Visual Explanations of the Explainers, to Gain Insight into Predictions from High-Dimensional Models

Janith Wanniarachchi
Monash University

Dianne Cook
Monash University

Kate Saunders
Monash University

Patricia Menendez
University of Melbourne

Thiyanga Talagala
University of Sri Jayewardenepura

Abstract Explainable Artificial Intelligence (XAI) methods such as SHAP, Counterfactuals and Anchors are used on black box models to understand the contributions from features. Traditional analysis of XAI methods focuses primarily on using numerical values to assess feature contributions. Visualising these explanations together with the observed data gives a clear picture of which facets of the model are explained by each method. In this paper, We propose a set of geometric representations for visualising XAI methods in a two dimensional data space. The proposed representations aim to assist in understanding the contribution of each feature as abstract concepts represented through XAI methods (i.e. the contribution indicated by SHAP is a different contribution to that indicated by LIME). We hope these representations will also aid in researchers who are applying XAI methods to their research problem without being affected by confirmation bias in picking methods that affirm their prior beliefs.

Table of contents

1 Introduction

Machine learning methods have become popular in tasks such as genome sequencing, climate modeling and natural language tasks. This shift is largely driven by the non-parametric nature of machine learning algorithms, which allows them to map connections between input variables with minimal human intervention or prior assumptions. However, since the analytical connections are learned implicitly rather than predetermined by the user, the underlying model behavior becomes highly opaque. As these models are increasingly deployed in public facing scenarios, understanding their inner workings is of paramount importance.

Explainable Artificial Intelligence (XAI) methods emerged in response to the increasing complexity and opaqueness of machine learning models and the need to understand how they work, particularly deep learning models. A central aim of XAI is to provide algorithmic and mathematical methods that gather insights into how black box models make decisions, addressing concerns related to trust, transparency, and accountability in AI systems [1].

XAI methods are primarily categorized based on their scope and robustness [2]. Based on the scope, XAI methods are categorized as global interpretability methods and local interpretability methods [3]. local methods are used to show the model's decision process or feature influence for a specific observation of interest. Global interpretability methods are developed by aggregating local explanations to demonstrate the effect of predictor variables on the entire dataset. Common examples for global methods are Partial Dependency Plots [4], [5], Accumulated Local Effect plots [6] while for local methods, SHAP [7], Anchors[8] and Counterfactuals[9] are common examples.

XAI methods are also categorized by whether the method is capable of explaining only a set family of machine learning models (e.g. tree based models, neural networks) or any black box model regardless of underlying architecture. In this categorization, XAI methods are divided into two categories: model-specific and model-agnostic. Model specific methods use underlying infor-

mation about the model's architecture to provide explanations while the model agnostic methods use the model as a function of the input data to approximate the internal workings of the model. Most XAI methods are model agnostic with a few model specific methods available specifically to deep learning models such as saliency maps [10] and learned feature visualization [11] where information on the internal gradients are available.

Model agnostic methods are robust to a wider range of models and therefore can be used in a multitude of contexts, while local interpretability methods provide a narrower inspection into the decision process of black box models.

Despite the advances in the current landscape of XAI methods, the explanations from these methods are inspected in isolation from the components that lead up to the explanation. Currently many of the XAI visualisation methods such as ModelStudio [12] focus primarily on showing feature contributions of the model based on different explanation methods [13]. While these visualisations are effective in communcating the numerical values from explanations, these visualisations also suffer from being unable to communicate how the underlying explanations relate to the model's view of the data and the data itself.

This isolation brings forward the issue of practitioners having to navigate a fragmented ecosystem of XAI techniques without a unified framework to guide them [7]. As different XAI methods probe different mathematical facets of a model, they inherently capture different aspects of the decision-making process. Consequently, when multiple methods are applied to the same pipeline, they frequently yield conflicting explanations. Resolving these disagreements currently forces practitioners to manually reconcile abstract algorithmic formulas with raw numerical outputs.

To bridge this gap, this paper introduces a novel framework of geometric representations for XAI methods. By mapping explanations to a shared geometric space, we visually unify the relationships between the raw data, the model's internal representations, and the resulting feature attributions, thereby providing a intuitive method to resolve algorithmic disagreements.

The remainder of this paper is organized as follows. In Section 2, we establish a foundational baseline by outlining a generalized predictive modeling workflow, followed by our proposed visualization approaches for this pipeline in Section 2.3. To ground our geometric framework, Section 2.1 introduces the benchmark datasets and Section 2.2 introduces model architectures used throughout our evaluations. Subsequently, Section 4 establishes the formal mathematical definitions of the XAI methods under review, setting the stage where we present the conceptual design and implementation of geometric XAI visualizations. Finally, Section 5 discusses the potential solutions to settle the disagreement between XAI explanations.

2 Understanding the building blocks behind the prediction workflow

In general, the workflow for prediction (as shown in Figure 1) is to take the set of observations and divide them into training and testing datasets. The training dataset is used to train the model, while the testing dataset is held out to evaluate how well the model generalizes to unseen data.

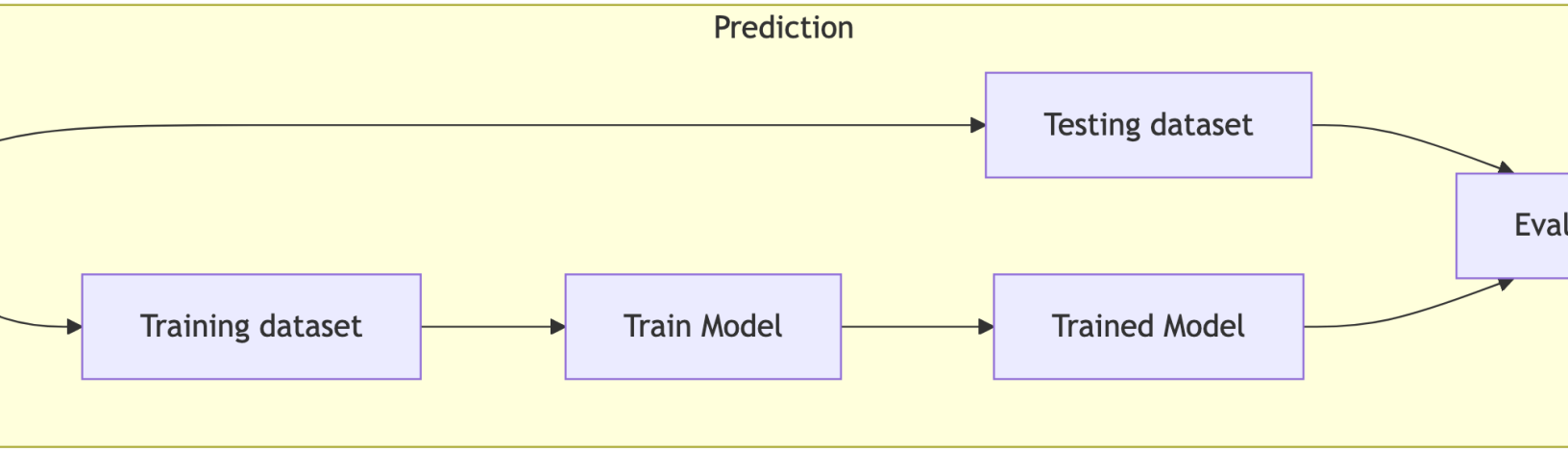

Figure 1: The workflow when the goal is to optimize for prediction

2.1 The dataset

In this paper, to demonstrate our solution we are going to be focusing on a simple two dimensional classification problem. The dataset contains two numeric variables each containing 5000 samples from a random uniform distribution between −1,1. The decision boundary for the two numeric variables is set based on the following rule.

$$f(x,y) = \begin{cases} A, \text{if } \ x < -0.5 \wedge y < -0.3 \\ A, \text{if } \ -0.5 \leq x < 0.4 \wedge y < 0.1 + 0.8x \\ A, \text{if } \ x \geq 0.4 \\ B, \text{if } \ y < -1 + 0.8x \\ B, \text{otherwise} \end{cases}$$

From this dataset we have picked the following points of interest (Figure 2) to demonstrate the disagreements between the XAI explainers, and the process of using the proposed geometric representations. These observations were selected as they are situated in areas where the model is giving non linear prediction boundaries.

