highly misleading. Specifically, high importance may be attributed to so-called suppressor variables lacking any statistical relation to the prediction target. This behavior has been confirmed empirically for a large array of XAI methods in Wilming et al. (2022). Here, we go one step further by deriving analytical expressions for the behavior of a variety of popular XAI methods on a simple two-dimensional binary classification problem involving Gaussian class-conditional distributions. We show that the majority of the studied approaches will attribute non-zero importance to a non-class-related suppressor feature in the presence of correlated noise. This poses important limitations on the interpretations and conclusions that the outputs of these XAI methods can afford.

1. Introduction

The field of 'explainable artificial intelligence' (XAI) is devoted to answering the broad question of why an automatic decision system put forward a certain prediction. This is often addressed by techniques that attribute a so-called 'importance' score to each feature of an individual test input. It is commonly agreed that being able to answer this question is necessary to create trust in and a better understanding of the behavior of such decision systems (Baehrens et al., 2010; Ribeiro et al., 2016; Binder et al., 2016; Lundberg & Lee, 2017; Fisher et al., 2019). In Haufe et al. (2014) and Wilming et al. (2022), it was shown that features which certain XAI methods determine to be important, e.g. by inspecting their corresponding weights of a linear model, may actually not have any statistical association with the predicted variable. As a result, the provided 'explanation' may not agree with prior domain knowledge of an expert user and might undermine that user's trust in the predictive model, even if it performs optimally. Indeed, a highly accurate model might exploit so-called suppressor features (Conger, 1974; Friedman & Wall, 2005), which can be statistically independent of the prediction target yet still lead to increased prediction performance. On the other hand, incorrect explanations may implant misconceptions about the data, the model and/or the relationship between the two into a user's mind, which could lead to misguided actions that could be harmful.

While Haufe et al. (2014) have introduced low-dimensional and well-controlled examples to illustrate the problem of suppressor variables for model interpretation, Wilming et al. (2022) showed empirically that the emergence of suppressors indeed poses a problem for a large group of XAI methods and diminishes their 'explanation performance'. Here, we go one step further and derive analytical expressions for commonly used XAI methods for a simple two-dimensional linear data generation process capable of creating suppressor variables by parametrically inducing correlations between features. In particular, we investigate which XAI approaches attribute non-zero importance to plain suppressor variables that are by construction independent of the prediction target and thereby violate a data-driven definition of feature importance recently put forward by Wilming et al. (2022).

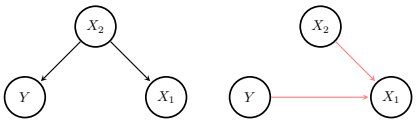
2. Related Work

XAI methods often analyze ML models in a post-hoc manner (Arrieta et al., 2020), where a trained model deemed to be 'non-interpretable', such as a deep neural network, is given, while the XAI methods attempt to 'reverse-engineer' its decision for a given input sample. A crucial limitation of

[1]Technische Universität, Berlin, Germany [2]Physikalisch-Technische Bundesanstalt, Berlin, Germany [3]Charité – Universitätsmedizin, Berlin, Germany. Correspondence to: Stefan Haufe <haufe@tu-berlin.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

the field of XAI is that it is still an open question what formal requirements *correct* explanations would need to fulfill and what conclusions about data, model, and their relationship the analysis of an importance map provided by XAI methods should afford. The lack of a clear definition of what problem XAI is supposed to solve led to multiple studies evaluating explanation methods (e.g. Doshi-Velez & Kim, 2017; Kim et al., 2018; Alvarez-Melis & Jaakkola, 2018; Adebayo et al., 2018; Sixt et al., 2020). Yet, these studies primarily employ auxiliary metrics to measure secondary quality aspects, such as the stability of the provided maps. For example, Yang & Kim (2019) investigate how importance maps for one model change relative to another model. Until recently, it has been considered difficult to define and evaluate the correctness of importance maps, because real-world datasets, which are ubiquitous in the ML community as benchmarks for supervised prediction tasks, do not offer access to the 'true' set of important features. However, several XAI benchmarks using controlled synthetic data have emerged in the past three years. Agarwal et al. (2022) propose a benchmark that can generate synthetic data and assess XAI methods on a broad set of evaluation metrics. The authors state that their framework predominantly serves the purpose of gaining a better understanding of a model's internal mechanics, which would primarily show the debugging capabilities of XAI methods rather than their ability to generate knowledge of 'real-world' effects. Sixt et al. (2020) provide a theoretical analysis of convergence problems of so-called saliency methods, especially Layer-wise Relevance Propagation (LRP, Bach et al., 2015), Deep Taylor Decomposition (DTD, Montavon et al., 2017), and DeepLIFT (Shrikumar et al., 2017). Notably, the provided derivations do not take the model's input data into account. Kindermans et al. (2018) use a minimal data generation example, to mainly motivate a discussion about drawbacks of saliency maps to finally propose novel explanation techniques based on the DTD framework. Janzing et al. (2020) consider a structural data generation model, promoting unconditional expectations as a value function for SHAP (Lundberg & Lee, 2017) by demonstrating that observational conditional expectations are flawed. In an extensive study on Partial Dependency Plots (Friedman, 2001) and M-plots (Apley & Zhu, 2020), Grömping (2020) theoretically analyse a regression task via a pre-defined regression model $\mathbb{E}(Y|\mathbf{x})$ with multivariate Gaussian distributed data. They argue that M-plots can lead to deceptive results, especially if machine learning models rely on interaction effects. Wilming et al. (2022) empirically study common post-hoc explanation methods using a carefully crafted dataset based on a linear data generation process. Here, all statistical dependencies and absolute feature importances are well defined, giving rise to ground-truth importance maps. This empirical study showed that most XAI methods indeed highlight suppressor features as important.



(a) Confounder (b) Suppressor and Collider

Figure 1. In (a), feature X_2 is a confounder variable influencing Y and another feature X_1, causing spurious associations. In contrast, in (b) X_2 is a so-called suppressor variable that has no statistical association with the target Y, although both influence feature X_1, which is called a collider.

2.1. Definition of Feature Importance

In this paper, we adopt a data-driven notion proposed by Wilming et al. (2022) as a tentative definition of feature importance. We consider a supervised learning task, where a model $f : \mathbb{R}^d \to \mathbb{R}$ learns a function between an input $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and a target $y^{(i)} \in \mathbb{R}$, based on training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$. Here, $\mathbf{x}^{(i)}$ and $y^{(i)}$ are realizations of the random variables $\mathbf{X}$ and Y, with joint probability density function $p_{\mathbf{X},Y}(\mathbf{x}, y)$. Then a feature X_j can be defined to be important if it has a statistical association to the target variable Y, i.e.

$$X_j \text{ is important} \Rightarrow X_j \not\perp\!\!\!\perp Y. \tag{1}$$

2.2. Suppressor Variables

To illustrate the characteristics of suppressor variables, consider a binary classification problem with two measured scalar input features x_1 and x_2, where x_1 carries all discriminative information, following Haufe et al. (2014). We design the input data such that x_1 holds the signal of interest $z \in \{-1, 1\}$, which is identical to the target variable $y = z$. Furthermore, during the measuring process, feature x_1 is inadvertently obfuscated by a *distractor* η: $x_1 = z + \eta$. The second feature only consists of the distractor signal, i.e. $x_2 = \eta$. Our goal is to learn a function that can discriminate between the two states $y = -1$ and $y = 1$ or, in other words, recover the signal of interest z. We can build a model solely based on feature x_1 to solve the classification problem, as x_1 is the only feature that contains information about $y = z$. Yet, the obfuscation of x_1 by the distractor η diminishes its predictive power. On the other hand, feature x_2 does not contain any information about $y = z$. Therefore, a model solely based on x_2 cannot reach above chance-level classification accuracy. However, a bivariate linear model with a weight vector $w = (1, -1)^\top$ can perfectly recover the signal of interest and, thereby, the target: $w^\top \mathbf{x} = z + \eta - \eta = z = y$. Additionally, Structural equation models (SEM) are depicting different ways in which a variable X_2 can influence the prediction of a target variable Y. In Figure 1a X_2 is a confounder variable influencing

Y and another feature X_1, causing spurious associations. Confounders can appear, for example, as watermarks in image classification tasks, as studied by Lapuschkin et al. (2019) and can reduce the generalization capabilities of a model to new data where confounders might be absent. However, in contrast, we consider suppressor variables X_2 (see Figure 1b) that have no statistical associations with a target variable Y, while X_1 is a collider variable, taking input from both Y and X_2. Here, we can establish the relation $\mathrm{P}(X_2 \mid X_1) \neq \mathrm{P}(X_2 \mid X_1, Y)$ showing a conditional dependency of the suppressor X_2 on the target Y. These conditional dependencies are used by multivariate methods to improve the accuracy of predictions. In practice, XAI methods do not distinguish whether a feature is a confounder or a suppressor, which can lead to misunderstandings about a model's performance and interpretation.

3. Methods

The purpose of this paper is to use a simple model of suppressor variables as a device to analyze the importances produced by a number of popular XAI methods, and to compare these importance scores to our data-driven definition of feature importance (1). In the following, we introduce notation that we will use throughout the text, define the data generation model, derive the Bayes optimal classifier, and provide further technical remarks.

