Informative Machine Learning Model Explanation Techniques

Ningsheng Zhao

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

January 2025

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Ningsheng Zhao**

Entitled: **Informative Machine Learning Model Explanation Techniques**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Lan Lin

_____ External Examiner
Dr. Kim Khoa Nguyen

_____ Examiner
Dr. Mazdak Nik-Bakht

_____ Examiner
Dr. Nizar Bouguila

_____ Examiner
Dr. Chun Wang

_____ Supervisor
Dr. Jia Yuan Yu

_____ Co-supervisor
Dr. Yong Zeng

Approved by _____
Dr. Farnoosh Naderkhani, Graduate Program Director

2025.01.14

_____
Dr. Mourad Debbabi, Dean of Faculty

Abstract

Informative Machine Learning Model Explanation Techniques

Ningsheng Zhao, Ph.D.
Concordia University, 2025

Explainable AI (XAI) is an emerging field focused on providing human-interpretable insights into complex and often black-box machine learning (ML) models. Shapley value attribution (SVA) is an increasingly popular XAI method that quantifies the contribution of each feature to a model's behavior, which can be either an individual prediction (local SVAs) or a performance metric (global SVAs). However, recent research has highlighted several limitations in existing SVA methods, leading to biased or incorrect explanations that fail to capture the true relationships between features and model behaviors. What's worse, these explanations are vulnerable to adversarial manipulation.

Additionally, global SVAs, while widely used in applied studies to gain insights into underlying information systems, face challenges when applied to ML models trained on imbalanced datasets, such as those used in fraud detection or disease prediction. In these scenarios, global SVAs can yield misleading or unstable explanations.

This thesis aims to address these challenges and improve the reliability and informativeness of SVA explanations. It makes three key contributions: 1) Proposing a novel error analysis framework that comprehensively examines the underlying sources of bias in existing SVA methods; 2) Introducing a series of refinement methods that significantly enhance the informativeness of SVA explanations, as well as their robustness against adversarial attacks; 3) Developing a standardization method for evaluating global model behaviors on imbalanced datasets, advancing the development of an explainable model monitoring system. Our experiments demonstrate that these methods substantially improve the ability of SVAs to uncover informative patterns in model behaviors, making them valuable tools for knowledge discovery, model debugging, and performance monitoring.

Acknowledgments

This journey has been long and challenging, and I have many people to thank for their support. First and foremost, I want to thank my supervisor, Dr. Jia Yuan Yu. His invaluable guidance, support, and advice have been with me every step of the way, and I wouldn't have reached this point without his encouragement and mentorship. His thoughtful feedback not only improved this thesis but also helped me grow as a researcher, and his continued belief in my work gave me the confidence to keep pushing forward. I feel truly fortunate to have had him as my mentor and hope to learn more from him in the years to come.

I also want to express my sincere gratitude to my co-supervisor, Dr. Yong Zeng, for always being there with support and patience. He made sure I stayed on track administratively and offered me practical advice when I needed it most. His understanding has been a steady source of reassurance throughout the second half of my PhD journey. I really appreciate everything he did to help me stay in such a good position to complete my thesis.

To all the members of my thesis committee—Dr. Kim Khoa Nguyen, Dr. Lan Lin, Dr. Chun Wang, Dr. Nizar Bouguila, and Dr. Mazdak Nik-Bakht—thank you for your invaluable time, insights, and helpful feedback. Your expertise has greatly improved the quality of my research.

I'm also deeply thankful to all the professors and lecturers who provided me with excellent courses, equipping me with the knowledge needed to undertake this research. In particular, I would like to extend my thanks to Dr. Luc Devroye, Dr. Frédéric Godin, Dr. Siamak Ravanbakhsh, Dr. Shai Ben-David, Dr. Changbao Wu, Dr. Ali Ghodsi, Dr.

Kun Liang, Dr. Xiaowen Zhou, and Dr. Wei Sun for their contributions to my academic development.

A special thanks to Mitacs and Daesys Inc. for their financial support, especially Dr. Krzysztof Dzieciolowski. Working with him has been an incredible learning experience. He shared so many practical insights from the business world and always provided helpful feedback. His advice challenged me to think in new ways and see beyond the academic scope of my research.

I'm also grateful to Fin-ML and BNPP Bank for their support. Fin-ML provided me with opportunities to present my work, as well as invaluable workshops and professional training that enriched my experience. BNPP Bank gave me hands-on experience through my internship, which was an unforgettable part of this journey.

Of course, I can't forget my girlfriend, Dr. Trang Bui. I cannot thank her enough for all that she's done. Her love, encouragement, and constant support have meant everything to me. She has always believed in me and offered thoughtful discussions—whether about research or life—that helped me through the toughest challenges. I am so lucky to have her company through all the ups and downs of this journey.

Lastly, to my parents, Xianguo Zhao and Huaifeng Zhang—thank you for always believing in me and staying strong and healthy throughout this journey. Your unwavering support gave me the courage to pursue my dreams, and I could not have done this without your understanding and sacrifice. I am forever grateful for your upbringing and unconditional love, and this thesis is dedicated to you. I also want to express my heartfelt thanks to my relatives for taking care of my parents in my absence. Your kindness and help brought me peace of mind. Thank you all, from the bottom of my heart.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Machine learning (ML) and artificial intelligence (AI) are increasingly integrated into our daily lives thanks to their powerful predictive performance. Because of their state-of-the-art accuracy, both scientific and industrial researchers and practitioners increasingly rely on them to handle intricate tasks. However, most ML models are black-box with complex structures and numerous parameters, making it difficult for humans to understand why certain predictions are made. For example, it can be difficult to understand the prediction generated for a single patient's diagnosis. While we usually prioritize how accurate these models are, there are certain applications where it is crucial to know how they actually work. Industries such as finance, healthcare, and criminal justice place a premium on model transparency due to its potential to identify undesired dependencies, build trust among users or organizations, and assess whether models operate like expert decision-makers.

To increase the transparency and trustworthiness of ML and AI, explainable AI (XAI) arises as an emerging field that aims to explain ML models. An increasingly popular XAI method is Shapley value attribution (SVA), which assigns importance scores to features regarding model behaviors (I. Covert, Lundberg, & Lee, 2020; Lundberg & Lee, 2017).

The literature suggests that SVA methods can be *true to the model* and/or *true to the data* (Chen, Covert, Lundberg, & Lee, 2023; Chen, Janizek, Lundberg, & Lee, 2020). SVA methods that are true to the model aim to understand the model's functional or algebraic dependencies on features. However, standard supervised ML learning models typically do not explicitly model dependencies between features (Janzing, Minorics, & Blöbaum, 2020; Watson, 2022). Moreover, in the presence of feature interdependence, a model can often be written in different algebraic forms that perform identically (Frye et al., 2020). Hence, even if an attribution is exactly true to the model, it still might not correctly represent the intrinsic relationships between features and the model's output. If knowledge discovery is our objective, we want SVAs to be *true to the data*, representing the model's informational dependencies on features. SVA methods that are true to the data put less emphasis on the particular model but more on the true underlying data-generating process (Chen et al., 2020).

In this work, we focus on the study of SVAs that are true to the data. Since they can explain ML models more informatively, we call them *informative SVAs*. In practice, SVAs have been widely used to assist decision explaining and model debugging. Moreover, researchers have recently begun applying SVAs to scientific discoveries. For example, SVA techniques have been used to identify risk factors for diseases and mortality (Alatrany, Khan, Hussain, Kolivand, & Al-Jumeily, 2024; Kırboğa & Kucuksille, 2023; Qiu et al., 2022; Snider, Patel, & McBean, 2021); gain valuable new insights into genetic or molecular processes (Janizek et al., 2021; Novakovsky, Dexter, Libbrecht, Wasserman, & Mostafavi, 2023; Yagin et al., 2023); and capture informative patterns for fraud detection (Psychoula et al., 2021), etc.

1.2 Problem Statement

Error-prone challenge While SVAs provide promising directions to improve the understanding of underlying information systems, concerns remain about their accuracy. Specifically, informative SVAs that are true to the data must be computed based on the true underlying distributions of the data, which are typically unknown in practice. Thus, we can only estimate these distributions using an observed dataset. However, the given dataset is usually too sparse to capture the complex distributions of high-dimensional or many-valued features, leading to significant estimation errors (Sundararajan & Najmi, 2020). To address data sparsity, a number of approaches have been proposed (Aas, Jullum, & Løland, 2021; Frye et al., 2020; Lundberg, Erion, & Lee, 2018; Mase, Owen, & Seiler, 2019). Nevertheless, (Chen et al., 2023) and (Yeh, Lee, Liu, & Ravikumar, 2022) demonstrate that all of these approaches suffer from some drawbacks that lead to undesirable errors. Hence, in practice, instead of estimating the true distribution, most built-in SVA tools are designed based on some distributional assumptions, such as feature independence assumption. However, untenable assumptions may also result in incorrect attributions (Frye et al., 2020), making SVAs vulnerable to model perturbation or adversarial attacks (Lin, Covert, & Lee, 2024; Slack, Hilgard, Jia, Singh, & Lakkaraju, 2020). In this sense, most of the existing SVA methods are unreliable and error-prone.

Data-imbalance challenge The informative SVA can be further categorized into *local SVA* and *global SVA*. Local SVAs focus on interpreting individual predictions, while global SVAs provide insights into the informative patterns across the entire dataset. Global SVAs are promisingly powerful tools for knowledge discovery and model monitoring. When conducting global SVAs, it is essential to choose a targeted performance metric, analyzing how each feature affects it (I. C. Covert, Lundberg, & Lee, 2021). For example, I. Covert et al. (2020) proposes a cross-entropy-based global SVA method to identify informative

factors, and detect feature corruptions. However, many machine learning models today are deployed on imbalanced classification datasets, such as those used for fraud detection, disease diagnosis, and risk identification. In such datasets, the distribution of classes is often skewed, ranging from slight biases to severe imbalances where the minority class may represent only a fraction of the total examples. This class imbalance presents a significant challenge for both model performance evaluation and explanation. On the one hand, traditional metrics like classification accuracy and cross-entropy tend to be less informative or uninformative about the minority class, potentially leading to misleading SVA explanations. On the other hand, confusion matrix-based metrics, such as f1_score, Matthews Correlation Coefficient (MCC), lift, and Precision-Recall curves (PRC), are highly sensitive to class imbalance. This sensitivity undermines their reliability as universal measures for model evaluation and monitoring, particularly when classifying highly imbalanced data. As a result, global SVAs based on these metrics may be unstable or unreliable explanations, especially when diagnosing and explaining model performance drifts. Hence, so far in the literature, few SVA methods have been proposed to understand features' impacts on those commonly used performance metrics.

1.3 Contributions and Thesis Outline

This thesis aims to address the aforementioned challenges in applying SVA methods for generating informative explanations in machine learning models. We will begin by establishing the necessary notations, concepts, and preliminaries in Chapter 2. In Chapter 3 and 4, we will focus on analyzing the sources of errors in SVA explanations and proposing solutions to refine these explanations. In Chapter 5, we will tackle the challenge of data imbalance in global SVA explanations, presenting some novel approaches to make SVAs more robust in such scenarios. Finally, the thesis will conclude with a summary of our contributions and potential avenues for future research. The core contributions of this work

are presented in Chapter 3, 4, and 5.

In Chapter 3, we propose a novel error theoretical analysis framework, in which the explanation errors of SVAs are decomposed into two components: observation bias and structural bias. We further clarify the underlying causes of these two biases and demonstrate that there is a trade-off between them. Based on this error analysis framework, we develop two novel concepts: over-informative and under-informative explanations. We demonstrate how these concepts can be effectively used to understand potential errors of existing SVA methods. In particular, we find that the widely deployed assumption-based SVAs can easily be under-informative due to the distribution drift caused by distributional assumptions. We propose a measurement tool to quantify such a distribution drift. Finally, our experiments illustrate how different existing SVA methods can be over- or under-informative.

In Chapter 4, we propose a series of refinement methods that combine out-of-distribution (OOD) detection and importance sampling techniques to alleviate the SVA biases discussed in Chapter 3. In essence, these methods aim to correct the distribution drift caused by distributional assumptions that are made to reduce data complexity. We apply our refinement methods to two widely-used SVAs: the marginal SVA and the surrogate model-based SVA. Our extensive experiments show that the proposed methods can not only achieve a significantly better balance between observational and structural biases but also protect the SVA explanations from adversarial attacks, thereby greatly enhancing the informativeness and reliability of both local and global SVA explanations.

In Chapter 5, we propose a standardization method for confusion matrix-based performance metrics called the *outperformance score (OPS)* function. Based on this function, we further propose a standardized global SVA method, *OPS-SAGE*. The OPS function is universal in the sense that it standardizes any given performance metric to a consistent range of $[0, 1]$ and provides a uniform interpretation. Essentially, the OPS function calculates the probability that the observed classification performance outperforms a random

performance given the class imbalance rate, while the OPS-SAGE attributes this probability to individual features. Both the OPS and OPS-SAGE are comparable across various performance metrics and test sets with differing imbalance rates. Our experiments on real datasets demonstrate the utility of our proposed methods, showing that the resulting model performance and feature importance scores remain robust to class imbalance rates.

1.4 Publications and Papers

There are five first-author papers related to this thesis. Three of them have already been published, while the remaining two are at different stages: one is ready for submission, and the other is still in preparation for submission, as detailed below.

- Ningsheng Zhao, Jia Yuan Yu, Trang Bui, and Krzysztof Dzieciolowski. A Transparent and Explainable Machine Learning Model Monitoring System. *In preparation for submission*.

- Ningsheng Zhao, Trang Bui, Jia Yuan Yu, and Krzysztof Dzieciolowski. Outperformance Score: A Universal Standardization for Confusion-Matrix Based Classification Performance Metrics. *Ready for submission with available manuscript*.

- Ningsheng Zhao, Jia Yuan Yu, Trang Bui, and Krzysztof Dzieciolowski. Correcting Biases of Shapley Value Attributions for Informative Machine Learning Model Explanations. In *ACM International Conference on Information and Knowledge Management, CIKM, 2024*.

- Ningsheng Zhao, Jia Yuan Yu, Krzysztof Dzieciolowski, and Trang Bui. Error Analysis of Shapley Value-Based Model Explanations: An Informative Perspective. In *AI Verification (SAIV 2024). Lecture Notes in Computer Science, vol 14846. Springer, Cham*.

- Ningsheng Zhao, Jia Yuan Yu, and Krzysztof Dzieciolowski. Classifier Rank - A New Classification Assessment Method. In *Proceedings of IADIS International Conference Big Data Analytics, Data Mining and Computational Intelligence, 2022*.

Chapter 2

Background

2.1 Notation for Data and Models

We seek to explain an ML model, denoted by $f : \mathcal{X} \rightarrow \mathcal{Y}$, which takes an instance $x = (x_1, \ldots, x_d)$ of d features, from the domain set $\mathcal{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_d)$, as input and outputs predictions for a target variable $Y \in \mathcal{Y} \subseteq \mathbb{R}$ (for classification, we typically focus on the predicted probability of a given class). In this paper, we use uppercase symbols X, Y to denote random variables, and lowercase symbols x, y to denote specific values. Furthermore, we use the notation X_S to refer to a sub-vector of X containing features in the subset $S \subseteq [d] \equiv \{1, \ldots, d\}$, and $X_{\bar{S}}$ to refer to its complementary sub-vector, which contains features from $\bar{S} = [d] \backslash S$. We assume that X and Y follow an *unknown* distribution $p(X, Y)$. Instead of the true distribution, we are provided with a dataset $\mathcal{D}_p(X, Y) = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ of N samples observed from $p(X, Y)$. This can be a training or testing set. Similarly, we use $\mathcal{D}_p(X_S, Y)$ to denote the portion of $\mathcal{D}_p(X, Y)$ containing only features in the subset S, and $\mathcal{D}_p(X, Y|X_S = x_S)$ to denote the portion containing samples that satisfy $X_S = x_S$. Thus, sub-dataset $\mathcal{D}_p(X_S, Y)$ is drawn from $p(X_S, Y)$ and $\mathcal{D}_p(X, Y|X_S = x_S)$ is drawn from $p(X, Y|X_S = x_S)$.

2.2 What is Model Explanation?

So far, there is no unified mathematical definition of *understandability, interpretability, explainablility*, and *explanation*. However, it is convenient to address the confusion about the distinction between these terms first. I prefer to summarize their definitions, based on literature (Arrieta et al., 2019) and (Miller, 2019), as the following:

- **Understandability** denotes the characteristic of a model to make a human understand its function without any need for explaining its internal structure or the algorithm means by which the model processes data internally (Arrieta et al., 2019).
- **Interpretability** is defined as a passive characteristic of a model referring to the degree to which a human observer can understand the cause of a decision (Arrieta et al., 2019; Miller, 2019). It is usually considered as two aspects:
    - **Intrinsic interpretability** refers to models that are considered interpretable due to their simple structure, such as decision trees and linear models.
    - **Post hoc interpretability** refers to the application of post hoc explanation methods after model training.
- **Explanation** is one mode in which an observer may obtain understanding, but clearly, there are additional modes that one can adopt, such as making decisions that are inherently easier to understand or via introspection (Miller, 2019).
    - **Local explanations** provide insights into individual predictions.
    - **Global explanations** provide insight into model performance across the entire dataset.
- **Explainablility** can be viewed as an active characteristic of a model, denoting any action or procedure taken by the model to clarify or detail its internal functions (Arrieta et al., 2019). However, in (Miller, 2019), the author equates 'interpretability'

with 'explainability'.

In a word, we seek to find explanation methods to increase the interpretability or explainability of a complex model so that a human observer can easily understand its output.

2.3 Feature Attribution

There are many possible ways to explain ML models, such as counterfactuals, exemplars, surrogate models, etc., but one extremely popular approach is *feature attributions*. Feature attributions explain the ML model f by quantifying each feature's contribution to a specific model output, which can be denoted by a vector $\phi = (\phi_i, \ldots, \phi_d)$, where each ϕ_i is called the *attribution score* or *importance score* of feature i. The model output could be either an individual prediction $f(x)$ for a specific sample x, or a performance metric $\mathrm{M}(f, \mathcal{D}_p(X, Y))$ evaluated across the entire dataset $\mathcal{D}_p(X, Y)$. In the former case, we term ϕ as ***local feature attribution***, whereas in the latter case, ϕ is referred to as ***global feature attribution***. For the example of linear models of form $f(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d$, each coefficient β_i can be viewed as a global feature attribution, while $\beta_i x_i$ is a reasonable local feature attribution on the given explicand. Hence, linear models are often considered interpretable because each feature is linearly related to the prediction via a single parameter.

Linear models offer a straightforward case for understanding the role of each feature by examining the model parameters. In such models, the coefficients directly indicate the influence of each feature on the prediction, providing a clear, interpretable relationship. However, this approach does not extend easily to more complex model types, such as neural networks, or tree-ensemble models. These models involve a large number of operations and interactions between features, making it difficult to interpret feature roles solely based on their parameters. To address this challenge, many researchers have turned to Shapley value explanations to summarize feature attributions for more complex models. To design

Shapley value explanation algorithms, we can adopt the ***removal-based framework***.

2.4 Removal-Based Framework

Many methods have been proposed to assign feature attributions, and almost all of them can be unified into the ***removal-based framework*** (I. C. Covert et al., 2021). This framework takes the idea that, to understand a feature's importance, remove it and see how the prediction changes. It includes three choices:

(1) (Feature removal) How does the method remove features from the model?

(2) (Model behavior) What model behavior does the method analyze?

(3) (Summary technique) How does the method summarize each feature's impact on the model?

2.4.1 Feature Removal

The principle behind removal-based explanations is to remove certain features to understand their impact on a model. However, a machine learning model requires all the input features to make predictions. Hence, each method requires a ***removal function*** of the form $f_S : \mathbb{R}^d \times 2^d \to \mathbb{R}$ to make predictions given an arbitrary subset of features. The removal function should agree with the original model f in the presence of all features, i.e., $f_{|d|}(x_{|d|}) = f(x)$. The following are some examples:

- (Marginalize with conditional) Remove features by marginalizing them out using their conditional distribution $p(X_{\bar{S}} \mid X_S = x_S)$:

$$f_S(x_S) = \mathbb{E}[f(X) \mid X_S = x_S]. \tag{1}$$

However, in practice, this approach is computationally challenging.

- (Marginalize with marginal) Remove features by marginalizing them out using their joint marginal distribution $p(X_{\bar{S}})$:

$$f_S(x_S) = \mathbb{E}_{X_{\bar{S}}}[f(x_s, X_{\bar{S}})]. \tag{2}$$

  Here, we assume that the features $X_1, \ldots, X_d$ are independent.

- (Mean values) Remove features by setting them to their mean values:

$$f_S(x_S) = f(x_S, \mathbb{E}[X_{\bar{S}}]). \tag{3}$$

  Here, we add an additional assumption of model linearity.

- (Zeros) Remove features by simply setting them to zeros:

$$f_S(x_S) = f(x_S, 0). \tag{4}$$

2.4.2 Model Behavior

To generate explanations, we have to determine which model behavior to be explained. Typically, there are two types of model behaviors:

- **Local model behaviors** related to individual predictions.
- **Global model behaviors** related to model performance on the entire dataset.

The model behavior can be quantified by a ***value function*** of the form $v : 2^d \mapsto \mathbb{R}$, which generates an output based on each subset of features $S \subseteq |d|$. The value function v is associated with the selected removal function f_S. The following are some examples:

Examples of local model behaviors:

- (Prediction) Analyze how holding out different features chances a model's prediction for an individual input x:

$$v_x(S) = f_S(x_S). \tag{5}$$

  For classification models, we can also use the log-odds ratio of the predicted probability.

- (Local loss) Consider the prediction loss, using a loss function $\mathbf{L}$, for an individual input x:

$$v_{xy}(S) = -\mathbf{L}(f_S(x_S), y). \tag{6}$$

  Those methods that choose this model behavior try to examine whether certain features make the prediction more or less correct.

Examples of global model behaviors:

- (Global loss) Consider the expected loss across the whole dataset:

$$v_{XY}(S) = -\mathbb{E}_{XY}\left[\mathbf{L}(f_S(X_S), Y)\right], \tag{7}$$

  such as MSE and cross-entropy.

- (Other performance metrics) In imbalanced classification like risk identification, instead of prediction loss, confusion-matrix-based performance metrics are preferred:

$$v_{XY}(S) = \mathbf{M}(f_S, \mathcal{D}_p(X_S, Y)), \tag{8}$$

  such as f1_score, MCC, lift, average_precision. Those methods that choose these global model behaviors aim to understand how much the model's performance degrades when certain features are withheld.

2.4.3 Summarizing Feature Attribution

The attribution score ϕ_i can be understood as each feature's contribution to the model behavior $v(S)$. However, there are total 2^d feature subsets S, plus all possible underlying feature interactions. We have too much information, but how to summarize them into a vector of d feature attributions $\phi = (\phi_1, \ldots, \phi_d)$? The following are some commonly used summarization techniques:

- (Exclude individual) Calculate the feature attribution by excluding individual features from the full set of features, i.e.,

$$\phi_i = v(|d|) - v(|d| \setminus \{i\}). \tag{9}$$

- (Include individual) Calculate the feature attribution by adding individual features to the empty set, i.e.,

$$\phi_i = v(\{i\}) - v(\emptyset). \tag{10}$$

- (Linear model) Fit a weighted linear model as a proxy for the value function v. Then, the feature attributions can be summarized using the learned coefficients:

$$\phi_1, \ldots, \phi_d = \text{argmin}_{\beta_0,\ldots,\beta_d} \sum_{S \subseteq D} \Pi(S) \left(\beta_0 + \sum_{i \in S} \beta_i - v(S) \right)^2, \tag{11}$$

  where Π is the weighting kernel. This method is also known as *Local Interpretable Model-agnostic Explanations (LIME)* (Ribeiro, Singh, & Guestrin, 2016).

- (Shapley values) Consider the model behavior as a cooperative game, then calculate the feature attributions using the Shapley values, see more details in the following section.

2.5 Cooperative Game and Shapley Value Attribution

2.5.1 Cooperative Game and Model Explanation

A *cooperative game* (Shapley, 1953) is a set function of the form $v : \mathcal{P}([d]) \mapsto \mathbb{R}$, describing the payoff achieved when a coalition of players $S \subseteq [d]$ participate in the game. Cooperative game theory research focuses on analyzing how payoffs can be distributed among players to incentivize their participation in the game, and predicting which coalitions will ultimately form. Cooperative game theory becomes increasingly important in model explanation problems because almost all the model behaviors (as discussed in Section 2.4.2) can be framed in terms of cooperative games. With this framework, the players are the model's features; the coalitions are subsets of those features; the payoff $v(S)$ corresponds to the model's output when using a coalition of features S; and the allocation of payoffs is the feature attribution that fairly reflects each feature's importance in model outputs. Specifically, the terminology in the scenario of a cooperative game is introduced as:

- *Players*: In model explanation, the features $[d] = \{1, \ldots, d\}$ act as the players in the cooperative game.
- *Coalitions*: Subsets of features S represent coalitions.
- *Payoff*: The payoff of a coalition is represented by the model's output $v(S)$ when only features in subset $S \subseteq [d]$ are considered.
- *Marginal Contribution*: a player i's *marginal contribution* to a coalition $S \in [d] \setminus \{i\}$ can be defined as the difference in the model's output when feature i is added to the coalition S, i.e., $v(S \bigcup \{i\}) - v(S)$. This measures how much value the feature adds to the coalition.

- *Allocations*: An allocation $\phi \in \mathbb{R}^d$ that assigns payoffs to each player is treated as feature attribution.

Under the removal-based framework discussed in Section 2.4, if a model behavior is viewed as a cooperative game, then each summarization technique can be understood in terms of allocation strategies for this game.

2.5.2 Shapley Value Attribution (SVA)

To be fair, allocation strategies must be designed based on each player's contribution to the game. The Shapley value (Shapley, 1953) is such a kind of fair allocation that calculates the average *marginal contribution*, $v(S \cup \{i\}) - v(S)$, of player i across all possible coalitions S that excludes i. They have been recently utilized to summarize each feature's contribution in model outputs (I. Covert et al., 2020; Lundberg & Lee, 2017). Specifically, using Shapley values, each feature i's importance score can be calculated as

$$\phi_i(v) = \sum_{S \subseteq [d] \setminus \{i\}} \pi(S) \left(v(S \cup \{i\}) - v(S) \right), \quad \text{where } \pi(S) = \frac{|S|!(d - |S| - 1)!}{d!}. \quad (12)$$

With this formula, the feature attribution $\phi(v) = (\phi_1(v), \ldots, \phi_d(v))$ is referred to as the *Shapley value attribution (SVA)*.

As discussed in Section 2.4, the SVA method can be characterized under the removal-based framework. Specifically, to design an SVA algorithm (also called a Shapley value *explainer*), we need to specify two components:

- A **removal function (RF)** $f_S(x_S)$ that can make predictions based on a sub-vector of input x_S instead of the full input vector x.

- A **value function** $v_{f_S}(S)$ associated with the selected RF f_S. For example, for local SVAs, we specify the value function as $v_{f_S}(S) = f_S(x_S)$, while for global SVAs, the

value function can be designed as $v_{f_S}(S) = \mathbf{M}(f_S, \mathcal{D}_p(X_S, Y))$ (see more discussions in (I. C. Covert et al., 2021)).

Example 1 (Local SVA)**.** *We consider a 3-dimensional case where we have an ML model f that makes predictions based on three input features $X = (X_1, X_2, X_3)$. We aim to design an SVA algorithm to assign importance scores (ϕ_1, ϕ_2, ϕ_3) to all three features to represent their impacts on an individual prediction $f(x_1, x_2, x_3)$. First, for each $S \in \mathcal{P}(\{1, 2, 3\}) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$, we specify the RF as $f_S(x_S) = f(x_S, 0)$. In other words, we remove features by setting their values as 0. Then, for the local SVA, the value function $v(S) = f(x_S, 0)$ as well. Finally, the importance scores ϕ_1, ϕ_2, ϕ_3 can be computed as*

$$\begin{aligned}\phi_1 =& \frac{1}{3}[f(x_1, 0, 0) - f(0, 0, 0))] + \frac{1}{6}[f(x_1, x_2, 0) - f(0, x_2, 0)] + \\ & \frac{1}{6}[f(x_1, 0, x_3) - f(0, 0, x_3)] + \frac{1}{3}[f(x_1, x_2, x_3) - f(0, x_2, x_3)]\end{aligned}$$

$$\begin{aligned}\phi_2 =& \frac{1}{3}[f(0, x_2, 0) - f(0, 0, 0)] + \frac{1}{6}[f(x_1, x_2, 0) - f(x_1, 0, 0)] + \\ & \frac{1}{6}[f(0, x_2, x_3) - f(0, 0, x_3)] + \frac{1}{3}[f(x_1, x_2, x_3) - f(x_1, 0, x_3)]\end{aligned}$$

$$\begin{aligned}\phi_3 =& \frac{1}{3}[f(0, 0, x_3) - f(0, 0, 0)] + \frac{1}{6}[f(x_1, 0, x_3) - f(x_1, 0, 0)] + \\ & \frac{1}{6}[f(0, x_2, x_3) - f(0, x_2, 0)] + \frac{1}{3}[f(x_1, x_2, x_3) - f(x_1, x_2, 0)].\end{aligned}$$

2.5.3 The Properties of SVAs

Fairness Axioms

Research in game theory (Hart, 1989; Roth, 1988) has proved that, based on a set of fairness axioms, Shapley values are the unique allocation of the total payoff $v([d])$ obtained

by the *grand coalition* $[d]$. Specifically, for a cooperative game v, the Shapley values are the unique credit allocation scheme that satisfies the following desirable properties:

(1) Efficiency: $\sum_{i=1}^{d} \phi_i(v) = v([d]) - v(\emptyset)$.

(2) Symmetry: if $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \in [d] \setminus \{i, j\}$, then $\phi_i(v) = \phi_j(v)$.

(3) Dummy: if $v(S) = v(S \cup \{i\})$ for all $S \in [d] \setminus \{i\}$, then $\phi_i(v) = 0$.

(4) Monotonicity: if $v(S \cup \{i\}) - v(S) \geq v'(S \cup \{i\}) - v'(S)$ for all $S \in [d] \setminus \{i\}$, then $\phi_i(v) \geq \phi_i(v')$.

(5) Linearity: if $v(S) = \sum_{k=1}^{n} c_k v_k(S)$, then for each player i, $\phi_i(v) = \sum_{k=1}^{n} c_k \phi_i(v_k)$.