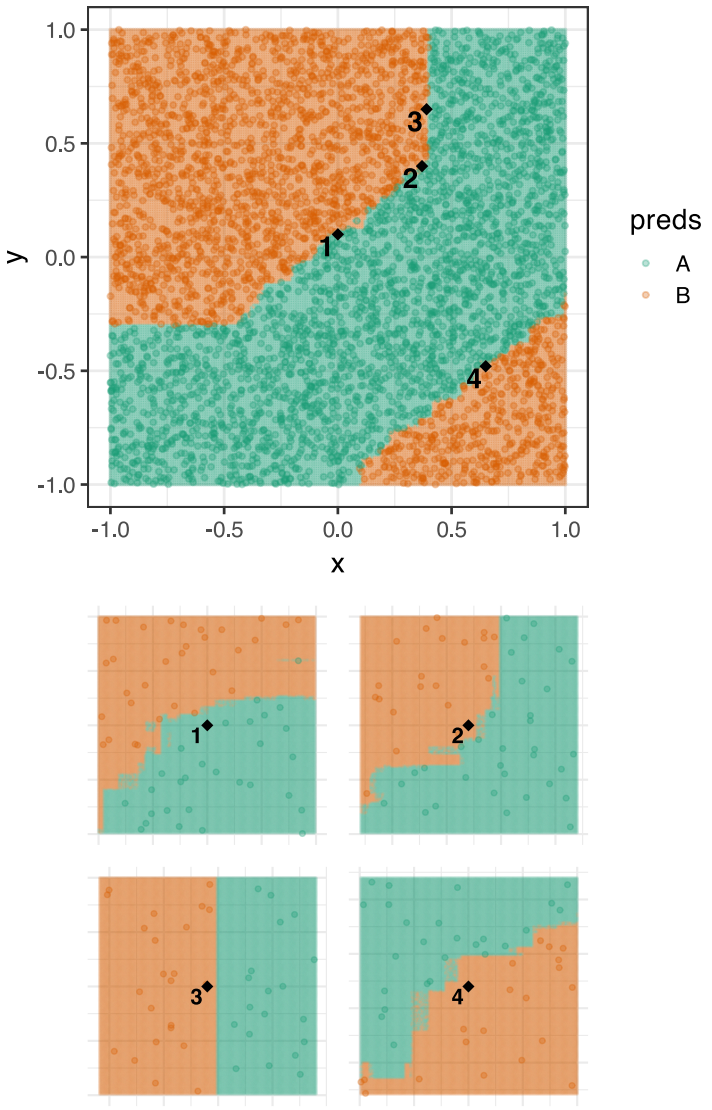

Figure 2: Four observations marked, for which we will use to show how the explainers explain the predictions made by the model.

2.2 The model

For the given binary classification task, we are going to fit a random forest model. A random forest is an ensemble machine learning model that combines the predictions of many decision trees to improve accuracy and robustness. It works by training each tree on a random subset of the training data, while also considering only a random subset of predictor variables at each split, which introduces diversity among the trees and reduces overfitting. For classification tasks, the final prediction is determined by majority voting across all trees, while for regression tasks it is typically the average of their predictions. By aggregating the outputs of multiple indepen-

dently constructed trees, random forests capture complex, non-linear relationships in data while maintaining strong predictive performance and resistance to noise and variability.

Random forest models were chosen for this task as they are less complicated while also illustrating the lack of interpretability in machine learning models, where the training algorithm is known yet the functional form of the model is not known.

The random forest we used in this case uses 10 trees and considers both variables at the same time.

2.3 Visualizing the prediction workflow

For the current problem, expanding upon the workflow in Figure 1, there are three primary components,along with one optional baseline,that should be visually inspected together. The optional baseline is the true decision boundary; while rarely available in practice, it can be included in simulated scenarios as a ground truth reference. The three core components consist of the training dataset, visualized to show the data distribution the model learned from; the testing dataset, which demonstrates how the model performs on held out data; and the learned decision surface. This surface represents the model's predictions generated over a dense, simulated grid encompassing all possible feature values, allowing us to visualize its behavior across the entire feature space.

An example is provided below to illustrate how these elements are visually integrated.

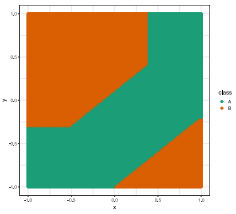

Figure 3: Actual decision boundary

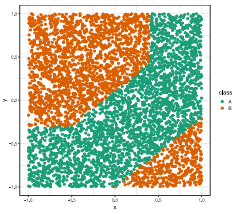

Figure 4: The training data

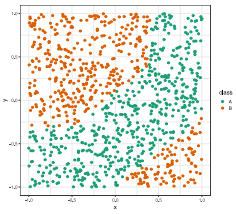

Figure 5: The testing data

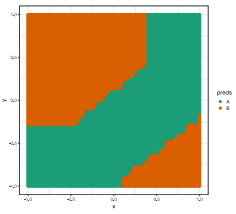

Figure 6: Model decision boundary

3 Generalizing the workflow for explanations

Previously, the workflow in Figure 1 focused on building a model for prediction. As the objective shifts toward explaining the inner mechanisms of that model, we must first define the purpose of an explanation. At its core, an explanation should answer a fundamental question: which features were most influential in driving the model's prediction for a specific observation?

In the modern era, where AI decisions directly impact individuals, understanding these local predictions is paramount. Furthermore, because model decision boundaries often become highly non-linear around difficult or edge-case observations [14], generating individual explanations to map this local behavior is essential. To ensure longevity amid the rapid development of new architectures, this paper focuses exclusively on local interpretability methods that are model-agnostic.

Local explanations are particularly valuable because AI systems increasingly support decisions that directly affect individual people, making observation-specific justification essential. Further-

more, restricting attention to model-agnostic methods ensures that the explanation framework remains applicable regardless of the underlying predictive model, avoiding dependence on architectures that may become obsolete as new machine learning methods emerge.

Compared to the purely predictive pipeline, the explanatory workflow illustrated in Figure 7 introduces several vital components. Beyond the trained model and the input data, the process now incorporates a specific target observation and a customized explanation generated for it. Although existing Explainable AI (XAI) algorithms differ considerably in their exact implementations, they can all be understood through a unified framework consisting of three fundamental stages: a perturbation mechanism, an optimization objective, and a final explanation output.