3.1. Linear Generative Model

We now slightly extend the generative data model of the former section 2.2 and provide a full specification of it. Again, we consider a binary classification problem with a two-dimensional feature space where feature x_1, by construction, is statistically associated with the target y, while feature x_2 fulfills the definition of a suppressor variable. Correlations between both features are introduced parametrically through a Gaussian noise process, as a result of which the Bayes optimal classifier generally needs to make use of the suppressor variable. We define H and Z as the random variables of the realizations η and z, respectively, to describe the linear generative model

$$\mathbf{x} = \mathbf{a}z + \eta, \quad y = z, \tag{2}$$

with $Z \sim \mathit{Rademacher}(1/2)$, $\mathbf{a} = (1,0)^\top$ and $H \sim N(\mathbf{0}, \Sigma)$ with a covariance matrix parameterized as follows:

$$\Sigma = \begin{bmatrix} s_1^2 & cs_1s_2 \\ cs_1s_2 & s_2^2, \end{bmatrix}, \tag{3}$$

where s_1 and s_2 are non-negative standard deviations and $c \in [-1, 1]$ is a correlation. The vector $\mathbf{a}$ is also called signal *pattern* (Haufe et al., 2014; Kindermans et al., 2018). With that, the generative model (2) induces a binary classification problem, where $\mathbf{X} = (X_1, X_2)$ is the random variable of the realization $\mathbf{x}$ with the joint density

$$p(\mathbf{x}) = \pi p_1(\mathbf{x} \mid Y = 1) + (1 - \pi)p_2(\mathbf{x} \mid Y = -1)\ , \tag{4}$$

and prior probabilities, $\pi = \mathrm{P}(Y = \pm 1) = 1/2$. The densities $p_{1/2}$ are the class-conditional densities which are both multivariate normal, with $\mathbf{X} \mid Y = y \sim N(\mu_i, \Sigma)$ for $y \in \{-1, 1\}$ and $i = 1, 2$ and have identical covariance matrix $\Sigma \in \mathbb{R}^{2\times 2}$ and expectations $\mu_1 = (1, 0)^\top$ and $\mu_2 = (-1, 0)^\top$. A graphical depiction of the data generated by our data model is provided in Figure 2.

3.2. Bayes Optimal Classifier

The classifier $g : \mathbb{R}^d \to \{-1, 1\}$ that minimizes the error $\mathrm{P}(g(\mathbf{X}) \neq Y)$ is called the Bayes optimal classifier and defined by $g(\mathbf{x}) = \mathbb{I}_{f^*(\mathbf{x})>1/2}$, with the conditional probability $f^*(\mathbf{x}) = \mathrm{P}(Y = 1|\mathbf{X} = \mathbf{x})$. For multivariate normal class-conditional densities, we can calculate the exact Bayes rule $f : \mathbb{R}^d \to \mathbb{R}$, which in this case is a linear discriminant function with $g(\mathbf{x}) = \mathbb{I}_{f(\mathbf{x})>0}$ and $f(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b$.

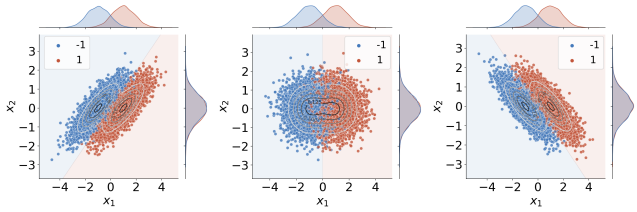
The generative data model, defined above in section 3.1, induces a binary classification problem yielding two class-conditional densities which are both multivariate normal. We solve the classification task in a Bayes optimal way if we assign $\mathbf{x}$ either to class $Y = 1$ or to class $Y = -1$ based on the minimal squared Mahalanobis distance $\delta^2(\mathbf{x}, \mu_i) = (\mathbf{x} - \mu_i)^\top\Sigma^{-1}(\mathbf{x} - \mu_i)$ between $\mathbf{x}$ and the two class means $\mu_i, i = 1, 2$. Then the concrete form of the linear Bayes rule is determined by the coefficients

$$w_1 = \alpha, \quad w_2 = -\alpha cs_1/s_2 \tag{5}$$

for $\alpha := (1 + (cs_1/s_2)^2)^{-\frac{1}{2}}$ and $||\mathbf{w}||_2 = 1$. Note, the classification problem is set up such that the linear decision rule requires no offset or bias term, i.e. $b = 0$. In Appendix A we provide further details for deriving the Bayes optimal decision rule f.

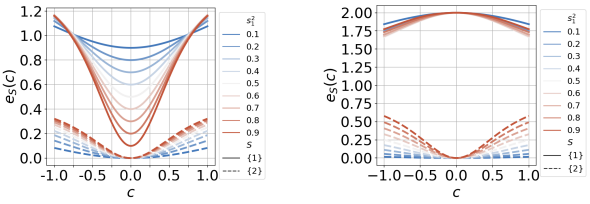
3.3. Notation

Throughout, $f : \mathbb{R}^d \to \mathbb{R}$ is a learned function, in our case the Bayes optimal classifier, where f usually represents the linear decision rule itself. The dimension of the input domain, $d \in \mathbb{N}$, is set to $d = 2$. We define an index set of all features $[d] := \{1, \ldots, d\}$, in order to define features of interest as a subset $S \subset [d]$, where x_S denotes the restriction of $\mathbf{x} \in \mathbb{R}^d$ to the index set S. Analogously, we define the complement $C = [d] \setminus S$, defining x_C as all other features that are not of interest in a particular explanation task. We also define the output of any XAI method as a mapping $e_S : \mathbb{R}^d \to \mathbb{R}$ representing the importance or 'relevance' assigned by the method to the feature set S.



(a) $c = 0.8$ (b) $c = 0$ (c) $c = -0.8$

Figure 2. Data sampled from the generative process (2) for different correlations c and constant variances $s_1^2 = 0.8$ and $s_2^2 = 0.5$. Boundaries of Bayes optimal decisions are shown as well. The marginal sample distributions illustrate that feature x_2 does not carry any class-related information.



(a) Faithfulness through pixel flipping (b) Permutation feature importance

Figure 3. Analytical approximations of faithfulness and permutation feature importance. Shown is a family of curves as a function of feature correlation $c \in [-1, 1]$ variance s_1^2 for constant variance $s_2^2 = 0.5$. Importance maps differ in offsets, indicating consistently higher importance for the informative feature x_1. Yet, both methods allocate importance also to the suppressor feature x_2 for $c > 0$. Analogous figures for different s_2^2 values are contained in the supplementary Figures 6 and 7.

4. Analysis of Common Explanation Methods

In the following, we provide a theoretical analysis of popular XAI methods. The linear generative model (2) is our device to assess those methods' behavior in the presence of suppressor features.

4.1. Gradient

A ML model's gradient itself is often used for explanations, as it describes the change of the model output as a function of the change of the input parameters (e.g. Gevrey et al., 2003; Selvaraju et al., 2017). For linear models, the gradient is identical to the model weights, and thus independent of the input sample. This might be in part a reason why linear models are sometimes described as 'glass-box' models, particularly when it comes to explaining complex non-linear models via linear surrogate models (e.g. Ribeiro et al., 2016). However, we can see that the Bayes optimal classifier's weights (5), which are the gradient of the optimal decision function f, clearly attribute non-zero importance to the suppressor variable x_2, which is inconsistent with the data-driven definition of feature importance (1).

4.2. Pattern

Haufe et al. (2014) argue that the coefficients of linear models are difficult to interpret. In particular, they may highlight suppressor variables. Instead, the authors propose a transformation to convert weight vectors into parameters $\mathbf{a}$ of a corresponding linear *forward model* $\mathbf{x} = \mathbf{a}f(\mathbf{x}) + \varepsilon$. The solution is provided by the covariance between the model output and each input feature: $a_j = \mathrm{Cov}(x_j, f(\mathbf{x})) = \mathrm{Cov}(x_j, f(\mathbf{x})) = \mathrm{Cov}(x_j, w^\top \mathbf{x})$, for $j = 1, \dots d$, which yields a global importance map

$$e_S(\mathbf{x}) := (\mathrm{Cov}(\mathbf{x}, \mathbf{x})w)_S \tag{6}$$

called *linear activation pattern* (Haufe et al., 2014). For the generative model (2) and the Bayes optimal classifier (5), we obtain

$$e_{\{1\}}(\mathbf{x}) = \alpha s_1^2 (1 - c^2) \,, \quad e_{\{2\}}(\mathbf{x}) = 0 \,. \tag{7}$$

Thus, the pattern approach does not attribute any importance to the suppressor feature x_2.

4.3. Faithfulness and Pixel Flipping

It is widely acknowledged that the correctness of any XAI method as well as the correctness of a given importance map is notoriously hard to assess. This is, because there exists no agreed upon definition of importance as well as because 'true' importances scores are rarely available when it comes to solving problems with learning algorithms. Nonetheless, surrogate metrics have been defined to work around this problem. These metrics are often referred to as 'faithfulness' and, rather than being based on fundamental properties of the data and/or model, they are often based on predictability arguments. Faithfulness is not a well-defined concept and has numerous notions, some of which are tied to specific XAI methods (Jacovi & Goldberg, 2020). As these metrics are often defined algorithmically, they can be regarded as XAI methods in their own right.

The most widely adopted notion of faithfulness is that the omission or obfuscation of an important feature will lead to a decrease in a model's prediction performance. One algorithmic operationalization to assess this is the 'pixel flipping' method (Samek et al., 2017). For linear models, the simplest form of flipping or removing features is just by setting their corresponding weights w_j to zero. With this, we can approximate the classification losses through squared errors as

$$e_S(\mathbf{x}) := \mathbb{E}\big((Y - f_{w_S=0}(\mathbf{x}))^2\big) - \mathbb{E}\big((Y - f(\mathbf{x}))^2\big) \,. \tag{8}$$

For features x_1 and x_2, we obtain

$$e_{\{1\}}(\mathbf{x}) = 2\alpha - \alpha^2 + \alpha^2 s_1^2(2c^2 - 1),$$
$$e_{\{2\}}(\mathbf{x}) = \alpha^2 c^2 s_1^2 \,, \tag{9}$$

as derived in Appendix C. We can observe that for non-zero correlation c, $e_{\{2\}}$ is non-zero; that is, pixel-flipping assigns importance to the suppressor feature x_2.