Weighted Least Square Characterization

As discussed in section 2.4.3, each feature's contribution can also be summarized by solving a weighted least square problem in Equation (11), where a linear additive model of the form

$$g(S) = \beta_0 + \sum_{i \in S} \beta_i \tag{13}$$

is fitted as a proxy for the cooperative game, i.e., $g(S) \approx v(S)$. Different selections of weighting kernel Π in Equation (11) may lead to different solutions of optimal coefficients $(\beta_1^*, \ldots, \beta_d^*)$. However, surprisingly, it has been proved by Lundberg and Lee (2017) that Shapley values are the only solution when the weighting kernel is defined as

$$\Pi(S) = \frac{d-1}{\binom{d}{|S|}|S|(d-|S|)}. \tag{14}$$

It is important to note that $\Pi(\emptyset) = \Pi([d]) = \infty$, which enforces constraints $\beta_0 = v(\emptyset)$, and $\sum_{i=1}^{d} \beta_i = v([d]) - v(\emptyset)$.

For simplicity, we denote the non-intercept coefficients as $\beta = (\beta_1, \ldots, \beta_d) \in \mathbb{R}^d$, and denote each subset using the corresponding binary vector $z \in \{0, 1\}^d$ so that $v(z) \equiv v(S)$

and $\Pi(z) \equiv \Pi(S)$ for $S = \{i : z_i = 1\}$. We then define a random variable Z with distribution $p(z) \propto \Pi(z)$ when $0 < \mathbf{1}^T z < d$ and $p(z) = 0$ otherwise. With this, we can write that the Shapley values can be calculated by solving the optimization problem

$$\begin{aligned} &\text{argmin}_\beta \, \mathbb{E}_Z \left[v(\mathbf{0}) + Z^T \beta - v(Z)) \right]^2 \\ &\text{s.t.} \quad \mathbf{1}^T \beta = v(\mathbf{1}) - v(\mathbf{0}). \end{aligned} \tag{15}$$

However, to solve this problem, we must take into account all 2^d coalitions. Hence, it is hard to calculate the exact Shapley values with the high-dimension d. In Appendix A, some popular estimation approaches are introduced to solve this problem.

Chapter 3

Error Analysis of SVA Explanations: An Informative Perspective

3.1 Introduction

Generally speaking, when explaining something, we draw upon our observations and existing knowledge structures. This concept also applies to model explainers. The dataset $\mathcal{D}_p(X)$, such as the training or testing sets, serves as the observations, and the chosen removal function f_S mentioned in section 2.5.2 acts as the knowledge structure. However, both of them could be biased. Specifically, $\mathcal{D}_p(X)$ is usually too sparse to represent complex distributions (Chen et al., 2023; Sundararajan & Najmi, 2020), and estimating f_S for all possible subsets S is NP-hard (Aas et al., 2021). Due to these two reasons, almost all existing SVA algorithms are error-prone and possibly computationally expensive, leading to incorrect explanations (see discussion in Chen et al. (2023)). To gain better insights into this problem, in this chapter, we establish a unified error analysis framework for SVAs.

Under the proposed error analysis framework, all explanation errors can be decomposed into two components: observation bias and structural bias. We analyze that observation bias arises due to the data sparsity, while structural bias results from unrealistic

structural assumptions. We further demonstrate the trade-off between observation bias and structural bias. Based on this trade-off, we propose two novel concepts to describe SVAs: over-informativeness (with large observation bias) and under-informativeness (with large structural bias). Using our proposed error analysis framework, we theoretically analyze the potential over- and under-informativeness of various existing SVA methods. Furthermore, for the widely deployed distributional assumption-based SVA methods, we provide a mathematical analysis that shows how these methods can cause distribution drifts and produce under-informative explanations. To evaluate this risk, we propose a measurement tool to quantify the distribution drift.

We verify our theoretical error analyses on the Bike Sharing dataset (Fanaee-T, 2013) and the Census Income dataset (Becker & Kohavi, 1996). The experimental results confirm our theoretical analysis that SVA methods that rely on structural assumptions tend to be under-informative, while excessive data smoothing methods can be sensitive to data sparsity, especially in low-density regions. This highlights the applicability of our error analysis framework, which can discern potential errors in many existing and future feature attribution methods.

Related work We provide a comprehensive analysis of potential explanation errors of SVA methods, while related works discussing SVA errors are primarily method-specific and example-based (Aas et al., 2021; Frye et al., 2020; Mase et al., 2019; Slack et al., 2020; Sundararajan & Najmi, 2020; Yeh et al., 2022). There has not been a comprehensive theoretical analysis of the errors of SVAs. Furthermore, here we focus on the problems of SVA methods for discovering the informational dependencies between features and the target, while others consider causal relationships (Janzing et al., 2020; Taufiq, Blöbaum, & Minorics, 2023) or the conceptual inadequacies of Shapley values for explanations (Huang & Marques-Silva, 2023; I. Kumar, Scheidegger, Venkatasubramanian, & Friedler, 2021; I. E. Kumar, Venkatasubramanian, Scheidegger, & Friedler, 2020).

3.2 Conditional RF & Informative SVAs

Under the removal-based framework in Section 2.4, the removal function (RF) f_S is leveraged to assess the impact of removing features in the complement subset $\bar{S}$ from the original model f. Thus, the choice of RF significantly influences the resulting feature attributions. Recent research (Chen et al., 2023, 2020; I. Covert et al., 2020; I. C. Covert et al., 2021) emphasize that, to ensure the SVAs *faithfully* capture the informational dependencies between model outputs and input features, we should select $f_S(x_S)$ to be the conditional expectation of model prediction $f(X)$ given the feature sub-vector $X_S = x_S$. Mathematically,

$$f_S(x_S) = \mathbb{E}[f(X)|X_S = x_S] = \mathbb{E}_{p(X_{\bar{S}}|X_S=x_S)}[f(x_S, X_{\bar{S}})]. \tag{16}$$

In this case, we call f_S the *conditional RF (CRF)*, and $\phi(v_{f_s})$ the *informative SVA* or *conditional SVA*. So far, we have introduced many concepts: model explanation, feature attribution, SVA, informative SVA, local SVA, and global SVA. To avoid confusion, their relations are illustrated in the framework of Figure 3.1.

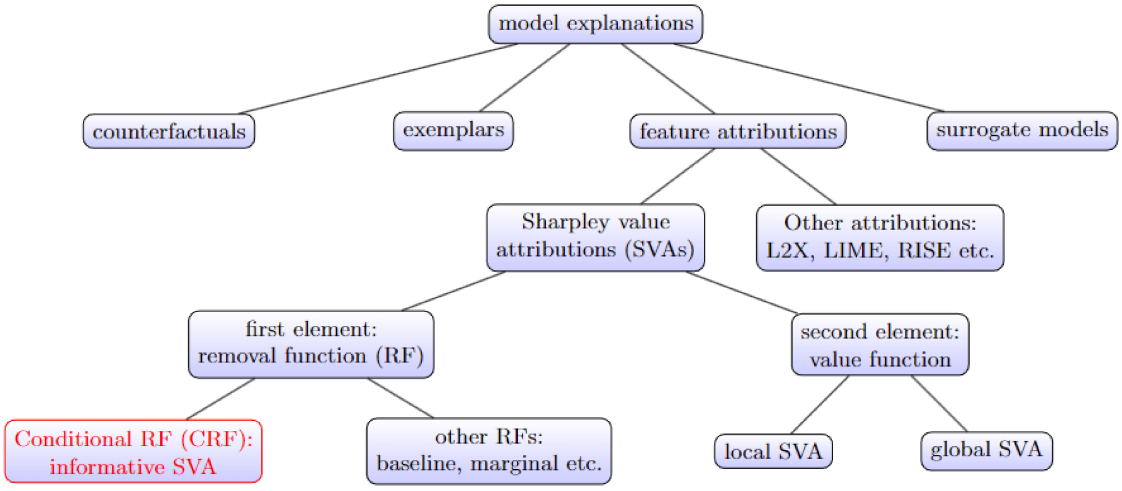


Figure 3.1: The framework of ML model explanations.

As shown in Equation (16), to compute the CRF f_S, we need to know the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$. However, it is typically unavailable in practice because the true underlying distribution $p(X)$ is unknown. Therefore, we can only estimate $f_S(x_S)$ using the given dataset $\mathcal{D}_p(X)$ (which we call the *explaining set*), such as the training set or testing set. Hence, the core objective of this study is **to estimate the CRF $f_S(x_S)$ for any arbitrary subset S given access to an ML model f and an explaining set $\mathcal{D}_p(X)$.** There are two main challenges associated with this estimation task:

NP-hard It is evident that the complete computation of SVA in Equation (12) requires the estimation of f_S for all possible $S \in \mathcal{P}([d])$ except $f_\emptyset$ and $f_{[d]}$. $f_\emptyset$ is a constant, and $f_{[d]} = f$. In the context of lower dimensions, such as the case when $d = 3$, it entails only $2^3 - 2 = 6$ function estimations: $f_{\{1\}}$, $f_{\{2\}}$, $f_{\{3\}}$, $f_{\{1,2\}}$, $f_{\{1,3\}}$ and $f_{\{2,3\}}$. As proposed by works (Lipovetsky & Conklin, 2001; Štrumbelj, Kononenko, & Šikonja, 2009), a straightforward way is to train six separate models on their corresponding sub-dataset $\mathcal{D}_p(X_1, Y)$, $\mathcal{D}_p(X_2, Y)$, $\mathcal{D}_p(X_3, Y)$, $\mathcal{D}_p(X_1, X_2, Y)$, $\mathcal{D}_p(X_1, X_3, Y)$ and $\mathcal{D}_p(X_2, X_3, Y)$, respectively. However, the number of required models grows exponentially with dimension d, leading to a significant computational challenge. For instance, when $d = 20$, it is impractical to train $2^{20} - 2 = 1048574$ models, especially for complex ML models like neural networks and tree ensemble models. Therefore, it is imperative to design a scalable estimation algorithm.

Data Sparsity In essence, for each $f_S(x_S)$, we need to estimate the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ using the explaining set $\mathcal{D}_p(X)$. For example, $f_S(x_S)$ can be empirically estimated from samples in the explaining set that match the condition $X_S = x_S$, as follows:

$$f_S(x_S) \approx \mathbb{E}_{x^{(n)} \sim \mathcal{D}_p(X|X_S = x_S)} f(x^{(n)}) = \frac{1}{\sum_{n=1}^{N} \mathbb{1}(x_S^{(n)} = x_S)} \sum_{n=1}^{N} f(x^{(n)}) \mathbb{1}(x_S^{(n)} = x_S). \tag{17}$$

However, in the explaining set, there could be very few or even no samples that match the condition $X_S = x_S$. In other words, the number $\sum_{n=1}^{N} \mathbb{1}(x_S^{(n)} = x_S)$ could be too low or even 0. This problem usually happens in problems that involve high-dimensional or many-valued features (Chen et al., 2023; Sundararajan & Najmi, 2020). For example, within a "bank dataset", it is unlikely to find any individual that exactly satisfies the condition: "$\texttt{credit\_score} = 3.879, \texttt{income} = \$112,643$".

3.3 The Estimation of CRF

While training separate models and empirical estimation are often impractical due to the above two challenges, various popular scalable methodologies have been proposed in recent research to estimate the CRF f_S (see discussion in I. C. Covert et al. (2021)). These methodologies can be categorized into two main approaches: smoothing the data and making distributional assumptions.

3.3.1 Data-Smoothing Approaches

To address the challenge of data sparsity, data-smoothing approaches focus on approximating the conditional expectation $\mathbb{E}[f(X)|X_S = x_S]$ by smoothing the provided explaining set $\mathcal{D}_p(X) = \{x^{(n)}\}_{n=1}^{N}$. The underlying rationale is the assumption that samples in the explaining set with feature sub-vectors $x_S^{(n)}$ similar to the target value x_S provide informative insights into the conditional expectation $\mathbb{E}[f(X)|X_S = x_S]$. Essentially, Smoothing relaxes the strict condition that exactly matches $X_S = x_S$. The explaining set can typically be smoothed using two popular ways: non-parametric kernel-based approaches or parametric model-based approaches.

Kernel-Based Non-Parametric Estimation

In section 3.2, we discuss that strict empirical estimation may be susceptible to data sparsity. To alleviate this, some relaxed empirical methods make the estimation using samples with similar feature values $X_S \approx x_S$, rather than exact match $X_S = x_S$. Hence, a similarity weight is assigned to each sample $x^{(n)}$ in the explaining set using a predefined kernel function $\kappa^{(n)}(x_S)$. The CRF f_S can then be approximated by the weighted average of the model predictions, as follows:

$$f_S(x_S) \approx \frac{1}{\sum_{n=1}^{N} \kappa^{(n)}(x_S)} \sum_{n=1}^{N} \kappa^{(n)}(x_S) f(x^{(n)}). \tag{18}$$

For instance, the cohort kernel (Mase et al., 2019) and Gaussian kernel (Aas et al., 2021) have been proposed as the similarity weight.

- With the cohort kernel, the similarity weight is defined as

$$\kappa^{(n)}(x_S) = \prod_{i \in S} \mathbb{1}(|x_i^{(n)} - x_i| \leq \sigma_i) = \begin{cases} 1, & \text{if } |x_i^{(n)} - x_i| \leq \sigma_i, \ \ \forall i \in S \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

  For each feature $i \in S$, the condition $X_i = x_i$ is relaxed into $|X_i - x_i| \leq \sigma_i$ with a selected *bandwidth* σ_i controlling the smoothness. For example, rather than using samples that exactly match "$income = \$112,643$", we use samples that satisfy "$|income - \$112,643| \leq \$5000$".

- With the Gaussian kernel, the similarity weight is defined as

$$\kappa^{(n)}(x_S) = \exp\left(-\frac{D(x_S, x_S^{(n)})^2}{2\sigma^2}\right), \tag{20}$$

  where $D(\cdot)$ represents a distance function and σ is the selected bandwidth. Notably,

the Gaussian kernel is a soft version of the cohort kernel.

Model-based parametric estimation

We can also use a parametric model to learn the valuable information provided by an arbitrary conditioned value $X_S = x_S$. The model could be the original model f if it has a tree structure, or a new model trained with the Empirical Risk Minimization (ERM) principle (Shalev-Shwartz & Ben-David, 2014) on the explaining set $\mathcal{D}_p(X)$.

- (Conditional Generative Model) Since only using samples from the explaining set $\mathcal{D}_p(X)$ may suffer from the data sparsity, recent researches (Belghazi, Oquab, & Lopez-Paz, 2019; Frye et al., 2020) proposed drawing samples from a conditional generative model. For instance, Frye et al, (Frye et al., 2020) introduce a masked variational autoencoder(MVA) model, comprising three integral components: an encoder $q_\xi(Z|X)$, a decoder $p_\theta(X|Z)$, and a masked encoder $e_\psi(Z|X_S)$ that maps an arbitrary sub-vector X_S to the latent space that agrees with the encoder $q_\xi(Z|X)$ as well as possible. Consequently, the CRF f_S can be approximated as

$$f_S(x_S) \approx \int \mathbb{E}_{p_\theta(X|Z=z)}[f(X)]e_\psi(Z = z|X_S = x_S)dz. \tag{21}$$

  Via the MVA model, the expected model prediction is estimated conditionally on the latent variable $Z = z$ inferred from the information $X_S = x_S$, rather than being directly conditioned on it.

- (Surrogate Model) Alternatively, Frye et al, (Frye et al., 2020) proposed the adoption of a supervised surrogate model $h_\theta(x_S)$ for the direct estimation of the CRF $f_S(x_S)$. The surrogate model is a neural network trained to match the original model's predictions, with masked features represented by zeros. The parameter set θ can be

estimated by minimizing the empirical MSE loss function:

$$\hat{\theta} = \text{argmin}_\theta \, \mathbb{E}_{x \sim \mathcal{D}_p(X)} \mathbb{E}_{S \sim Shapley} [h_\theta(x_S) - f(x)]^2. \tag{22}$$

It has been demonstrated that both the MVA model and the surrogate model offer the same flexibility across diverse distributions, but the surrogate model may be more effective in practice (Chen et al., 2023; Frye et al., 2020). However, it is worth noting that the complete training process for both models demands an exponential number of samples, rendering them susceptible to the curse of dimensionality.

- (Tree-Structured Model) TreeSHAP (Lundberg et al., 2020) is a specific SVA for tree-structured models. TreeSHAP roughly approximates the conditional expectation $\mathbb{E}[f(X)|X_S = x_S]$ by averaging the predictions from all possible leaves that are not against the condition $X_S = x_S$, weighted by the proportion of the explaining set $\mathcal{D}_p(X)$ falling in those leaves. Essentially, this procedure relaxes the condition $X_S = x_S$ into a set of branches induced by the conditioned value. For instance, considering a stump with only two edges "$X_1 < 10$" and "$X_1 \geq 10$", we approximate $\mathbb{E}[f(X)|X_1 = 8] \approx \mathbb{E}[f(X)|X_1 < 10]$.

3.3.2 Distributional Assumptions-Based Approaches

Data-smoothing methods usually leverage samples drawn from the true underlying distribution $p(X)$ but with relaxed conditions. Conversely, distributional assumptions-based approaches focus on samples that strictly adhere to the condition $X_S = x_S$ but from an assumed distribution $q(X_{\bar{S}}|X_S = x_S)$. This assumed distribution is a rough approximation of the true conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ based on certain robust assumptions. Once $q(X_{\bar{S}}|X_S = x_S)$ is defined, $f_S(x_S)$ can be empirically approximated through samples

drawn from it, as follows:

$$f_S(x_S) \approx \mathbb{E}_{x'_{\bar{S}} \sim q(X_{\bar{S}}|X_S=x_S)}[f(x_S, x'_{\bar{S}})]. \tag{23}$$

Here, $(x_S, x'_{\bar{S}})$ represents a *hybrid sample* where the removed features in $\bar{S}$ take replacement values sampled from the assumed distribution $q(X_{\bar{S}}|X_S = x_S)$.

Baseline RF with Constant Assumption

The simplest way to remove features in subset $\bar{S}$ is to replace their values with a fixed *baseline* x^b, i.e., we let $X_{\bar{S}} = x^b_{\bar{S}}$ (Chen et al., 2023; Sundararajan & Najmi, 2020). This essentially assumes that the missing features are constant values, rather than random variables following a complex conditional distribution. Formally, we define the assumed removal distribution $q(X_{\bar{S}}|X_S = x_S) = \mathbb{1}(X_{\bar{S}} = x^b_{\bar{S}})$, and then the approximation formula for f_S in Equation (23) simplifies to:

$$f_S(x_S) \approx f(x_S, x^b_{\bar{S}}). \tag{24}$$

This is also called the *baseline RF*. While the constant assumption might be overly restrictive, it greatly streamlines the computational process. This makes the baseline RF a practical choice, particularly for scenarios where computational resources are limited.

Marginal RF with Feature Independence Assumption

Another common way is to assume feature independence (I. Covert et al., 2020; Lundberg & Lee, 2017). For any given subset S, if we assume that the sub-vector X_S is independent of its complement $X_{\bar{S}}$, then the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ becomes the joint marginal distribution $p(X_{\bar{S}})$, and the CRF becomes the *marginal RF*. In this case, the empirical estimate can be easily implemented using the entire dataset $\mathcal{D}_p(X_{\bar{S}})$ rather

than matched sub-dataset $\mathcal{D}_p(X_{\bar{S}}|X_S = x_S)$:

$$f_S(x_S) \approx \mathbb{E}_{p(X_{\bar{S}})}[f(x_S, X_{\bar{S}})] \approx \frac{1}{N}\sum_{n=1}^{N} f(x_S, x_{\bar{S}}^{(n)}). \tag{25}$$

It is even possible to assume independence among all features, i.e., $q(X_{\bar{S}}|X_S = x_S) = \prod_{i\in\bar{S}} p(X_i)$, which is referred to as the *product of marginal RF* (Datta, Sen, & Zick, 2016). However, this assumption is significantly stronger but offers no additional computational benefit, so it is seldom used in practice.

Parametric RF with Parametric Assumption

We can also assume $p(X)$ is a parametric distribution, such as Gaussian or uniform distribution. As discussed in literature (Aas et al., 2021; Chen et al., 2020; Janzing et al., 2020), if we assume a multivariate Gaussian distribution, the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ can then be written by a closed-form formula (see Appendix B). In this case, we the approximated RF the *Gaussian RF*. However, this formula requires a matrix inversion with complexity $O(|S|^3)$, which is computationally expensive in high-dimensional cases. Another approach is the *uniform RF*, which assumes that each removed feature follows an independent uniform distribution (Aas et al., 2021; Chen et al., 2023). However, it makes even stronger assumptions than the product of marginal, thus also seldom used in practice.

3.4 Observation Bias & Structural Bias Trade-Off

The SVA in Equation (12) is a function of the value function $v(S)$. Furthermore, the value function is intrinsically related to the CRF, which is estimated based on the explaining set and the selected approach framework. As a result, errors in estimating the CRF will directly cause errors in evaluating the value function, leading to errors in SVAs.

3.4.1 Overfitting and Underfitting of the RF

We use the notation $\hat{f}_S^{(N)}$ to denote an estimated CDF based on an explaining set of size N. Let $\hat{f}_S = \lim_{N\to\infty} \hat{f}_S^{(N)}$ be the limit of the estimate when using an infinitely large explaining set. For instance, Frye et al. (2020) proposed adopting a supervised surrogate model $h_\theta(x_S)$ for the estimation of the CDF $f_S(x_S)$. In this case, $\hat{f}_S^{(N)}(x_S) = h_{\hat{\theta}^{(N)}}(x_S)$ and $\hat{f}_S(x_S) = h_{\theta^*}(x_S)$, where $\hat{\theta}^{(N)}$, θ^* are obtained by minimizing the empirical MSE and true MSE, respectively. In essence, $\hat{f}_S^{(N)}$ is an estimate of $\hat{f}_S$, and $\hat{f}_S$ is a proxy for the true CDF f_S.

The error associated with an estimated RF $\hat{f}_S^{(N)}$ can be decomposed into two components: the *estimation error* and the *approximation error* (Shalev-Shwartz & Ben-David, 2014), expressed as:

$$\begin{aligned} \hat{f}_S^{(N)} - f_S &= (\hat{f}_S^{(N)} - \hat{f}_S) + (\hat{f}_S - f_S) \\ &= \epsilon_{estimation} + \epsilon_{approximation}. \end{aligned} \tag{26}$$

The estimation error quantifies the risk of utilizing a finite dataset for the CRF estimation. This type of error can be highly sensitive to data sparsity but can be mitigated by either smoothing the data (Sundararajan & Najmi, 2020) or increasing the data size. The estimated RF $\hat{f}_S^{(N)}$ is said to be *overfitting* at a point $X_S = x_S$ if it exhibits a significant absolute estimation error $|\hat{f}_S^{(N)}(x_S) - \hat{f}_S(x_S)|$.

On the other hand, the approximation error measures the level of risk associated with making distributional or modeling assumptions. In this case, the estimated RF $\hat{f}_S^{(N)}$ is said to be *underfitting* at a point $X_S = x_S$ if it demonstrates a significant absolute approximation error $|\hat{f}_S(x_S) - f_S(x_S)|$. It is worth noting that underfitting cannot be alleviated through an increase in data size, but can be exacerbated by excessive data smoothing.

3.4.2 Explanation Error Decomposition

Since we use $\hat{f}_S^{(N)}$ to estimate the true CRF f_S, the true value function v_{f_S} is estimated by $v_{\hat{f}_S^{(N)}}$. The difference between these two value functions causes explanation errors for the SVAs in Equation (12). Using similar ideas as in Section 3.4.1, we propose to decompose the explanation error into

$$\begin{aligned} \phi(v_{\hat{f}_S^{(N)}}) - \phi(v_{f_S}) &= \Big(\phi(v_{\hat{f}_S^{(N)}}) - \phi(v_{\hat{f}_S})\Big) + \Big(\phi(v_{\hat{f}_S}) - \phi(v_{f_S})\Big) \\ &= \textit{observation bias} + \textit{structural bias}. \end{aligned} \tag{27}$$

We call the first component $\phi(v_{\hat{f}^{(N)}}) - \phi(v_{\hat{f}})$ the *observation bias*, which occurs because we make explanations based on only a finite number of observations of the whole distribution. Next, we call the second component $\phi(v_{\hat{f}}) - \phi(v_f)$ the *structural bias*, arising from the utilization of an imperfect or limited knowledge structure to make explanations. While observation bias is caused by the estimation error, structural bias arises from the approximation error (see Equation (26)).

Observation bias may become substantial when the explaining set is too sparse to accurately capture the complex underlying distribution. To mitigate this, we can make simplifying structural assumptions to approximate f_S, for example, by using a surrogate model or an assumed distribution. However, imposing assumptions may cause the approximation to be inadequate. For example, using a surrogate model $h_\theta(x_S)$ with complexity $|\theta|$ may be insufficient to encompass a perfect θ^* that satisfy $h_{\theta^*} = f_S$. Moreover, making unrealistic distributional assumptions may drift the true underlying distribution $p(X)$ to a different one $q(X)$. Therefore, there is typically a trade-off between observation bias and structural bias in estimating the CRF using a finite explaining set. Figure 3.2 gives an illustration of this trade-off.

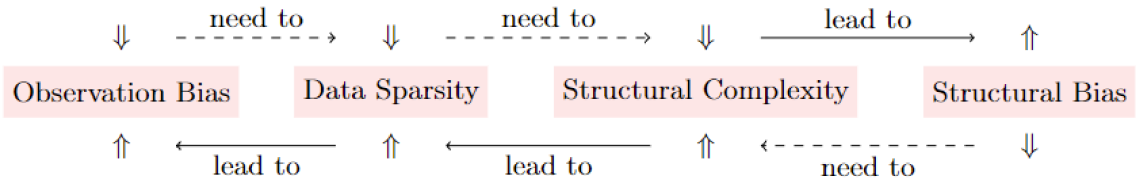


Figure 3.2: An illustration of the trade-off between observation bias and structural bias. On one hand, to reduce observation bias, it is necessary to alleviate the data sparsity, which requires us to decrease the structural complexity of the CRF approximation. However, this simplification of structural complexity might concurrently lead to an increase in structural bias. On the other hand, to reduce structural bias, we may need to increase the structural complexity, which inevitably entails an aggravation of the data sparsity, consequently increasing the observation bias.

3.4.3 Over-informative Explanation

When the absolute value of observation bias $|\phi(v_{\hat{f}_S^{(N)}}) - \phi(v_{\hat{f}_S})|$ is large, we say that the corresponding SVA is *over-informative*. Over-informativeness often manifests in high-dimensional data and low-density regions, where the provided explaining set is typically too sparse to represent the whole population. Consequently, the estimated RF $\hat{f}_S^{(N)}$ can easily be overfitting, resulting in an undesirable observation bias. When the SVA is over-informative, it may erroneously assign importance to uninformative or noisy features. To better illustrate the concept of over-informative SVAs, we present a toy example on two-dimensional data below.

Example 2 (Over-informative SVA)**.** *Consider model $f(x_1, x_2) = 10x_2$ based on two independent features, X_1 and X_2. Suppose $X_1 \sim \mathcal{N}(0, 1)$ and $X_2 \sim \mathcal{N}(0, 1)$. Now, consider the case where we do not know the true distribution of (X_1, X_2), and we only observe a dataset of 100 samples $\{(x_1^{(1)}, x_2^{(1)}), \ldots, (x_1^{(100)}, x_2^{(100)})\}$. Suppose this dataset contains an outlier $(x_1, x_2) = (5, 1)$, where the value $X_1 = 5$ is notably greater than that of all other samples. The objective is to explain the prediction $f(5, 1) = 10$. According to the Shapley value formula in Equation (12), in order to obtain feature attribution ϕ, we need to estimate the CRFs $f_{\{\emptyset\}}, f_{\{1\}}, f_{\{2\}}$. Let us consider the empirical estimates (Sundararajan & Najmi,*

2020) of these CRFs at $(5, 1)$*, which are:*

$$\hat{f}^{(100)}_{\{\emptyset\}}(x_\emptyset) = \frac{1}{100}\sum_{i=1}^{100} f(x_1^{(i)}, x_2^{(i)}) = \frac{1}{100}\sum_{i=1}^{100} 10x_2^{(i)} \approx 0,$$
$$\hat{f}^{(100)}_{\{1\}}(5) = \frac{\sum_{i=1}^{100} \mathbb{I}(x_1^{(i)} = 5) f(x_1^{(i)}, x_2^{(i)})}{\sum_{i=1}^{100} \mathbb{I}(x_1^{(i)} = 5)} = 10,$$
$$\hat{f}^{(100)}_{\{2\}}(1) = \frac{\sum_{i=1}^{100} \mathbb{I}(x_2^{(i)} = 1) f(x_1^{(i)}, x_2^{(i)})}{\sum_{i=1}^{100} \mathbb{I}(x_2^{(i)} = 1)} = 10.$$

With these estimates, using Equation (12)*, we can calculate* $\hat{\phi}_1 \approx 5$*. This implies that* X_1 *contributes half to the prediction* $f(5, 1) = 10$*. However, it is clear that, in reality,* X_1 *is an uninformative feature for* f *and* ϕ_1 *should always be 0. This error occurs because we observe only one sample with* $X_1 = 5$ *in the dataset, making the empirical estimator* $\hat{f}^{(100)}_{\{1\}}$ *overfitting at* $(5, 1)$*. Since the true CRF is* $f_{\{1\}} = 0$*, the estimation error is 10, causing the observation bias to be 5. In this case, the SVA score* $\hat{\phi}_1$ *is over-informative and it erroneously assigns importance to irrelevant features.*

3.4.4 Under-informative Explanation

Conversely, when the absolute value of structural bias $|\phi(v_{\hat{f}_S}) - \phi(v_{f_S})$ is large, we say that the corresponding SVA is *under-informative*. In practice, making unreasonable assumptions is the primary reason for under-informativeness. When the SVA is under-informative, it may underestimate or even ignore some relevant mutual information between input features and model outputs. For example, Chen et al. (2020) demonstrates that assuming feature independence can result in highly correlated features receiving considerably different importance scores. We give a toy two-dimensional example below to illustrate an under-informative SVA.