The process begins with a perturbation mechanism, which generates a simulated collection of synthetic observations in the immediate neighborhood of the target instance. This collection, defined as the local perturbation dataset, maps out the feature space around the specific observation being analyzed. The method of generating these neighbors varies across algorithms, ranging from random local sampling in LIME and feature masking in SHAP, to selective perturbation outside candidate rules in Anchors and optimization-based searches in Counterfactual explanations.

Next, the black-box model evaluates these simulated neighbors to produce local predictions. These outputs are fed into an optimization objective, which aligns the perturbed data and their predictions according to a method-specific criterion. In traditional statistics, parametric models inherently possess a functional form that explicitly outlines the contribution of each predictor. Machine learning models, however, are complex procedural algorithms lacking such transparency. To bridge this gap, the optimization stage frequently involves fitting a simple, interpretable surrogate model—referred to as the local parametric model. Whether it is a generalized linear model (GLM), a ridge regression, or a shallow decision tree, this surrogate mimics the black-box model's behavior within the local neighborhood. While not every XAI method requires an explicit surrogate model (SHAP computes additive attributions directly, Anchors optimize rules, and Counterfactuals search for alternatives), all rely on applying an optimization criterion to local predictions.

Ultimately, the optimized representation is returned as the final explanation. Depending on the underlying XAI method, this output takes an interpretable form tailored to the user's needs, such as feature importance coefficients, Shapley values, decision rules, or a modified counterfactual input. Together, these three stages provide a cohesive abstraction for seemingly disparate model-agnostic methods.

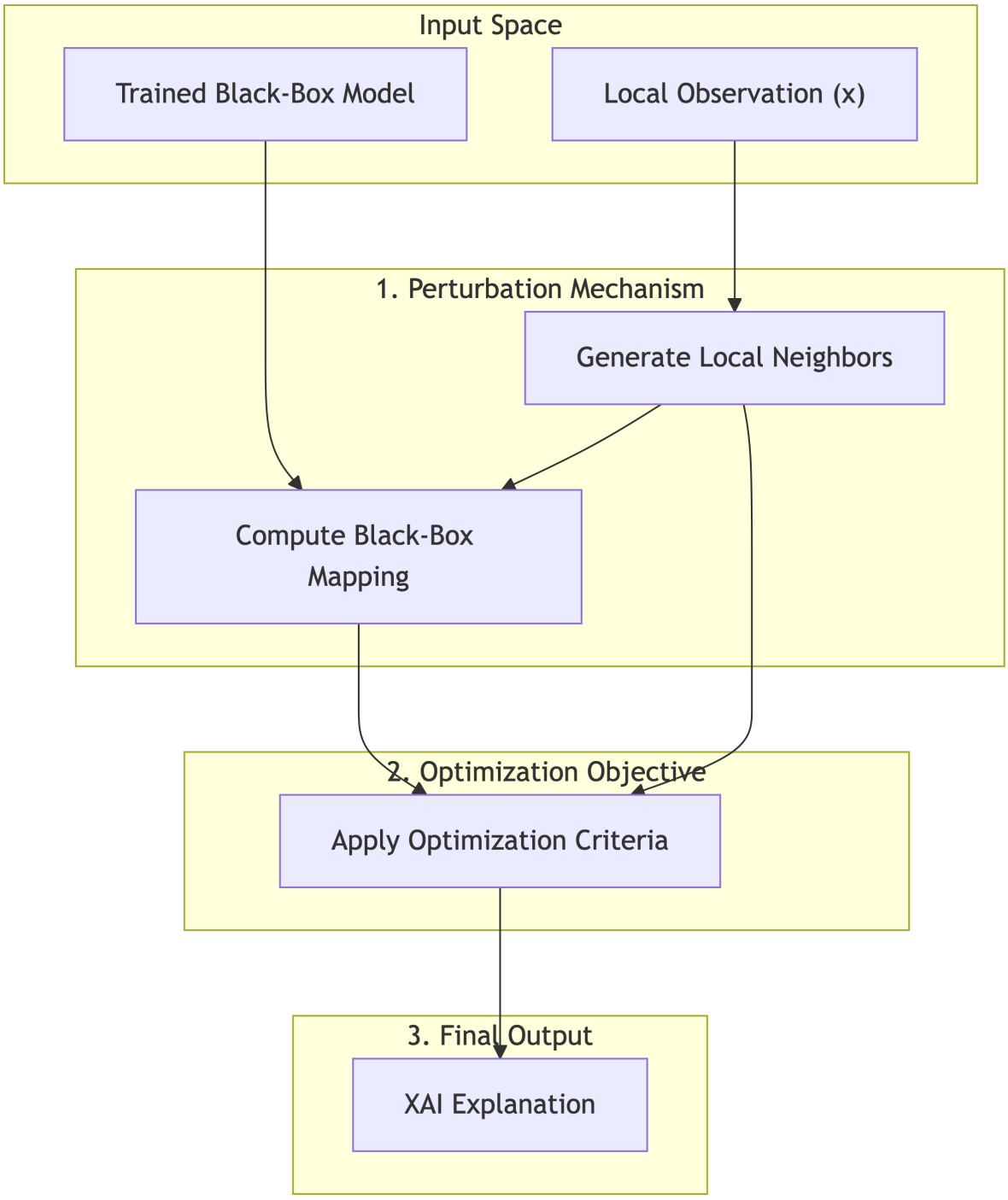

Figure 7: The generalized workflow when optimizing for local model explanation

4 XAI methods and their geometrical representation

The XAI methods discussed in this paper are built up of the components that can be generalized to contain, a perturbation mechanism, an optimization objective and a final output.

A perturbation mechanism in this context is a method that generates new instances around the local observation through perturbation. Afterwards using perturbed data and the original black box model, each XAI method will try to solve an optimization criteria that is related to the question that the XAI method is trying to answer. The returned output of each XAI method will differ, due to the nature of the question that each method is trying to answer.

From the plethora of available local interpretability methods, our focus will be on the methods of SHAP, Counterfactuals, LIME and Anchors as they are model agnostic and therefore can be applied to a wide variety of models [2]. These four methods have a similar structure that showcase different aspects of the model such as understanding the influence of features on a given prediction and the model boundary associated around the local neighborhood of the prediction.

4.1 SHAP

SHAP explanations are also model coefficients estimated to show the change in the expected prediction when the model is with and without the given feature. For a given observation, SHAP explanations can be represented as "forces" or unit vectors pulling the observations in the directions of each dimension. While these unit vectors aren't necessarily pushing the observation to another place, it is a depiction of the influence that each feature is having on the observation.

4.1.a Technical Definition

SHAP values are based on Shapley values defined for cooperative games in game theory. We first present the foundations of Shapley values and translate the components used in the derivation to create SHAP values. In a cooperative game, players are working together to maximize a reward and suppose we need to find out the how the reward should be distributed among each player. Shapley values are the unique solution to this problem that distribute the reward appropriately while satisfying a set of conditions [15].