4.4. Permutation Feature Importance

Proposed by Breiman (2001), the permutation feature importance (PFI) for features x_S measures the drop in classification performance when the associations between x_S and the corresponding class labels is broken via random permutation of the values of x_S. As in pixel flipping, a significant drop in performance defines an important feature (set). Let $\pi_S(\mathbf{x})$ be the randomly permuted version of $\mathbf{x}$, where features with indices in S are permuted and the remaining components are untouched. The randomly permuted features $\pi_S(\mathbf{x})$ and x_S are independent and identically distributed now, which leads to the following approximation of PFI:

$$e_S(\mathbf{x}) \coloneqq \mathbb{E}\big((Y - f(\pi_S(\mathbf{x})))^2\big) - \mathbb{E}\big((Y - f(\mathbf{x}))^2\big) \,. \tag{10}$$

For features x_1 and x_2, we obtain

$$e_{\{1\}}(\mathbf{x}) = 2\alpha + 2\alpha^2 c^2 s_1^2 \quad e_{\{2\}}(\mathbf{x}) = 2\alpha^2 c^2 s_1^2 \,. \tag{11}$$

Thus, similar to faithfulness, PFI assigns non-zero importance to x_2 if $|c| > 0$. This similarity is expanded upon in Appendix D, and a graphical depiction of that behavior is presented for both methods in Figure 3.

4.5. Partial Dependency Plots

Partial dependency (PD) plots are a visualization tool for (learned) high-dimensional functions, aiming to foster a deeper understanding of the relations between their in- and outputs. PD plots also became widely appreciated in the XAI community, where they have been proposed as model-agnostic 'interpretation' or 'explanation' tools (e.g., Molnar, 2020). For a group of features of interest x_S and remaining features x_C, the partial dependency function is the average function

$$e_S(\mathbf{x}) \coloneqq \mathbb{E}_{x_C}\big(f(\mathbf{x})\big) = \int_{\mathbb{R}} f(x_S, x_C) p(x_C) \mathrm{d}x_C \,, \tag{12}$$

where $p(x_C)$ denotes the marginal probability density function, or 'marginal expectation', of x_C. The Bayes optimal decision (5) allows us to directly state the partial dependency functions for features x_1 and x_2 as

$$e_{\{1\}}(\mathbf{x}) = \alpha x_1 \quad e_{\{2\}}(\mathbf{x}) = -\alpha c s_1 s_2^{-1} x_2 \,. \tag{13}$$

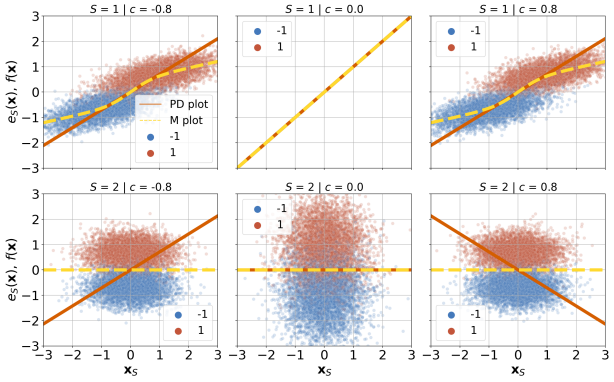


Figure 4. The Partial Dependency Plots (black solid line) and M-plots (red dashed line) for different correlations (columns), and different features x_1 (upper row) and x_2 (bottom row) corresponding for Figure fig:1. The background shows a scatter plot of the corresponding predictions $f(\mathbf{x})$ vs. the feature of interest $\mathbf{x}_S$. The Partial Dependency Plots and M-plots both 'follow' the 'trend' of the samples showing an apparent dependency on the feature x_1 (upper row). For feature x_2 the scatter plots show no structural direction, where we would suspect no 'directional response' from explanation methods like those shown by the M-plots. While PD plots show a dependency on x_2. The figures depict cropped versions; the scatter plots and explanation functions extend beyond the axes' limits for some plots.

These results indicate that the PD function does vary as a function of the suppressor feature x_2. This is further illustrated in Figure 4, which shows PD plots with corresponding scatter plots of the log odds $f(\mathbf{x})$ as a function of the feature of interest x_S. The partial dependency function for x_2 is heavily influenced by the correlation of x_1 and x_2 and only vanishes for $c = 0$, indicating that PD plots are indeed merely a tool to visualize relations between in- and outputs of a function rather than providing 'explanations' compatible with the data-driven definition of feature importance (1). This is in line with works reporting problematic behavior of PD plots when applied to strongly correlated data (Apley & Zhu, 2020; Molnar, 2020).

Marginal Plots For exploratory analyses of tabular datasets, it is common to start by visually assessing simple scatter plots of the target variable as a function of individual features. As such, it is common to fit curves to pairs of in- and outputs (x_1, y) and (x_2, y). This can be done by estimating the conditional expectations $\mathbb{E}\big(Y|X_1 = x_1\big)$ or $\mathbb{E}\big(Y|X_2 = x_2\big)$. A variation of this is to replace output parameters by their model predictions, leading to conditional expectations $e_S(\mathbf{x}) \coloneqq \mathbb{E}\big(f(x_S, x_C)|X_S = x_S\big)$, which were coined M-plots by Apley & Zhu (2020). Their Calculation requires the conditional expectations $\mathbb{E}\big(X_2|X_1 = x_1\big) = \frac{cs_2}{s_1} h(x_1)$ and $\mathbb{E}\big(X_1|X_2 = x_2\big) = \frac{cs_1}{s_2} x_2$, where

$$h(x_1) \coloneqq (x_1 - 1)\vartheta(2x_1/s_1^2) + (x_1 + 1)(1 - \vartheta(2x_1/s_1^2)) \tag{14}$$

and with $\vartheta(x) := (1+\exp(-x))^{-1}$ as the sigmoid function. For the generative model (2) and corresponding Bayes optimal classifier with weights (5) , the conditional expectations for the model given x_1 or x_2, respectively, amount to

$$e_{\{1\}}(\mathbf{x}) = \alpha x_1 - \alpha c^2 h(x_1) \quad e_{\{2\}}(\mathbf{x}) = 0\,. \tag{15}$$

This is shown in Appendix E. Thus, the M-plot assigns a vanishing conditional expectation value to the suppressor variable x_2, which is also confirmed visually in Figure 4 (bottom row). As such, M-plots appear to be suitable tools to identify important features according to definition (1). However, M-plots have been reported to lead to misinterpretations of main effects if y depends on x_1 and x_2, especially when there is an interaction between the two features (Grömping, 2020). Studying the case of interacting features, however, goes beyond the scope of this paper.

4.6. Shapley Values

Another class of XAI methods leverages game theoretic considerations to assign importance scores to individual features. Originally introduced by Shapley (1953), the concept of distributing gains of a coalition game among players fairly was extended by Lipovetsky & Conklin (2001) and Lundberg & Lee (2017), who propose the use of Shapley values (Shapley, 1953) as a procedure to quantify the contribution of a feature to a decision function by considering all possible combinations of features. One can quantify the contribution of a feature x_j to a coalition of features S via the Shapley value

$$e_{\{j\}} = \sum_{S\subseteq[d]\setminus\{j\}} \gamma_d(S)\left[v(S\cup\{j\}) - v(S)\right]\,, \tag{16}$$

with the weighting factor γ_d representing the proportion of coalitions S not including the jth feature, defined as $\gamma_d(S) = {|S|!(d-|S|-1)!}/{d!}$. The value function $v : 2^{[d]} \to \mathbb{R}$, with $v(\emptyset) = 0$, is a set function that assigns a quantity of 'worth' to a coalition and can have many forms. But, for our analysis, we are focusing on the choices made by Lipovetsky & Conklin (2001); Lundberg & Lee (2017) and Aas et al. (2021). In general, the purpose of the value function $v(S) := g_S(\mathbf{x}_S)$, $g_S : \mathbb{R}^{|S|} \to \mathbb{R}$ is to measure the impact of a reduced subset of feature values x_S on the model output. In the following paragraphs, we analyze three different value functions to assess: (1) their impact on feature attribution within the Shapley value framework, and (2) the consequences for models relying on suppressor variables.

Coefficient of Multiple Determination In the Shapley value regression context, Lipovetsky & Conklin (2001) leverage the coefficient of determination (Hoffman, 1960) as a value function, which we decompose as $R^2 = \sum_{j=1}^{d} w_j r_j$. Here, w_j are the learned model weights, and $r_j := (X^\top y)_j$ defines the sample correlation between feature x_j and target y, for standardized features x_j. We can directly define R^2 for a subset of features as $g_S(\mathbf{x}_S) := R^2_S = \sum_{j\in S} w_j r_j$, and utilize it as value function $v(S) := g_S(x_S)$, which can be interpreted as shares of the overall R^2. If we recall the data generation process (2) and consider the covariances $\mathrm{Cov}(Y, X_1) = 1$, and $\mathrm{Cov}(Y, X_2) = 0$, respectively, we can state the marginal Pearson correlations $\rho_{Y,X_1} = (s_1^2+1)^{-1/2}$ and $\rho_{Y,X_2} = 0$ directly, without relying on the sample correlations r_j.