Example 3 (Under-informative SVA)**.** *Suppose we are given two features* X_1 *and* X_2*, where* $X_1 = 2X_2$*, representing the same factor in two different units, e.g., price in different*

currencies or temperature in different scales. Consider two linear models $f_1(x_1, x_2) = 10x_1 + x_2$ *and* $f_2(x_1, x_2) = x_1 + 19x_2$*, which both equals* $21x_2$*. In essence,* f_1 *and* f_2 *are the same models with different algebraic forms. However, under the feature independence assumption, they can be explained in two different ways. Assume* $\mathbb{E}[X_1] = \mathbb{E}[X_2] = 0$ *and suppose we are interested in explaining the same prediction* $f_1(2, 1) = f_2(2, 1) = 21$*. Using the SVA formula for linear models under independent feature assumptions*[1]*, we can calculate* $\hat{\phi}_1 = 20, \hat{\phi}_2 = 1$ *for* f_1*, and* $\hat{\phi}_1 = 2, \hat{\phi}_2 = 19$ *for* f_2*. That means* X_1 *is given dominantly high feature attribution for* f_1 *while* X_2 *is given dominantly high feature attribution for* f_2*. In reality,* X_1 *and* X_2 *should receive the same attribution score, i.e.,* $\phi_1 = \phi_2$*, because they provide the same information. In this case, both explanations are under-informative due to the unrealistic feature independence assumption.*

In summary, SVA could be over-informative if it is estimated based on insufficient observations. Meanwhile, it could also be under-informative if it is approximated based on unrealistic structural assumptions. In the following sections, we use the error analysis framework proposed in Equation (27) to analyze the over- and under-informativeness of existing CRF estimation methods. As discussed in Chapter 3.3, these methods can be categorized into two main approaches: smoothing the data and making distributional assumptions.

3.4.5 Explanation Error Analysis of Data-Smoothing Approaches

To address the challenge of data sparsity, one effective method is to smooth the explaining set. Typically, the data can be smoothed using either non-parametric kernel-based approaches or parametric model-based approaches. However, excessive data smoothing can lead to serious structural bias. Unfortunately, it is unclear to what extent the explaining

[1]Following (Lundberg & Lee, 2017), given a linear model $f(x) = \sum_{j=1}^{d} \beta_j x_j + \beta_0$, under the feature independence assumption, the SVA for the jth feature can be calculated as $\phi_j = \beta_j(x_j - \mathbb{E}[X_j])$.

set should be smoothed (Sundararajan & Najmi, 2020). Below we analyze the potential explanation errors of some popular data smoothing methods.

Empirical CRF : the structural bias is zero because the empirical estimator will converge to the true CRF when the data size goes to infinity. However, the empirical CRF is usually seriously over-informative when data sparsity exists (as illustrated in Example 1).

Non-parametric kernel-based approaches : for this type of approach, the extent of data smoothing is controlled by the bandwidth(s) of the kernel, which could be set either too conservatively, resulting in over-informativeness, or too generously, leading to under-informativeness. Moreover, the selected kernel function might not correctly define the similarity between samples (Chen et al., 2023), causing undesirable structural bias.

Parametric model-based approaches : for both the conditional generative model and supervised surrogate model proposed in (Frye et al., 2020), the extent of data smoothing is controlled by the complexity of the selected neural networks. Over-informativeness and under-informativeness respectively coincide with the overfitting and underfitting of the trained neural network. However, controlling the overfitting and underfitting of this trained neural network is challenging. First, since the neural network is trained on an exponential number of all possible sub-datasets $\mathcal{D}_p(X_S)$, it is sometimes difficult to ensure learning optimality within an acceptable computation time (Chen et al., 2023). As a result, non-optimal learning may result in structural bias. Furthermore, even if a neural network is well-trained, it might still be overfitting under data sparsity in low-density regions (see examples in (Yeh et al., 2022)), causing observational bias.

TreeSHAP : this is a specific SVA method for tree-structured models. TreeSHAP is usually under-informative. First, it utilizes the predefined tree structure of the original model, which was trained under unclear assumptions about feature dependencies (Aas et al., 2021).

Second, it approximates the conditional expectation $\mathbb{E}[f(X)|X_S = x_S]$ by averaging the predictions from all leaves that are not against the condition $X_S = x_S$. Essentially, this procedure relaxes the condition $X_S = x_S$ into a set of weaker conditions. For instance, with a stump containing two leaves "$X_1 < 10$" and "$X_1 \geq 10$", we approximate $\mathbb{E}[f(X)|X_1 = 8]$ by $\mathbb{E}[f(X)|X_1 < 10]$. This relaxation of conditions introduces structural bias.

3.4.6 Explanation Error Analysis of Distributional Assumptions-Based Approaches

Besides smoothing the data, an alternative way to mitigate data sparsity is to approximate the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ with an assumed distribution $r(X_{\bar{S}})$. In this work, we call $r(X_{\bar{S}})$ the *removal distribution*, as it is the assumed distribution for removed feature subset $X_{\bar{S}}$. As discussed in Section 3.3.2, there are four common removal distributions:

(1) *Baseline*: $r(X_{\bar{S}}) = \mathbb{1}(X_{\bar{S}} = x^b_{\bar{S}})$, assuming $X_{\bar{S}}$ has a constant value $x^b_{\bar{S}}$.

(2) *Marginal*: $r(X_{\bar{S}}) = p(X_{\bar{S}})$, assuming X_S and $X_{\bar{S}}$ are independent.

(3) *Product of marginal*: $r(X_{\bar{S}}) = \prod_{i \in \bar{S}} p(X_i)$, assuming each feature in $\bar{S}$ is independent.

(4) *Uniform*: $r(X_{\bar{S}}) = \prod_{i \in \bar{S}} u_i(X_i)$, where u_i denotes a uniform distribution over $\mathcal{X}_i$. In this case, each feature in $\bar{S}$ is assumed to be independently and uniformly distributed.

With $p(X_{\bar{S}}|X_S = x_S) \approx r(X_{\bar{S}})$, the CRF f_S in formula (16) can be approximated as

$$\hat{f}_S(x_S) = \mathbb{E}_{r(X_{\bar{S}})}[f(x_S, X_{\bar{S}})] = \int f(x_S, x'_{\bar{S}}) r(X_{\bar{S}} = x'_{\bar{S}}) dx'_{\bar{S}}, \tag{28}$$

which can be empirically estimated by

$$\hat{f}_S^{(N)}(x_S) = \frac{1}{N}\sum_{n=1}^{N} f(x_S, x_{\bar{S}}^{(n)}), \tag{29}$$

using an explaining set $\mathcal{D}_r(X) = \{(x^{(n)})\}_{n=1}^N$ drawn from $r(X)$.

Observational bias: The purpose of making assumptions is to reduce the distribution complexity, and thus the observation bias. In particular, to estimate the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ for any arbitrary x_S, we require a dataset with complexity $O(|\mathcal{X}|)$. This complexity will change when using an assumed removal distribution $r(X_{\bar{S}})$. Table 3.1 summarizes the data complexity requirement for the above four removal distributions.

Table 3.1: The complexity of different removal functions.

Removal distribution	**Formula**	**Data complexity required**
Conditional	$p(X_{\bar{S}}\|X_S = x_S)$	$O(\|\mathcal{X}\|)$
Baseline	$\mathbb{1}(X_{\bar{S}} = x_{\bar{S}}^b)$	$O(1)$
Marginal	$p(X_{\bar{S}})$	$O(\|\mathcal{X}_{\bar{S}}\|)$
Product of marginals	$\prod_{i\in\bar{S}} p(X_i)$	$O\left(\prod_{i\in\bar{S}} \|\mathcal{X}_i\|\right)$
Uniform	$\prod_{i\in\bar{S}} u_i(X_i)$	$O\left(\prod_{i\in\bar{S}} \|\mathcal{X}_i\|\right)$

From Table 3.1, we can see that the baseline removal distribution simplifies the conditional distribution into a constant value, thus having a zero observation bias. The marginal removal distribution also decreases the data complexity requirement from $O(|\mathcal{X}|)$ into $O(|\mathcal{X}_{\bar{S}}|)$. However, not all the distributional assumptions can ensure a decrease in complexity, even though the assumptions are strong. For example, both products of marginal and uniform removal distributions require a dataset with a complexity of $O\left(\prod_{i\in\bar{S}} |\mathcal{X}_i|\right)$, which might not be necessarily lower than the complexity requirement of conditional distribution (i.e., $O(|\mathcal{X}|)$) in the presence of dependencies among features.

Structural bias: By reducing the data complexity requirement, making some distributional assumptions can reduce the observation bias. However, if these assumptions are far from the true underlying distribution, they could also engender considerable structural bias. Specifically, distributional assumptions can make the true joint distribution $p(X)$ drift towards a different distribution $q(X)$, where $q(X_{\bar{S}}|X_S = x_S) = r(X_{\bar{S}})$. To analyze the structural bias induced by distributional drift, we introduce the following definitions.

Definition 1. *A sample x is defined as an **out-of-distribution (OOD) sample** of $p(X)$, denoted as $x \notin p(X)$, if $p(X = x) = 0$. Conversely, if $p(X = x) > 0$, it is defined as an **in-distribution sample** of $p(X)$, denoted as $x \in p(X)$.*

Definition 2. *The **OOD rate** of $q(X)$ to $p(X)$ is defined as the proportion of samples drawn from $q(X)$ that are OOD samples of $p(X)$, denoted as $\mathbf{Pr}\{X \notin p(X)|X \in q(X)\}$.*

For an arbitrary value x_S observed from $p(X_S)$, the instance $x = (x_S, x'_{\bar{S}})$ where $x'_{\bar{S}} \sim r(X_{\bar{S}})$ is called a hybrid sample (Chen et al., 2023). As a result of the distribution drift, hybrid samples $(x_S, x'_S) \sim q(X)$ could be either in-distribution or OOD samples of $p(X)$. Thus, we can derive the approximation error of the CRF estimator $\hat{f}_S(x_S)$ in Equation (28) as

$$
\begin{aligned}
&\hat{f}_S(x_S) - f_S(x_S) \\
=& \int_{(x_S, x'_{\bar{S}}) \in q(X)} f(x_S, x'_{\bar{S}}) r(X_{\bar{S}} = x'_{\bar{S}}) dx'_{\bar{S}} - f_S(x_S) \\
=& \int_{(x_S, x'_{\bar{S}}) \notin p(X)} f(x_S, x'_{\bar{S}}) r(X_{\bar{S}} = x'_{\bar{S}}) dx'_{\bar{S}} \quad + \\
& \int_{(x_S, x'_{\bar{S}}) \in p(X)} f(x_S, x'_{\bar{S}}) r(X_{\bar{S}} = x'_{\bar{S}}) dx'_{\bar{S}} - f_S(x_S) \\
=& \int_{(x_S, x'_{\bar{S}}) \notin p(X)} f(x_S, x'_{\bar{S}}) r(X_{\bar{S}} = x'_{\bar{S}}) dx'_{\bar{S}} \quad + \\
& \int_{(x_S, x'_{\bar{S}}) \in p(X)} f(x_S, x'_{\bar{S}}) \left[r(X_{\bar{S}} = x'_{\bar{S}}) - p(X_{\bar{S}} = x'_{\bar{S}}|X_S = x_S) \right] dx'_{\bar{S}}.
\end{aligned} \tag{30}
$$

Therefore, the approximation error of assumption-based RFs stems from two sources: (i) the inclusion of OOD samples in the approximation; and (ii) changes in the probability density of in-distribution samples. The OOD sample-related approximation error may contribute to a large proportion of structural bias, especially when the OOD rate is high. In practice, some OOD samples may be senseless. For instance, the OOD samples could represent a bank client who is 20 years old but has 25-year working experience, or a clinic patient whose systolic blood pressure is lower than his diastolic blood pressure. Moreover, adversarial attacks have been designed in the literature (Slack et al., 2020) to arbitrarily manipulate model explanations (feature attributions). Under our error analysis framework, it is easy to see that these attacks essentially target the OOD sample-related approximation error in Equation (30), intentionally modifying the structural bias.

3.5 OOD Measurement of Distribution Drift

In practice, assumption-based RFs, such as the baseline RF and marginal RF, are widely used thanks to their simple implementations (Lin et al., 2024). For these methods, explanation errors mainly arise from structural bias caused by distributional assumptions, which are unchangeable once the assumptions are made. Hence, evaluating structural bias or under-informativeness resulting from distributional assumptions is crucial. However, it is impossible to directly measure the structure bias because the true conditional RF f_S is unknown. As discussed in Section 3.4.6, structural bias arises from distribution drift, which usually leads to the use of OOD samples in estimating SVAs. Therefore, we can alternatively assess structural bias or under-informativeness by measuring how much the distribution drifts, and how high the OOD rate is.

3.5.1 Distribution Drift

Let $\mathbf{S}$ be a random variable on domain $\mathcal{P}([d]) \setminus [d]$ (i.e., the power set of $[d]$ excluding $[d]$, which is the set of all possible subsets involved in the computation of SVA scores for all d features).

Lemma 1. *For each* $S \in \mathcal{P}([d]) \setminus [d]$, $\mathbf{Pr}\{\mathbf{S} = S\} = \frac{1}{d \cdot \binom{d}{|S|}}$.

Proof. According to Equation (12), the Shapley value feature attribution of the ith feature ϕ_i is essentially the weighted average of feature i's marginal contribution over all possible subsets $S \subseteq [d] \setminus \{i\}$, with weights equal $\pi(S)$. In the context of all d features, a subset S only appears when computing SVA scores for features that are not in S. There are $d - |S|$ such features. Therefore, the probability function of $\mathbf{S}$ can be derived as

$$\mathbf{Pr}\{\mathbf{S} = S\} = \frac{d - |S|}{d}\pi(S) = \frac{d - |S|}{d} \cdot \frac{|S|!(d - |S| - 1)!}{d!} = \frac{1}{d \cdot \binom{d}{|S|}}. \qquad \square$$

$\square$

Given $\mathbf{S} = S$ and an instance x, we have

$$p(X = x|\mathbf{S} = S) = p(X_S = x_S)p(X_{\bar{S}} = x_{\bar{S}}|X_S = x_S).$$

By assuming a removal distribution $r(X_{\bar{S}})$ on the conditional distribution $p(X_{\bar{S}} = x_{\bar{S}}|X_S = x_S)$, the distribution drift into

$$q\left(X = x|\mathbf{S} = S\right) = p(X_S = x_S)r(X_{\bar{S}} = x_{\bar{S}}). \tag{31}$$

Then, considering all possible subsets S, the marginal density of a hybrid sample $x \sim q(X)$

can be computed as

$$q\left(X=x\right)=\frac{1}{d}\sum_{S\in\mathcal{P}([d])\setminus[d]}\frac{1}{\binom{d}{|S|}}p(X_S=x_S)r(X_{\bar{S}}=x_{\bar{S}}). \tag{32}$$

If the assumed removal distribution $r(X_{\bar{S}}) \neq p(X_{\bar{S}}|X_S = x_S)$, there will be a distribution drift from $p(X)$ to $q(X)$. For example, when using baseline and marginal removal distributions, the true distribution $p(X)$ could drift into $q^{baseline}(X)$ and $q^{marginal}(X)$, respectively, where

$$q^{baseline}(X)=\frac{1}{d}\sum_{S\in\mathcal{P}([d])\setminus[d]}\frac{1}{\binom{d}{|S|}}p(X_S)\mathbb{1}(X_{\bar{S}}=x^b_{\bar{S}}), \qquad \text{and} \tag{33}$$

$$q^{marginal}(X)=\frac{1}{d}\sum_{S\in\mathcal{P}([d])\setminus[d]}\frac{1}{\binom{d}{|S|}}p(X_S)p(X_{\bar{S}}). \tag{34}$$

3.5.2 OOD Detection and OOD Classifier

To detect the OOD samples, Slack et al. (2020) proposed training a binary classifier $\texttt{ood\_score}(x)$ to predict whether a given sample x belongs to $p(X)$ or $q(X)$. Specifically, we first generate a M-size dataset $\mathcal{D}_q(X)$ from $q(X)$ and label it as 0. This dataset is then combined with the provided explaining set $\mathcal{D}_p(X)$ labeled as 1 to train the classifier. The classifier returns an OOD score, approximating the probability that the input x comes from $p(X)$. A hybrid sample (x_S, x'_S) is considered an OOD sample if $\texttt{ood\_score}(x_S, x'_S)$ is smaller than a selected threshold t.

Furthermore, let $C = \texttt{ood\_score}(X)$ denote the OOD score random variable, and let $p(C)$, $q(C)$ denote the distributions of C induced by $p(X)$, $q(X)$ respectively. If no distribution drift occurs, i.e., $q(X) = p(X)$, then we have $q(C) = p(C)$. Conversely, if $q(C) \neq p(C)$, then $q(X) \neq p(X)$, indicating a distribution drift. Thus, to detect the distribution drift, we propose comparing the distribution drift by examining the distributions of OOD scores C calculated on $\mathcal{D}_p(X)$ and $\mathcal{D}_q(X)$. One possible way to compare the two

distributions is to visualize their density histograms in a single plot (see Figure 3.3 and Figure 3.3 for an example). Another way is to quantify the distribution drift by calculating the *total variation distance* (Devroye, Györfi, & Lugosi, 1996):

$$D_{TV}[p(C), q(C)] = \frac{1}{2} \int_0^1 |p(C = c) - q(C = c)| dc. \tag{35}$$

The total variation distance can be conveniently estimated by half the absolute sum of density difference in all bins between the two density histograms.

3.6 Experiments

In this section, we conduct experiments to verify the error analyses we performed on existing SVA methods in previous sections. First, we demonstrate how to apply the method we proposed in Section 3.5.2 to detect and measure the distribution drifts caused by different distributional assumptions that have been used in the literature. Next, we will show that this distribution drift can lead to under-informative attributions, which assign significantly different important scores to highly correlated features. Finally, we demonstrate how data sparsity can cause over-informative attributions, which assign highly important scores to irrelevant or noisy features.

Dataset To assure the generalizability of our conclusions, we conduct our experiments on two datasets. Our first dataset is the Bike Sharing Dataset, which contains 17,389 records of hourly counts of bike rentals in 2011-2012 in the Capital Bike Sharing system (Fanaee-T, 2013). The dataset comprises a set of 11 features, following an unknown joint distribution. The objective is to predict the number of bikes rented during a specific hour of the day, based on various features related to time and weather conditions, such as hour, month, humidity, and temperature. The second dataset that we use is the Census Income (also

known as Adult) dataset, which contains information such as age, work class, education, etc. of 48,842 adults (Becker & Kohavi, 1996). The goal is to predict whether an adult's income exceeds 50,000 dollars. The dataset is extracted from the 1994 Census database. In each dataset, samples with missing data are removed.

For the Bike Sharing dataset, we aim to explain an xgBoost regressor trained on a training set of 15,379 samples and tested on a testing set of 2,000 samples. In addition, we split the Census Income dataset into a training set of 32,561 samples and a testing set of 4,000 samples. Our goal for the Census Income dataset is to explain an xgBoost classifier trained and tested on the respective sets.

3.6.1 Distribution Drift Detection

In this section, we will demonstrate how different distributional assumptions caused distribution drifts and estimate the corresponding OOD rates. Besides the training and test datasets described above, we generate four sets of hybrid samples by using four different removal distributions: uniform, product of marginal, marginal, and baseline. To make the results comparable, we calculate the OOD scores of the four hybrid sample sets using a *single* OOD classifier. Such an OOD classifier is trained using samples from the training set (labeled as 1) and hybrid samples generated from uniform removal distribution (labeled as 0). Note that this OOD classifier is still valid for OOD detection on hybrid samples generated from the other distributions because those samples are in-distribution of the uniform removal distribution.

The trained OOD classifier is then used to calculate OOD scores C for all real samples from both the training and testing sets, as well as for all hybrid samples in the four generated sets. We plot density histograms of these OOD scores in Figure 3.3 (for the Bike Sharing dataset) and Figure 3.4 (for the Census Income dataset). The total variance

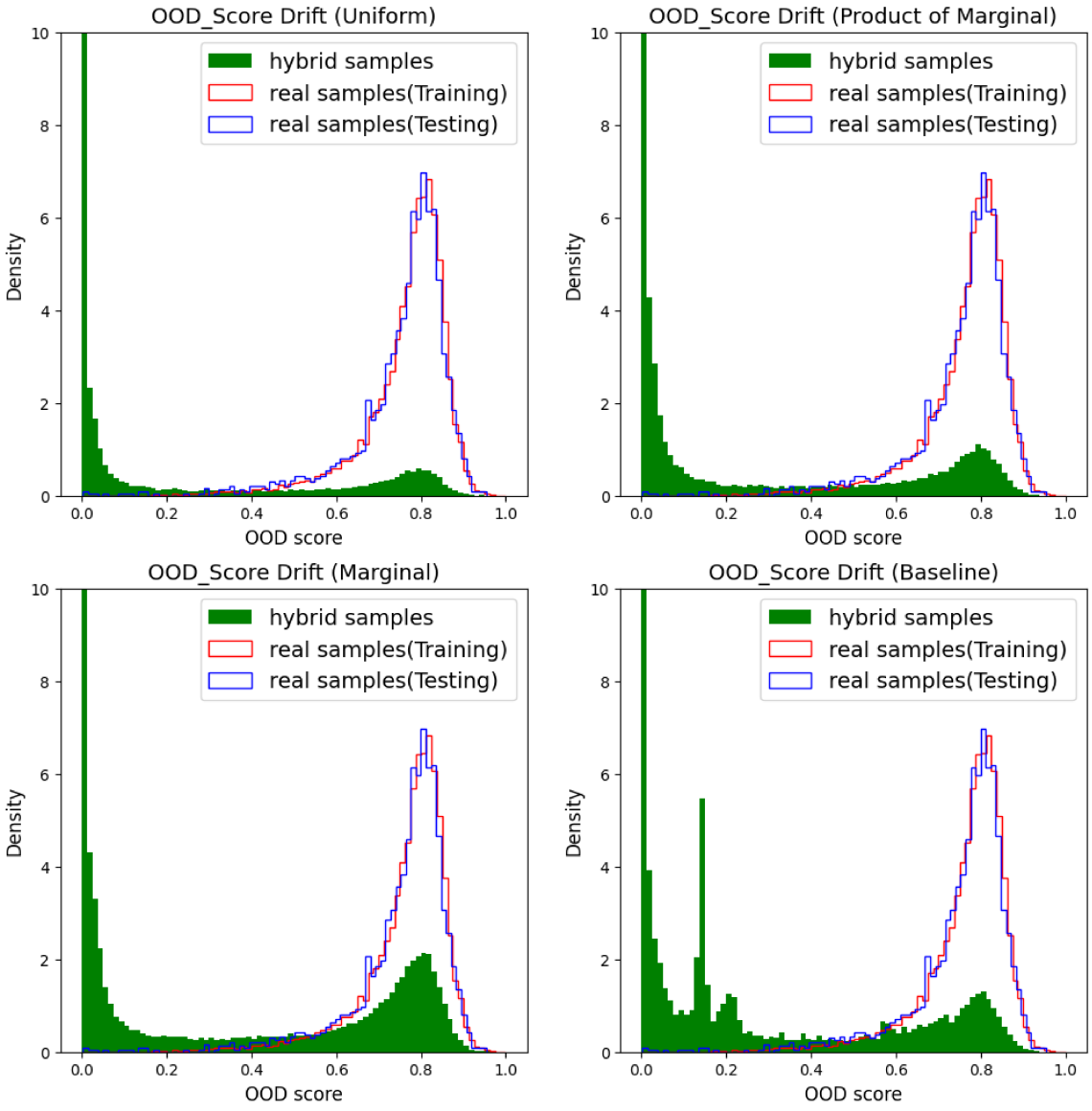


Figure 3.3: The density histograms of OOD scores on real samples and hybrid samples for the Bike Sharing Dataset.

distances between the OOD score distributions calculated from the training samples versus the generated hybrid samples are given in Table 3.2. First, we observe that the OOD density histograms of the training and test samples overlap, which implies that there is no distribution drift detected between the training and testing sets of both datasets. Second, we observe that all four removal distributions introduce noticeable distribution drifts, together with a considerable number of OOD samples. This is particularly evident for the uniform

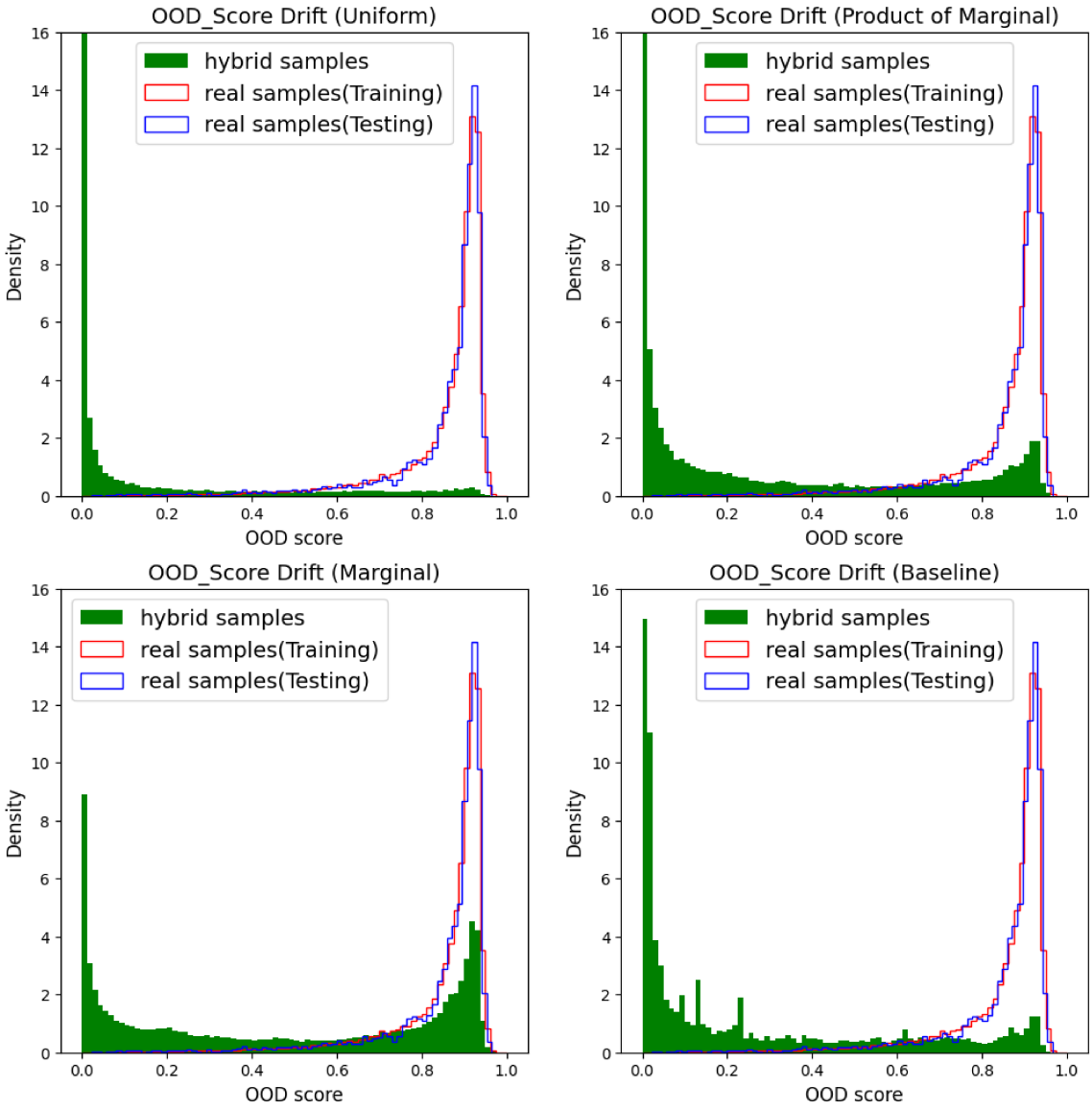


Figure 3.4: The density histograms of OOD scores on real samples and hybrid samples for the Census Income Dataset.

and product of marginal removal distributions, where the OOD rates are exceptionally high when adopting a threshold of 0.3 (0.866 and 0.757 for the Bike Sharing dataset, and 0.901 and 0.69 for the Census Income dataset, respectively). In contrast, the marginal removal distribution seems to exhibit the least distribution drift, ($D_{TV} = 0.578$ in the Bike Sharing dataset and $D_{TV} = 0.524$ in the Census Income dataset, respectively). Finally, the fact that the total variance distances are all greater than 50% for all removal distributions in both

Table 3.2: The OOD rates and total variance distance

Removal distribution	**OOD rate (t=0.3)**	**Total Variance Distance**
Bike Sharing Dataset		
Uniform	0.866	0.868
Product of Marginal	0.757	0.77
Marginal	0.538	0.578
Baseline	0.666	0.696
Census Income Dataset		
Uniform	0.901	0.903
Product of Marginal	0.69	0.729
Marginal	0.448	0.524
Baseline	0.756	0.804

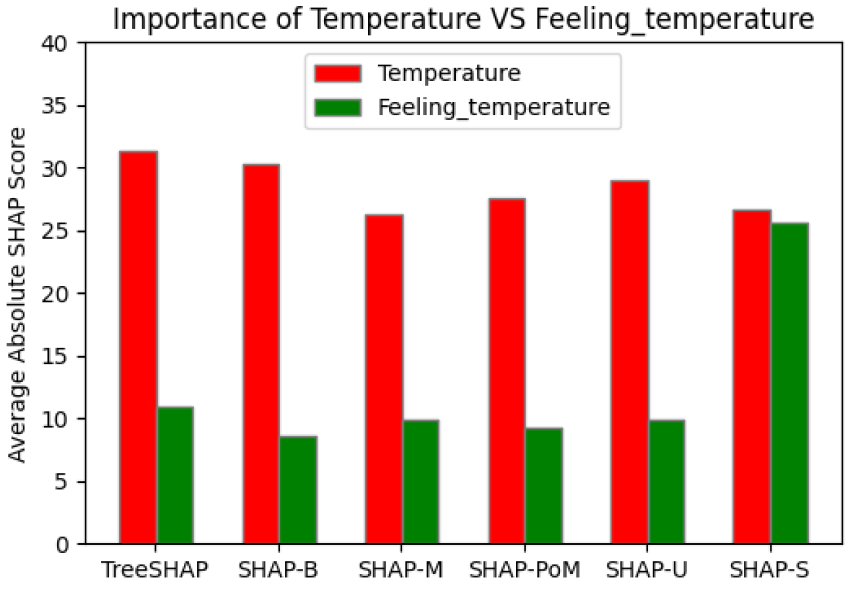
datasets highlights the severity of the distribution drifts.