For a coalition game of N players, let F be the superset of all possible combinations of players such that $F = \{\{1\}, ..., \{N\}, ..., \{1, 2\}, ..., \{N, N-1\}, ..., \{1, ..., N\}\}$. Let $v : 2^N \to \mathbb{R}$ be the value function that maps each subset of possible player combinations to a reward. Let A be a set containing N number of sets in it as $A = \{a_1, ..., a_n\}$ where each a_i is a set. Then $|A|$ is defined as the number of items in the set, $A \setminus \{i\}$ is defined the set A without the element i in any of the a_i subsets and $a_i \cup \{i\}$ is defined as the set a_i with i included as an item.

The payoff or Shapley value for each i^{th} player, ψ_i, is given by the following equation,

$$\psi_i = \frac{1}{N} \sum_{S \subseteq F \setminus \{i\}} \frac{v(S \cup \{i\}) - v(S)}{\binom{N-1}{|S|}}.$$

The calculated Shapley values are the only solutions with the following properties

1. Efficiency: The sum of Shapely values of all agents is equal to the total for the grand coalition (i.e. $\sum_{f \in F} \psi_i = v(F)$)
2. Symmetry: If i and j are equivalent then $\psi_i = \psi_j$.
3. Linearity: Linear combinations of value function should provide equivalent linear combinations (i.e. $\psi_i(v + w) = \psi_i(v) + \psi_i(w)$, $\psi_i(av) = a\psi(v)$)

In cases where explaining $\boldsymbol{x}_i$ is difficult, we resort to using an interpretable representation of the observation given by $\boldsymbol{x}'_i$. For example in SHAP $\boldsymbol{x}'_i$ is a binary vector representing whether or not we want to include that feature's contribution to the overall prediction. The interpretable representation $\boldsymbol{x}'_i$ is obtained through a mapping function $h_{\boldsymbol{x}_i}$ such that $h_{\boldsymbol{x}_i}(\boldsymbol{x}'_i) = \boldsymbol{x}_i$ and the mapping function is defined for each observation.

For SHAP we will be fitting a linear model around the given observation using the interpretable representation $\boldsymbol{x}'_i$. To observe the effect of each feature we would create binary vectors $\boldsymbol{z}'_i$ that are similar to $\boldsymbol{x}'_i$ (i.e. with certain features included and excluded). Using the generated $\boldsymbol{z}'_i$ s an interpretable model g of the following form will be built:

$$g(\boldsymbol{x}'_i) = \phi_0 + \sum_{j=1}^{p} \phi_i \boldsymbol{x}'_{i,j} \quad \text{where} \quad \boldsymbol{x}'_{i,j} \quad \text{is the} \quad j^{\text{th}} \quad \text{feature of} \quad \boldsymbol{x}'_i.$$

The built interpretable model g should satisfy the following properties to provide the coefficients of feature influence.

1. The model must be faithful to the original model (Local accuracy).

$$f(h_{\boldsymbol{x}_i}(\boldsymbol{x}'_i)) = g(\boldsymbol{x}'_i) = \phi_0 + \sum_{j=1}^{p} \phi_i \boldsymbol{x}'_{i,j}$$

2. The model gives zero as the feature importance value if the feature is missing from the interpretable representation i.e. $\boldsymbol{x}'_{i,j} = 0 \implies \phi = 0$.

3. For two blackbox models f_1 and f_2 where feature i has more influence on the observation in f_1 compared to f_2, we would need the interpretable model to be consistent and set $\phi_{i,1} \geq \phi_{i,2}$ where $\phi_{i,j}$ is the i^{th} coefficient from the j^{th} model ($j = 1, 2$).

If we consider the components of $\boldsymbol{x}'_i$ as players in a cooperative game and the value function as the model prediction when the features are present, then by using the formula for Shapley values, the coefficients of the interpretable model is given by,

$$\phi_j(f, \boldsymbol{x}_i) = \frac{1}{p} \sum_{\boldsymbol{z}'_i \subseteq \boldsymbol{x}'_i} \frac{f(\boldsymbol{z}'_i) - f(\boldsymbol{z}'_i \setminus j)}{\binom{p-1}{|\boldsymbol{z}'_i|}}$$

Evaluating the expectation for every subset of features directly is computationally expensive and therefore numerous model agnostic methods have been proposed to approximate the SHAP values by sampling from the power set of features[16] with model specific methods for tree based models being able to obtain exact SHAP values [17].

4.1.b Intuition

Intuitively, SHAP values measure the contribution of each predictor to the difference between the predicted value and the baseline or global expected value. The bigger the value, by magnitude indicates a stronger influence. Values can be positive and negative, which means that the predictors operate in opposite directions (like negative correlation).

4.1.c Geometric representation of SHAP

From an abstract visual standpoint we can draw a coefficient as a force acting upon the model's prediction itself. Therefore we propose the SHAP coefficients to be visualized as axis parallel arrows scaled by the magnitude of the coefficient. It should be noted that we assume that the data is normalized when visualizing.

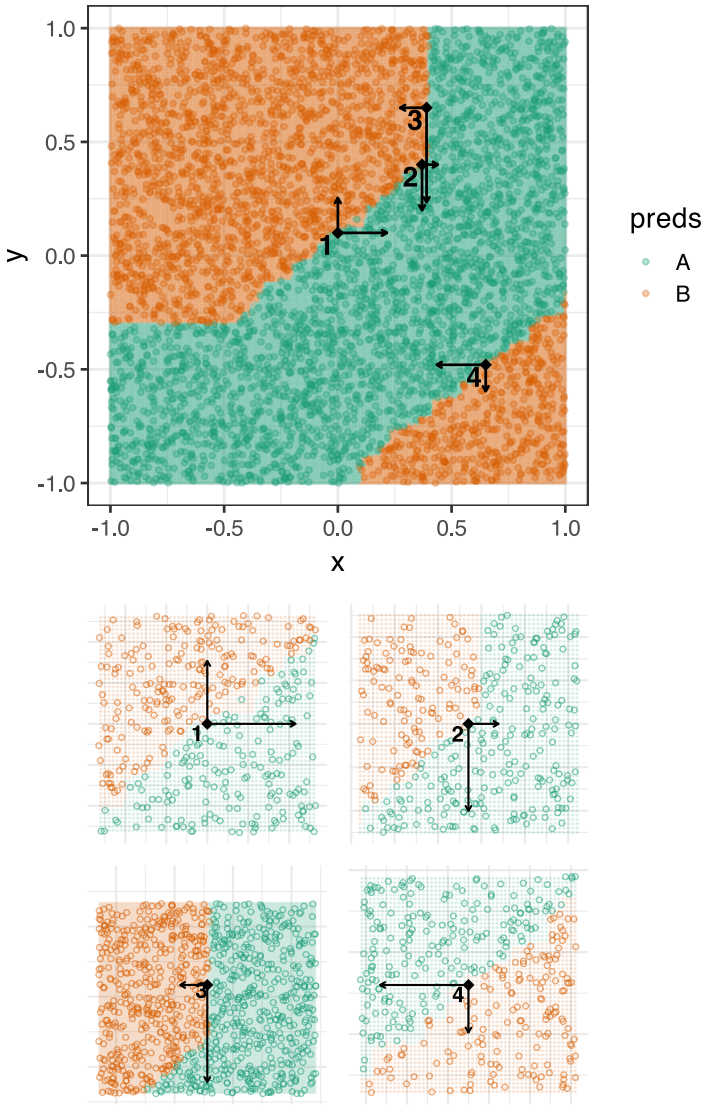

Figure 8: Geometric representation of SHAP in two dimensions

Table 1

4.2 LIME

4.2.a Technical Definition

Let us define $\mathcal{G}$ as the set of all possible models in the form of linear models or decision trees. The interpretable model used to derive the LIME explanations would then be $g \in \mathcal{G}$ defined in the domain of $\{0,1\}^d$. Even amongst interpretable models in $\mathcal{G}$, there would be varying degrees of complexity and we define $\Omega(g)$ as a measure of complexity of g which changes based on the model.