First, we consider the case of calculating the Shapley values $e_{\{j\}}$ with respect to the R^2_S value function, and, as originally intended by Lipovetsky & Conklin (2001), three hypothetically trained models: One bivariate model, here the Bayes rule (5), and two univariate models $f_{\{1\}}(\mathbf{x}) = \hat{w}x_1$ and $f_{\{2\}}(\mathbf{x}) = \tilde{w}x_2$. We specify $e_{\{1\}}, e_{\{2\}}$ as

$$e_{\{1\}}(\mathbf{x}) = \frac{\alpha+1}{2(s_1^2+1)^{1/2}} \quad e_{\{2\}}(\mathbf{x}) = \frac{\alpha-1}{2(s_1^2+1)^{1/2}}, \tag{17}$$

where the rules $f_{\{1\}}(\mathbf{x}) = x_1$ and $f_{\{2\}}(\mathbf{x}) = x_2$, with $\hat{w} = 1$ and $\tilde{w} = 0$ correspond to the optimal decisions for the univariate models. We can observe that the Shapley values are 'governed' by the factor α of the bivariate model. As long as $c \neq 0$, it holds that $\alpha \neq 1$, and this method attributes importance to the suppressor feature x_2. Now, we approximate this procedure using only the bivariate model containing all variables – this is the 'common' scenario, as it can be quite computationally expensive to train new models on many feature subsets. Using the Shapley value framework together with the R^2 measure, we obtain

$$e_{\{1\}}(\mathbf{x}) = \alpha(s_1^2+1)^{-1/2}, \quad e_{\{2\}}(\mathbf{x}) = 0\,. \tag{18}$$

Since $e_{\{2\}} = 0$, we can conclude that R^2 measure in combination with Shapley values is an appropriate value function for assessing feature importance for our linear data generation process (2). This, and the work of the following section, is expanded upon in Appendix F.

SHAP Lundberg & Lee (2017) propose the conditional expectation for a suitable approximation of f, but for computational reasons the authors decided to approximate it with the non-conditional expectation, assuming feature independence. This is called the SHAP (Shapley additive explanations) approach. Later, Aas et al. (2021) suggested an estimation method for the conditional expectation, extending SHAP by actively incorporating potential dependencies among features. We start by defining the value function via the marginal expectation $g_S(\mathbf{x}_S) := \mathbb{E}_{x_C}\big(f(x_S, x_C)\big)$, and with the results of Section 4.5, we obtain the Shapley values

$$e_{\{1\}}(\mathbf{x}) = \alpha x_1, \quad e_{\{2\}}(\mathbf{x}) = -\alpha c s_1 s_2^{-1} x_2\,. \tag{19}$$

This, in essence, resembles the partial dependency functions (13). In a similar way, we calculate the Shapley values for the set function defined via the conditional expectation $g_S(\mathbf{x}_S) \coloneqq \mathbb{E}\big(f(x_S, x_C)|X_S = x_S\big)$ as

$$\begin{aligned} e_{\{1\}}(\mathbf{x}) &= \alpha x_1 - \frac{\alpha c^2}{2} h(x_1) - \frac{\alpha c s_1}{2 s_2} x_2 \\ e_{\{2\}}(\mathbf{x}) &= \frac{\alpha c^2}{2} h(x_1) - \frac{\alpha c s_1}{2 s_2} x_2 \,, \end{aligned} \tag{20}$$

where h is defined in (14). Thus, the Shapley value $e_{\{2\}}$ does not just reflect an attribution of importance to the suppressor variable x_2 but is also affected by feature x_1 if $c \neq 0$.

4.7. Counterfactual Explanations

Wachter et al. (2017) propose an explanation framework based on counterfactual explanations, which we can think of as statements depicting an "alternative world". Formally, we have a given instance $\xi \in \mathbb{R}^d$ and the desired outcome y^*, and try to find a minimizer

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \max_{\lambda} \lambda (f(\mathbf{x}) - y^*)^2 + \delta(\mathbf{x}, \xi) \,, \tag{21}$$

for $\lambda \in \mathbb{R}$ and a suitable distance function δ (Wachter et al., 2017). To find a counterfactual sample according to (21) for our linear model $f(\mathbf{x}) = w^\top \mathbf{x}$, it is sufficient to consider points that are located on the linear decision boundary $f(\mathbf{x}^*) = 0$ of the Bayes optimal classifier (5), since the decision can be flipped in any epsilon-neighborhood around any such point. The closest such counterfactual $\mathbf{x}^*$ for a given instance ξ is the point that has minimal distance to ξ in the Euclidean sense. We can also think of that point as the orthogonal projection of ξ onto the decision hyperplane via its orthogonal subspace

$$\langle \xi - au, u \rangle = 0 \quad \text{with} \quad \mathbf{x}^* \coloneqq \xi - au \,, \tag{22}$$

where u is an element of the orthogonal complement of w, and $a \in \mathbb{R}$. Then, with $u = (cs_1/s_2, 1)^\top$ and $a = \langle \xi, u \rangle / \|u\|_2^2$, the counterfactual explanation $\mathbf{x}^*$ results in

$$\begin{aligned} x_1^* &= \beta(\xi_1 - \xi_2 c s_1 s_2^{-1}) \\ x_2^* &= \beta c s_1 s_2^{-1} (\xi_2 c s_1 s_2^{-1} + \xi_1) \,, \end{aligned} \tag{23}$$

with $\beta \coloneqq ((c s_1 s_2^{-1})^2 + 1)^{-1}$. Thus, to change the decision of the Bayes optimal classifier with minimal interventions, a shift from ξ to $\mathbf{x}^*$ would be required, and this shift would not only involve a change in the informative feature x_1 but also in the suppressor feature x_2 (see also Figure 5 for a graphical depiction). Based on this result it may be, erroneously, concluded from this counterfactual explanation, that feature x_2 has a correlation with or even a causal influence on the classifier decision.

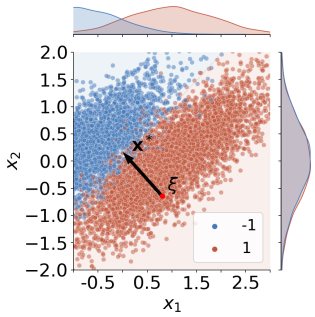


Figure 5. Counterfactual x^* for a given instance of interest ξ in the generative setting $c = 0.8$, $s_1^2 = 0.8$, and $s_2^1 = 0.5$. As can be seen, for $|c| > 0$, reaching a counterfactual decision always involves a manipulation of the suppressor feature x_2.

4.8. FIRM

Another post-hoc method to assess the importance of features of an arbitrary function $f : \mathbb{R}^d \to \mathbb{R}$ is the feature importance ranking measure (FIRM) proposed by Zien et al. (2009). Inspired by the feature sensitivity measure of Friedman (2001), the authors utilize the conditional expectation $\mathbb{E}\big(f(\mathbf{x})|X_S = x_S\big)$ and define the importance ranking measure as

$$e_S(\mathbf{x}) \coloneqq \mathrm{Var}(\mathbb{E}\big(f(\mathbf{x})|X_S\big))^{\frac{1}{2}} \,. \tag{24}$$

Computing this expression, in general, is infeasible since we need access to the data distribution. For the generative model (2) it is possible to prove that

$$\begin{aligned} e_{\{1\}}(\mathbf{x}) &= \alpha \mathrm{Var}(X_1 - c^2 h(X_1))^{\frac{1}{2}} \\ &\geq \frac{\alpha}{2} \left(2\vartheta(2/s_1^2) - 1\right) \\ e_{\{2\}}(\mathbf{x}) &= 0 \,. \end{aligned} \tag{25}$$

A derivation of the lower bound is provided in Appendix G. As also noted in Haufe et al. (2014), the variability of $e_{\{2\}}$ is zero, indicating that FIRM does not assign importance to suppressor features.

4.9. Integrated Gradients

Integrated gradients (Sundararajan et al., 2017) belongs to the family of path methods (Friedman, 2004), which aggregate a model's gradients along a predefined path or curve $\gamma : [0, 1] \to \mathbb{R}^d$ with $\gamma(0) = \mathbf{x}'$ and $\gamma(1) = \mathbf{x}$. If we think of images, then $\mathbf{x} \in \mathbb{R}^d$ can be an image we seek an explanation for, and $\mathbf{x}'$ represents a corresponding baseline image, where a black image $\mathbf{x}' \equiv 0$ is a common choice. For the curve $\gamma : t \mapsto \mathbf{x}' + t(\mathbf{x} - \mathbf{x}')$, a general baseline $\mathbf{x}'$, and a model f, the integrated gradient importance map is given by (Sundararajan et al., 2017)

$$e_{\{j\}}(\mathbf{x}) \coloneqq (x_j - x_j') \int_{[0,1]} \frac{\partial f(\mathbf{x}' + t(\mathbf{x} - \mathbf{x}'))}{\partial x_j} \, \mathrm{d}t. \tag{26}$$

For the Bayes optimal linear classifier (5), the importance scores for features x_1 and x_2 are given by

$$e_{\{1\}}(\mathbf{x}) = \frac{\alpha}{2}(x_1^2 - (\mathbf{x}')^2),$$
$$e_{\{2\}}(\mathbf{x}) = -\frac{\alpha c s_1}{2 s_2}(x_2^2 - (\mathbf{x}')^2)\ , \tag{27}$$

respectively. Thus, independent of the baseline $\mathbf{x}'$ (provided that $\mathbf{x}' \neq \mathbf{x}$), the integrated gradients for the suppressor feature x_2 are non-zero for $|c| > 0$.