3.6.2 Under-informativeness Audit

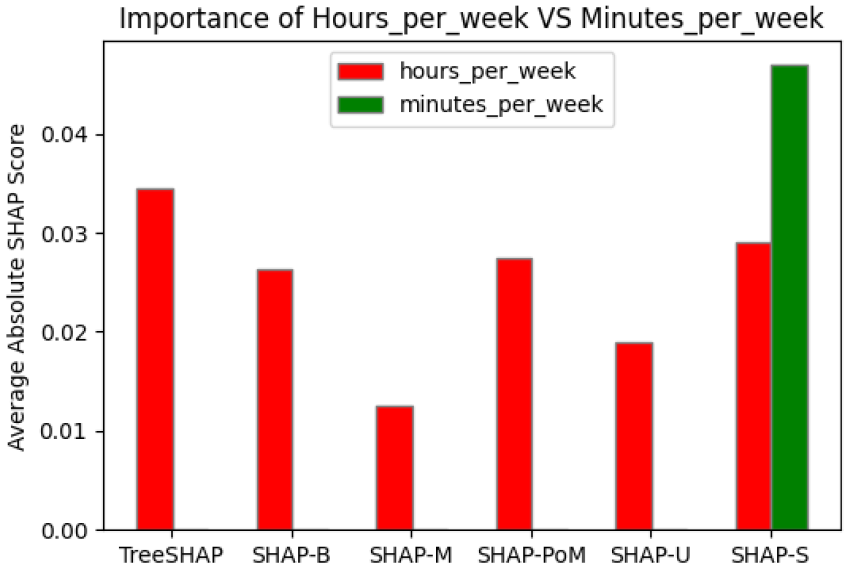
In Section 3.6.1, we showed that assumption-based methods caused severe distribution drifts. In this section, we will demonstrate that these distribution drifts can contribute to under-informative attributions.

For both datasets, we explain model predictions on 100 samples using SVAs calculated from five different RFs, namely SHAP-B (with baseline RF), SHAP-M (with marginal RF), SHAP-PoM (with product of marginal RF), SHAP-U (with uniform RF) and SHAP-S (with surrogate model-estimated CRF). In addition, TreeSHAP is also used to explain the predictions of xgBoost models on each dataset.

Intuitively, an informative feature attribution should (1) assign similar attribution scores to the two highly correlated features "`Temperature`" and "`Feeling_Temperature`" with Pearson correlation of 0.99 for the Bike Sharing dataset as they convey almost the same information; (2) assign exactly the same attribution score to features "`Hours_per_week`"



(a) Bike Sharing Dataset



(b) Census Income Dataset

Figure 3.5: Under-informativeness Audit on 100 predictions. (a) the average absolute SHAP scores of features "`Temperature`" and "`Feeling_Temperature`" (ideally, they should receive similar scores); (b) the average absolute SHAP scores of features "`Hours_per_week`" and "`Minutes_per_week`" (ideally, they should receive exactly the same score).

and "Minutes_per_week" for the Census Income Dataset because they hold the same information but in different scales.

From Figure 3.5a, we can observe that TreeSHAP, SHAP-B, SHAP-M, SHAP-PoM, and SHAP-U all assign much higher importance scores to feature "Temperature" than "Feeling_Temperature". Moreover, in Figure 3.5b, TreeSHAP, SHAP-B, SHAP-M, SHAP-PoM, and SHAP-U only assign importance to feature "Hours_per_week" and ignore feature "Minutes_per_week". This is because these methods do not consider the dependencies among features, leading to under-informative attributions. In contrast, SHAP-S trains a surrogate model to learn feature correlations, thus able to allocate similar importance scores to "Temperature" and "Feeling_Temperature". For the Census Income dataset, even though SHAP-S mitigates the problem of under-informativeness by assigning importance to both "Hours_per_week" and "Minutes_per_week", however, these scores are not the same. This indicates that the SHAP-S still produces structural bias and does not completely resolve the under-informativeness problem for the Census Income dataset.

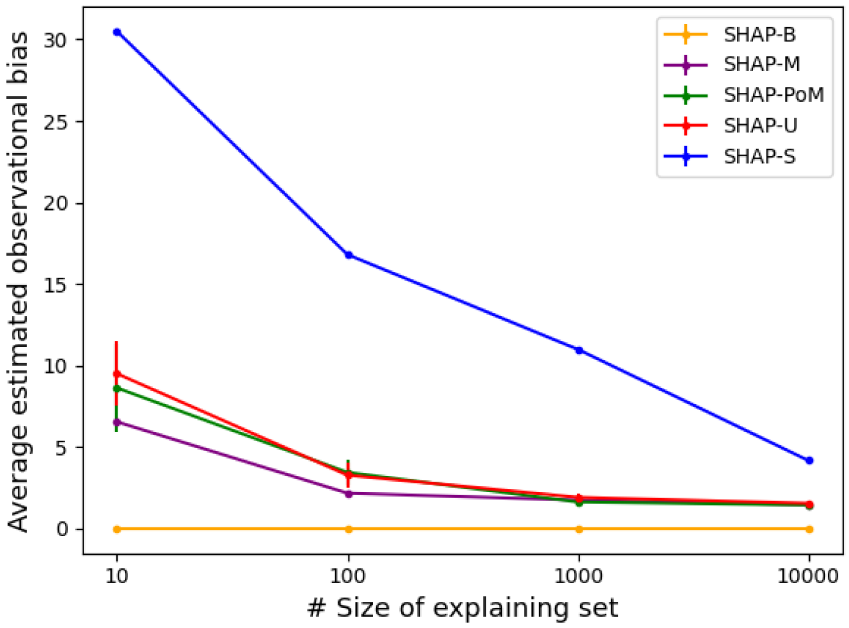
3.6.3 Over-informativeness Audit

In this section, we turn our attention to over-informativeness and observation bias. Recall that, the observation bias in Equation (27) is $\phi(v_{\hat{f}_S^{(N)}}) - \phi(v_{\hat{f}_S})$ where $\hat{f}_S = \lim_{N\to\infty} \hat{f}_S^{(N)}$. However, since we do not have an infinite explaining set, we cannot evaluate the observational bias directly. In this experiment, we estimate $\hat{f}_S$ by $\hat{f}_S^{(M)}$, where $\hat{f}_S^{(M)}$ is estimated using the whole training sets of both datasets. That is, $M = 15,379$ for the Bike Sharing dataset and $M = 32,561$ for the Census Income dataset. For random explaining sets with $N \in \{10, 100, 1000, 10000\}$, we estimate the average absolute observation bias in the SVAs of 100 predictions, namely

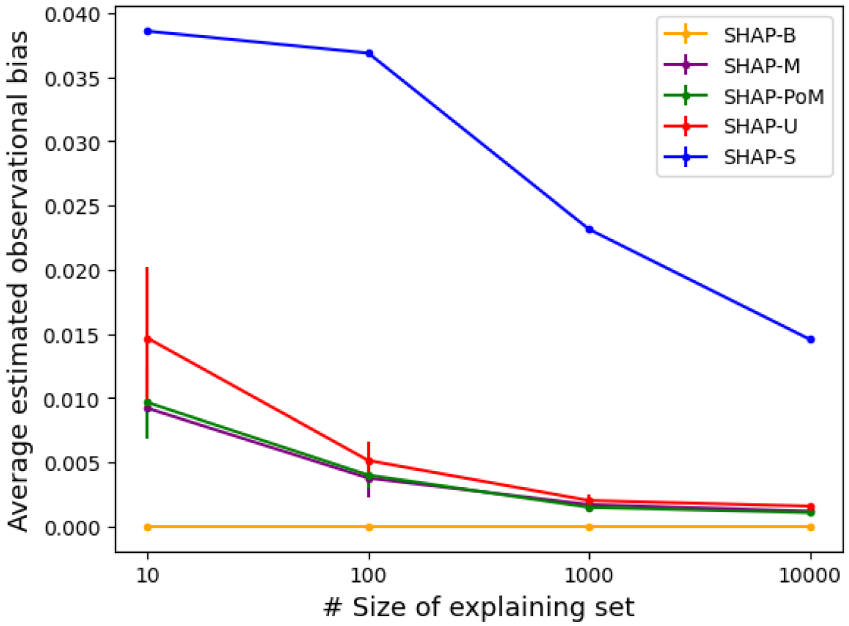
$$\frac{1}{100}\frac{1}{d}\sum_{i=1}^{10}\sum_{j=1}^{d} |\phi_{ij}(v_{\hat{f}_S^{(N)}}) - \phi_{ij}(v_{\hat{f}_S^{(M)}})|,$$

where ϕ_{ij} is the SVA of the jth feature in the ith prediction. The results are plotted in Figure 3.6. We observe similar trends in both datasets. Generally, observation bias decreases when the size of the explaining set increases. This illustrates the relationship between observation bias and data sparsity. However, different methods exhibit different sensitivity to data sparsity. Specifically, SHAP-B always has 0 observation bias, which agrees with our analysis in Section 3.4.6. For SHAP-M, SHAP-PoM, and SHAP-U, observation bias quickly stabilizes at $N = 1,000$. In contrast, SHAP-S shows high sensitivity to data sparsity, especially for the Census Income Data, at $N = 10,000$, the observation bias of SHAP-S is still much higher than those of other methods. Note that both datasets that we use contain less than 20 features. If the data is high-dimensional, SHAP-S will be more impacted by data sparsity, producing higher observation bias.

As discussed in Section 3.4.5, even if the surrogate model has an overall good fit on a large explaining set, SHAP-S can still be over-informative on low-density regions where data sparsity persists. To verify this remark, we generate a noisy feature from a mixed Gaussian distribution: $Z \sim \mathcal{N}(0,1)$ with probability 0.999 and $Z \sim \mathcal{N}(10,1)$ otherwise. For each dataset, we train a surrogate model on the whole training set with this noisy feature added. Even when the explaining set is large, the values from $\mathcal{N}(10,1)$ are still sparse, so the surrogate model is easy to overfit at points with $Z \sim \mathcal{N}(10,1)$. To see this, we use the SHAP-S feature attribution that utilizes the trained surrogate model to explain 100 predictions where $Z \sim \mathcal{N}(0,1)$ versus where $Z \sim \mathcal{N}(10,1)$. The feature attribution results are plotted in Figure 3.7. We can see that, in both datasets, even with a surrogate model trained on a large explaining set, SHAP-S still assigns high importance to noisy features if given predictions with $Z \sim \mathcal{N}(10,1)$. This noisy feature should be given 0 importance because it is sampled independently from all other features.

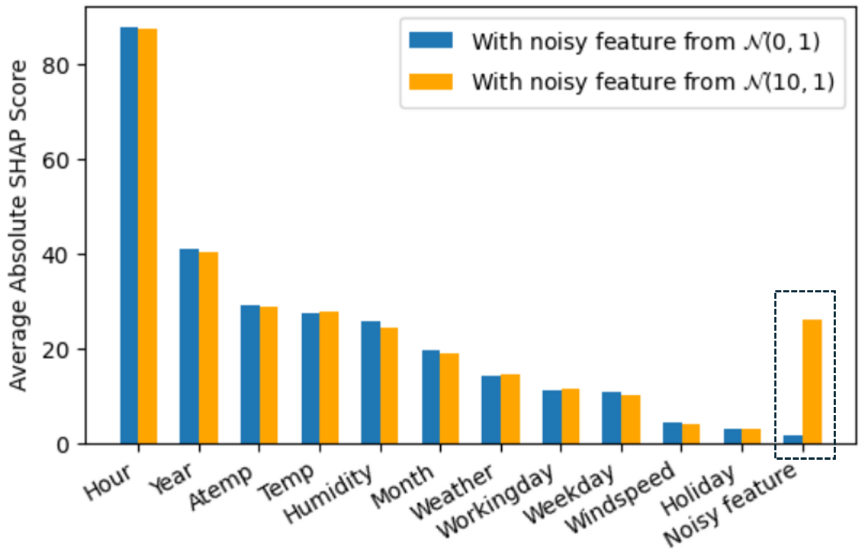


(a) Bike Sharing Dataset

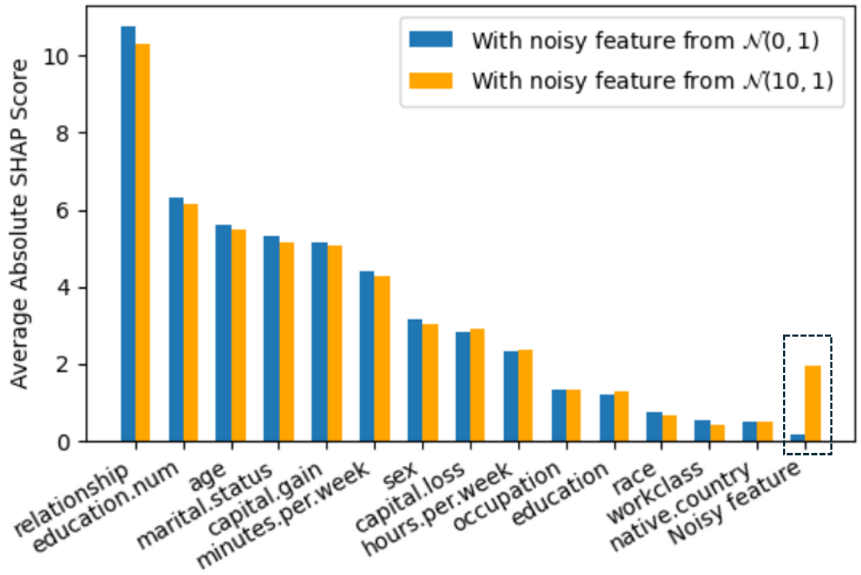


(b) Census Income Dataset

Figure 3.6: The change in average estimated observation bias of the SVAs as the size of the explaining set changes.



(a) Bike Sharing Dataset



(b) Census Income Dataset

Figure 3.7: Average absolute feature attributions given by SHAP-S on 100 predictions where the noisy feature comes from either $\mathcal{N}(0, 1)$ or $\mathcal{N}(10, 1)$.

3.7 Conclusions

In this chapter, we introduced a unified error analysis framework for informative SVAs. Our framework stems from the estimation and approximation errors arising from estimating the conditional removal function. These errors correspond to observation and structural

bias, which generate feature attributions that are respectively over- or under-informative. We apply our error analysis to discern potential errors in various existing SVA techniques. Carefully designed experimentation verifies our theoretical analysis. Future work can utilize our error analysis framework to develop new SVA methods that can effectively mitigate both under- and over-informativeness.

Chapter 4

Correcting Biases of SVAs for Informative Model Explanations

4.1 Introduction

Recent research has pointed out that existing SVA methods are error-prone and cannot capture the true informational structure (Chen et al., 2023; Sundararajan & Najmi, 2020). For example, Frye et al. (2020) demonstrated that SVAs based on the feature independence assumption can ignore model dependence on relevant features. They proposed a surrogate model to capture dependencies among features. However, Yeh et al. (2022) shows that this method can generate unreasonable explanations in low-density regions. In Chapter 3, we further analyzed that these explanation errors stem from a trade-off between two biases: observation bias due to data sparsity and structural bias due to untenable distributional assumptions. These biases lead to explanation errors, causing what we term over- and under-informative explanations.

In this chapter, we aim to provide solutions to reduce the observation and structural biases of existing SVA methods. Our key idea is to correct the distribution drift resulting from structural assumptions that are placed to reduce data complexity requirements. By

doing so, we can obtain a better trade-off between observation and structural biases. To realize this vision, we propose a novel combination of out-of-distribution (OOD) detection techniques and important sampling methods to refine two existing SVA methods, one based on the feature independence assumption (Lundberg & Lee, 2017; Sundararajan & Najmi, 2020), and the other based on a surrogate model (Frye et al., 2020). Our contributions can be summarized as follows.

- We propose in-distribution and OODIS refinement methods to remedy the distribution drift caused by the feature independence assumption.
- We propose an OODIS refinement method to reduce the sensitivity to data sparsity of the SVA method based on surrogate models.
- We provide a computational trick to calculate the importance sampling weights based on an OOD classifier without requiring complex density estimations.
- Our experiments verify that the proposed methods can greatly improve the informativeness of SVAs in both local and global explanation tasks.

4.2 Quick Reviews

We consider a supervised learning setting where a target variable $Y \in \mathcal{Y}$ is to be predicted based on an input variable $X = (X_1, \ldots, X_d)$ that consists of d features. Let p denote the underlying distribution, and $\mathcal{X}_p$ denote the domain set of X. We aim to explain an ML model $f : \mathcal{X}_p \to \mathcal{Y}$ by quantifying each feature's contribution using the SVA method in Equation (12). As discussed in Section 2.5.2, to design an SVA algorithm, we need to specify a removal function $f_S(x_S)$ which makes predictions based on a subset of input x_S. Furthermore, to enable the SVAs to capture the informational dependencies between model outputs and input features, $f_S(x_S)$ should be defined as the conditional removal function

(CRF) as:

$$
\begin{aligned}
f_S(x_S) &:= \mathbb{E}_{p(X_{\bar{S}}|X_S=x_S)}[f(x_S, X_{\bar{S}})] \\
&= \int f(x_S, x'_{\bar{S}}) p(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S) dx'_{\bar{S}}.
\end{aligned} \tag{36}
$$

CRF Estimation It is challenging to estimate the CRF $f_S(x_S)$ because the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ is typically unavailable in practice. Therefore, we need to estimate it using a given explaining set $\mathcal{D}_p(X) \equiv \{(x^{(n)})\}_{n=1}^N$. In practice, empirical estimates using the explaining set may be susceptible to data sparsity, particularly in the context of high-dimensional and many-valued features (Chen et al., 2023; Sundararajan & Najmi, 2020). Besides, estimating the CRF f_S for all possible subsets S is NP-hard (Aas et al., 2021). As discussed in Section 3.3, several methods have been proposed to approximate the CRF f_S, which either smooth the explaining set or make distributional assumptions. In this work, we focus on two popular approaches: the *surrogate model* and the *marginal removal function (MRF)*. In the following, we use $\hat{f}_S^{(N)}$ to denote the estimate of f_S using one of the two approaches on an explaining set of size N. Let $\hat{f}_S = \lim_{N\to\infty} \hat{f}_S^{(N)}$ denote the approximated CRF when we have access to an infinite amount of data.

- **(Surrogate Model)** We can train a neural network $h_\theta(x_S)$ as a surrogate for the CRF $f_S(x_S)$. This neural network is trained to mimic the original model's predictions, with removed features represented by zeros. The set of neural network parameters θ can be estimated by minimizing the empirical loss function $\mathbf{L}(\cdot)$:

$$
\hat{\theta} = \operatorname{argmin}_\theta \mathbb{E}_{x\sim\mathcal{D}_p(X)} \mathbb{E}_{S\sim Shapley} \mathbf{L}\left(h_\theta(x_S), f(x)\right). \tag{37}
$$

  In this case, we have $\hat{f}_S^{(N)} = h_{\hat{\theta}}$ and $\hat{f}_S = h_\theta$.

- **(MRF)** Alternatively, we can approximate the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ by making feature independence assumptions. Specifically, we can assume X_S

and $X_{\bar{S}}$ to be independent, then the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ can be approximated by the marginal distribution $p(X_{\bar{S}})$. With $p(X_{\bar{S}}|X_S = x_S) \approx p(X_{\bar{S}})$, the CRF f_S in formula (36) can be approximated as

$$\hat{f}_S(x_S) = \mathbb{E}_{p(X_{\bar{S}})}[f(x_S, X_{\bar{S}})] = \int f(x_S, x'_{\bar{S}}) p(X_{\bar{S}} = x'_{\bar{S}}) dx'_{\bar{S}}, \tag{38}$$

which can be empirically estimated by

$$\hat{f}_S^{(N)}(x_S) = \frac{1}{N} \sum_{n=1}^{N} f(x_S, x_{\bar{S}}^{(n)}) \tag{39}$$

using the explaining set $\mathcal{D}_p(X)$.

Distribution Drift The purpose of making the feature independence assumption is to reduce the data complexity requirement and thus mitigate data sparsity. However, in practice, the feature independence assumption is typically untenable, which could engender undesirable distribution drift. Specifically, the computation of both the CRF in Equation (36) and the MRF in Equation (38) involve the hybrid sample $(x_S, x'_{\bar{S}})$. In both cases, $x_S \sim p(X_S)$. When $x'_{\bar{S}} \sim p(X_{\bar{S}}|X_S = x_S)$, the hybrid sample $(x_S, x'_{\bar{S}})$ follows the true underlying distribution $p(X)$. However, as shown in Section 3.5.1, when $x'_{\bar{S}} \sim p(X_{\bar{S}}) \neq p(X_{\bar{S}}|X_S = x_S)$ as in the MRF method of Equation (38), the hybrid sample $(x_S, x'_{\bar{S}})$ follows a different distribution $q^{marginal}(X)$, which can be written as

$$q^{marginal}(X) = \frac{1}{d} \sum_{S \in \mathcal{P}([d]) \setminus [d]} \frac{1}{\binom{d}{|S|}} p(X_S) p(X_{\bar{S}}). \tag{40}$$

In other words, the distributional assumption made by the MRF method causes the data distribution to drift from $p(X)$ to $q^{marginal}(X)$. Such a distribution drift can cause hybrid samples $(x_S, x'_{\bar{S}}) \in q^{marginal}(X)$ involved in the CRF estimation to be **out-of-distribution**

(OOD) samples of $p(X)$, i.e., $(x_S, x'_{\bar{S}}) \notin \mathcal{X}_p$ or $p(x_S, x'_{\bar{S}}) = 0$. Especially in the presence of many high-correlated features, $q^{marginal}$ may have a very high **OOD rate** to $p(X)$, denoted as $R_p(q^{marginal}) = \mathbb{E}_{q^{marginal}(X)}[\mathbb{1}(X \notin \mathcal{X}_p)]$. A high OOD rate suggests a severe distribution drift, which could lead to considerable estimation error of f_S, and thus explanation error of the corresponding SVA.

Explaination Error Since we use $\hat{f}_S^{(N)}$ to estimate the true CRF f_S, the true SVA $\phi(v_{f_S})$ in Equation (12) is estimated by $\phi(v_{\hat{f}_S^{(N)}})$. However, most existing CRF estimation methods are error-prone, so the explanation error $\phi(v_{\hat{f}_S^{(N)}}) - \phi(v_{f_S})$ may occur due to the difference $\hat{f}_S^{(N)} - f_S$. As discussion in Chapter 3, the error $\hat{f}_S^{(N)} - f_S$ can be decomposed into two components: the *estimation error*, $\hat{f}_S^{(N)} - \hat{f}_S$, and the *approximation error*, $\hat{f}_S - f_S$. Correspondingly, the explanation error can also be decomposed into two components: the ***observation bias***, $\phi(v_{\hat{f}^{(N)}}) - \phi(v_{\hat{f}})$, which is associated with the estimation error $\hat{f}_S^{(N)} - \hat{f}_S$, arising from the utilization of insufficient observations (i.e., data sparsity); the ***structural bias***, $\phi(v_{\hat{f}}) - \phi(v_f)$, which is associated with the approximation error $\hat{f}_S - f_S$, stemming from the distribution drift caused by structural assumptions.

Both observation bias and structural bias can reduce the informativeness of SVAs. To describe SVAs suffering from observation bias and/or structural bias, we employ the terminologies defined in the following:

Definition 3 (**Over-informativeness**). *A SVA is over-informative if it has a large absolute value of observation bias* $|\phi(v_{\hat{f}_S^{(N)}}) - \phi(v_{\hat{f}_S})|$.

Definition 4 (**Under-informativeness**). *A SVA is under-informative if it has a large absolute value of structural bias* $|\phi(v_{\hat{f}_S}) - \phi(v_{f_S})|$.

In essence, the informativeness of SVAs is determined by the accuracy of the selected estimate $\hat{f}_S^{(N)}$. Hence, in this work, we aim to enhance the informativeness of SVAs by improving the accuracy of the surrogate model and MRF methods.

4.3 Proposed Methods

4.3.1 In-Distribution Refinement

The MRF method is widely used in practice to calculate SVA (I. Covert et al., 2020; Lundberg & Lee, 2017) because of its simple implementation. In particular, the feature independence assumption helps reduce the data complexity requirement and thus the observation bias (as discussed in Chapter 3). On the other hand, the feature independence assumption is typically untenable, leading to considerable structural bias and under-informative SVAs. In essence, the MRF $\hat{f}_S$ in Equation (38) is a rough approximation of the CRF f_S in Equation (36), and the approximation error can be derived as

$$\begin{aligned}
&\hat{f}_S(x_S) - f_S(x_S) \\
=&\int_{(x_S, x'_{\bar{S}}) \notin \mathcal{X}_p} f(x_S, x'_{\bar{S}}) p(X_{\bar{S}} = x'_{\bar{S}}) dx'_{\bar{S}} \quad + \\
&\int_{(x_S, x'_{\bar{S}}) \in \mathcal{X}_p} f(x_S, x'_{\bar{S}}) \left[p(X_{\bar{S}} = x'_{\bar{S}}) - p(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S) \right] dx'_{\bar{S}}.
\end{aligned} \tag{41}$$

This equation shows that the approximation error of the MRF method stems from two sources: (i) the inclusion of OOD samples in the approximation; and (ii) the changes in the probability density of in-distribution samples. The first part, which is caused by OOD samples, may contribute to a large proportion of the approximation error. Therefore, the first refinement method we propose to reduce the approximation error is to remove OOD samples from the MRF computation. With OOD samples removed, the MRF can be modified as

$$\hat{f}_S(x_S) = \mathbb{E}_{p(X_{\bar{S}})} \left[f(x_S, X_{\bar{S}}) \mathbb{1} \left((x_S, X_{\bar{S}}) \in \mathcal{X}_p \right) \right]. \tag{42}$$

Here, we call the approximated removal function in Equation (42) the *in-distribution MRF*.

To detect the OOD samples, as proposed by Slack et al. (2020), we can train an OOD classifier $\texttt{ood\_score}(x)$ to predict whether a given sample x comes from $p(X)$ or from

$q^{marginal}(X)$ in Equation (40). When $\texttt{ood\_score}(x)$ is high, x is more likely to come from $p(X)$ rather than $q^{marginal}(x)$. A hybrid sample $(x_S, x'_{\bar{S}})$ is considered an OOD sample if $\texttt{ood\_score}(x_S, x'_{\bar{S}})$ is smaller than a selected threshold t. The in-distribution MRF can then be empirically estimated by the in-distribution hybrid samples:

$$\hat{f}_S^{(N)}(x_S) = \frac{1}{N}\sum_{n=1}^{N} f(x_S, x_{\bar{S}}^{(n)}) \mathbb{1}\left(\texttt{ood\_score}(x_S, x_{\bar{S}}^{(n)}) \geq t\right). \tag{43}$$

The selection of threshold t There is a trade-off when selecting the threshold t. As t increases, on the one hand, more and more true OOD samples are screened out from the estimation, helping reduce the structural bias of the corresponding SVA; on the other hand, the sample size for the estimation is reduced (e.g., some true in-distribution samples could be misclassified as OOD samples and removed) and thus the observation bias may increase.

4.3.2 OOD Importance Sampling (OODIS) Refinement

The most effective way to mitigate over-informativeness is to increase the sample size. Nevertheless, it is infeasible to sample directly from an unknown distribution $p(X_{\bar{S}}|X_S)$. To address this problem, we can sample from an assumed distribution $q(X_{\bar{S}}|X_S)$. However, as discussed in Section 4.2, distributional assumptions may induce distribution drift leading to structural bias and under-informativeness. We thus propose using an importance sampling method, which can be derived from the CRF formula in Equation (36):

$$\begin{aligned}
f_S(x_S) = & \int f(x_S, x'_{\bar{S}}) p(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S) dx'_{\bar{S}} \\
= & \int f(x_S, x'_{\bar{S}}) \frac{p(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S)}{q(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S)} q(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S) dx'_{\bar{S}} \\
= & \frac{q(X_S = x_S)}{p(X_S = x_S)} \int \frac{p\left(X = (x_S, x'_{\bar{S}})\right)}{q\left(X = (x_S, x'_{\bar{S}})\right)} f(x_S, x'_{\bar{S}}) q(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S) dx'_{\bar{S}} \\
= & \frac{1}{w_S(x_S)} \mathbb{E}_{x'_{\bar{S}} \sim q(X_{\bar{S}} | X_S = x_S)} \left[w(x_S, x'_{\bar{S}}) f(x_S, x'_{\bar{S}}) \right],
\end{aligned} \tag{44}$$

where

$$w(x) = \frac{p\left(X = x\right)}{q\left(X = x\right)}, \tag{45}$$

and

$$\begin{aligned} w_S(x_S) &= \frac{p(X_S = x_S)}{q(X_S = x_S)} \\ &= \int \frac{p\left(X = (x_S, x'_{\bar{S}})\right)}{q(X_S = x_S)} dx'_{\bar{S}} \\ &= \int \frac{p\left(X = (x_S, x'_{\bar{S}})\right)}{q\left(X = (x_S, x'_{\bar{S}})\right)} q(X_{\bar{S}} = x'_{\bar{S}} | X_S = x_S) dx'_{\bar{S}} \\ &= \mathbb{E}_{x'_{\bar{S}} \sim q(X_{\bar{S}} | X_S = x_S)} \left[w(x_S, x'_{\bar{S}})\right]. \end{aligned} \tag{46}$$

In Equation (44), $w(x_S, x'_{\bar{S}})$ acts as the *importance weight* of a hybrid sample $(x_S, x'_{\bar{S}})$, and $w_S(x_S)$ acts as the *normalizing constant*.

The selection of $q(X)$ An appropriate assumed distribution $q(X)$ should fulfill three essential criteria: (1) It should facilitate easy sampling from the conditional distribution $q(X_{\bar{S}}|X_S = x_S)$; (2) For any subset S and any $x \in \mathcal{X}_p$, the probability density $q(X_{\bar{S}} = x_{\bar{S}}|X_S = x_S) > 0$, ensuring that all plausible in-distribution instances have positive probabilities of being sampled; and (3) the OOD rate $R_p(q^{marginal})$ should not be too high to ensure sufficient in-distribution instances being sampled. Based on these three criteria, we suggest choosing $q^{marginal}(X)$ for the importance sampling.

OODIS MRF When selecting $q(X) = q^{marginal}(X)$ or $q(X_{\bar{S}}|X_S = x_S) = p(X_{\bar{S}})$, we call the removal function in Equation (44) the *OODIS MRF*, which is exactly equal to the CRF f_S, indicating a zero approximation error. The OODIS MRF can be empirically

estimated from the explaining set $\mathcal{D}_p(X)$ by:

$$\hat{f}_S^{(N)}(x_S) = \frac{\sum_{n=1}^{N} w(x_S, x_{\bar{S}}^{(n)}) f(x_S, x_{\bar{S}}^{(n)})}{\sum_{n=1}^{N} w(x_S, x_{\bar{S}}^{(n)})}. \tag{47}$$

Essentially, this is a modified version of the MRF estimator in Equation (39). The distribution drift induced by the feature independence assumption is rectified by the importance sampling technique. However, this correction comes at a cost: when the OOD rate $R_{q^{marginal}}(p)$ is high, the data sparsity problem that was well mitigated by the feature independence assumption could again be magnified to some extent. Specifically, many hybrid samples $(x_S, x_{\bar{S}}^{(n)})$ could be screened out due to being identified as OOD samples and receiving trivial or zero importance weights. This may lead to an increase in the observation bias of the corresponding SVA.