To develop a sense of the local model boundary we would need to sample data points around the given data instance $\boldsymbol{x}_i$. We define $l_{\boldsymbol{x}_i}(\boldsymbol{z})$ as the proximity measure between a generated instance $\boldsymbol{z}$ and $\boldsymbol{x}_i$.

The interpretable model that we define in the local neighborhood needs to have a model boundary as similar as possible to the black box model. Therefore, we define $\mathcal{L}(f, g, l_{\boldsymbol{x}_i})$ as a measure of how unfaithful (dissimilar in terms of a measure of accuracy) g is to the original model f in the local neighborhood defined by $l_{\boldsymbol{x}_i}$.

LIME works by trying to find the simplest model within the local neighborhood that is as similar as possible to the original black box model [18]. Therefore, the local explanation made by LIME is then given by,

$$\arg\min_{g \in \mathcal{G}} \mathcal{L}(f, g, l_{\boldsymbol{x}_i}) + \Omega(g).$$

In practical implementations, the surrogate model is typically a sparse linear model such as linear regression for regression tasks or logistic regression for classification, often regularized using LASSO to encourage sparsity and improve interpretability. Decision trees may also be used as local surrogates in some variants, but linear models remain the standard due to their simplicity and direct interpretation through feature coefficients.

Generalized Linear Models (GLMs) extend ordinary linear regression by allowing the response variable to follow distributions from the exponential family and by linking the expected value of the response to a linear predictor through a link function. A GLM consists of three components: a random component specifying the distribution of the response variable, a systematic component given by the linear predictor, and a link function connecting the two. Mathematically, the linear predictor is defined as:

$$\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$$

where η_i is the linear predictor for observation i, β_0 is the intercept and β_j are the regression coefficients. The expected response is then related to this predictor through the link function:

$$g(\mu_i) = \eta_i$$

where $\mu_i = E[Y_i]$ and $g(.)$ is the chosen link function. For binary classification, logistic regression is a common GLM where the logit link is used:

$$\log\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$$

4.2.b Intuition

Intuitively, LIME provides the coefficients of a linear model. Larger values in magnitude indicate the more influential predictors. Positive and negative values indicate that the predictors operating in opposite directions.

4.2.c Geometric representation of LIME

To address this issue, we have decided to incorporate to the existing visualisation of the machine learning model in data space, a secondary plot showing a zoomed in version of the local neighborhood and the model's faithfulness to the original model. The perturbed dataset is built on top of the predictions from the model on the perturbed data. Therefore showing the local model's faithfulness to the original model is performed by visualising the local model in its respective space.

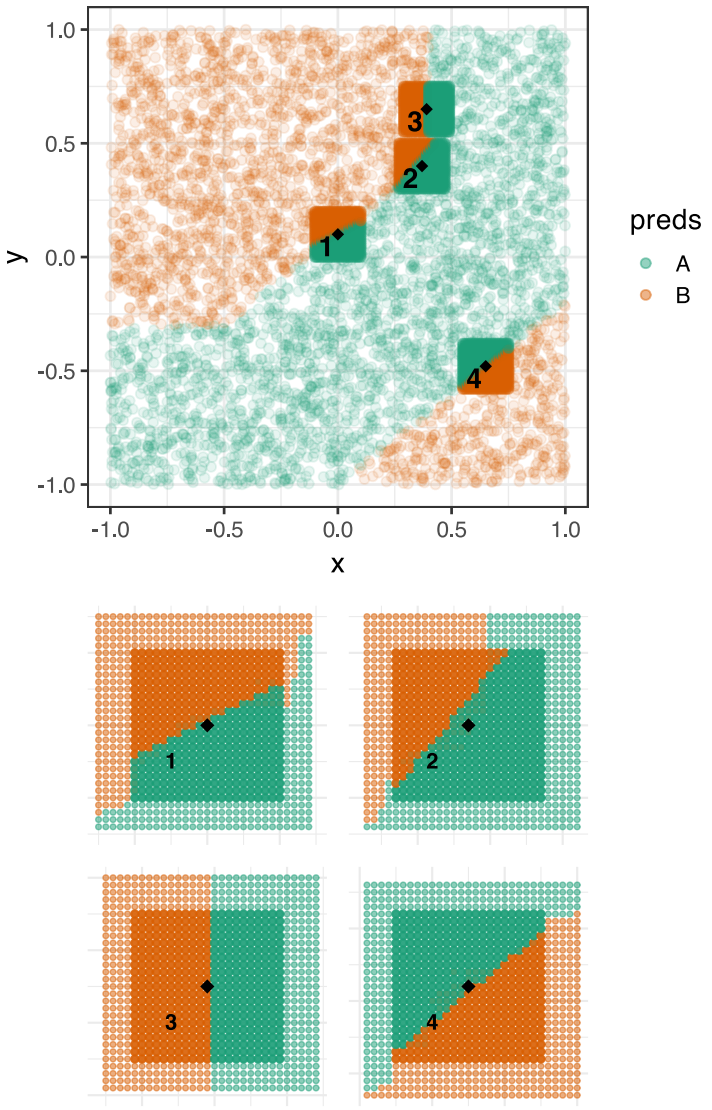

Figure 9: Geometric representation of LIME in two dimensions

Table 2: Table of LIME values

As showcased in Figure 7 our aim is to showcase the local perturbation dataset that we use in creating the local model within the plot. The lighter shade of colors are the predicted values from the model and then the deeper shade of colors are the predictions from the local model on the perturbation dataset.

To interpret the LIME results, the local model's boundary has to be considered as an indication of which features were most important for the local model. Intuitively, drawing a line from the point of interest in the direction orthogonal to the local model's decision boundary would give an indication of which features are more important.

4.3 Counterfactuals

4.3.a Technical Definition

A counterfactual explanation for a given observation is the closest unobserved yet plausible data point within the distribution of the dataset that has the desired model output. The main focus of counterfactuals is to show where the given observation needs to go to cross the model boundary with the least amount of changes to the features.

There are many methods of generating counterfactual explanations. In this paper, we will be utilizing Multi Objective Counterfactuals [19] as it has a more intuitive process compared to other approaches. A counterfactual explanation $\boldsymbol{x}_i^{(c)}$ for a given $\boldsymbol{x}_i$ and a desired outcome value $y_i^{(\text{exp})}$, is defined as an observation satisfying the following conditions:

1. $y_i^{(\text{exp})} \approx f(\boldsymbol{x}_i^{(c)})$.
2. $\boldsymbol{x}_i$ and $\boldsymbol{x}_i^{(c)}$ are close to each other in the data space.
3. $\boldsymbol{x}_i^{(c)}$ differs from $\boldsymbol{x}_i$ only in a few components.
4. $\boldsymbol{x}_i^{(c)}$ is a plausible data point according to the distribution of each dimension.