4.10. LIME

The idea of LIME (Ribeiro et al., 2016) is to 'explain' a model's decision for a given instance $\mathbf{x}$ by sampling data points in the vicinity of $\mathbf{x}$ and using these samples to build a 'glass-box' model, which is assumed to be more easily interpretable. Typically, a linear model is chosen as a surrogate model. In the scenario studied here, the Bayes optimal model (5) is already linear with non-zero weight w_2. Thus, we would expect that a local linear approximation would show the same behavior. Indeed, Garreau & von Luxburg (2020) show that for a 'linear black-box' model and a Gaussian *i.i.d.* sampling procedure from $N(\mu, \sigma^2 \mathrm{I}_d)$, the local weights $\hat{w}_j$ estimated by LIME are approximately proportional to the partial derivatives of f. Since these derivates reduce to the weights (5) of the Bayes optimal linear classifier in the studied setting, we have $w_j \propto \hat{w}_j$. Therefore, LIME resembles the global model and attributes non-zero importance to the suppressor variable x_2.

4.11. Saliency Maps, LRP and DTD

Saliency map explanations estimate how a prediction $f(\mathbf{x})$ is influenced when moving along a specific direction in the input space. If the direction is along the model's gradient, this is known as sensitivity analysis (Baehrens et al., 2010; Simonyan et al., 2014). Several explanation techniques for neural networks are based on this approach (e.g. DeConvNet and Guided BackProp), primarily distinguishing themselves by their treatment of rectifiers (Kindermans et al., 2018; Zeiler & Fergus, 2014; Springenberg et al., 2015). For single-layer neural networks without rectifiers, that is, linear models, the saliency maps of these explanation methods reduce to the gradient itself (cf. Section 4.1). Layerwise relevance propagation (LRP, Bach et al., 2015) and its generalization Deep Taylor Decomposition (DTD, Montavon et al., 2017) are methods that propagate a quantity termed 'relevance' from output to input neurons backwards through a neural network, following a set of rules. The DTD approach develops, for each layer l of a neural network, a first-order Taylor expansion around a root point $\mathbf{x}_0$, which gives rise to a relevance score for each neuron j with the propagation rule $e_{\{j\}}(\mathbf{x}) \coloneqq R_j^{l-1} = w \odot (\mathbf{x} - \mathbf{x}_0)(w^\top \mathbf{x})^{-1} R_j^l$, where $\odot$ is the Hadamard product. Choosing an appropriate root point is essential in the DTD framework, and Kindermans et al. (2018) notice that by estimating the distractor η and understanding it as root point $x_0 = \eta$, DTD recovers the pattern estimator for linear models proposed by Haufe et al. (2014). Kindermans et al. (2018) derive the signal estimator $S_{\mathbf{a}} = \mathrm{Cov}(\mathbf{x}, y) w^\top \mathbf{x}$, yielding the DTD propagation rule

$$e_{\{j\}}(\mathbf{x}) = (w \odot \mathbf{a})_j \tag{28}$$

for $j = 1, 2$ (cf. Eg. (2)) (see Appendix H). Kindermans et al. (2018) refer to this propagation rule as *PatternArribution*, or *PatterNet* in case where only the activation patterns $\mathbf{a}_j$ are back-propagated. In this case, DTD indeed achieves that no relevance gets attributed to suppressor features in a linear setting. Notably, it has also been shown that in more complex learning scenarios and depending on root points, DTD can generally yield almost any explanation (Kohlbrenner et al., 2020; Montavon et al., 2018; Sixt & Landgraf, 2022).

5. Discussion

The field of XAI is seen as a critical part of a to-be-developed infrastructure that should guarantee the safety of future ML-based high-stake decision systems and create trust in such systems. However, the current state of XAI lacks precise specifications of the problem to be solved by XAI methods. Operationalizations of XAI are, therefore, notoriously difficult to validate theoretically and empirically, which currently prohibit their use for quality assurance.

Two proclaimed use cases of XAI are model and dataset debugging (Lapuschkin et al., 2019), and feature discovery (e.g. Jiménez-Luna et al., 2020; Tran et al., 2021). However, it remains unclear how well contemporary XAI methods can provide evidence in each of these use cases that is beyond anecdotal. If XAI outputs are ill-defined or simply unfit for purpose, this could turn an anticipated benefit of their use even into a disadvantage. For example, characterizing features as important that have no statistical association with the prediction target could give rise to psychological biases and circular reasoning. A pathologist presented with an importance or saliency map for a histological image may try to identify familiar patterns in the map while potentially being tempted to ignore less familiar structures. In the worst case, this could mutually reinforce false prior beliefs between researchers and developers of XAI methods.

Suppressors as Benchmarks for XAI

Wilming et al. (2022) argue that humans often implicitly assume an actual statistical association between a feature and the prediction target when being offered the 'explanation' that the feature in question is important. This gives rise to a purely data-driven yet concrete definition of feature importance based on a statistical dependency on the prediction

target. We use this definition to construct a standard binary classification problem with Gaussian class-conditional distributions. By introducing noise correlations within this model, we create a suppressor variable, which has no statistical relation to the target but whose inclusion in any model will lead to better predictions (Haufe et al., 2014).

We view this simple, yet very insightful, classification problem primarily as a minimal counterexample, where the existence of suppressor variables challenges the assumptions of many XAI methods as well as the assumptions underlying metrics such as faithfulness, which are often considered a gold-standard for quantitative evaluation and an appropriate surrogate for 'correctness'. Indeed, authors have shown empirically that XAI methods can lead to suboptimal 'explanation performance' even when applied to linear data with suppressor variables (Wilming et al., 2022). Here, we complement the study of Wilming et al. (2022) by deriving analytical expression of popular XAI methods employing a two-dimensional linear binary classification problem that has the same problem structure as the 64-dimensional problem presented by Wilming et al. (2022). These analytical expressions allow us to study the factors that lead to non-zero importance attribution, and to expose the mathematical mechanism by which different properties of the data distribution influence XAI methods. Our results demonstrate that outputs of explanation methods must be interpreted in combination with knowledge about the underlying data distribution. Conversely, it may be possible that XAI methods with improved behavior could be designed by reverse-engineering the analytical importance functions e_S.

We found that several XAI methods are incapable of nullifying the suppressor feature, i.e., assigning non-zero importance to it, when correlations between features are present. This is the case for the naive pixel flipping and the PFI methods representing operationalization of faithfulness, but also for actively researched methods like SHAP, LIME, and counterfactuals, as well as partial dependency plots. Note that these methods can typically also not be 'fixed' by just ranking features according to their importance scores and considering only the top features 'important'. In fact, we can devise scenarios where the weight w_2 corresponding to the suppressor variable x_2 is more than twice as high as the weight w_1 (see Appendix B and Haufe et al. (2014)), which may lead to the misconception that the feature x_2 is 'twice' as important as feature x_1. XAI methods based on the Shapley value framework yield particular diverging results, as the strong influence of the value function is reflected in the diversity of analytical solutions. SHAP-like approaches, based on the conditional or marginal expectations 4.6, show how heavily dependent such methods are on the correlation structure of the dataset. In contrast, the M-Plot approach, FIRM, PATTERN, and the Shapley value approach using the R^2 value function, deliver promising results by assigning exactly zero importance to the suppressor variable. This positive result can be attributed to the fact that all methods make explicit use of the statistics of the training data including the correlation structure of the data. This stands in contrast to methods using only the model itself to assign importance to a test sample.

5.1. Limitations

Here we studied a linear generative model and used a univariate data-driven definition of feature importance to design our ground truth data. In real-world scenarios, we do not expect that suppressor variables are always perfectly uncorrelated with the target. In Appendix A we provide deliberations for the case where the suppressor variable $x_2 = \varepsilon z + \eta_2$ consists of a small portion $\varepsilon \in \mathbb{R}$ of the signal z as well. However, in this case, it is not exactly clear what numerical value for the importance we can assume as ground-truth, other than zero. Furthermore, modern machine learning model architectures excel in dealing with highly complex non-linear data involving, among other characteristics, feature interactions. Most XAI methods have been designed to 'explain' the predictions of such complex models. To better understand the behavior of both machine learning models and XAI methods in such complex settings, future work needs to focus on non-linear cases, and develop clear definitions of feature importance in complex settings.

6. Conclusion

We study a two-dimensional linear binary classification problem, where only one feature carries class-specific information. The other feature is a suppressor variable carrying no such information yet improving the performance of the Bayes optimal classifier. Analytically, we derive closed-form solutions for the outputs of popular XAI methods, demonstrating that a considerable number of these methods attribute non-zero importance to the suppressor feature that is independent of the class label. We also find that a number of methods do assign zero significance to that feature by accounting for correlations between the two features. This signifies that even the most simple multivariate models cannot be understood without knowing essential properties of the distribution of the data they were trained on.

Acknowledgements

This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 758985), the German Federal Ministry for Economy and Climate Action (BMWK) in the frame of the QI-Digital Initiative, and the Heidenhain Foundation. We thank Jakob Runge for a fruitful discussion.

References

Aas, K., Jullum, M., and Løland, A. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 9525–9536. Curran Associates Inc., 2018.

Agarwal, C., Krishna, S., Saxena, E., Pawelczyk, M., Johnson, N., Puri, I., Zitnik, M., and Lakkaraju, H. OpenXAI: Towards a transparent evaluation of model explanations. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Alvarez-Melis, D. and Jaakkola, T. S. On the Robustness of Interpretability Methods. 2018.

Apley, D. W. and Zhu, J. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B*, 82(4): 1059–1086, September 2020.

Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11: 1803–1831, aug 2010.

Binder, A., Bach, S., Montavon, G., Müller, K.-R., and Samek, W. Layer-Wise Relevance Propagation for Deep Neural Network Architectures. In Kim, K. J. and Joukov, N. (eds.), *Information Science and Applications (ICISA) 2016*, Lecture Notes in Electrical Engineering, pp. 913–922. Springer, 2016.

Breiman, L. Random Forests. *Machine Learning*, 45(1): 5–32, 2001.