OODIS Surrogate Model The OODIS method not only can address the distribution drift problem of the MRF, but it can also moderate the data sparsity problem of the surrogate model $h_\theta(x_S)$. In particular, rather than the explaining set $\mathcal{D}_p(X)$, we train $h_\theta(x_S)$ on a larger generated dataset $\mathcal{D}_q(X)$ (e.g., $\mathcal{D}_{q^{marginal}}(X)$), where each sample x weighted by $w(x)$. In other words, instead of the loss function in Equation (37), the parameter set θ can be estimated by minimizing the following weighted loss function:

$$\hat{\theta} = \text{argmin}_\theta \, \mathbb{E}_{x \sim \mathcal{D}_q(X)} \mathbb{E}_{S \sim Shapley} w(x) \mathbf{L}\left(h_\theta(x_S), f(x)\right). \tag{48}$$

We call this the *OODIS surrogate model*. This proposed method can alleviate the sparsity problem of the surrogate model by increasing the in-distribution sample size. With a sufficiently large sample set $\mathcal{D}_q(X)$, not only the original observed samples in the explaining set, i.e., $x \in \mathcal{D}_p(X)$, but also many other non-observed in-distribution samples (i.e., $x \in \mathcal{X}_p \setminus \mathcal{D}_p(X)$) can be generated and used in the estimation of $h_\theta(x_S)$. Meanwhile, OOD

samples are discounted via the importance weight $w(x)$.

The estimation of $w(x)$

Directly estimating $w(x)$ using Equation (45) entails complex density estimation of $p(X)$ and $q(X)$. However, in the following, we demonstrate how we approximate $w(x)$ using an OOD classifier without the need to estimate any density functions. Specifically, given an OOD classifier $\texttt{ood\_score}(x)$ trained on a combination of an M-size dataset $\mathcal{D}_q(X)$ and an N-size explaining set $\mathcal{D}_p(X)$, according to Bayes' theorem, we can write

$$\begin{aligned}\texttt{ood\_score}(x) &\approx \mathbf{Pr}\{X \in \mathcal{X}_p | X = x\} \\ &= \frac{p(X=x)\mathbf{Pr}\{X \in \mathcal{X}_p\}}{p(X=x)\mathbf{Pr}\{X \in \mathcal{X}_p\} + q(X=x)\mathbf{Pr}\{X \in \mathcal{X}_q\}}.\end{aligned}$$

Then $w(x)$ can be approximated as

$$\begin{aligned}w(x) = \frac{p\left(X=x\right)}{q\left(X=x\right)} &= \frac{\mathbf{Pr}\{X \in \mathcal{X}_q\}}{\mathbf{Pr}\{X \in \mathcal{X}_p\}} \cdot \frac{\mathbf{Pr}\{X \in \mathcal{X}_p | X = x\}}{1 - \mathbf{Pr}\{X \in \mathcal{X}_p | X = x\}} \\ &\approx \frac{M}{N} \cdot \frac{\texttt{ood\_score}(x)}{1 - \texttt{ood\_score}(x)} \\ &\propto \frac{\texttt{ood\_score}(x)}{1 - \texttt{ood\_score}(x)}.\end{aligned} \tag{49}$$

Equation (49) offers a computationally efficient means of estimating $w(x)$ without explicit density estimations.

The approximation of $w(x)$ in Equation (49) is solely determined by the OOD classifier $\texttt{ood\_score}(x)$. Therefore, the correctness of the trained $\texttt{ood\_score}(x)$ directly impacts the estimation error and approximation error of both the OODIS MRF and the OODIS surrogate model, thus the observation bias and structural bias of the corresponding SVAs. A highly smooth $\texttt{ood\_score}(x)$ tends to classify most samples in $\mathcal{D}_q(X)$ as in-distribution samples, potentially including some true OOD samples. In this case, the observation

bias decreases while the structural bias increases. Conversely, an excessively overfitted $\texttt{ood\_score}(x)$ might classify most non-observed samples (i.e., $x \notin \mathcal{D}_p(X)$) as OOD samples, even though some of them are indeed in-distribution. In this case, the data sparsity mitigation of the OODIS method is minimal, and the OODIS MRF and the OODIS surrogate model are not very different from the empirical CRF estimator and the ordinary surrogate model.

4.4 Experiments

In this section, we conduct two sets of experiments to demonstrate that our proposed in-distribution and OODIS refinement methods can produce more informative SVAs. The first experiment focuses on local feature attributions that explain model predictions, while the second focuses on global feature attributions that explain model performance on the entire dataset.

4.4.1 Informative Local Explanations of Model Predictions

Experiment settings: Our first set of experiments is conducted on the Bike Sharing Dataset Fanaee-T (2013) with 17,389 records of hourly counts of bike rentals in 2011-2012 in the Capital Bike Sharing system. There are 11 features following an unknown joint distribution $p(X)$. The explanation task of interest is to assign feature attributions for the predictions made by an xgBoost regression model trained on a training set of 15,379 samples and tested on a test set of 2,000 samples. We compare the performance of our refinement methods applied on the MRF (SHAP-M) and surrogate model (SHAP-S) to popular existing SVA methods, namely TreeSHAP (Lundberg et al., 2020, 2018) and baseline SHAP[1] (SHAP-B) (Sundararajan & Najmi, 2020).

[1]SHAP-B has the removal function $\hat{f}_S(x_S) = f(x_S, x^b_{\bar{S}})$ for a baseline value x^b. Specifically, in our experiments, we use the data mean as the baseline.

Under-informativeness Correction

First, we demonstrate that our proposed refinement methods can effectively correct the distribution drift caused by the feature independence assumption of the MRF method, and thus address the corresponding under-informativeness.

Distribution drift correction: To detect the distribution drift, we use the methods proposed in Section 3.5. Specifically, we train an OOD classifier $\texttt{ood\_score}(x)$ (xgBoost) on a dataset that contains samples observed from the original dataset (labeled as 1) and hybrid samples from the assumed distribution $q^{marginal}(X)$ (labeled as 0). Based on this OOD classifier, we apply the in-distribution and OODIS refinement methods on the above hybrid samples (generated from $q^{marginal}(X)$) to generate two other sets of hybrid samples: one by removing the estimated OOD samples (with $t = 0.3$); another by resampling according to the importance weight $w(x)$. The distribution of $\texttt{ood\_score}(x)$ for samples x from each of the four data sets are compared using density histograms in Figure 4.1. If those data sets are produced from the same distribution (i.e., no distribution drift), the corresponding density histograms of $\texttt{ood\_score}(x)$ should be identical. We can see that the histogram for hybrid samples generated from $q^{marginal}(X)$ (shaded green) greatly deviates from the histogram for samples from the original dataset (red line). In contrast, hybrid samples generated using our refinement methods closely resemble the original samples, with the OODIS refinement method (blue line) performing slightly better than the in-distribution counterpart (yellow line). Furthermore, the *total variation distance (TVD)* (in Equation 35) between the histograms generated by the sample set from $q^{marginal}(X)$ and the original dataset is as high as 0.633, implying a significant distribution drift. The proposed in-distribution and OODIS refinement methods greatly reduce this value to 0.196 and 0.082, respectively. This suggests that the in-distribution refinement can drastically mitigate the distribution drift, and the OODIS refinement can even almost perfectly eliminate the distribution drift.

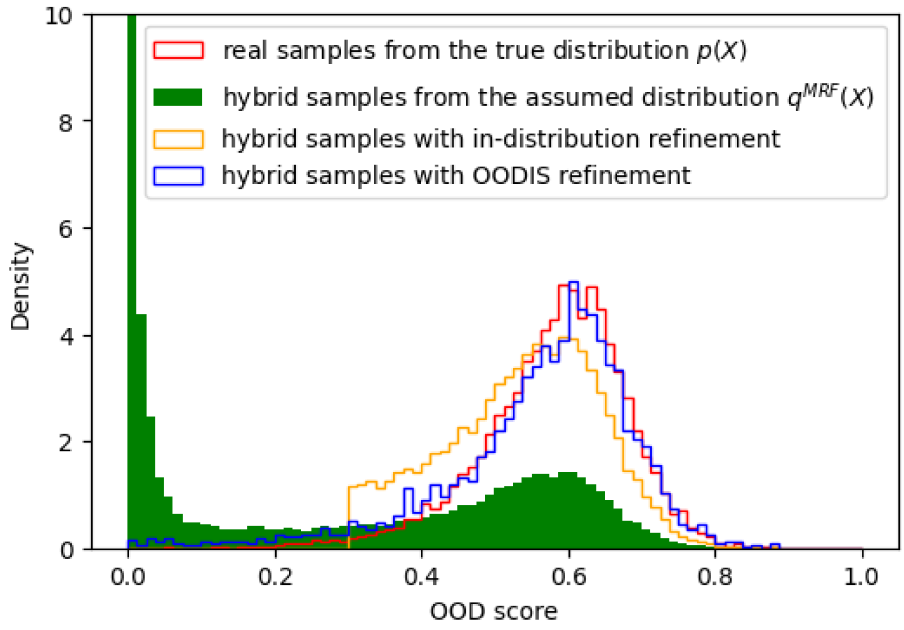


Figure 4.1: The density histograms of `ood_score`(x) on real samples and hybrid samples.

Defense against adversarial attack: Slack et al. (2020) demonstrate that SVA methods with simplified assumptions on feature dependency can be sensitive to adversarial attacks. Specifically, these methods can produce significantly different SVAs for two similarly performing ML learning models, one trained on the original training dataset, and another trained on the perturbed off-manifold region of the training set. To demonstrate that our refinement methods can increase the robustness against such an adversarial attack, besides the original model f, we train an adversarial model f' in the following way. First, we train an OOD classifier based on the training set. Subsequently, approximately 4000 OOD samples with an OOD score lower than a threshold of 0.001 are randomly drawn from the corresponding marginal distribution $q^{marginal}(X)$. For these OOD samples, the values of the target variable "`number_of_bike_rented`" are set as 800 if the feature "`Hour`" falls within the range [0,5]; otherwise, they are set as 0. This is an adversarial attack because it intentionally creates a false high positive correlation between night-time "`Hour`" and "`number_of_bike_rented`". The adversarial model f' is trained on the combined dataset

of the original training set and the generated OOD samples. The resulting model f' has almost the same performance as the model f trained only on the original training set, with the MSE loss on the test set being 1797.45 and 1752.52, and the corresponding R-squared (R^2) being 0.9433 and 0.9448, respectively. Moreover, the Pearson correlation of the two models' predictions on the test set is 0.986, and the Spearman correlation is 0.980. This indicates that the adversarial model makes similar predictions to the original model on the in-distribution samples of the test set. Desirably, an SVA method using the test set as the explaining set should produce similar SHAP scores to the feature "`Hour`" at night time. However, as shown in Figure 4.2, TreeSHAP, SHAP-B, and SHAP-M fail to circumvent

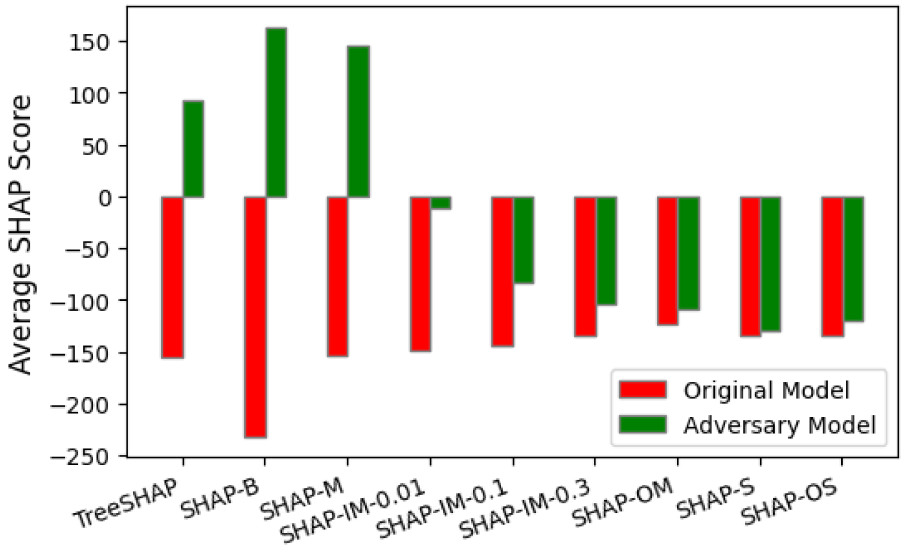


Figure 4.2: The average SHAP score of feature "Hour" over night-time samples in the explaining (test) set.

the adversarial attack, as they assign positive average SHAP scores to night-time `Hour` for f'. This is unreasonable because the number of bike rentals should substantially decrease at night time. On the other hand, our proposed OODIS MRF (SHAP-OM) and OODIS surrogate model (SHAP-OS) methods successfully produce comparable performance to the SHAP-S method, which is known to be robust against adversarial attacks (Frye et al.,

2020; Lin et al., 2024). We also observe that the in-distribution MRF method (SHAP-IM) can correct the positive SVA of the adversarial model to negative. The performance improves as the threshold t increases from 0.01 to 0.3.

Attributions for similar informative features: Next, let us consider the two features "`Temperature_C`" and "`Temperature_F`", which contains exactly the same information, temperature, but in different units, degree Celcius and degree Fahrenheit, respectively. Therefore, an informative model explanation should assign identical importance scores to these two features. However, as shown in Figure 4.3, TreeSHAP, SHAP-B, and SHAP-M

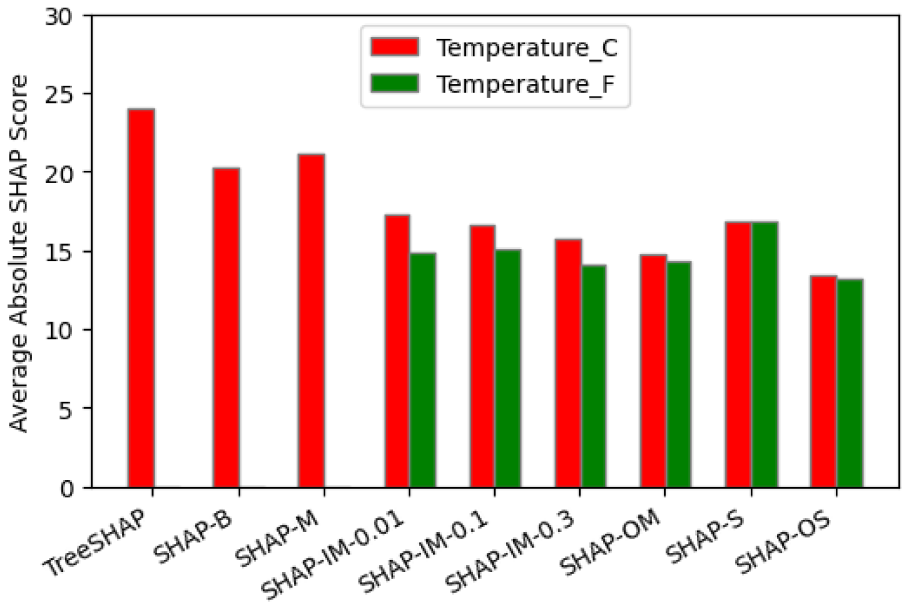


Figure 4.3: The average absolute SHAP scores of features "`Temperature_C`" and "`Temperature_F`" (ideally, they should receive the identical scores).

only assign importance to "`Temperature_C`" and assign zero importance to "`Temperature_F`". This is because these methods ignore/simplify feature dependency, resulting in under-informative explanations. On the contrary, all of our refinement methods, including the in-distribution refinement with a very small threshold $t = 0.01$, successfully correct such a structural bias from SHAP-M by assigning similar importance scores to the two features.

Over-informative Correction

In Section 4.4.1, SHAP-S performs well on all under-informative experiments. However, SHAP-S is not a perfect SVA method as it can be sensitive in low-density regions. To demonstrate this, we add into the explaining set a noisy feature Z generated from $\Gamma(5, 40)$ distribution. The feature is generated independently of both the target variable and the other features, therefore, it is not informative for the target variable and should receive zero importance. Figure 4.4 shows the average absolute feature attributions assigned to Z when

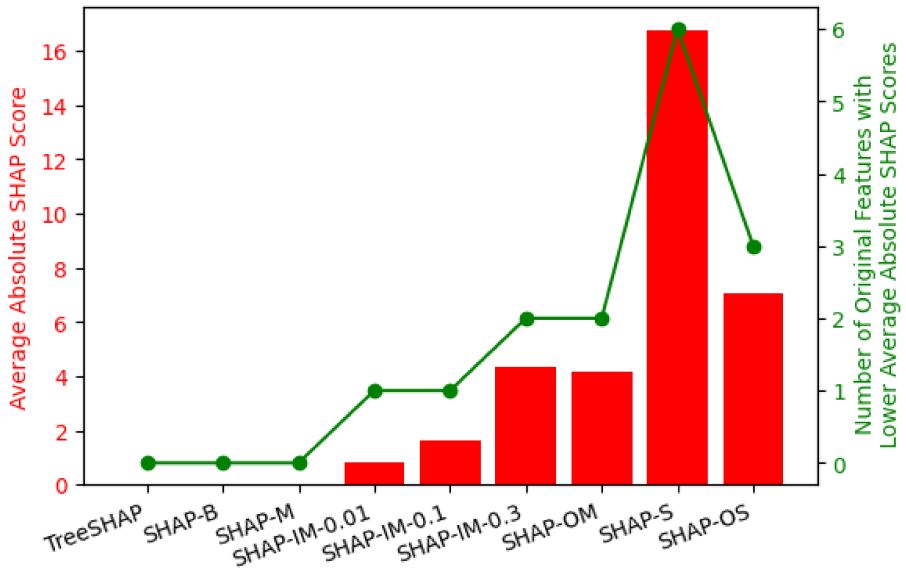


Figure 4.4: Average absolute feature attribution given for the noisy feature Z on low-density samples with $Z > 300$.

we explain model predictions in a low-density region where $Z > 300$. While TreeSHAP, SHAP-B, and SHAP-M correctly give zero importance to Z, SHAP-S incorrectly assigns high SHAP scores to Z at an average of 16.766, which is higher than 6 of the original 11 features in the Bike Sharing dataset. Our OODIS refinement method greatly reduces this sensitivity for SHAP-S, returning an average absolute SHAP score of 7.057, which is higher than 3 of the original 11 features. Although the proposed in-distribution and OODIS MRF methods do not return the perfect zero attribution, they still significantly outperform

SHAP-S.

The reason why SVA methods assign nonzero importance scores to noisy features on low-density regions is that they have high observation bias. Recall that observation bias is caused by data sparsity. To demonstrate this, we use different methods to calculate the SVAs for 100 model predictions based on random explaining sets of size $N = 10, 100, 1000, 10000$. We then estimate the average absolute observation bias of these SVAs by

$$\frac{1}{100}\frac{1}{d}\sum_{i=1}^{100}\sum_{j=1}^{d}|\phi_{ij}(v_{\hat{f}_S^{(N)}}) - \phi_{ij}(v_{\hat{f}_S^{(M)}})|,$$

where $M = 15,389$ is the size of the entire training set. Since we do not have an infinite explaining set, we use $\hat{f}_S^{(M)}$ to estimate $\hat{f}_S = \hat{f}_S^{(\infty)}$. The results are shown in Figure 4.5. We can see that SHAP-S has a high observation bias compared to other methods, even

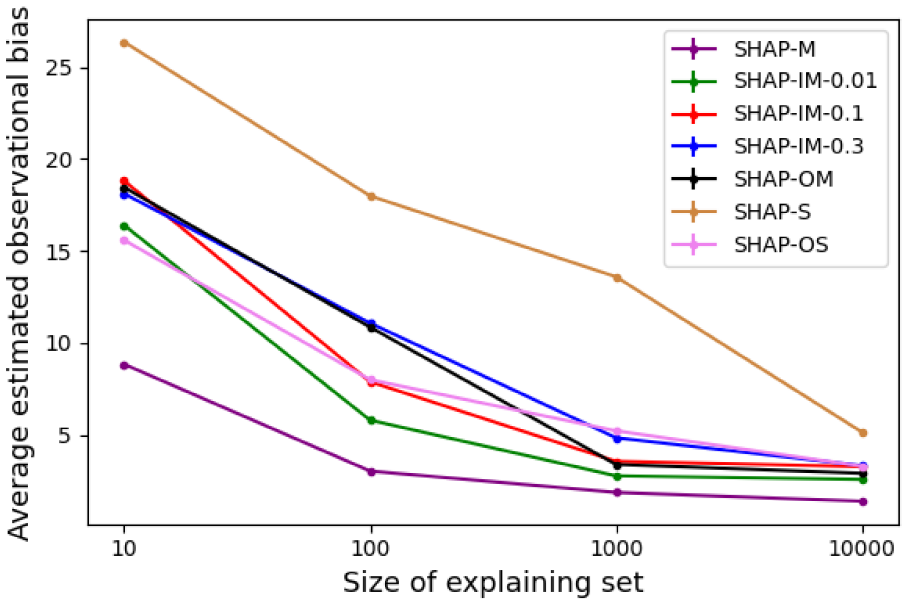


Figure 4.5: Estimated observation bias of different SVA methods.

when the explaining set is large. However, our OODIS refinement method (SHAP-OS) can greatly reduce the observation bias from SHAP-S. This is because the proposed OODIS surrogate model can estimate the CRF f_S more accurately, evidenced by much lower MSE

losses shown in Figure 4.6. Notably, the OODIS surrogate model trained on a 100-sized ex-

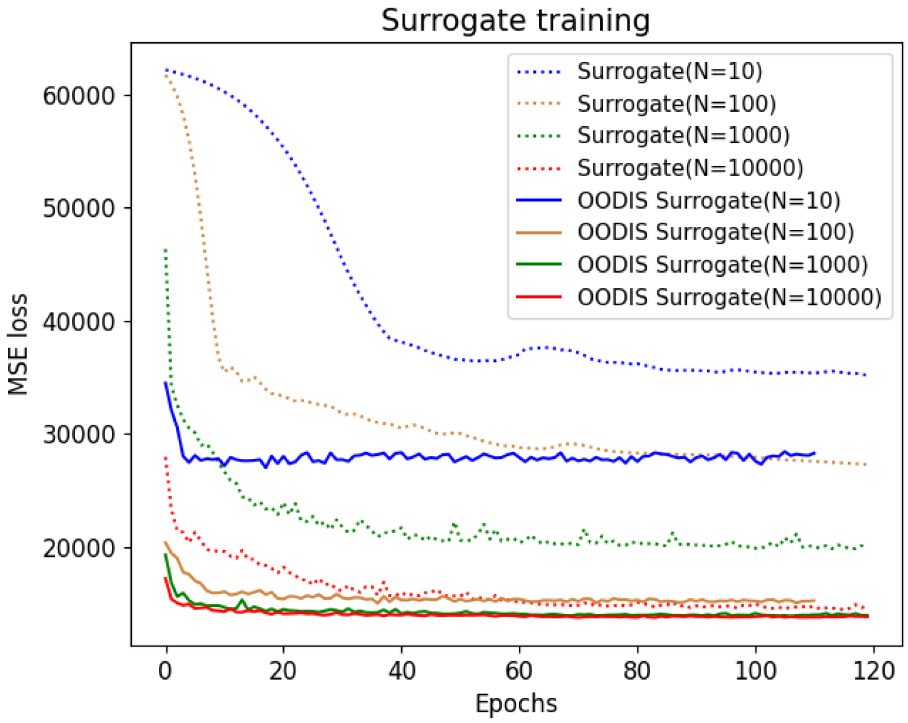


Figure 4.6: Training errors of SHAP-S and OODIS surrogate model.

plaining set can achieve similar performance to the original surrogate model trained on the 10,000-sized explaining set. On the other hand, the in-distribution and OODIS refinement can increase the observation bias of the SHAP-M method to a certain extent, the reason of which is discussed in Section 4.3. This problem subsides as the explaining set increases in size. However, this small increase in observation bias is acceptable and worthwhile, as it considerably decreases the structural bias (as shown in section 4.4.1), thereby reducing the total explanation error.

Together with the experiments in Section 4.4.1, this experiment emphasizes how our refinement methods can attain a better trade-off between observation and structural biases, resulting in less under-informative and less over-informative explanations.

4.4.2 Informative Global Explanations - Gene Retrieval

Experimental setting This experiment explores the application of global feature attributions for scientific discovery, exemplified by a gene retrieval task (Chen et al., 2020). In particular, we employ the Mechanisms of Action (MoA) Prediction Dataset from Kaggle (inversion et al., 2020), containing hundreds of gene expression features. In particular, we randomly select 20 true genes X from the dataset, then simulate another 20 synthetic genes X' that are highly correlated to the selected true genes (Pearson correlations > 0.99). Additionally, we introduce 200 noisy genes Z randomly generated from Gaussian distribution $\mathcal{N}(0, 1)$. This results in a dataset consisting of 40 informative genes (including synthetic genes X') and 200 uninformative genes. A synthetic disease target variable is generated using the 40 informative genes following one of the two models below:

- **Linear model:**

$$Y = \texttt{sigmoid}\left(\frac{1}{40}\sum_{i=1}^{40} X_i\right) > 0.5 \tag{50}$$

- **Nonlinear model:**

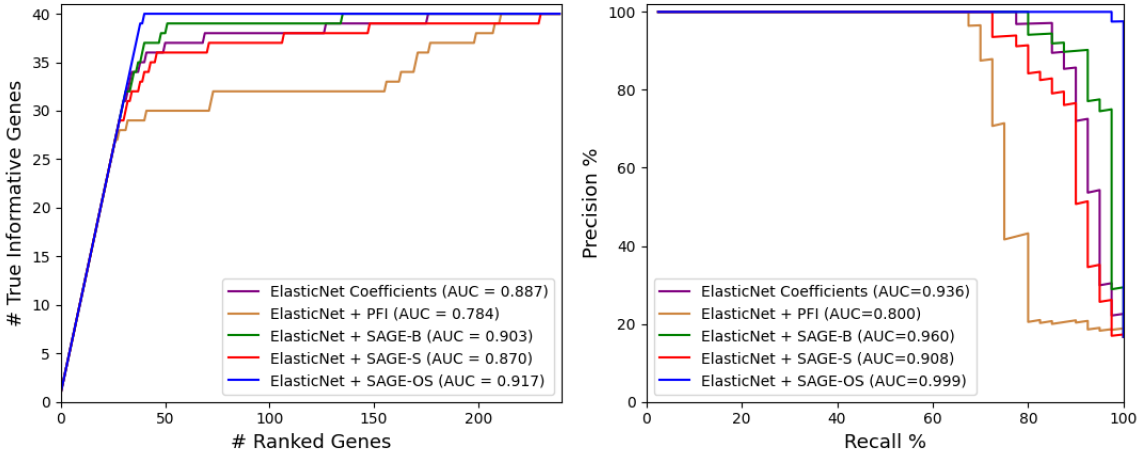
$$Y = \left(\frac{1}{40}\sum_{i=1}^{40} \cos X_i^2\right) > 0.9 \tag{51}$$

Our objective is to retrieve the 40 genes that are informative to the target disease variable by explaining a classification model trained on 1,500 samples of the generated dataset and tested on another 1,000 samples. For Y generated from Model (50), we consider a logistic ElasticNet model (test accuracy 0.958). For Y generated from Model (51), due to poor classification performance of the logistic ElasticNet model (test accuracy 0.648), we additionally consider an xgBoost classification model (test accuracy 0.879).

Global Explanation Methods Building upon the work of I. Covert et al. (2020), we employ the cross-entropy-based SVA, known as *Shapley additive global importance (SAGE)*, to identify the informative genes. SAGE offers insights into intrinsic relationships between

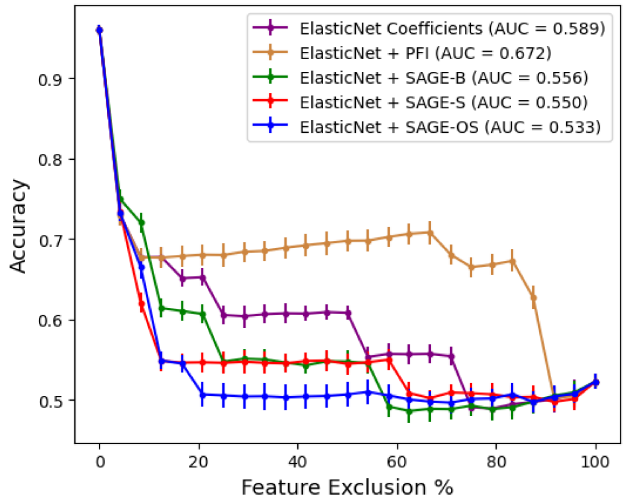
features and the target variable when both the original model f and the CRF f_S are optimal. To achieve this, we compute SAGE values for all 240 genes using three removal functions: the baseline (SAGE-B), the surrogate model (SAGE-S), and the OODIS surrogate model (SAGE-OS). Due to the high computational cost in the global setting, the MRF, in-distribution MRF, and OODIS MRF are not considered in this experiment. We compare the performance of the SAGE methods to two commonly used global feature importance methods, namely the absolute value of the feature coefficients in the logistic ElasticNet model (since regression models are inherently interpretable) and the permutation feature importance method (PFI) (Fisher, Rudin, & Dominici, 2019; Pedregosa et al., 2011).

Evaluation Metrics We evaluate the gene retrieval performance using three evaluation metrics: the *total operating characteristic (TOC)* curve (Pontius & Si, 2014), the *precision-recall (PR)* curve (Manning, 2009; Raghavan, Bollmann, & Jung, 1989), and the exclusion curve (Jethani, Sudarshan, Covert, Lee, & Ranganath, 2022; Petsiuk, Das, & Saenko, 2018). Specifically, the TOC and PR curves are employed to compare the true and retrieved genes where a higher area under the curve (AUC) suggests better performance. While the TOC and PR curves offer valuable metrics, their practical application is limited in real-world cases where the ground truth set of informative genes is often unknown. To address this, the *exclusion curve* is proposed, which essentially tracks the degradation of model performance (e.g., classification accuracy) as features are progressively excluded based on their assigned importance. If the global feature attribution method accurately captures the true information each feature holds regarding the target variable, the exclusion curve should exhibit a significant drop in performance as informative features are removed. This translates to a lower AUC for the exclusion curve, referred to as the exclusion AUC by Jethani et al. (2022). In this experiment, the exclusion curve monitors validation accuracy, derived from averaging 50 validation results through Monte-Carlo Cross Validation (Xu & Liang, 2001).