We will define the above conditions as objective functions. Let $o_1(\hat{y}, y_i^{(\text{exp})})$ be a function that quantifies the distance between $\hat{y}$ and $y_i^{(\text{exp})}$. Let $o_2(\boldsymbol{x}_i^{(c)}, \boldsymbol{x}_i)$ define the distance between the given observation and the counterfactual explanation that is going to be generated. Let $o_3(\boldsymbol{x}_i^{(c)}, \boldsymbol{x}_i)$ be the number of changed features between the given observation and the counterfactual explanation using the L_0 norm. Let $o_4(\boldsymbol{x}_i^{(c)}, k)$ be the distance between the counterfactual explanation and the k nearest observed points from the dataset $\mathcal{D}$. Each of these objective functions correspond to the four conditions that a counterfactual should satisfy.

Finding $\boldsymbol{x}_i^{(c)}$ can then be translated to a minimization task as,

$$\arg\min_{\boldsymbol{x}_i^{(c)}} \left[\, o_1(f(\boldsymbol{x}_i^{(c)}), y_i^{(c)}), o_2(\boldsymbol{x}_i^{(c)}, \boldsymbol{x}_i), o_3(\boldsymbol{x}_i^{(c)}, \boldsymbol{x}_i), o_4(\boldsymbol{x}_i^{(c)}, \boldsymbol{x}_i) \right] .$$

4.3.b Intuition

The values returned for a counterfactual reflect the distance to get from the point of interest to its counterfactual, along each predictor. A smaller value would indicate more influence, because changing the value on that predictor by a small amount could lead to a change of predicted class. Differing signs are meaningful, in the same was as the other explainers, that the predictors operate in opposite directions.

4.3.c Geometric representation of Counterfactuals

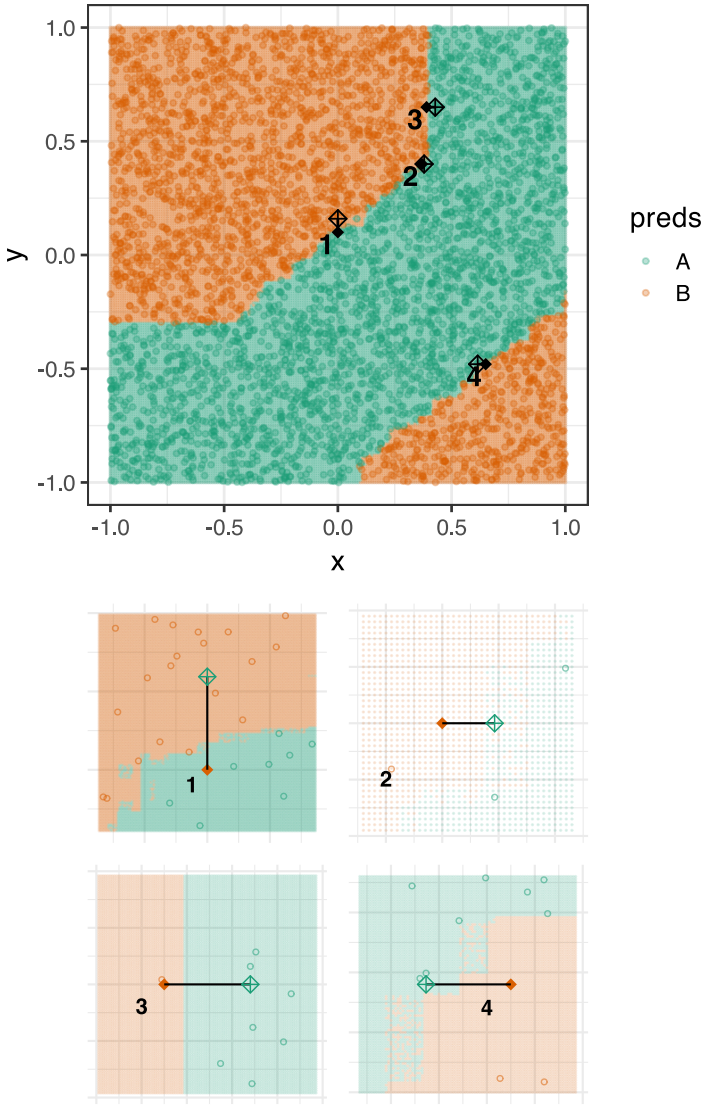

Figure 10: Geometric representation of Counterfactuals in two dimensions

Table 3

A Counterfactual explanation is another data point itself. However the issue is that the generated data point is unobserved yet plausible. While it is possible to simply plot the point that is given by the counterfactual as a new point it should be highlighted that this point does not exist in the dataset $\mathcal{D}$ and instead is a plausible point. Also it may help to show that the model boundary is somewhere between the line connecting the observation and the counterfactual. We might not know the exact location of the model boundary and finding the exact location would not be trivial. But counterfactuals provides a reasonable guesstimate of where the model boundary would lie.

4.4 Anchors

4.4.a Technical Definition

Anchors are defined as a rule or a set of predicates that satisfy the given instance and is a sufficient condition for $f(x_i)$ with high probability. A predicate is a logical condition that an observation may or may not satisfy. For example a predicate $a = \{x_{i,1} > 2\}$ would be true if the first component of $\boldsymbol{x}_i$ is greater than 2.

For a list of predicates to be an anchor it should satisfy the given instance (i.e. be true for the given instance) while also maximizing two criteria, precision and coverage.

The precision of an anchor $\mathcal{A}$ is defined formally as,

$$\text{Prec}(\mathcal{A}) = \mathbb{E}_{z|\mathcal{A}}[\mathbb{1}_{f(x)=f(z)}]$$

where z is a generated instance in the local neighborhood. Intuitively, the precision of an anchor translates to the expected proportion of observations in the local neighborhood of $\boldsymbol{x}_i$ with the same class as the one of $\boldsymbol{x}_i$ given that the generated observations satisfy the anchor $\mathcal{A}$

For a given tolerance level τ and error rate δ, a list of predicates $\mathcal{A}$ is considered an anchor if

$$\Pr(\text{Prec}(\mathcal{A}) \geq \tau) \geq 1 - \delta.$$

The coverage of an anchor A is defined as the expected proportion of the generated local neighborhood is within the defined anchor $\mathcal{A}$,

$$\text{Coverage}(\mathcal{A}) = \mathbb{E}_z[\mathcal{A}(z)].$$

Since a list of predicates is considered to be an anchor if it maximizes the coverage and precision, finding an anchor for a given instance can be defined as the solution to the following optimization problem,

$$\max_{\mathcal{A} \text{ s.t. } \Pr(\text{Prec}(\mathcal{A}) \geq \tau) \geq 1-\delta} \text{Coverage}(\mathcal{A})$$

The target would then be to maximize the coverage while ensuring that the precision is above a tolerance level.

4.4.b Intuition

Intuitively, the anchors are the relative fraction of the lengths of the sides of the filled box. A smaller value indicates a more influential predictor, because we can only extend the box a small amount in that direction, to get a more pure box.