Conger, A. J. A Revised Definition for Suppressor Variables: A Guide To Their Identification and Interpretation , A Revised Definition for Suppressor Variables: A Guide To Their Identification and Interpretation. *Educational and Psychological Measurement*, 34(1):35–46, 1974.

Doshi-Velez, F. and Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*, 2017.

Fisher, A., Rudin, C., and Dominici, F. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

Friedman, E. J. Paths and consistency in additive cost sharing. *International Journal of Games Theory*, 32(4), 2004.

Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5): 1189–1232, 2001.

Friedman, L. and Wall, M. Graphical Views of Suppression and Multicollinearity in Multiple Linear Regression. *The American Statistician*, 59(2):127–136, 2005.

Garreau, D. and von Luxburg, U. Explaining the explainer: A first theoretical analysis of lime. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1287–1296. PMLR, August 2020.

Gevrey, M., Dimopoulos, I., and Lek, S. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling*, 160(3):249–264, 2003.

Grömping, U. Model-agnostic effects plots for interpreting machine learning models. *Reports in Mathematics, Physics and Chemistry*, (1/2020), 2020.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., and Bießmann, F. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, 87:96–110, 2014.

Hoffman, P. J. The paramorphic representation of clinical judgment. *Psychological Bulletin*, 57(2):116–131, 1960.

Jacovi, A. and Goldberg, Y. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, Online, July 2020. Association for Computational Linguistics.

Janzing, D., Minorics, L., and Bloebaum, P. Feature relevance quantification in explainable AI: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916. PMLR, 2020.

Jiménez-Luna, J., Grisoni, F., and Schneider, G. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584, 2020.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, pp. 2668–2677. PMLR, 2018.

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., and Dähne, S. Learning how to explain neural networks: Patternnet and patternattribution. In *International Conference on Learning Representations*, 2018.

Kohlbrenner, M., Bauer, A., Nakajima, S., Binder, A., Samek, W., and Lapuschkin, S. Towards best practice in explaining neural network decisions with lrp. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, 2019.

Lipovetsky, S. and Conklin, M. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

Lundberg, S. M. and Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017.

Molnar, C. *Interpretable Machine Learning*. Independently published, 2020.

Montavon, G., Bach, S., Binder, A., Samek, W., and Müller, K.-R. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition. *Pattern Recognition*, 65:211–222, 2017.

Montavon, G., Samek, W., and Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018. ISSN 1051-2004.

Ribeiro, M. T., Singh, S., and Guestrin, C. " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, November 2017.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Shapley, L. S. A Value for n-Person Games. In Kuhn, H. W. and Tucker, A. W. (eds.), *Contributions to the Theory of Games (AM-28), Volume II*, pp. 307–318. Princeton University Press, 1953.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning Important Features Through Propagating Activation Differences. 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

Sixt, L. and Landgraf, T. A rigorous study of the deep taylor decomposition. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Sixt, L., Granz, M., and Landgraf, T. When explanations lie: Why many modified BP attributions fail. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9046–9057. PMLR, 13–18 Jul 2020.

Springenberg, J., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic Attribution for Deep Networks. In *ICML*, 2017.

Tran, K. A., Kondrashova, O., Bradley, A., Williams, E. D., Pearson, J. V., and Waddell, N. Deep learning in cancer diagnosis, prognosis and treatment selection. *Genome Medicine*, 13(1):152, 2021.

Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31:841, 2017.

Wilming, R., Budding, C., Müller, K.-R., and Haufe, S. Scrutinizing xai using linear ground-truth data with suppressor variables. *Machine learning*, 111(5):1903–1923, 2022.

Yang, M. and Kim, B. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*, 2019.

Zeiler, M. D. and Fergus, R. Visualizing and Understanding Convolutional Networks. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. (eds.), *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pp. 818–833. Springer International Publishing, 2014.

Zien, A., Krämer, N., Sonnenburg, S., and Rätsch, G. The Feature Importance Ranking Measure. In Buntine, W., Grobelnik, M., Mladenić, D., and Shawe-Taylor, J. (eds.), *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pp. 694–709. Springer, 2009.

Štrumbelj, E. and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3): 647–665, 2014.

A. Bayes optimal classifier

The generative data model, defined in (2), induces a binary classification problem yielding two class-conditional densities which are both multivariate normal, with $\mathbf{X} \mid Y = y \sim N(\mu_i, \Sigma)$ for $y \in \{-1, 1\}$ and $i = 1, 2$, and have identical covariance matrix $\Sigma \in \mathbb{R}^{2\times 2}$ and expectations $\mu_1 = (1, 0)^\top$ and $\mu_2 = (-1, 0)^\top$. We solve the classification task in a Bayes optimal way if we assign $\mathbf{x}$ either to class $Y = 1$ or to class $Y = -1$ based on the minimal squared Mahalanobis distance $\delta^2(\mathbf{x}, \mu_i) = (\mathbf{x} - \mu_i)^\top \Sigma^{-1}(\mathbf{x} - \mu_i)$ between $\mathbf{x}$ and the two class means $\mu_i, i = 1, 2$. As described we have equal covariance matrices Σ for both classes, thus, the Bayes rule becomes linear and we can assign $\mathbf{x}$ to class $Y = 1$, if $\mathbf{w}^\top(\mathbf{x} - \mu) \geq 0$, where $\mathbf{w} \coloneqq \Sigma^{-1}(\mu_1 - \mu_2)$ and $\mu \coloneqq \frac{1}{2}(\mu_1 + \mu_2)$. The concrete form of the Bayes optimal rule $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ with weights $\mathbf{w}^\top = (w_1, w_2)^\top$ is determined by the coefficients

$$w_1 = \alpha, \quad w_2 = -\alpha c s_1 / s_2 \tag{29}$$

for $\alpha \coloneqq (1 + (cs_1/s_2)^2)^{-\frac{1}{2}}$ and $||w||_2 = 1$ and $b = 0$. The inverse of the covariance matrix Σ is given by

$$\Sigma^{-1} = \frac{1}{s_1^2 s_2^2 (1 - c^2)} \begin{bmatrix} s_2^2 & -cs_1 s_2 \\ -cs_1 s_2 & s_1^2 \end{bmatrix}. \tag{30}$$

Furthermore, we consider a version of generative data model (2) where we parameterize, via a scalar $\varepsilon \in \mathbb{R}$, the dependency between the suppressor variable and the target

$$\mathbf{x} = \mathbf{a}_\varepsilon z + \eta, \quad y = z, \tag{31}$$

with $Z \sim Rademacher(1/2)$, $\mathbf{a}_\varepsilon = (1, \varepsilon)^\top$ and $H \sim N(\mathbf{0}, \Sigma)$. The induces binary classification problem slightly changes with class-conditional distributions $\mathbf{X} \mid Y = y \sim N(\mu_i, \Sigma)$ for $y \in \{-1, 1\}$ and $i = 1, 2$ and updated expectations $\mu_1 = (1, \varepsilon)^\top$ and $\mu_2 = (-1, \varepsilon)^\top$. Then for the optimal Bayes rule, we yield the weights and offset

$$w_1 = \alpha, \quad w_2 = -\alpha c s_1 / s_2 \quad b = \varepsilon \alpha c s_1 / s_2 \,. \tag{32}$$

B. Ranking

Let the correlation $c = -0.8$ and the varainces be $s_1^2 = 1$ and $s_2^2 = 0.15$, then we yield the coefficients $w_1 \approx 0.42$ and $w_2 \approx 0.90$. Using the coefficient as importance scores would rank the suppressor variable x_2 twice as high as the class-dependent variable x_1.

C. Faithfulness

Throughout the appendix, let the random variables $\mathbf{X} = (X_1, X_2)$ and Y be defined as in Section 3 and $f(\mathbf{x}) = w_1 x_1 + w_2 x_2$. For the Pixel-Flipping method, we consider the error

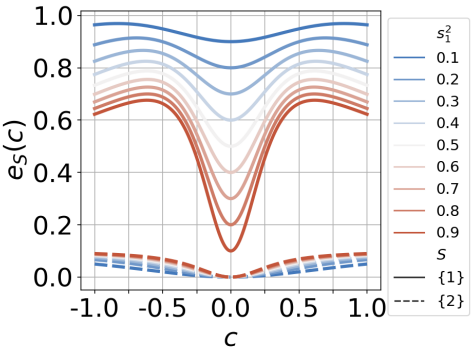
$$e_S(\mathbf{x}) \coloneqq \mathbb{E}\big((Y - f_{w_S=0}(\mathbf{x}))^2\big) - \mathbb{E}\big((Y - f(\mathbf{x}))^2\big). \tag{33}$$

Now, let us consider

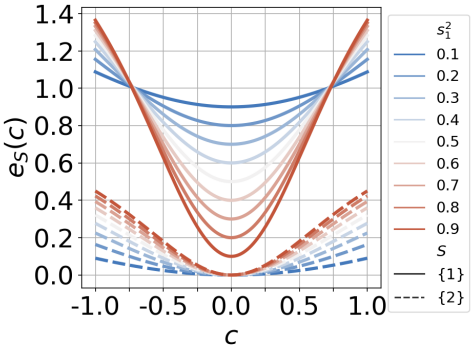
$$\begin{aligned} \mathbb{E}\big((Y - f(\mathbf{x}))^2\big) &= \mathrm{P}(Y = 1)\mathbb{E}\big((Y - f(\mathbf{x}))^2 \mid Y = 1\big) \\ &\quad + \mathrm{P}(Y = -1)\mathbb{E}\big((Y - f(\mathbf{x}))^2 \mid Y = -1\big) \\ &= \frac{1}{2}\mathbb{E}\big((1 - f(\mathbf{x}))^2 \mid Y = 1\big) + \frac{1}{2}\mathbb{E}\big((1 + f(\mathbf{x}))^2 \mid Y = -1\big) \\ &= 1 - 2w_1 + w_1^2(s_1^1 + 1) + w_2^2 s_2^2 + 2w_1 w_2 c s_1 s_2. \end{aligned} \tag{34}$$