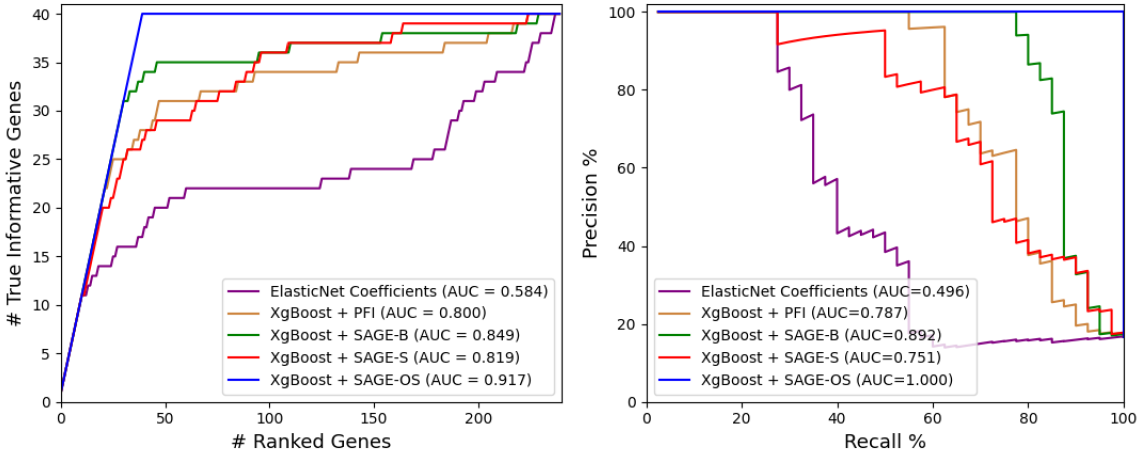
(a) TOC curve

(b) PRC curve



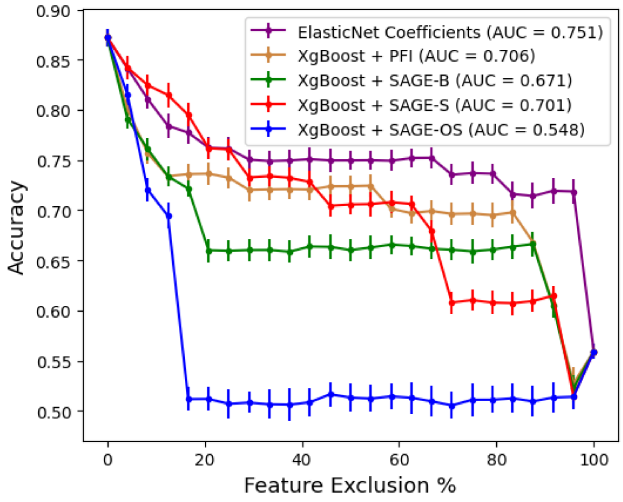
(c) Exclusion curve

Figure 4.7: Gene retrieval results when the target variable is generated from the linear model in Equation (50).



(a) TOC curve

(b) PRC curve



(c) Exclusion curve

Figure 4.8: Gene retrieval results when the target variable is generated from the nonlinear model in Equation (51).

Results Our results are plotted in Figures 4.7 and 4.8. In the case where the target disease variable is generated from a linear model (50), our proposed SAGE-OS method has the best performance on all three evaluation curves. In more detail, the top 40 genes with the highest global feature importance scores produced by the proposed SAGE-OS method contain 39 out of the true 40 genes, implying a near-perfect gene retrieval performance. The only missing gene is closely ranked at the 41st place. The ElasticNet Coefficient and SAGE-B methods also produce good results, with their top 40 genes containing 35 and 36 true genes, respectively. This is because the true model is linear and the ElasticNet model considers feature correlation to a certain extent (Chen et al., 2020). Furthermore, by excluding the top 20% important genes evaluated using the SAGE-OS method, the model accuracy decreases to 0.506. In comparison, this value is 0.55, 0.61, 0.65, and 0.68 when using the SAGE-S, SAGE-B, ElasticNet coefficient, and PFI methods, respectively. In the nonlinear case, our proposed SAGE-OS method significantly outperforms other methods on all three evaluation curves. It perfectly recovers all the true 40 genes by assigning them the highest importance scores. In contrast, the performance of ElasticNet coefficients deteriorates greatly as the true model becomes nonlinear: the top 40 genes with the highest absolute coefficients value contain only 17 true genes. In this case, excluding the top 20% important genes with SAGE-OS reduces the model accuracy to 0.51, while the model accuracy remains above 0.65 when using other global feature attribution methods.

4.5 Related Work

Our paper aims to correct explanation errors caused by existing SVA methods. Many papers have discussed how existing SVA methods make unreasonable explanations (Chen et al., 2023; Huang & Marques-Silva, 2023; I. E. Kumar et al., 2020). However, these papers did not provide remedies for the illustrated problems. Other papers proposed new SVA methods (Aas et al., 2021; Frye et al., 2020; Mase et al., 2019; Sundararajan & Najmi,

2020), however, these methods only target one specific problem, leaving the others unresolved (Chen et al., 2023). In Chapter 3, we formalized a comprehensive error analysis framework for SVA methods, where the explanation errors are decomposed into observation and structural biases. Our refinement methods target both these biases, resulting in more informative SVAs.

There have also been papers (Slack et al., 2020; Taufiq et al., 2023; Yeh et al., 2022) that use OOD detection techniques. Specifically, Slack et al. (2020) uses an OOD classifier to create adversarial attacks on SVA methods. Yeh et al. (2022) use an OOD classifier for density estimations that help avoid such adversarial attacks. Although the on-manifold restricted method proposed by Taufiq et al. (2023) is similar to our in-distribution method, they focus on interventional SVAs while we focus on observational SVAs.

Finally, although OOD detection and importance sampling are well-established methods in the literature, to the best of our knowledge, we are the first to combine these two methods and apply them to model explanation tasks using SVAs.

4.6 Conclusion

In this chapter, we seek to correct the observation and structural biases produced by existing SVA methods, resulting in under- and over-informative explanations. To do so, we propose two novel solutions: the in-distribution and out-of-distribution importance sampling (OODIS) refinement methods. The two methods aim to alleviate the data sparsity and untenable feature independence assumption problem by increasing the sample size while correcting the distribution drift. Via extensive experiments, we show that our refinement methods outperform existing SVA methods in both local and global explanation tasks, producing local feature attributions that are simultaneously less over- and under-informative, and retrieving correct global feature-target informational dependency. In future work, we wish to apply our methods to facilitate business decisions and scientific discovery.

Chapter 5

A Universal Standardization for Global Model Behaviors on Imbalanced Data

5.1 Introduction

While Chapters 3 and 4 focus on addressing challenges related to the first element, *feature removal*, of the *removal-based framework* (detailed in Section 2.4), this chapter shifts its focus to the second element: *model behavior*. In the context of SVA explanations, specifying the model behavior to be explained is crucial. This behavior could represent an individual prediction (e.g., in local SVA) or a performance metric evaluated over the entire dataset (e.g., in global SVA). In this chapter, we concentrate on global model behavior, with a particular emphasis on classification performance metrics.

Classification is an important ML task with wide applications in fields such as finance, healthcare, business, and more. In practice, a classifier is trained on huge amounts of data, thus interpretable and reliable performance metrics are important to evaluate and track the classifier's performance and notify the developer if the classifier needs to be retrained.

There have been many performance metrics proposed in the literature (Fawcett, 2006; Japkowicz & Shah, 2011; Powers, 2020) to evaluate classifiers, such as precision, f1_Score,

Matthews Correlation Coefficient (MCC), Receiver Operating Characteristic (ROC) curve, Precision-Recall curve (PRC), lift curve, etc. However, researchers have not agreed on a single general-purpose classification performance metric (Chicco, Tötsch, & Jurman, 2021). In fact, in different applications, different performance metrics are preferred. For example, while the ROC curve is preferred for balanced data, the PRC and f1_Score are more favorable in imbalanced datasets (Saito & Rehmsmeier, 2015). On the other hand, Chicco et al. (2021) recommends the MCC for reliable classification. Since each performance metric has a different range of values, which depends on the test set, and a different interpretation, there are no universal criteria to evaluate the goodness of classifiers and model drift detection.

Furthermore, most classification tasks are trained and tested on imbalanced data, e.g., fraud detection, disease diagnosis, or custom recommendations. As many performance metrics are sensitive to the imbalance rates, it is hard to evaluate and monitor classification performance using the preferred performance metrics if the test sets have different imbalance rates. Such sensitivity is studied by Luque, Carrasco, Martín, and de las Heras (2019), however, they did not suggest a method to address the problems caused by class imbalance. Koyejo, Natarajan, Ravikumar, and Dhillon (2014) propose a generalization of performance metrics, however, their focus is on utilizing such metrics to train classifiers rather than evaluate them. They also do not consider the issue of class imbalance.

In this work, we aim to provide a universal standardization to evaluate classification performance that can applied to any confusion matrix-based performance metrics while alleviating the dependency on class imbalance. Our key idea is to evaluate classification performance relative to all possible performances given the test set. In particular, we provide a detailed mathematical formulation, justifications, explanations, and experiments to demonstrate how such an idea can be implemented. Our contributions are summarized as follows.

- We formalize a mathematical framework for confusion matrix-based classification performance metrics. We categorize them into labeling and scoring metrics and derive the common properties of each type of metric.

- We propose a universal standardization called the **outperformance score function**, or **OPS function**, which is essentially the head probability of the observed performance with respect to a distribution of possible performances given the class imbalance rates. The function can be applied to any confusion matrix-based performance metrics, and the resulting **outperformance scores** share the same scale and similar interpretation. Hence, our outperformance scores, as well as their corresponding global SVA, termed **OPS-SAGE**, can be compared across datasets and metrics of the same category.

- We present how we select the assumed distribution of performances that are used to compare to the observed performance. We further demonstrate how to calculate the outperformance scores by setting reasonable distributions of possible performance given the imbalanced rates.

- We illustrated via experiments how our method can be used on different performance metrics. We find that our proposed outperformance scores are robust to class imbalance and thus can be used to compare or track performances over different datasets. Furthermore, the OPS-SAGE can be applied to explain the model drift.

5.2 Classifier Performance Metrics

5.2.1 Preliminary

Let us consider a binary classification problem, where the goal is to predict label $Y \in \{0, 1\}$ based on features $X = (X_1, ..., X_d) \in \mathcal{X} \subset \mathbb{R}^d$ using the observed data set $\mathcal{D}_{\text{train}} =$

$\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where y_i is the true class label of input data point $\mathbf{x}_i$, $i = 1, ..., m$. A classifier $\hat{f}$ can be obtained by training a classification model f on the data set $\mathcal{D}_{\text{train}}$. Depending on the classification model f, for an input $\mathbf{x} \in \mathcal{X}$, the classifier $\hat{f}$ can output either a binary label $\hat{f}(\mathbf{x}) \in \{0, 1\}$, or a score $\hat{f}(\mathbf{x}) \in [0, 1]$. This score is usually related to the probability that the instance associated with input $\mathbf{x}$ has class membership 1 instead of 0. It also can be thought of as the quantified relevance to the positive class (label 1). When a binary label is produced, the predicted class membership $\hat{y}$ is equal to the classifier's output $\hat{f}(\mathbf{x})$. On the other hand, when a score is produced, a threshold t must be applied to determine the instance's predicted class membership

$$\hat{y} = \begin{cases} 1 & \text{if } \hat{f}(\mathbf{x}) \geq t \\ 0 & \text{otherwise} \end{cases}. \tag{52}$$

The performance of the classifier $\hat{f}$ can be evaluated using a test data set $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of size n. Specifically, the classifier's outputs $\{\hat{f}(\mathbf{x}_i)\}_{i=1}^n$ are compared to the true labels $\{y_i\}_{i=1}^n$ using some performance metrics. Many performance metrics are defined based on the *confusion matrix*.

5.2.2 Confusion Matrix

For each instance i in the test set $\mathcal{D}_{\text{test}}$, there are four possible combinations of the actual label y_i and the predicted label $\hat{y}_i$, that is $\{(1, 1), (0, 1), (1, 0), (0, 0)\}$. The confusion matrix is a 2×2 table that counts the number of instances in the test set $\mathcal{D}_{\text{test}}$ that fall into each of these four value combinations. As illustrated Table 5.1, the confusion matrix consists of four elements $\{n_1, n_2, n_3, n_4\}$, where $n_1 = \sum_{i=1}^n \mathbb{I}(y_i = 1, \hat{y}_i = 1)$, counting the *true positives* (TP); $n_2 = \sum_{i=1}^n \mathbb{I}(y_i = 0, \hat{y}_i = 1)$, counting the *false positive* (FP); $n_3 = \sum_{i=1}^n \mathbb{I}(y_i = 1, \hat{y}_i = 0)$, counting the *false negative* (FN); and $n_4 = \sum_{i=1}^n \mathbb{I}(y_i =$

Table 5.1: A confusion matrix

		True labels	
		1	0
Predicted labels	1	n_1	n_2
	0	n_3	n_4

$0, \hat{y}_i = 0)$, counting the *true negative* (TN). It is easy to see that $n_1 + n_2 + n_3 + n_4 = n$. From the confusion matrix, we can calculate the *Type-I error* α, *Type-II error* β and the *class imbalance rate* (also called *prevalence*) π as follows.

$$\alpha = \frac{n_2}{n_2 + n_4}, \quad \beta = \frac{n_3}{n_1 + n_3}, \quad \text{and} \quad \pi = \frac{n_1 + n_3}{n}. \tag{53}$$

WLOG, we say that class imbalance happens when π is low, that is, when there are much fewer instances in the positive class (true label 1) than in the negative class (true label 0).

Lemma 2. *For a confusion matrix,* $\{n, \pi, \alpha, \beta\}$ *is a re-parametrization of* $\{n_1, n_2, n_3, n_4\}$ *when* $\pi \in (0, 1)$ *(i.e., when* α *and* β *are well-defined).*

Proof. We can write $\{n_1, n_2, n_3, n_4\}$ as functions of $\{n, \pi, \alpha, \beta\}$, i.e.

$$n_1 = n\pi(1-\beta), \quad n_2 = n\pi\beta, \quad n_3 = n(1-\pi)\alpha, \quad \text{and} \quad n(1-\pi)(1-\alpha).$$

The Jacobian for this transformation is

$$\begin{bmatrix} \pi(1-\beta) & n(1-\beta) & 0 & -n\pi \\ \pi\beta & n\beta & 0 & n\pi \\ (1-\pi)\alpha & -n\alpha & n(1-\pi) & 0 \\ (1-\pi)(1-\alpha) & -n(1-\alpha) & -n(1-\pi) & 0 \end{bmatrix}.$$

The determinant of the Jacobian is $n^3\pi(1-\pi)$, which is non-zero when $\pi \neq 0$ and $\pi \neq 1$. Thus, the Jacobian is nonsingular, and $\{n, \pi, \alpha, \beta\}$ is a re-parametrization of

$\{n_1, n_2, n_3, n_4\}$. □

Lemma 2 shows that we can identify a confusion matrix based on the sample size $n \in \mathbb{N}$, the class imbalance rate $\pi \in (0, 1)$, the Type-I error rate $\alpha \in [0, 1]$ and the Type-II error rate $\beta \in [0, 1]$. This is convenient, compared to $\{n_1, n_2, n_3, n_4\}$, especially when we want to focus on the class imbalance rate π. As mentioned in Section 5.2.1, many classifier performance metrics are calculated based on the confusion matrix. We classify them into two categories: *labeling metrics* and *scoring metrics*.

5.2.3 Labeling Metrics

Labeling metrics M_L are computed from a *single* confusion matrix C of the predicted and actual labels $\{(y_i, \hat{y}_i)\}_{i=1}^n$. Examples of labeling metrics are *recall*, *precision*, and *f1_score* (Powers, 2020), and the *Matthews correlation coefficient* (MCC) (Chicco et al., 2021), the formulas of which are given in Table 5.2. We can see that these metrics are only dependent on π, α, β, but independent of the test data size n. This illustrates the usefulness of the new parametrization.

Table 5.2: Formulas of some labeling metrics terms of two parameterizations of the confusion matrix.

Metrics	**In terms of** $\{n_1, n_2, n_3, n_4\}$	**In terms of** $\{n, \pi, \alpha, \beta\}$
Recall	$\frac{n_1}{n_1+n_3}$	$1-\beta$
Precision	$\frac{n_1}{n_1+n_2}$	$\frac{\pi(1-\beta)}{\pi(1-\beta)+(1-\pi)\alpha}$
f1_score	$\frac{2n_1}{2n_1+n_2+n_3}$	$\frac{2\pi(1-\beta)}{\pi(2-\beta)+(1-\pi)\alpha}$
MCC	$\frac{n_1\times n_4-n_2\times n_3}{\sqrt{(n_1+n_2)(n_1+n_3)(n_4+n_2)(n_4+n_3)}}$	$\frac{1-\alpha-\beta}{\sqrt{(1-\alpha+\frac{\pi}{1-\pi}\beta)(1-\beta+\frac{\pi}{1-\pi}\alpha)}}$

5.2.4 Scoring Metrics

When predicting labels of an instance given input $\mathbf{x}$, instead of directly outputting a binary label $\hat{y} \in \{0, 1\}$, many machine learning classifiers return a numeric score $p =$

$\hat{f}(\mathbf{x}) \in [0,1]$. The predicted label is obtained by specifying a threshold $t \in (0,1)$ and then applying it to Equation (52). Thus, given the true labels $\{y_i\}_{i=1}^n$ and classifier output $\{\hat{f}(\mathbf{x}_i)\}_{i=1}^n$, a threshold t uniquely defines a confusion matrix C. Although the threshold $t = 0.5$ is often used, there is no stringent criterion of threshold selection due to two main reasons. First, the prediction score $p = \hat{f}(\mathbf{x})$ is not well-calibrated, because it does not represent the correct prediction likelihood despite being often interpreted as the predicted probability (Guo, Pleiss, Sun, & Weinberger, 2017; Naeini, Cooper, & Hauskrecht, 2015; Niculescu-Mizil & Caruana, 2005). Second, the threshold should be flexibly selected with regard to different applications. For example, a high threshold is usually used in custom marketing to reduce the Type-1 error. On the other hand, a low threshold is preferred in risk detection to control for the Type-2 error.

This motivates the use of scoring metrics that summarize classifiers' overall performance for all possible thresholds. Essentially, they visualize certain trade-offs for all possible thresholds in curves. For example, the *Receiver Operating Characteristics* (ROC) curve (Fawcett, 2006) visualizes the trade-off between a low Type-1 error rate α and a high recall $1-\beta$. Another example is the *Precision-Recall curves* (PRC) (Powers, 2020) visualizes the trade-off between a high precision and a high recall. Table 5.3 summarizes the formulas of popular scoring metrics curves. We again see that the formulas of the x- and y-axes for these curves do not involve n.

Table 5.3: Formulas of some scoring metrics terms of $\{n, \pi, \alpha, \beta\}$.

Curve	ϕ_x: x-axis	ϕ_y: y-axis	ideal AUC
ROC	Type I error: α	recall: $1-\beta$	1
PRC	recall: $1-\beta$	precision: $\frac{\pi(1-\beta)}{\pi(1-\beta)+(1-\pi)\alpha}$	1
Lift Curve	percentage:[1] $\pi(1-\beta)+(1-\pi)\alpha$	lift: $\frac{(1-\beta)}{\pi(1-\beta)+(1-\pi)\alpha}$	$1-\log\pi$
Gain Curve	percentage: $\pi(1-\beta)+(1-\pi)\alpha$	recall: $1-\beta$	$1-\frac{\pi}{2}$

It is not trivial or possible to find the formula (relating the x- and y-axis) of the curve.

[1] Percentage means the percentage of instances predicted as positive by the classifier

Hence, in practice, the curves are approximately plotted from points computed based on a sequence of J thresholds $1 = t_1 > t_2 > \ldots > t_{J-1} > t_J = 0$. In this case, we can say that the scoring metrics $\mathbf{M}_S$ are approximated from a sequence of confusion matrices $C_1, C_2, ..., C_J$ obtained by decreasingly varying the thresholds t from 1 to 0.

To summarize the curve into a single numeric value, people often calculate the *area under the curve*, i.e., AUC. For all curves, the larger the AUC the better the performance. The last column of Table 5.3 shows the ideal AUC when the predicted labels perfectly match the true labels. We can divide the AUC to the ideal value to obtain the Normalized AUC or NAUC, whose values range from 0 to 1, i.e.,

$$NAUC = \frac{AUC}{\text{ideal } AUC} = \frac{AUC}{AUC \text{ induced by the ideal classifier}}. \tag{54}$$

The NAUC measures how close the selected classifier is to the ideal classifier that predicts everything correctly.

5.2.5 Issues with Confusion Matrix-Based Performance Metrics

Dependence on Imbalance Rates From Tables 5.2 and 5.3, we can find that some metrics, such as ROC, depend only on the classification errors $\{\alpha, \beta\}$, but not on π, thus they are robust to class imbalance. However, due to the loss of information for class distribution, they might be misleading and less informative. For example, as analysis in (Lobo, Jiménez-Valverde, & Real, 2008), ROC has some drawbacks including the decoupling from the class skew. On the other hand, most other metrics such as f1_score, MCC, and PRC, include π in their formulas and hence are more informative and preferable when the data is imbalanced. For example, the study in (Saito & Rehmsmeier, 2015) shows that PRC is more informative than ROC when evaluating classification performance on imbalanced data. However, through π, these metrics may be sensitive to class imbalance. Specifically,

π can affect the calibrations of these metrics, making it difficult to measure the quality of a classifier. **A low value does not necessarily indicate a poor quality**. For example, the value of $f1 = 0.6$ suggests a poor classifier when data is balanced, i.e., $\pi = 0.5$. When $\pi = 0.1$, on the contrary, $f1 = 0.6$ suggests a good classifier. As a result, a value decrease is not necessarily a quality degradation. Changes in π can cause significant fluctuations in these metrics, making them unreliable to detect drifts in a classifier's performance. In applications, the prevalence π of the test set, such as the COVID infection rate, often varies over time. Hence, it is not good practice to monitor classifier performance by comparing the values of these metrics on test sets of different time periods. Moreover, π can even change the range of these metrics, making them **not legitimate to be averaged over multiple classes or test sets with different class ratios or imbalance rates**. As a result, it is difficult to extend these performance metrics to multi-class evaluation and apply them for cross-validation.

No Free Lunch There is an ongoing discussion about what metrics should be used to evaluate the performance of classification models (Chicco et al., 2021; Hossin & Sulaiman, 2015). In fact, different applications may prefer different performance metrics. For example, lift curve is preferred in custom recommendation, and PRC is preferred in event identification. Since different performance metrics have different ranges of values, there is no universal standard to evaluate the classification performance.

5.3 A Universal Standardization

To address the problems discussed in Section 5.2.5, we aim to mitigate the sensitivity of certain performance metrics on class imbalance rates and thus provide a universal standard for classification performance evaluation.

Definition 5. *A **Universal Standardization** of classification performance evaluation is a methodology that ensures performance metrics are consistently calibrated across different test sets, regardless of data imbalance. Specifically, a **universally standardized metric** must satisfy the following criteria:*

(1) a decrease in its value always suggests a quality degradation;

(2) a common evaluation criterion (e.g., a fixed threshold) can be uniformly applied across all test sets to judge a classifier's quality;

(3) its values can be meaningfully averaged across multiple classes or test sets, regardless of class distributions or imbalance rates.

In this work, we propose that universal standardization can be achieved by evaluating the relative performance of a classifier with respect to a distribution of possible performances conditioned on the class imbalance rates. To gain more insight into this conception, let's consider a metaphor. We can think about classifiers as students, and the corresponding learning task as a certain course. To evaluate their performance for this course (learning task), the students (classifiers) are usually tested on some given exams (test sets). However, the average value of the testing grades (metrics) always decreases as exams (test sets) become more difficult (imbalanced). Hence, it is not a good idea to judge how good a student (classifier) is depending simply on the grade (metric) without considering the effect of exam difficulty (data imbalance). Let's assume a difficult exam to be the university entrance examination, then what is the common admission criterion? The answer is student rank. For example, a student, who achieves a grade of only 70/100 but is ranked in the top 5% of all the candidates, can also be considered as an excellent student and will be selected by most universities.

Similarly, to fairly judge how good a classifier is, we need to consider all performances that can be possibly achieved given the test set. In other words, we need to consider the

distribution of possible performances conditioned on the test set. Note the information that the test set contributes to a confusion matrix is completely captured by π and n. However, as seen in Sections 5.2.3 and 5.2.4, most confusion-based metrics do not depend on n. Conditioning on n may make the standardization unnecessarily dependent on n. Specifically, the space of the possible performances being compared to, which corresponds to the space of possible values of α and β, depends on n. This may cause the standardization to be incomparable across datasets and metrics.

We thus consider two classes of confusion matrix-based performance metrics which depend on only the Type I and Type II errors and the class imbalance rate. Let $\mathbf{M}_L$ denote a labeling metric and $\mathbf{M}_S$ denote a scoring metric. As discussed in Sections 5.2.3 and 5.2.4, labeling metrics are a function of a single confusion matrix while scoring metrics can be estimated by a sequence large enough number of confusion matrices. Thus, $\mathbf{M}_L$ is a function of α, β and π, and $\mathbf{M}_S$ can be estimated by $\widehat{\mathbf{M}}_S(\alpha_1, \alpha_2, ..., \alpha_J, \beta_1, \alpha_2, ..., \beta_J, \pi)$, where (α_j, β_j, π) encodes the information from confusion matrix C_j corresponding to a threshold t_j. WLOG, we assume that the higher value of $\mathbf{M}$ indicates better classification performance (for those matrices $\mathbf{M}$ that the lower value indicates better performance, we can consider $-\mathbf{M}$, or $\frac{1}{\mathbf{M}}$ if $\mathbf{M} > 0$). Let uppercase notation A, B and P denote the random variable version of α, β and π, respectively.

Definition 6 (Outperformance Score Function)**.** *Consider a classifier $\hat{f}$ evaluated on a given dataset $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i)\}$ using a performance metrics $\mathbf{M}$, which return a performance score μ. The **outperformance score (OPS) function** can then be defined as:*

$$\text{OPS}_{\mathbf{M}}(\mu; \pi) = \Pr\{\mathbf{M} < \mu | P = \pi\}. \tag{55}$$

Specifically, for a labeling metric $\mathbf{M}_L$,

$$\text{OPS}_{\mathbf{M}_L}(\mu; \pi) = \Pr\{\mathbf{M}_L(A, B, P) < \mu | P = \pi\},$$

and for a scoring metric $\mathbf{M}_S$ *estimated using* J *confusion matrices,*

$$\text{OPS}_{\widehat{\mathbf{M}}_S}(\mu;\pi) = \Pr\{\widehat{\mathbf{M}}_S(A_1,...,A_J,B_1,...,B_J,P) < \mu | P = \pi\},$$

where the probabilities are taken with respect to an assumed joint distribution given P *of* (A,B) *or* $(A_1,...,A_J,B_1,...,B_J)$*, respectively.*

Essentially, the outperformance score is the probability that the observed performance outperforms a random performance given the class imbalance rate. Thus, when applied to any confusion matrix-based classification performance metrics, the returned outperformance score always ranges within $[0,1]$, and the higher the outperformance score, the better the classification performance. This provides a universal standard to evaluate classifiers and a universal interpretation. As a probability, the outperformance score has the following linear property:

Property 6.1 (Linear Property)**.** *If performance metrics* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *satisfy* $\mathbf{M}_2 = a\mathbf{M}_1 + b$ *where* $a > 0$ *and* b *are constants, then* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *has the same outperformance scores.*

Proof. Let $\mathbf{M}_1$ and $\mathbf{M}_2$ return values of μ_1 and $\mu_2 = a\mu_1 + b$, respectively. By definition 6, we have

$$\begin{aligned}
\text{OPS}_{\mathbf{M}_2}(\mu_2;\pi) &= \Pr\{\mathbf{M}_2 < \mu_2 | P = \pi\} \\
&= \Pr\{a\mathbf{M}_1 + b < a\mu_1 + b | P = \pi\} \\
&= \Pr\{\mathbf{M}_1 < \mu_1 | P = \pi\} \\
&= \text{OPS}_{\mathbf{M}_1}(\mu_1;\pi).
\end{aligned}$$

□

Given $P = \pi$, the property also holds if a and b are functions of π. This linear property

is useful because many existing performance metrics have linear relations. For example, lift $=$ precision$/\pi$, that is, their OPS function returns the same value for the same classifier.

In Definition 6, the outperformance score is taken with respect to an assumed joint distribution of Type I and Type II errors given the imbalance rate. Such an assumed distribution needs to satisfy the properties and constraints of confusion matrices. In the following, we describe how we choose such distributions for labeling and scoring metrics.

5.4 Outperformance Score of Labeling Metrics

Note that labeling matrices are functions of a single confusion matrix. For independent instances $(y_i, \hat{y}_i)_{i=1}^n$, in which the probabilities that an instance i falls into each of the four categories $\{(1,1),(0,1),(1,0),(0,0)\}$ are equal, given the class imbalance rate, Type I and Type II errors are independent. Indeed, the cells of confusion matrices that summarize such instances follow an equiprobability multinomial distribution. Since conditioning on π implies conditioning on subtotals n_1+n_3 and n_2+n_4, properties of multinomial distributions dictate that subvectors (n_1, n_3) and (n_2, n_4) are independent. Together with Equation (53), this implies that for fixed test size and given the imbalance rate, Type I and Type II errors are independent.

We thus consider the case where Type I error A and Type B error B independently follow $\mathrm{Unif}[0,1]$ distribution when calculating the outperformance score of labeling metrics. This implies that all possible performances are equally likely. We discussed in Section 5.3 that n can limit possible values of Type I and Type II errors. Hence, by assuming A and B independently follow $\mathrm{Unif}[0,1]$ distribution, we implicitly assume infinite test size n. Thus, a careful interpretation of the outperformance score (for labeling metrics) is "the probability that the classifier outperforms random (equally likely) performances, given that it gives a similar performance on an infinitely large test set with class imbalance rate π".