4.4.c Geometric representation of Anchors

Anchors are defined as a list of boundary conditions. These boundary conditions should be represented as a p-dimensional hybercube in the data space. For two dimensions the boundary would be a square while for high dimensions the generated boundary would be a p-dimensional hypercube. However, simply drawing a cube would not suffice to communicate a key information that Anchors provides: a majority of the area within the cube would contain points that have a

similar outcome to the given observation. Therefore it is necessary to highlight the contents of the box compared to the points outside the box.

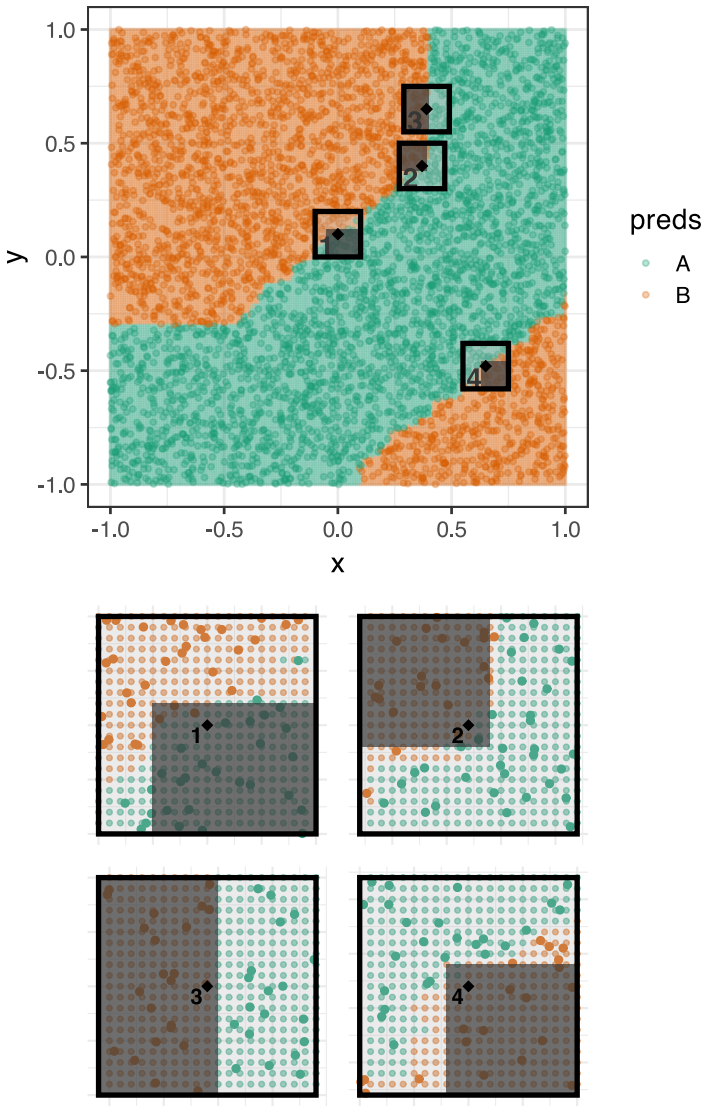

Figure 11: Geometric representation of Anchors in two dimensions

Table 4

When trying to visualize anchors we tried to incoporate the parts that we showcased in the previous figure about the local dataset around the observation of interest. So in this plot we have the big black box that shows the area where the perturbation distribution was created. That is the area of interest which we used to focus our model on. And then we broke this box into smaller sub boxes and we tried to see which sub box contains the most of the points that have the same predicted point for the observation of interest. But how do we interpret this? The main thing to notice is the direction and that is what we need to focus on When interpreting anchors we try to see the direction that the anchor has

gone to in relation to the perturbation distributions area. Aside from these examples there are obvious quirks, for example in point 2, we have that the anchor has gone in the same direction indicating that the variables have similar influence to the point 6, which is also the case in point 4. Then we have point 5 and point 1 which show that the model uses x and y as more important variables to the predicition for those specific instances.

As a summary, each XAI method obtains explanation in three different formats: SHAP provide coefficients of a linear additive model, Counterfactuals provide a new data point (or alternatively the changes made to the features to cross the model boundary) and finally Anchors provides a set of boundaries containing points with similar outcomes as the given observation.

5 Disagreements

5.1 Extracting and standardizing information from explanations

Explanations themselves are not inherently comparable across methods, as they measure fundamentally different facets of the underlying model's behavior. For instance, SHAP and LIME output feature attribution scores in the form of linear coefficients. In contrast, geometric and rule-based methods rely on spatial metrics: counterfactuals are defined by the distance between the selected instance and the identified counterfactual, while anchors rely on the distances from the instance to the edges of a decision bounding box.

For counterfactuals, to extract the importances we compute the absolute difference between the counterfactual point coordinates (x_c, y_c) and the original point coordinates (x, y):

$$d_x = \mid x_c - x \mid$$

$$d_y = \mid y_c - y \mid$$

For anchor bounding boxes, the importance is obtained from the width of this bounded region, calculated as the absolute difference between the upper and lower bounds for the respective feature. In this example we would be having two features.

$$\Delta x_{anchor} = \mid x^{(1)} - x^{(2)} \mid$$

Before scaling, we project the values based on their explainability method m. Local model based methods (LIME, SHAP) take the absolute value, while distance based methods (Counterfactuals, Anchors) take the inverse of the value to align their directional meaning (larger value = more important / tighter bound). It also includes a safeguard against division by zero.

For a given feature value x evaluated by method m:

$$x_{proj}^{(m)} = \begin{cases} \mid x \mid & \text{if } \; m \in \{\text{LIME}, \text{SHAP}\} \\ 0 & \text{if } \; m \in \{\text{Counterfactuals}, \text{Anchors}\} \;\; \text{and} \;\; x = 0 \\ \frac{1}{x} & \text{if } \; m \in \{\text{Counterfactuals}, \text{Anchors}\} \;\; \text{and} \;\; x \neq 0 \end{cases}$$

(The exact same piecewise function applies to $y_{proj}^{(m)}$)

The primary challenge lies in standardizing these disparate outputs, bridging the gap between linear coefficients and spatial distances, so they can be compared across methods.

Finally, the projected values are standardized using min-max scaling, calculated **independently within each method group** m to ensure values are comparable across different explainability techniques on a $[0, 1]$ scale.

$$x_{minmax}^{(m)} = \frac{x_{proj}^{(m)} - \min_i(x_{proj,i}^{(m)})}{\max_i(x_{proj,i}^{(m)}) - \min_i(x_{proj,i}^{(m)})}$$

$$y_{minmax}^{(m)} = \frac{y_{proj}^{(m)} - \min_i(y_{proj,i}^{(m)})}{\max_i(y_{proj,i}^{(m)}) - \min_i(y_{proj,i}^{(m)})}$$

Because the outputs of these XAI methods are derived through fundamentally different mathematical processes, their raw values cannot be directly compared. To enable a relative inspection of feature significance, the importances from the distance based methods are transformed by inversing the importance as the lower values in Counterfactuals and Anchors indicate that the feature is more important.

Afterwards, we normalized these outputs by using a min-max normalization step to ensure that for each XAI method the resulting importances across observations are scaled between 0 and 1, enabling the comparison between methods across observation.