We take a closer look at the conditional expectation $\mathbb{E}\big((1 - f(\mathbf{x}))^2 \mid Y = 1\big)$ and observe

$$\begin{aligned} \mathbb{E}\big((1 - f(\mathbf{x}))^2 \mid Y = 1\big) &= \mathbb{E}\big(1 \mid Y = 1\big) - 2\mathbb{E}\big(f(\mathbf{x}) \mid Y = 1\big) + \mathbb{E}\big(f(\mathbf{x})^2 \mid Y = 1\big) \\ &= 1 - 2w_1 + w_1^2(s_1^1 + 1) + w_2^2 s_2^2 + 2w_1 w_2 c s_1 s_2, \end{aligned} \tag{35}$$



(a) Faithfulness for $s_2^2 = 0.1$



(b) Faithfulness for $s_2^2 = 0.9$

Figure 6.

where

$$\begin{aligned}\mathbb{E}\big(f(\mathbf{x}) \mid Y=1\big) &= w_1,\\ \mathbb{E}\big(f(\mathbf{x})^2 \mid Y=1\big) &= \mathrm{Var}\left(f(\mathbf{x}) \mid Y=1\right) + \mathbb{E}\big(f(\mathbf{x}) \mid Y=1\big)^2\\ &= w_1^2 s_1^2 + w_2^2 s_2^2 + w_1^2 + 2w_1 w_2 c s_1 s_2.\end{aligned} \tag{36}$$

By using (35) and (36) we can compute the value for $\mathbb{E}\big((1+f(\mathbf{x}))^2 \mid Y=-1\big)$ analogously

$$\mathbb{E}\big((1+f(\mathbf{x}))^2 \mid Y=-1\big) = 1 - 2w_1 + w_1^2(s_1^1+1) + w_2^2 s_2^2 + 2w_1 w_2 c s_1 s_2. \tag{37}$$

Similarly, we reive the results for $\mathbb{E}\big((Y - f_{w_S=0}(\mathbf{x}))^2\big)$ using the obfuscated decision rule $f_{w_S=0}$. Finally, for the weights $w_1 = \alpha$ and $w_2 = \alpha c s_1 / s_2$ we yield the importance values (9)

$$\begin{aligned}e_{\{1\}}(\mathbf{x}) &= \mathbb{E}\big((Y - f_{w_1=0}(\mathbf{x}))^2\big) - \mathbb{E}\big((Y - f(\mathbf{x}))^2\big)\\ &= 1 + w_2^2 s_2^2 - 1 - 2w_1 + w_1^2(s_1^1+1) + w_2^2 s_2^2 + 2w_1 w_2 c s_1 s_2\\ &= 2\alpha - \alpha^2 + \alpha^2 s_1^2 (2c^2 - 1)\\ e_{\{2\}}(\mathbf{x}) &= \mathbb{E}\big((Y - f_{w_2=0}(\mathbf{x}))^2\big) - \mathbb{E}\big((Y - f(\mathbf{x}))^2\big)\\ &= 1 - 2w_1 + w_1^2(s_1^2+1) - 1 - 2w_1 + w_1^2(s_1^1+1) + w_2^2 s_2^2 + 2w_1 w_2 c s_1 s_2\\ &= \alpha^2 s_1^2 c^2\end{aligned} \tag{38}$$

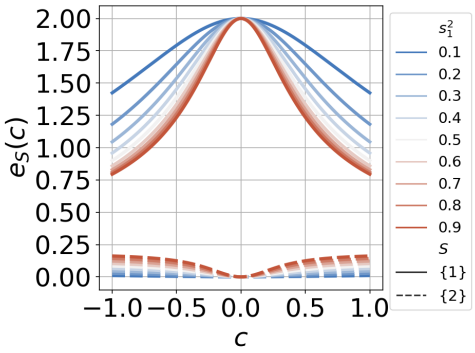
D. Permutation Feature Importance

Analogously to the computation of the Faithfulness values we can compute the Permutation Feature Importance values. Note, we compute the Permutation Importance value in a relatively naive way, where we understand the permutation $\pi_S(\mathbf{x})$ as 'breaking' the correlations with the remaining features and the target. We do not provide a probabilistic definition of a permutation operator. We just use a direct translation of how we would implement feature permutation in practice. We already computed the value of $\mathbb{E}\big((Y - f(\mathbf{x}))^2\big)$, therefore it is sufficient to consider

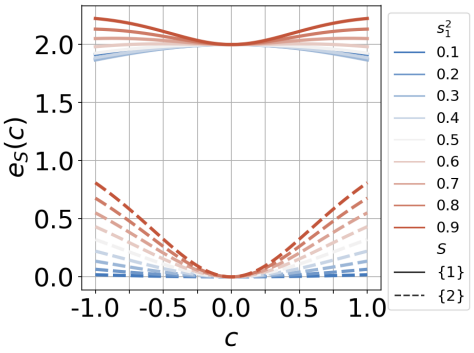
$$\begin{aligned}\mathbb{E}\big((Y - f(\pi_{\{1\}}(\mathbf{x})))^2\big) &= \mathbb{E}\big(Y^2\big) - 2\mathbb{E}\big(Y f(\pi_{\{1\}}(\mathbf{x}))\big) + \mathbb{E}\big(f(\pi_{\{1\}}(\mathbf{x}))^2\big)\\ &= 1 + w_1^2(s_1^2+1) + w_2^2 s_2^2,\end{aligned} \tag{39}$$

where $\mathbb{E}\big(Y^2\big) = 1$ by the properties of the Rademacher distribution and $\mathbb{E}\big(Y f(\pi_{\{1\}}(\mathbf{x}))\big) = 0$, because we set $\mathrm{Cov}(Y, \pi_{\{1\}}(X_1)) = 0$ and $\mathrm{Cov}(\pi_{\{1\}}(X_1), X_2) = 0$. Moreover

$$\begin{aligned}\mathbb{E}\big(f(\pi_{\{1\}}(\mathbf{x}))^2\big) &= w_1^2 \mathbb{E}\big(\pi_{\{1\}}(x_1)\big) - 2w_1 w_2 \mathbb{E}\big(\pi_{\{1\}}(x_1) x_2\big) + w_2^2 \mathbb{E}\big(x_2^2\big)\\ &= w_1^2(s_1^2+1) + w_2^2 s_2^2.\end{aligned} \tag{40}$$



(a) Permutation Importance for $s_2^2 = 0.1$



(b) Permutation Importance for $s_2^2 = 0.9$

Figure 7.

Analogously for $\pi_{\{2\}}(\mathbf{x})$ we obtain

$$\mathbb{E}\big((Y - f(\pi_{\{2\}}(\mathbf{x})))^2\big) = 1 - 2w_1 + w_1^2(s_1^2 + 1) + w_2^2 s_2^2. \tag{41}$$

E. Conditional expectations

In order to compute the M-plots $\mathbb{E}\big(w_1 x_1 + w_2 x_2 | X_S = x_S\big)$, we will first consider the conditional expectations $\mathbb{E}\big(X_1 | X_2 = x_2\big)$ and $\mathbb{E}\big(X_2 | X_1 = x_1\big)$. Starting with $\mathbb{E}\big(X_1 | X_2 = x_2\big)$, by the law of total expectation, we can write

$$\begin{aligned}\mathbb{E}\big(X_1 | X_2 = x_2\big) = \mathbb{E}\big(X_1 \mid X_2 = x_2, Y = 1\big)\,\mathrm{P}(Y = 1 \mid X_2 = x_2) \\ + \mathbb{E}\big(X_1 \mid X_2 = x_2, Y = -1\big)\,\mathrm{P}(Y = -1 \mid X_2 = x_2).\end{aligned} \tag{42}$$

Since $\mathrm{P}(X_1, X_2 \mid Y = 1) \sim \mathcal{N}((1, 0)^\top, \Sigma)$ and $\mathrm{P}(X_1, X_2 \mid Y = -1) \sim \mathcal{N}((-1, 0)^\top, \Sigma)$, we can straightfowardly compute the conditional expectations

$$\begin{aligned}\mathbb{E}\big(X_1 \mid X_2 = x_2, Y = 1\big) &= \frac{c s_1}{s_2} x_2 \\ \mathbb{E}\big(X_1 \mid X_2 = x_2, Y = -1\big) &= \frac{c s_1}{s_2} x_2.\end{aligned} \tag{43}$$

And by Bayes' theorem and using the notation of the joint density p (see (4)), we can write

$$\mathrm{P}(Y = 1 \mid X_2 = x_2) = \frac{p_{1,X_2}(x_2)\,\mathrm{P}(Y = 1)}{p_{1,X_2}(x_2)} = \frac{1}{2}, \tag{44}$$

with marginal density p_{1,X_2} of the marginal distribution of p in X_2, i.e. $p_{1,X_2}(x_2) = \varphi(x_2/s_1)$ with standard normal density φ. In accordance with (44) we obtain $\mathrm{P}(Y = -1 \mid X_2 = x_2) = 1/2$. Combining the results (42), (43) and (44) we yield

$$\mathbb{E}\big(X_1 | X_2 = x_2\big) = \frac{c s_1}{s_2} x_2. \tag{45}$$

Again, by the law of total expectation, for $\mathbb{E}\big(X_1 | X_2 = x_2\big)$, we can compute the conditional expectations $\mathbb{E}\big(X_2 \mid X_1 = x_1, Y = 1\big)$ and $\mathbb{E}\big(X_2 \mid X_1 = x_1, Y = -1\big)$ in a straightforward manner by the argument used for (43)

$$\begin{aligned}\mathbb{E}\big(X_2 \mid X_1 = x_1, Y = 1\big) &= \frac{c s_2}{s_1}(x_1 - 1) \\ \mathbb{E}\big(X_2 \mid X_1 = x_1, Y = -1\big) &= \frac{c s_2}{s_1}(x_2 + 1).\end{aligned} \tag{46}$$