The above-described distribution of labeling matrices can be represented by a simple

directed acyclic graph (DAG) in Figure 5.1. Given $P = \pi$ and assuming $A, B \sim \text{Unif}[0, 1]$,

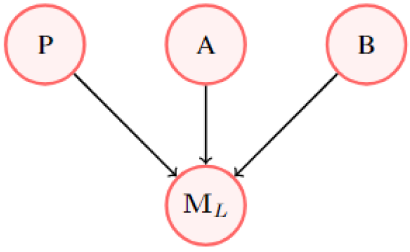


Figure 5.1: DAG for labeling matrices.

the outperformance score can be computed using variable elimination as:

$$\text{OPS}_{\mathbf{M}_L}(\mu; \pi) = \Pr\{\mathbf{M}_L(A, B, P) < \mu | P = \pi\} = \int_0^1 \int_0^1 \mathbb{I}(\mathbf{M}_L(\pi, \alpha, \beta) < \mu) d\alpha d\beta, \tag{56}$$

where $\mathbb{I}(\cdot)$ denote the indicator function. For some labeling metrics $\mathbf{M}_L$, the outperformance score can be derived analytically, such as f1_score as shown in Example 4. For others, the outperformance scores can be estimated using the trapezoidal rule or Monte Carlo approximation, such as MCC as shown in Appendix C.1.

Example 4 (Outperformance Score of f1_score)**.** *A popular labeling metric is f1_score, whose formula can be written as*

$$f1(\pi, \alpha, \beta) = \frac{2\pi(1-\beta)}{\pi(2-\beta) + (1-\pi)\alpha}. \tag{57}$$

The OPS function of f1_score can be computed as:

$$\text{OPS}_{f1}(\mu; \pi) = \begin{cases} \frac{(1+\pi)\mu}{2\pi(2-\mu)}, & \text{if } 0 \le \mu \le \frac{2\pi}{1+\pi} \\ \frac{(1+\pi)\mu}{2\pi(2-\mu)} - \frac{[(1+\pi)\mu - 2\pi]^2}{2\pi(1-\pi)\mu(2-\mu)}, & \text{if } \frac{2\pi}{1+\pi} < \mu \le 1. \end{cases} \tag{58}$$

For example, the trivial positive classifier (TPC), for who $\alpha = 1$ and $\beta = 0$, always returns

$f1(\pi, 1, 0) = \frac{2\pi}{1+\pi}$, *and thus the following outperformance score*

$$\text{OPS}_{f1}(\frac{2\pi}{1+\pi}; \pi) = \frac{1+\pi}{2}. \tag{59}$$

If a classifier cannot reach this baseline, then we prefer to select the TPC.

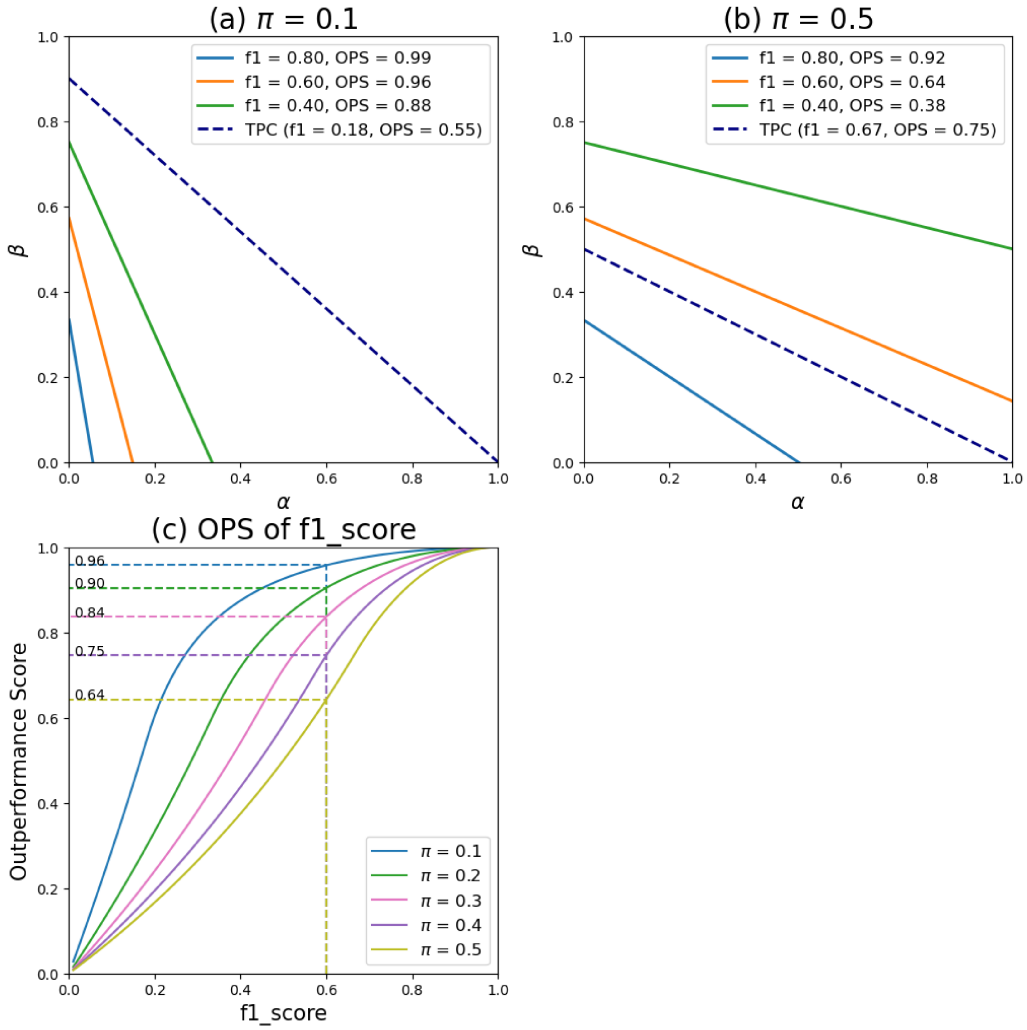


Figure 5.2: Geometric representation of OPS_{f1} when (a) $\pi = 0.1$; (b) $\pi = 0.5$. And (c) plots the OPS function of f1_score given different π.

The graphical representation of OPS_{f1} *can be found in Figure 5.2 (a) and (b). Given* $P = \pi$*, the performance space can be projected onto the unit square* $[0, 1]^2$ *that includes all possible pairs of* (α, β)*. For a given* $\mu \in (0, 1)$*, the straight line* $f1(\pi, \alpha, \beta) = \mu$

divides the unit square into two parts, where the left part indicates better performance ($f1 > \mu$), and right part indicates worse performance ($f1 < \mu$). The outperformance score $\text{OPS}_{f1}(\mu;\pi)$ *is the area of the right part, representing the proportion of all possible performances that are outperformed by $f1 = \mu$. Moreover, as shown in Figure 5.2 (c), the imbalance rate π changes the distribution of f1_score, and for the same value $f1 = 0.6$, the outperformance score and thus the quality of the classifier increases as π decreases. Hence, lower values of the f1_score could be satisfactory for highly imbalanced data while inadequate for balanced data.*

5.5 Outperformance Score of Scoring Metrics

Recall that scoring metrics are estimated using a sequence of confusion matrices corresponding to a sequence of thresholds $t_1, ..., t_J$. Given π, this in turn corresponds to a sequence of Type I errors $\alpha_1, ..., \alpha_J$ and a sequence of Type II errors $\beta_1, ..., \beta_J$. According to Equation (52), if $t_j < t_k$, then the number of predicted positive labels using threshold t_j will be greater than that using t_k. Thus, $\alpha_j > \alpha_k$ and $\beta_j < \beta_k$. This shows the trade-off between preferably low Type I and Type II errors when choosing the threshold. Nevertheless, when calculating the outperformance score for scoring metrics, the assumed joint distribution of $(A_1, A_2, ..., A_J, B_1, B_2, ..., B_J)$ should respect the property that As and Bs have opposite ordering, that is if $A_j < A_k$, then $B_j > B_k$.

It is challenging to specify such a distribution so that all possible performances are equally likely. Moreover, if such a distribution exists, it would also be difficult to derive analytic formulas for the OPS function. Thus the outperformance scores will be estimated using Monte Carlo approximation. To do it, we propose a *Directed Binary Tree* (DBT) distribution, which is easy to sample from, to represent the joint distribution of $(A_1, A_2, ..., A_J, B_1, B_2, ..., B_J)$. In more details, let $E = (A, B)$ and $\theta = (l_A, u_A, l_B, u_B)$ where (l_A, u_A) is the possible range of A and (l_B, u_B) is the possible range of B. We can

draw a sample from the DBT distribution by first drawing the Type I and Type II errors (α_1, β_1) independently from Unif$[0, 1]$. The obtained sample α_1 and β_1 each divides the range $[0, 1]$ into two parts. These parts can be used to draw the next two sets of errors: (α_2, β_2) drawn uniformly from the left part of α_1 and right part of β_1, and (α_3, β_3) from the right part of α_1 and left part of β_1. The next four samples can be drawn similarly using (α_2, β_2) and (α_3, β_3), and so on. This procedure returns a sample $(\alpha_1, ..., \alpha_J, \beta_1, ..., \beta_J)$ that satisfy the opposite ordering we discussed above. We summarize the sampling procedure in Algorithm 1 and visualize it using a directed binary tree, as illustrated by Figure 5.3.

Algorithm 1: Sampling from the Directed Binary Tree distribution

Input : Number of Iteration K

Output: A sample from the Directed Binary Tree distribution of $J = 2^{K-1} + 1$ points

$ActiveRange \leftarrow$ list that contains (0, 1, 0, 1);
$NewActiveRange \leftarrow$ empty list;
$SampleSet \leftarrow$ empty list;
for $k \leftarrow 1$ **to** K **do**
  foreach $l \in ActiveRange$ **do**
    Sample $\alpha \sim$ Unif$[l_A, u_A]$ and $\beta \sim$ Unif$[l_A, u_A]$;
    Add $e = (\alpha, \beta)$ into $SampleSet$;
    Add $\theta_l = (l_A, \alpha, \beta, u_B)$ and $\theta_r = (\alpha, u_A, l_B, \beta)$ into $NewActiveRange$;
  end
  $ActiveRange \leftarrow NewActiveRange$;
  $NewActiveRange \leftarrow$ empty list;
end
return $SampleSet$.

With G samples from the binary tree distribution, we can estimate the outperformance score to be

$$\text{OPS}_{\widehat{\mathbf{M}_L}}(\mu; \pi) \approx \frac{1}{G} \sum_g \mathbb{I}\left(\widehat{\mathbf{M}}_L(\alpha_1^{(g)}, ..., \alpha_J^{(g)}, \beta_1^{(g)}, ..., \beta_J^{(g)}, \pi) < \mu\right). \tag{60}$$

In practice, a large number of samples can be generated once and stored to calculate any outperformance scores in the future.

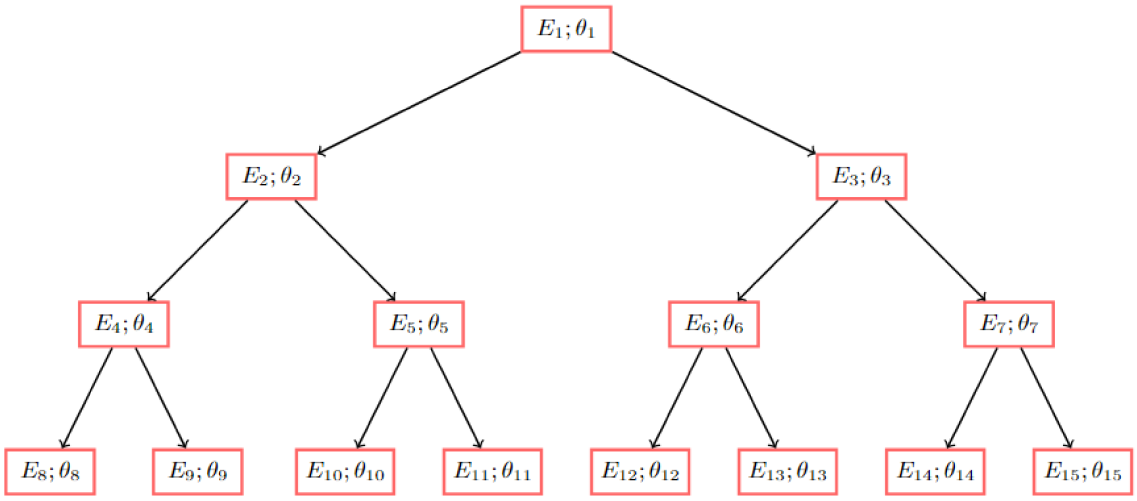


Figure 5.3: A Directed Binary Tree model with depth=3.

We are usually interested in two kinds of scoring metrics: one is the area of the curve, i.e., AUC, and the other is a specific point (u, v) on the curve. Typically, we can view u as the profit and v as the cost. To achieve a profit of u, what level of performance does the cost of v indicate? For the gth sample from the DBT distribution $(\alpha_1^{(g)}, ..., \alpha_J^{(g)}, \beta_1^{(g)}, ..., \beta_J^{(g)})$, the AUC can be approximated by linearly connecting points $(u_j^{(g)}, v_j^{(g)})$ on the curve corresponding to $(\alpha_j^{(g)}, \beta_j^{(g)})$. However, it is a little tricky to get the coordinate v given a specific coordinate u using the sample $(\alpha_1^{(g)}, ..., \alpha_J^{(g)}, \beta_1^{(g)}, ..., \beta_J^{(g)})$ because we can not ensure that there is $1 \leq j \leq J$ such that $(\alpha_j^{(g)}, \beta_j^{(g)})$ correspond to the exact value u. To address this issue, we can do a linear interpolation as follows. First find the nearest left and right simulated neighbors such that $u_l^{(g)} \leq u \leq u_r^{(g)}$ as well as their corresponding errors $(\alpha_l^{(g)}, \beta_l^{(g)})$ and $(\alpha_r^{(g)}, \beta_r^{(g)})$. The linear interpolation can be done by

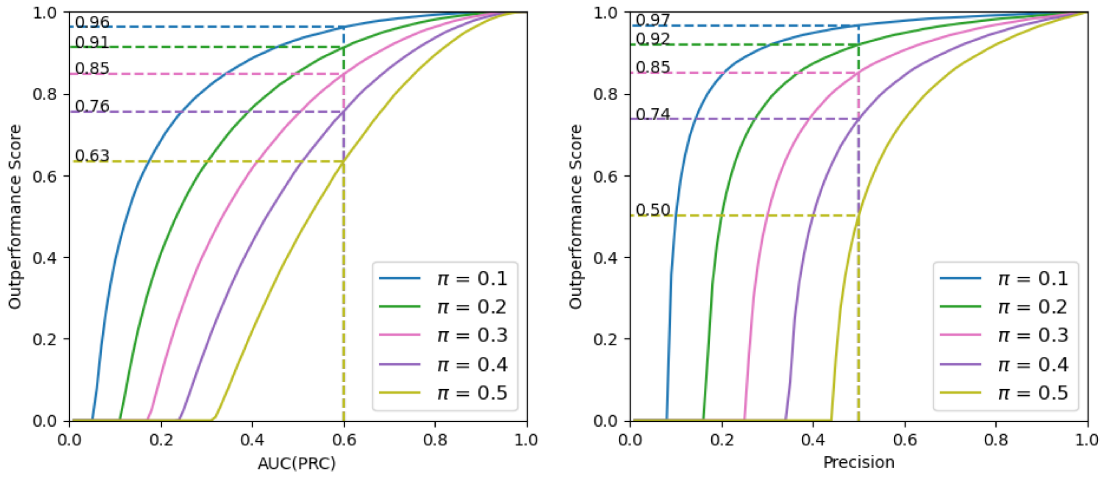
$$\hat{\alpha} = \alpha_l^{(g)} + (\alpha_r^{(g)} - \alpha_l^{(g)}) \frac{u - u_l^{(g)}}{u_r^{(g)} - u_l^{(g)}}, \text{ and } \hat{\beta} = \beta_l^{(g)} + (\beta_r^{(g)} - \beta_l^{(g)}) \frac{u - u_l^{(g)}}{u_r^{(g)} - x_l^{(g)}}. \quad (61)$$

Then, use the formula of v to estimate v from $\hat{\alpha}$ and $\hat{\beta}$.

Example 5 (Outperformance Score of PRC)**.** *One commonly used scoring metric for evaluating classifier performance on imbalanced data is PRC, which illustrates the trade-off between recall (x-axis) and precision (y-axis). The OPS function for the AUC of the PRC can be expressed as*

$$\text{OPS}_{AUC(PRC)}(\mu;\pi) = \Pr\{AUC < \mu | P = \pi\}, \tag{62}$$

and is shown in in Figure 5.4a for various imbalance rates π, ranging from 0.1 to 0.5. We



(a) OPS of AUC(PRC)

(b) OPS of Precision | Recall=0.8

Figure 5.4: The OPS functions of PRC: (a) AUC, and (b) a specific point given Recall=0.8, conditional on different imbalance rates.

can see that the OPS function, and therefore the distribution of AUC(PRC), is sensitive to the imbalance rate π. Specifically, for a given AUC value, such as 0.6, the outperformance score increases as the data becomes more imbalanced. Moreover, for a specific point (u, v) on the PRC, we can compute the outperformance score of $precision = v$ given $recall = u$ with the following OPS function

$$\text{OPS}_{precision|recall}(v,u;\pi) = \Pr\{precision < v | recall = u, P = \pi\}. \tag{63}$$

Similarly, as depicted in Figure 5.4b, for a given point (e.g., (0.8, 0.5)), the outperformance score increases as π decreases. This indicates that, in imbalanced data, achieving high recall does not necessarily require high precision, and lower precision may still reflect good performance.

In addition to the PRC, we also discuss the outperformance score of another widely used metric, the lift curve, in Appendix C.2.

5.6 A Universally Standardized Global Feature Importance

In the above, we introduced the outperformance score as a form of universal standardization for classification performance evaluation. Can a similar universal standardization be applied to the evaluation of feature importance? The answer is affirmative. By utilizing the SVA method in conjunction with the outperformance score, we can establish a universally standardized framework for feature importance.

As discussed in Section 2.5.2, for global SVAs, we need to design a value function $v_{f_S}(S) = \mathbf{M}(f_S, \mathcal{D}_p(X_S, Y))$ targeted to a specific performance metric $\mathbf{M}$. Essentially, the global SVA $\phi = (\phi_1, \ldots, \phi_d)$ attributes the achieved performance score $\mathbf{M}(f, \mathcal{D}(X, Y))$ into each feature X_i. Hence, the performance metric $\mathbf{M}$ and global SVA ϕ are measured in the same scale. Since the outperformance score is a universally standardized metric, selecting it as the target metric $\mathbf{M}$ implies that the corresponding global SVA will also function as a universally standardized metric for measuring feature importance. To align with the terminology introduced by I. Covert et al. (2020), we refer to the outperformance score based global SVA as *OPS-SAGE*[2]. Since OPS-SAGE is a universally standardized feature importance score, a reduction in its value signifies a corresponding decline in the importance of the associated feature, regardless of any data imbalance.

[2]SAGE: Shapley additive global importance, which is originally targeted to the cross entropy by I. Covert et al. (2020).

Property 6.2. *If a model f exhibits a performance drift, $\mathbf{M}(f, \mathcal{D}) - \mathbf{M}(f, \mathcal{D}')$, across two test sets $\mathcal{D}$ and $\mathcal{D}'$, then the SVA for this performance drift, denoted as $\phi_{\mathbf{M}(f,\mathcal{D})-\mathbf{M}(f,\mathcal{D}')}$, is equal to the SVA drift $\phi_{\mathbf{M}(f,\mathcal{D})} - \phi_{\mathbf{M}(f,\mathcal{D}')}$ corresponding to $\mathbf{M}$. In other words, the drift in the model's performance can be explained by the drift in the features' importance.*

The proof of Property 6.2 is straightforward from the linearity property of Shapley value (see Section 2.5.2). This property allows us to analyze model degradation (i.e., a decline in outperformance score) by simply examining the change in the OPS-SAGE scores. In particular, when the OPS-SAGE is an informative SVA (see Section 3.2), its variation reflects a change in the underlying information patterns.

5.7 Experiments

In this section, we demonstrate the use of our proposed outperformance score and OPS-SAGE method to evaluate and explain three types of classification performance: prediction performance (with a threshold), risk identification performance, and recommendation performance. The experiments are conducted on two datasets: the Heart Disease Health Dataset (Teboul, 2022) and the Loan Default Dataset (NIKHIL, 2019). However, in this section, we focus exclusively on analyzing the experimental results from the Heart Disease Health Dataset, while the results for the Loan Default Dataset are presented in Appendix D.

Heart Disease Dataset This dataset contains 253,680 responses from the health-related telephone survey conducted by the Behavioral Risk Factor Surveillance System in 2015. The goal of the classification task is to predict whether an individual has heart disease using 21 input variables. To exemplify the universal standardization property of the outperformance score and OPS-SAGE, we generate three distinct groups of individuals as testing sets for evaluation, as described below:

- (General group) This test set is randomly sampled from the original dataset, with approximately 9% of individuals labeled as having heart disease (i.e., positive class).
- (Old-age group) This test set is randomly sampled from individuals that are more than 70 years old, with around 19% of them having heart disease.
- (Hospital group) This test set is independently sampled in each class, with a class ratio of 3:7 (i.e., 30% of them having heart disease). This group is presumed to represent individuals visiting a hospital, as it has a relatively higher proportion of positive cases.

The summary of these test sets is given in Table 5.4. An XgBoost classifier is trained on the remaining data samples. The performance of the classifier is then evaluated on each of the three test groups, with detailed analysis provided in the subsequent sections.

Table 5.4: Summary of the test sets.

Info	General Group	Old-age Group	Hospital Group
Sample Size	9,000	9,043	9,206
Class Imblance Rates	0.091	0.19	0.3

5.7.1 Evaluate and Explain Prediction Performance with a Threshold

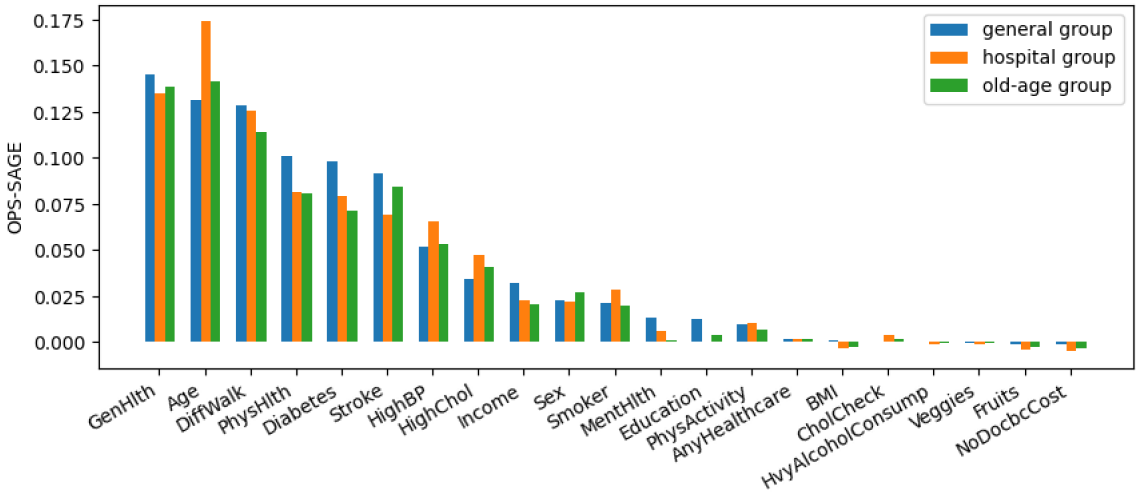
The Xgboost classifier can make predictions by selecting a threshold t, and its performance can be evaluated by labeling metrics. In this experiment, we select the threshold $t = 0.19$, which is determined by maximizing the f1_score of a validation set. To evaluate the prediction performance with this threshold, we consider the f1_score, MCC, and their outperformance scores, i.e., OPS(f1) and OPS(MCC). The values of these metrics are presented in Table 5.5. We can see that the Xgboost classifier returns similar results for both the f1_score (0.408 and 0.453) and MCC (0.348 and 0.3) on the general and old-age groups, while significantly higher values are observed for the hospital group, with an

Table 5.5: The Xgboost classifier's prediction performance given $t = 0.19$.

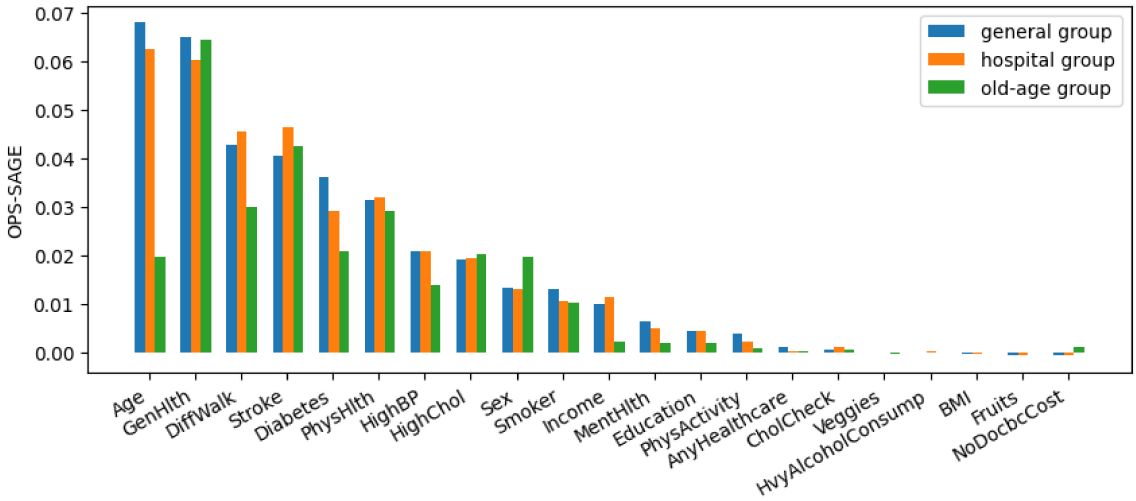
Test Sets	f1	OPS(f1)	MCC	OPS(MCC)
General Group ($\pi = 0.091$)	0.408	0.892	0.348	0.874
Old-age Group ($\pi = 0.19$)	0.453	0.799	0.3	0.779
Hospital Group ($\pi = 0.3$)	0.614	0.85	0.468	0.859

f1_score of 0.614 and MCC of 0.468. However, do these differences indicate model degradation on the general and old-age groups? Additionally, given that these metrics appear relatively low, does this imply that the quality of the XgBoost classifier is poor? These questions are difficult to answer based solely on the f1_score and MCC, as both are not universally standardized and are highly dependent on data imbalance or prevalence π. In contrast, the proposed outperformance score method provides a more reliable answer to these questions. The results in Table 5.5 show that, despite the lower f1_score and MCC, the outperformance scores for the general group are slightly higher than those for the hospital group, with OPS(f1): $0.892 > 0.85$, and OPS(MCC): $0.874 > 0.859$. Furthermore, the fact that these outperformance scores exceed 0.85 demonstrates that the XgBoost classifier outperforms at least 85% of all possible performance outcomes, indicating good quality. On the other hand, for the old-age group, even though the f1_score and MCC are similar to those for the general group, the outperformance scores are notably lower, with OPS(f1) of 0.799 and OPS(MCC) of 0.779. This suggests that the XgBoost classifier performs slightly worse when predicting heart disease in older individuals.

As discussed in Section 5.6, the achieved outperformance scores can be attributed to each feature using the proposed OPS-SAGE method (all OPS-SAGEs mentioned in the following are informative SVAs, as we select the OODIS surrogate model in equation 47 as the CRF estimator). The Xgboost classifier's OPS-SAGE scores w.r.t. OPS(f1) and OPS(MCC) are presented in Figure 5.5. We can see that these OPS-SAGE scores are measured on the same scale, allowing for direct comparison across all three test groups, despite variations in class imbalance. Specifically, in Figure 5.5a, we observe that the three



(a) OPS-SAGE w.r.t. OPS(f1) given $t = 0.19$



(b) OPS-SAGE w.r.t. OPS(MCC) given $t = 0.19$

Figure 5.5: The Xgboost classifier's OPS-SAGE w.r.t. (a) OPS(f1), and (b) OPS(MCC), given $t = 0.19$

groups exhibit similar risk patterns w.r.t. OPS(f1), where features such as `GenHlth`, `Age`, `Diffwalk`, `Physhlth`, `Diabetes` and `Stroke` are the primary risk factors. However, for the old-age group, the importance scores for features like `Diffwalk`, `PhysHlth`, `Diabetes`, and `Income` decrease to some extent. According to Property 6.2, the drift in the importance of these features contributes to the model's performance degradation (i.e., a reduction in OPS(f1)) for the old-age group. Moreover, Figure 5.5 reveals that the general and hospital

groups display similar risk patterns w.r.t. OPS(MCC), while the old-age group exhibits a noticeably distinct risk pattern. This drift in risk pattern leads to a reduction in OPS(MCC) for the old-age group. Particularly, the importance of the feature `Age` decreases significantly, accounting for nearly half of the performance drift.

5.7.2 Evaluate and Explain Risk Identification Performance

A robust risk identification classifier should be able to achieve a possibly high precision while maintaining a required risk coverage rate (i.e., recall). The precision-recall curve (PRC) is commonly used to evaluate a classifier's performance in risk identification. In this experiment, we assess two types of risk identification performance: overall performance across the entire recall range and specific performance at a fixed recall of 0.9. The following metrics are used for evaluation:

Metrics for overall risk identification performance:

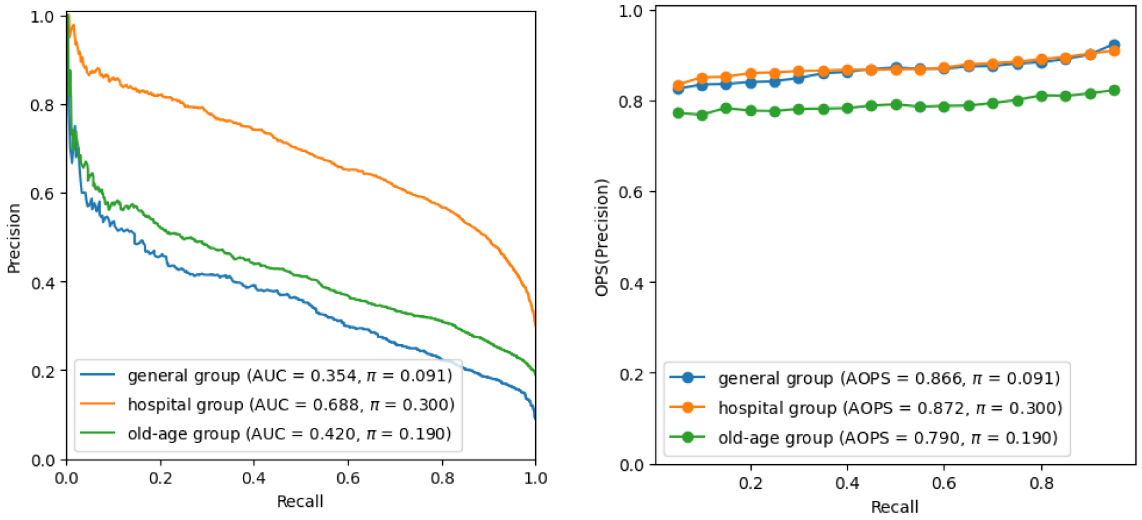
- **AUC(PRC)**: the area under the PRC, often interpreted as the average precision across the recall range.
- **OPS(AUC)**: the outperformance score of AUC(PRC).
- **AOPS(Precision)**: the average outperformance score of precision across the recall range.