The normalized values are shown in Table 5 which contains the original importance given by each XAI method along with the final importance after normalization.

Table 6 summarizes the primary features identified by each respective technique, highlighting the varied interpretations of the model's behavior. The results reveal a significant divergence among the explanations;

This systemic variance underscores a critical insight: there is no universally "correct" explanation. Instead, determining the appropriate XAI method is highly context-dependent, ultimately governed by the specific problem formulation and the distinct goals of the end-user.

Table 5: The importance of the two features towards the prediction along with the normalized importance.

Point	method	Original x	Original y	x Importance	y Importance
1	lime	–43.47	88.87	0.00	0.95
	SHAP	0.22	0.15	1.00	0.12
	Counterfactuals	0.00	0.06	0.00	1.00
	Anchors	0.15	0.12	0.00	1.00
2	lime	–66.44	54.72	0.02	0.58
	SHAP	0.07	–0.20	0.00	0.28
	Counterfactuals	0.01	0.00	1.00	0.00
	Anchors	0.12	0.12	0.69	1.00
3	lime	–1001.34	0.00	1.00	0.00
	SHAP	–0.12	–0.42	0.32	1.00
	Counterfactuals	0.04	0.00	0.24	0.00
	Anchors	0.11	0.20	1.00	0.00
4	lime	76.34	–93.65	0.03	1.00
	SHAP	–0.22	–0.12	1.00	0.00
	Counterfactuals	0.04	0.00	0.26	0.00
	Anchors	0.12	0.12	0.69	1.00

Table 6: A summary of the most important features for each point

Point	LIME	SHAP	Counterfactuals	Anchors
1	y	x	y	y
2	y	y	x	y
3	x	y	x	x
4	y	x	x	y

6 Discussion

visualising the explanations of XAI methods as geometric objects within the data space itself provides a more intuitive and comprehensive understanding of each method. This approach allows users to see how different methods interpret the same data points and their relationships, making the abstract concepts more tangible. By representing high-dimensional data and XAI

explanations in a visual format, users can better grasp the underlying patterns and influences driving model decisions.

Disagreements between XAI methods can often be clarified by examining their geometric representations. Different XAI methods may offer varying explanations due to their distinct approaches to estimation and approximation. By visualising these methods geometrically, it becomes easier to identify the root causes of their differences and understand how each method interprets the model's behavior. This visual approach can reveal the underlying reasons for discrepancies, helping users to reconcile different perspectives and select the most appropriate method for their specific needs.

Bringing the human perspective into XAI is essential for a thorough understanding of both the data and the model. Human intuition and domain knowledge play a crucial role in interpreting complex data and model outputs. By incorporating interactive visualisation tools, users can explore data and model explanations more effectively, leveraging their expertise to uncover insights that automated methods might miss. This human-centric approach ensures that the interpretations are not only technically sound but also practically relevant.

7 Supplementary material

The case for interactivity, So far we have discussed the importance of establishing a set of geometric representations to better visualise XAI expalanations. However, the last layer of the stack that is ommited from Figure 7 is the human (be it the model developer, a stakeholder or a decision maker). Incorporating the user's feedback and ideas in to the

Since we are looking at each individual observation separately understanding the model boundary requires us to generate multiple visualisations and compare between them. We can instead use interactive techniques to bring the human perspective present the different aspects of the model to the end user directly. The end user can actively participate in the selection of observations which creates an internal feedback loop to the end user on the choice of XAI method and local observation.

Human interactivity is crucial in enabling the understanding of XAI methods. Since we are looking at each individual observation separately, understanding the model boundary requires us to generate multiple visualisations and compare between them. We can instead use interactive techniques to bring the human perspective and present the different aspects of the model to the end user directly. Interactive tools allow users to explore and manipulate the visual representations of XAI methods, facilitating a deeper and more intuitive understanding of model explanations. Interactivity can include features such as clicking, zooming and shared state management enabling users to investigate specific aspects of the data and model behavior. This hands-on approach helps bridge the gap between complex algorithmic outputs and human comprehension. The end user can actively participate in the selection of observations which creates feedback on the choice of XAI method and local observation.

Combining visualisation and interactivity methods, we suggest a novel approach where each geometric representation is visualised in an interactive environment. By allowing the user to

explore, compare and contrast observations and explainers users can understand the different facets of the black box model that each XAI method reveals. Our proposed solution integrates advanced visualisation techniques with interactive interfaces to create a dynamic, user-friendly platform for exploring XAI methods. This approach allows users to switch between different XAI methods, observe how explanations change with different data points, and gain insights into the model's decision process. By providing an interactive environment, we aim to make the understanding of complex AI models more accessible and actionable for developers and end users alike.

Bibliography

[1] S. Ali *et al.*, "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," *Information Fusion*, vol. 99, p. 101805, 2023, doi: https://doi.org/10.1016/j.inffus.2023.101805.

[2] C. Molnar, *Interpretable machine learning: a guide for making black box models explainable*, Second edition. Munich, Germany: Christoph Molnar, 2022.

[3] C. Molnar, G. Casalicchio, and B. Bischl, "Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges," vol. 1323, 2020, pp. 417–431. Accessed: Oct. 16, 2023. [Online]. Available: http://arxiv.org/abs/2010.09337

[4] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[5] B. M. Greenwell and others, "pdp: An R package for constructing partial dependence plots.," *R J.*, vol. 9, no. 1, p. 421, 2017.

[6] D. W. Apley and J. Zhu, "Visualizing the effects of predictor variables in black box supervised learning models," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 82, no. 4, pp. 1059–1086, 2020.

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-Precision Model-Agnostic Explanations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.11491.

[9] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR." Accessed: Jun. 21, 2024. [Online]. Available: http://arxiv.org/abs/1711.00399

[10] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[11] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[12] H. Baniecki and P. Biecek, "modelStudio: Interactive Studio with Explanations for ML Predictive Models," *Journal of Open Source Software*, vol. 4, no. 43, p. 1798, 2019, [Online]. Available: https://doi.org/10.21105/joss.01798

[13] H. Baniecki, D. Parzych, and P. Biecek, "The grammar of interactive explanatory model analysis," *Data Mining and Knowledge Discovery*, pp. 1–37, 2023, [Online]. Available: https://doi.org/10.1007/s10618-023-00924-w

[14] G. James, D. Witten, T. Hastie, R. Tibshirani, and others, *An introduction to statistical learning*, vol. 112. Springer, 2013.

[15] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied stochastic models in business and industry*, vol. 17, no. 4, pp. 319–330, 2001.

[16] M. Mayer and D. Watson, "kernelshap: Kernel SHAP." 2024. [Online]. Available: https://github.com/ModelOriented/kernelshap

[17] J. Yang, "Fast treeshap: Accelerating shap value computation for trees," *arXiv preprint arXiv:2109.09847*, 2021.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," 2016, doi: 10.48550/ARXIV.1602.04938.

[19] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-Objective Counterfactual Explanations," in *Parallel Problem Solving from Nature – PPSN XVI*, vol. 12269, T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann, Eds., Cham: Springer International Publishing, 2020, pp. 448–469. doi: 10.1007/978-3-030-58112-1_31.