Furthermore, by Bayes' theorem, we know

$$\begin{aligned} \mathrm{P}(Y=1 \mid X_1 = x_1) &= \frac{p_{1,X_1}(x_1)\,\mathrm{P}(Y=1)}{\frac{1}{2}p_{1,X_1}(x_1) + \frac{1}{2}p_{2,X_1}(x_1)} \\ &= \vartheta(2x_1/s_1^2), \end{aligned} \tag{47}$$

where p_{1,X_1} and p_{2,X_1} are the marginal densities of the marginal distribution of p in X_1, namely $p_{1,X_1}(x_1) = \varphi((x_1-1)/s_1)$ and $p_{1,X_1}(x_1) = \varphi((x_1+1)/s_1)$, and sigmoid function $\vartheta : \mathbb{R} \to \mathbb{R}$. Similarly, $\mathrm{P}(Y=-1 \mid X_1 = x_1) = 1 - \vartheta(2x_1/s_1^2)$. The combination of (46) and (47) amounts to (15).

F. Shapley values

With $d = 2$ and set of feature indices $[d]$, we consider the Shapley values

$$e_{\{j\}} = \sum_{S \subseteq [d] \setminus \{j\}} \gamma_d(S)\left[v(S \cup \{j\}) - v(S)\right]. \tag{48}$$

We define the value function v via a set function $v(S) \coloneqq g_S(\mathbf{x}_S)$, $g_S : \mathbb{R}^{|S|} \to \mathbb{R}$. For feature sets with two features, the Shapley values are given by

$$\begin{aligned} e_{\{1\}} &= \frac{1}{2}\big(g_{\emptyset\cup\{1\}} - g_\emptyset\big) + \frac{1}{2}\big(g_{\{1,2\}} - g_{\{2\}}\big) \\ e_{\{2\}} &= \frac{1}{2}\big(g_{\emptyset\cup\{2\}} - g_\emptyset\big) + \frac{1}{2}\big(g_{\{1,2\}} - g_{\{1\}}\big), \end{aligned} \tag{49}$$

with their corresponding weights

$$\gamma_2(\emptyset) = 1/2, \quad \gamma_2(\{1\}) = 1/2, \quad \gamma_2(\{2\}) = 1/2. \tag{50}$$

In the considered scenarios we use different set functions depending on the particular XAI approach, but set $g_\emptyset = 0$. Now, we state the value functions we used to compute the corresponding Shapley values for each feature.

Coefficient of multiple determination For Paragraph 4.6 we employed the value function $g_S(\mathbf{x}_S) \coloneqq R^2_S = \sum_{j \in S} w_j r_j$ and for the corresponding subsets S we yield

$$\begin{aligned} g_\emptyset(\mathbf{x}) = 0 \quad g_{\{1,2\}}(\mathbf{x}) = w_1/(s_1^2+1)^{1/2} \quad g_{\{2\}}(\mathbf{x}) = 0 \\ g_{\{1\}}(\mathbf{x}) = w_1/(s_1^2+1)^{1/2} \\ g_{\{1\}}(\mathbf{x}) = \hat{w}_1/(s_1^2+1)^{1/2} \text{ (for the 'sub-model' } f_{\{1\}}(x_1) = \hat{w}_1 x_1). \end{aligned} \tag{51}$$

SHAP Using the set function $g_S(\mathbf{x}_S) \coloneqq \mathbb{E}_{x_C}(f(\mathbf{x}_S, \mathbf{x}_C))$ and computing the Shapley values for a linear model reduces to

$$g_{S\cup\{j\}}(\mathbf{x}_S) - g_S(\mathbf{x}_S) = w_j(x_j - \mathbb{E}(x_j)), \tag{52}$$

therefore, do not dependent on S (cf. Štrumbelj & Kononenko, 2014). For the set function $g_S(\mathbf{x}_S) \coloneqq \mathbb{E}\big(f(x_S, x_C)|X_S = x_S\big)$ consider the derivations provided by Aas et al. (2021).

G. FIRM

To derive the lower bound for $e_{\{1\}}(\mathbf{x})$ we first observe that for $x_1 > 0$

$$\begin{aligned} x_1 - h(x_1) &= x_1 - \left[(x_1-1)\vartheta(2x_1/s_1^2) + (x_1+1)(1-\vartheta(2x_1/s_1^2))\right] \\ &= 2\vartheta(2x_1/s_1^2) - 1 \\ &> 0 \end{aligned} \tag{53}$$

where $\vartheta(x) := (1 + \exp(-x))^{-1}$ denotes the sigmoid function as before. Thus, we may estimate $e_{\{1\}}(\mathbf{x})$ from below by

$$\begin{aligned} e_{\{1\}}(\mathbf{x})^2 &= \mathrm{Var}(\mathbb{E}\big(f(\mathbf{x}) \mid X_1\big)) \\ &= \alpha^2 \mathbb{E}\big((X_1 - c^2 h(X_1))^2\big) \\ &\geq \alpha^2 \mathbb{E}\big((X_1 - h(X_1))^2 \mid X_1 > 1\big)\, \mathrm{P}(X_1 > 1) \\ &= \alpha^2 \mathbb{E}\big((2\vartheta(2X_1/s_1^2) - 1)^2 \mid X_1 > 1\big)\, \mathrm{P}(X_1 > 1) \\ &\geq \alpha^2 (2\vartheta(2/s_1^2) - 1)^2\, \mathrm{P}(X_1 > 1 \mid Y = 1)\, \mathrm{P}(Y = 1) \\ &= \frac{\alpha^2}{4} \left(2\vartheta(2/s_1^2) - 1\right)^2 . \end{aligned} \tag{54}$$

Taking the square root on either side now yields the lower bound

$$e_{\{1\}}(\mathbf{x}) \geq \frac{\alpha}{2} \left(2\vartheta(2/s_1^2) - 1\right) . \tag{55}$$

H. LRP and DTD

The Deep Taylor decomposition (DTD, Montavon et al., 2017) as a generalization of layerwise relevance propagation (LRP, Bach et al., 2015) summarizes this family of explanation methods via the general propagation rule

$$e_{\{j\}}(\mathbf{x}) := R_j^{l-1} = \frac{w \odot (\mathbf{x} - \mathbf{x}_0)}{w^\top \mathbf{x}} R_j^l \, . \tag{56}$$

In applications of DTD, the choice of a suitable root point $\mathbf{x}_0$ is of critical importance. Here, Kindermans et al. (2018) observe that in order to extract the 'signal' from the data we have to remove the distractor η by choosing a signal estimator $S_\mathbf{a}(\mathbf{x}) = x - \eta$, i.e. we pick the root point $\mathbf{x}_0 = \eta$ and implicitly estimate the distractor $\hat{\eta} = x - S_\mathbf{a}(x)$. Furthermore, a good signal estimator should yield high values of the quality measure

$$\rho(S_\mathbf{a}) := 1 - \max_v \frac{v^\top \mathrm{Cov}(\hat{\eta}, y)}{\left(\sigma^2_{v^\top \hat{\eta}} \sigma_y^2\right)^{1/2}} , \tag{57}$$

i.e. a signal estimator $S_\mathbf{a}$ is optimal if we have a vanishing correlation between $\hat{\eta}$ and y. With these observations Kindermans et al. (2018) assume a linear dependency between the signal $S_\mathbf{a}$ and the target y yielding a signal estimator

$$S_\mathbf{a}(x) = \mathbf{a} w^\top \mathbf{x} \, . \tag{58}$$

Now, consider

$$\begin{aligned} & \mathrm{Cov}(\hat{\eta}, y) = 0 \\ \Leftrightarrow\ & \mathrm{Cov}(\mathbf{x} - S_\mathbf{a}(\mathbf{x}), y) = 0 \\ \Leftrightarrow\ & \mathrm{Cov}(\mathbf{x}, y) = \mathrm{Cov}(S_\mathbf{a}(\mathbf{x}), y) \\ \Leftrightarrow\ & \mathrm{Cov}(\mathbf{x}, y) = \mathrm{Cov}(\mathbf{a} w^\top \mathbf{x}, y) \, , \end{aligned} \tag{59}$$

and for $\mathrm{Cov}(\mathbf{a} w^\top \mathbf{x}, y) = \mathbf{a}\, \mathrm{Cov}(y, y)$ we yield the activation pattern $\mathbf{a} = \mathrm{Cov}(\mathbf{x}, y)/\sigma_y^2$. Applying the signal estimater $S_\mathbf{a}$ to the propagation rule (56) means replacing $(\mathbf{x} - \mathbf{x}_0)$ by $(\mathbf{a} w^\top \mathbf{x})$, and with $R_j^l = \mathbb{I}_{f(\mathbf{x})>0}$ we yield the explanation

$$e_{\{j\}}(\mathbf{x}) = \left(\frac{w \odot (\mathbf{a} w^\top \mathbf{x})}{w^\top \mathbf{x}} \right)_j = (w \odot \mathbf{a})_j \ , \tag{60}$$

which is called the *PatternAttribution* method by Kindermans et al. (2018). The *PatterNet* method only back-propagates the activation patterns $\mathbf{a}_j$.