Metrics for risk identification performance at recall=0.9:

- **Precision**: the fraction of true positives among the predicted positives.
- **OPS(Precision)**: the outperformance score of precision.

The values of these metrics are presented in Table 5.6, with the PRC shown in Figure 5.6a. Additionally, we also introduce the OPS(Precision)-Recall curve (OPRC) in Figure 5.6b, which visualizes the OPS(Precision) at 20 points across the recall range. The

OPRC can be viewed as a standardized version of the PRC. The results indicate that, based



(a) PRC

(b) OPRC

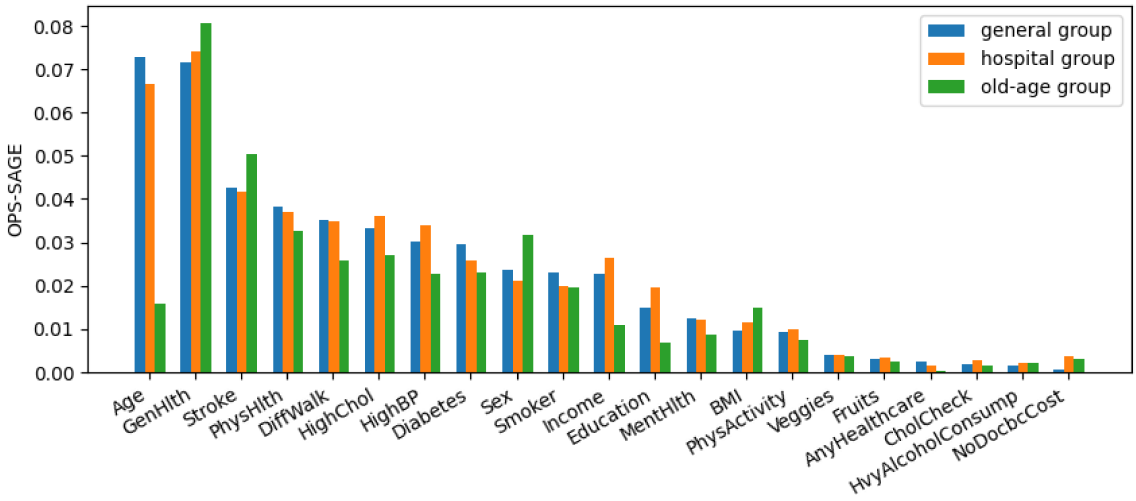
Figure 5.6: Visualize the Xgboost classifier's overall risk identification performance with (a) Precision-Recall curve (PRC), and (b) OPS(Precision)-Recall curve (OPRC).

Table 5.6: The Xgboost classifier's risk identification performance.

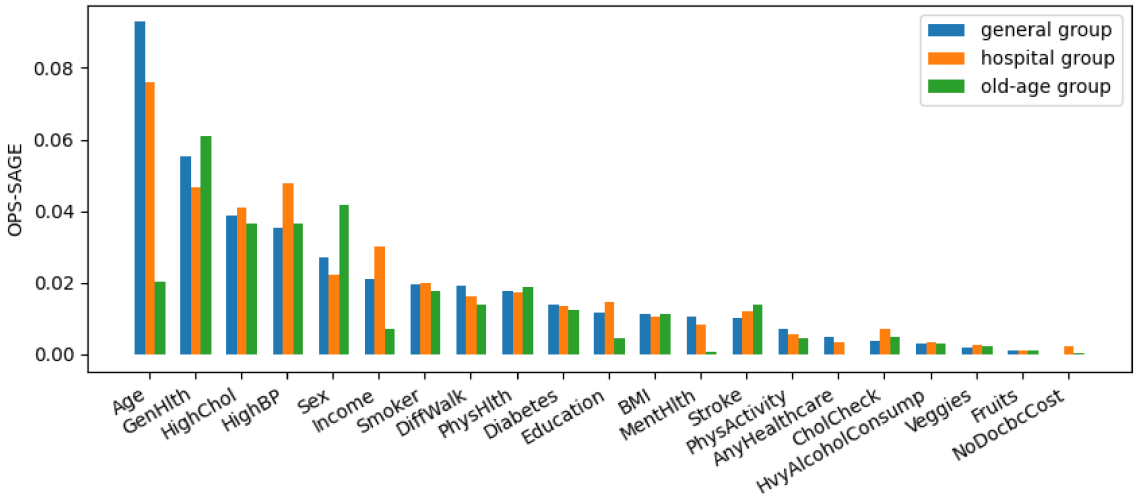
	General Group ($\pi = 0.091$)	**Old-age Group** ($\pi = 0.19$)	**Hospital Group** ($\pi = 0.3$)
Overall risk identification performance			
AUC(PRC)	0.354	0.42	0.688
OPS(AUC)	0.869	0.797	0.909
AOPS(Precision)	0.866	0.79	0.872
Risk identification performance at Recall=0.9			
Precision	0.183	0.264	0.495
OPS(Precision)	0.901	0.815	0.902

on the PRC, the XgBoost classifier performs best on the hospital group and worst on the general group, regardless of whether the overall performance or the specific performance at recall=0.9 is considered. Clearly, both the AUC(PRC) and the precision (at recall=0.9) decrease as the test groups become more imbalanced. However, when applying the proposed OPS function to standardize the PRC, the values of OPS(AUC), AOPS(Precision),

and OPS(Precision) (at recall=0.9) demonstrate that the XgBoost classifier performs similarly well in identifying heart disease in both the general and hospital groups, but shows relatively poorer performance in the old-age group.



(a) OPS-SAGE w.r.t. OPS(AUC) of PRC



(b) OPS-SAGE w.r.t. OPS(Precision) at Recall=0.9

Figure 5.7: The Xgboost classifier's OPS-SAGE w.r.t. (a) OPS(AUC) of PRC, and (b) OPS(Precision) at Recall=0.9.

The conclusion above can be explained through the proposed OPS-SAGE method. As illustrated in Figure 5.7a and 5.7b, the risk patterns for the general and hospital groups

appear similar, while they differ noticeably for the old-age group. In particular, the importance of feature `Age` is greatly diminished in the old-age group. This reduction occurs because all individuals in this group are at high-risk ages, making `Age` a less informative factor in identifying heart disease risk within this cohort. Furthermore, risk patterns may vary depending on the specific targeted metric. For example, the feature `Stroke` is highly informative for the OPS(AUC) of PRC (as shown in Figure 5.7a), yet it is far less informative for the OPS(Precision) at recall=0.9 (as shown in Figure 5.7b). This discrepancy arises because `Stroke` is overall a strong indicator of heart disease (i.e., $\mathbf{Pr}\{heart\ disease|stroke\}$ is significant), but only a small fraction of heart-diseased individuals have a stroke (i.e., $\mathbf{Pr}\{stroke|heart\ disease\}$ is low). As a result, the feature `Stroke` becomes less critical for achieving high recall like 0.9.

From Table 5.6, we also observe that the precision is very low at recall=0.9 across all three groups. In practice, risk identification is often conducted on highly imbalanced datasets, and to capture the majority of the true risks, a significant number of false positives are typically generated. However, this does not necessarily indicate poor risk identification performance. As long as the selected model achieves a high outperformance score, it can still be deemed qualified. For instance, to identify 90% of heart disease cases in the general group, the XgBoost classifier yields a low precision of 0.183, yet achieves a high OPS(Precision) of 0.901, reflecting excellent performance.

5.7.3 Evaluate and Explain Recommendation Performance

In recommendation tasks, we typically care more about the precision of the recommended items from the dataset, without the need for a recall threshold such as 0.9. The lift curve is often employed to measure how much better the selected classifier performs compared to the random classifier, whose precision always equals the prevalence π. We also

consider two types of recommendation performance: the overall performance for recommending all possible percentages of items from the dataset, and the specific performance when recommending a fixed number of 500 items (denoted as K=500, for example, we need to recommend 500 individuals to receive specific treatment for heart disease). The following metrics are considered for evaluation:

Metric for overall recommendation performance:

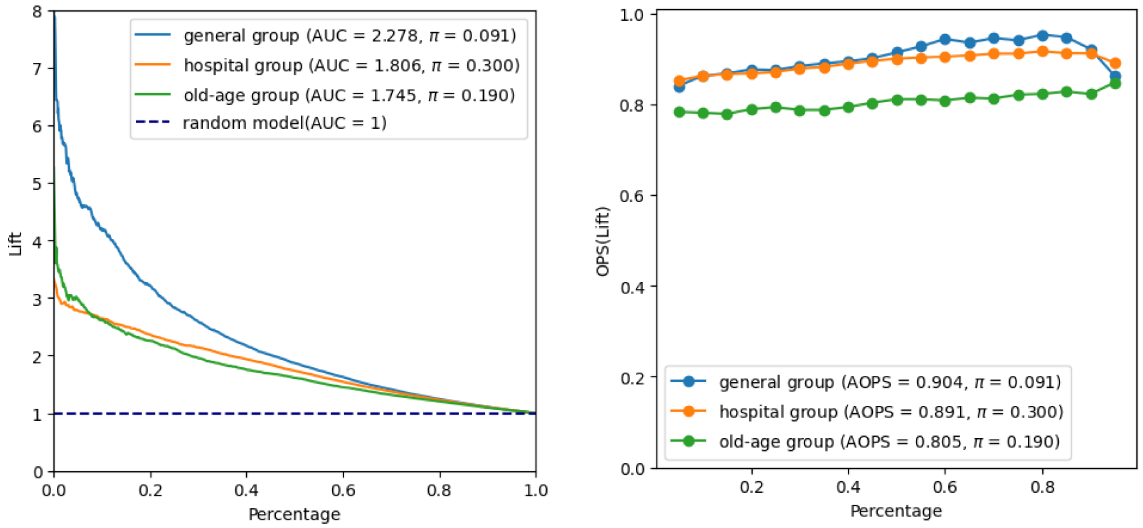
- **AUC(Lift curve)**: the area under the lift curve, which is often interpreted as the average lift across the entire percentage range.
- **NAUC(Lift curve)**: normalized AUC (as shown in Equation 54) of the lift curve, quantifying how close the classifier is to the ideal classifier.
- **OPS(AUC or NAUC)**: the outperformance score for AUC(Lift curve) or NAUC(Lift curve).
- **AOPS(Lift or Precision)**: the average outperformance score of lift or precision across the entire percentage range.

Metric for recommendation performance at K=500:

- **Precision**: the fraction of true positive items among the total recommended items.
- **Lift**: defined as $\frac{\text{precision}}{\pi}$, measuring how many times better the selected classifier is compared to the random classifier.
- **OPS(Lift or Precision)**: the outperformance score of lift or precision.

It is important to note that, according to Property 6.1, OPS(AUC) is equivalent to OPS(NAUC) for any scoring metric curves, and OPS(Lift) is equivalent to OPS(Precision) when recommending a fixed number or percentage of items. The results of these metrics are summarized in Table 5.7, with the lift curve illustrated in Figure 5.8a. Similar to the OPRC,

we also introduce the OPS(lift) curve in Figure 5.8b, which visualizes the OPS(Lift) or OPS(Precision) at 20 points across the percentage range. The OPS(lift) curve can be viewed as a standardized version of the lift curve.



(a) Lift curve

(b) OPS(lift) curve

Figure 5.8: Visualize the Xgboost classifier's overall recommendation performance with (a) lift curve, and (b) OPS(lift) curve.

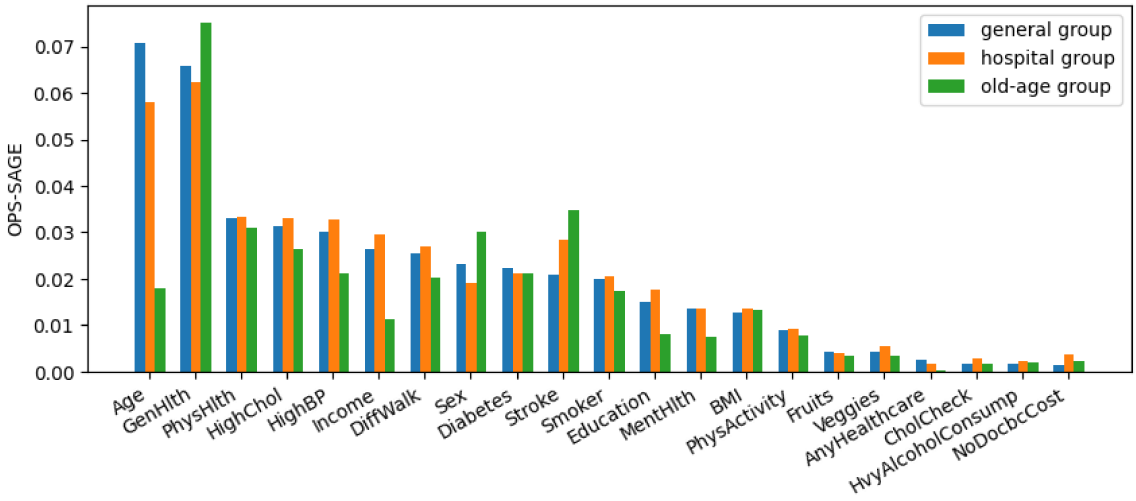
Table 5.7: The Xgboost classifier's recommendation performance.

	General Group ($\pi = 0.091$)	**Old-age Group** ($\pi = 0.19$)	**Hospital Group** ($\pi = 0.3$)
Overall recommendation performance			
AUC(Lift curve)	2.278	1.745	1.806
NAUC(Lift curve)	0.67	0.656	0.82
OPS(AUC or NAUC)	0.915	0.841	0.929
AOPS(Lift or Precision)	0.904	0.805	0.891
Recommendation performance at K=500			
Precision	0.418	0.558	0.83
Lift	4.61	2.937	2.766
OPS(Lift or Precision)	0.84	0.782	0.852

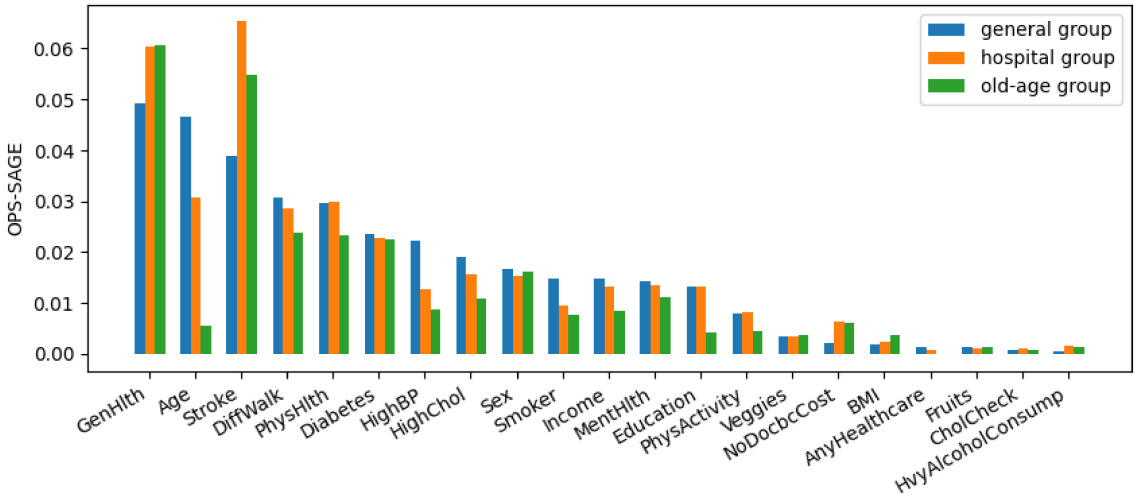
To begin, we evaluate the overall recommendation performance. The results of the AUC(Lift curve) indicate that the Xgboost classifier performs best on the general group,

with similar performance across the old-age and hospital groups. However, the NAUC(Lift curve) results show that the classifier performs best on the hospital group, with similar performance on the general and old-age groups. This discrepancy between the two metrics can be confusing. A similar issue arises when evaluating the specific recommendation performance at K=500 using precision and lift. However, this confusion can be resolved by standardizing these metrics with the proposed OPS function. Once standardized, AUC and NAUC share the same outperformance score, and so do precision and lift. The values of OPS(AUC or NAUC) and OPS(Lift or Precision) reveal that the Xgboost classifier performs similarly well on the general and hospital groups, with slightly lower performance on the old-age group. This conclusion aligns with the findings in Section 5.7.2 regarding risk identification performance.

Similarly, the proposed OPS-SAGE method can be employed to attribute recommendation performance to features, as demonstrated in Figure 5.9. Notably, there is a significant reduction in the importance of the feature `Age` for the old-age group, reflecting the same rationale discussed in Section 5.7.2. Moreover, the general and hospital groups exhibit similar informative patterns regarding overall recommendation performance, but there is a slight divergence in patterns for the recommendation performance at K=500. Specifically, for the hospital group, the Xgboost classifier relies more on features like `Stroke` and `GenHlth`, and less on features such as `Age` and `HighBP`, when recommending 500 individuals for heart disease treatment compared to the general group. Furthermore, across all three groups, the feature `Stroke` plays a more important role in recommending 500 individuals than in the overall recommendation performance. This is logical, as individuals with a history of stroke are more likely to have heart disease and should be prioritized when only a small percentage of people are recommended for treatment. These observations further validate that the OPS-SAGE method provides reasonable and reliable explanations.



(a) OPS-SAGE w.r.t. OPS(AUC) of Lift curve



(b) OPS-SAGE w.r.t. OPS(lift) given K=500

Figure 5.9: The Xgboost classifier's OPS-SAGE w.r.t. (a) OPS(AUC) of Lift curve, and (b) OPS(Lift or Precision) at K=500.

5.8 Conclusions

In this chapter, we introduced a novel standardization method for confusion matrix-based classification performance metrics, termed *outperformance scores*. These scores are interpreted as the probability that a classifier's observed performance outperforms a

distribution of potential performances, given the imbalance rates. The proposed outperformance scores allow practitioners to standardize their preferred performance metrics, facilitating consistent evaluation and comparison of classifiers across diverse datasets. Additionally, we proposed a uniformly standardized global feature importance method, *OPS-SAGE*, which attributes the outperformance score to individual features. OPS-SAGE enables practitioners to uncover and compare informative patterns underlying different datasets. Our experiments demonstrate the robustness of the proposed methodology in handling datasets with varying imbalance rates, making it a valuable tool for monitoring and explaining model performance drifts across different data groups. Furthermore, since the outperformance scores are directly comparable across labels with varying imbalance rates, they can be (weighted) averaged to evaluate multi-label classifiers effectively.

One limitation of our method is that outperformance scores for labeling and scoring metrics rely on different distributions of potential performances. Thus, it may not be appropriate to compare the outperformance score applied on a labeling metric to the outperformance score applied on a scoring metric. Future research can try to look deeper into this issue.

Chapter 6

Discussion

This thesis has explored novel methods to address key challenges of Shapley value attribution (SVA), with a focus on informative SVA that is true to the data. We tackled two central issues in generating informative explanations: (1) the error-prone nature of SVAs due to reliance on estimated distributions from sparse datasets and untenable distributional assumptions, and (2) the challenges caused by class imbalance in global SVA methods.

To address the first challenge, in Chapter 3, we developed an error theoretical analysis framework to decompose explanation errors into observation bias and structural bias. This decomposition enabled a deeper understanding of how existing SVA methods may become under-informative due to distributional assumptions, or over-informative due to data sparsity. In Chapter 4, our proposed refinement methods, combining out-of-distribution (OOD) detection and importance sampling techniques, significantly reduced these biases, offering more robust explanations across both local and global SVA settings.

For the second challenge of class imbalance, in Chapter 5, we introduced the *outperformance score (OPS)* function to standardize confusion matrix-based performance metrics. Building on this function, we further proposed a novel standardized global SVA, named the OPS-SAGE. These contributions provide a uniform interpretation of model performance across varying levels of class imbalance, enhancing the robustness and interpretability of

global SVA explanations in real-world scenarios.

Our experiments demonstrated that the proposed methods improved the reliability and informativeness of SVA explanations, particularly in high-dimensional, sparse, and imbalanced data contexts. The proposed error framework, refinement techniques, and OPS-SAGE showed substantial improvements over traditional SVA methods, contributing to the broader goal of making machine learning models more transparent and reliable for practical applications in fields such as healthcare and finance.

While this work advances the XAI field by addressing several core challenges in SVA methods, there remains room for future research. We highlight some promising directions as follows:

- (Causal Inference in SVA) In this work, we primarily focus on informative SVA methods, but there is growing interest in developing causal SVA techniques that go beyond informational dependency to identify true cause-and-effect relationships between features and model outcomes. Integrating principles from causal inference, such as counterfactual analysis, into the existing SVA framework could lead to more robust, causally informative explanations. Such methods would be particularly useful in high-stakes domains, such as healthcare, where understanding the underlying causal mechanisms is essential for making informed decisions.

- (SVA in Complex Data Structures) Most of the current SVA methods focus on single-modal data types, such as tabular or image data. However, many real-world problems involve various data, combining text, images, and structured data, which introduces new challenges for SVA methods. Future work could explore how SVAs can be extended or adapted in these complex data structures, ensuring that attributions remain meaningful and interpretable across different varieties of data.

- (Applications in Scientific Discovery) While we have demonstrated the utility of SVA in scientific discovery, such as identifying risk factors for diseases, more work

remains to be done in this area. Future research could focus on refining the application of SVAs in specific domains, such as genomics, neuroscience, and climate science, where understanding complex informative patterns among variables is crucial for new discoveries. Furthermore, interdisciplinary collaborations between AI researchers and domain experts could help tailor SVA methods to the specific needs of different scientific fields.

Appendix A

The Estimation of Shapley Values

A.1 Monte-Carlo Sampling Algorithm

The Monte-Carlo Sampling Algorithm approximates the effect of removing a feature from the model by integrating over samples from the training data (Molnar, 2020; Štrumbelj & Kononenko, 2014). Specifically, let x be the explicand for explaining, x^* be a random data point from the training data, $S_{-i} \subseteq D \setminus \{i\}$ be a random coalition excluding feature i, $\bar{S}_{-i} = \{D \setminus \{i\}\} \setminus S_{-i}$ be the set of missing features excluding feature i. Then, we can construct two mixed instance $x_{+i} = [x_{S_{-i}}, x_i, x^*_{\bar{S}_{-i}}]$, and $x_{-i} = [x_{S_{-i}}, x^*_i, x^*_{\bar{S}_{-i}}]$ by replacing the missing feature values of x with the corresponding ones in x^*. By using this method, we can draw a sample set $\{x^m_{+i}, x^m_{-i}\}_{m=1}^M$ to approximate the Shapley value as

$$\hat{\phi}_i = \frac{1}{M} \sum_{m=1}^{M} \left(f(x^m_{+i}) - f(x^m_{-i}) \right). \tag{64}$$

The detailed steps of the algorithm are as following:

- Input: number of samples M, instance of interest x, feature index i, training data X, and model f

- Initialize $\hat{\phi}_i = 0$
- for $m = 1$ to M do:
  - (1) Draw random instance x^* from X
  - (2) Choose a random permutation o of $D = \{1, \ldots, d\}$
  - (3) order x: $x_o = (x_{o[1]}, \ldots, x_i, \ldots, x_{o[d]})$
  - (4) order x^*: $x^*_o = (x^*_{o[1]}, \ldots, x^*_i, \ldots, x^*_{o[d]})$
  - (5) construct two mixed instance
    - $x_{+i} = (x_{o[1]}, \ldots, x_i, \ldots, x^*_{o[d]})$
    - $x_{-i} = (x_{o[1]}, \ldots, x^*_i, \ldots, x^*_{o[d]})$
  - (6) compute $\Delta = f(x_{+i}) - f(x_{-i})$
  - (7) update $\hat{\phi}_i = \hat{\phi}_i + \Delta$
- Return $\frac{\hat{\phi}_i}{M}$

Note that we have to repeat the above sampling estimate d times for each feature.

A.2 Estimation via Linear Regression

Using the Lagrangian method, the solution of the optimization problem in (15) can be directly derived as the following closed-form formula:

$$\beta^* = A^{-1}\left(b - \mathbf{1}\frac{\mathbf{1}^T A^{-1} b - v(\mathbf{1}) + v(\mathbf{0})}{\mathbf{1}^T A^{-1}\mathbf{1}}\right), \tag{65}$$

where

$$A = \mathbb{E}[ZZ^T] \quad \text{and} \quad b = \mathbb{E}\left[Z\left(v(Z) - v(\mathbf{0})\right)\right].$$

We have that the Shapley value $\phi_i(u) = \beta_i^*$. However, unfortunately, this formula cannot be directly calculated in practice without evaluating $v(z)$ for all 2^d coalitions z.

A.2.1 KernelSHAP

Rather than considering all 2^d coalitions, KernelSHAP (Lundberg & Lee, 2017) proposes to subsample a set of coalitions, and then optimize an approximate objective. Specifically, KernelSHAP first samples n i.i.d. coalitions $(z_1, \ldots, z_n)$ from the distribution $p(Z)$, then estimates Shapley values of cooperative game u by solving the following problem:

$$\begin{aligned} &\text{argmin}_\beta \frac{1}{n} \sum_{i=1}^{n} \left(v(\mathbf{0}) + z_i^T \beta - v(z_i)\right)^2 \\ &\text{s.t.} \quad \mathbf{1}^T \beta = v(\mathbf{1}) - v(\mathbf{0}). \end{aligned} \tag{66}$$

Similarly, with the Lagrangian method, the solution $\hat{\beta}_n$ is straightforward as the following

$$\hat{\beta}_n = \hat{A}_n{}^{-1} \left(\hat{b}_n - \mathbf{1} \frac{\mathbf{1}^T \hat{A}_n^{-1} \hat{b}_n - v(\mathbf{1}) + v(\mathbf{0})}{\mathbf{1}^T \hat{A}_n^{-1} \mathbf{1}} \right), \tag{67}$$

where

$$\hat{A}_n = \frac{1}{n} \sum_{i=1}^{n} z_i z_i^T \quad \text{and} \quad \hat{b}_n = \frac{1}{n} \sum_{i=1}^{n} z_i \left(v(z_i) - v(\mathbf{0})\right).$$

First of all, we can conclude that $\hat{\beta}_n$ is a consistent estimator that converges to the correct Shapley values β^* given a sufficiently large number of samples, i.e.,

$$\lim_{n \to \infty} \hat{\beta}_n = \beta^*.$$

This can be proved by the strong law of large numbers, which implies that

$$\lim_{n \to \infty} \hat{A}_n = A \quad \text{and} \quad \lim_{n \to \infty} \hat{b}_n = b.$$

Nevertheless, $\hat{\beta}_n$ may not be an unbiased estimator, even though both $\hat{A}_n$ and $\hat{b}_n$ are unbiased. The reason is that it is difficult to characterize the interaction between $\hat{A}_n$ and $\hat{b}_n$ in Equation (67), such as $\mathbb{E}[\hat{A}_n^{-1}\hat{b}_n]$.

A.2.2 Unbiased KernelSHAP

As shown by I. Covert and Lee (2020), the $d \times d$ matrix A can be exactly calculated with the known distribution $p(Z)$. Specifically, the diagonal entries A_{ii} can be computed by $A_{ii} = \mathbb{E}[Z_i^2] = \mathbb{E}[Z_i] = p(Z_i = 1)$, and the off-diagonal entries A_{ij} can be computed by $A_{ij} = \mathbb{E}[Z_i Z_j] = p(Z_i = Z_j = 1)$. Therefore, instead of the estimator $\hat{A}_n$, we can directly use A's exact form and approximate β^* by estimating only b. Using the exact value of $\mathbb{E}[Z]$, b can be estimated by

$$\bar{b}_n = \frac{1}{n}\sum_{i=1}^{n} z_i v(z_i) - \mathbb{E}[Z]v(\mathbf{0}).$$

Replacing b with $\bar{b}_n$ in Equation (65), we have an alternative estimator for β^*:

$$\bar{\beta}_n = A^{-1}\left(\bar{b}_n - \mathbf{1}\frac{\mathbf{1}^T A^{-1}\bar{b}_n - v(\mathbf{1}) + v(\mathbf{0})}{\mathbf{1}^T A^{-1}\mathbf{1}}\right). \tag{68}$$

Same as $\hat{\beta}_n$, this is obviously a consistent estimator, i.e., $\lim_{n\to\infty}\bar{\beta}_n = \beta^*$. However, unlike $\hat{\beta}_n$, the estimator $\bar{\beta}_n$ is also unbiased, i.e., $\mathbb{E}[\bar{\beta}_n] = \beta^*$, because of its linear dependence on an unbiased estimator $\bar{b}_n$. Hence, $\bar{\beta}_n$ is referred to as *unbiased KernelSHAP*.

In addition, by the *central limit theorem (CLT)*, it is easy to find that $\bar{b}_n$ converges in distribution to a multivariate Gaussian,

$$\bar{b}_n \xrightarrow{D} \mathcal{N}(b, \frac{\Sigma_{\bar{b}}}{n}), \tag{69}$$

where $\Sigma_{\bar{b}} = \text{Cov}(Zv(Z))$. Since $\bar{\beta}_n$ is a linear function of $\bar{b}_n$, it also converges to a multivariate Gaussian,

$$\bar{\beta}_n \xrightarrow{D} \mathcal{N}(\beta^*, \frac{\Sigma_{\bar{\beta}}}{n}), \tag{70}$$

where the covariance $\Sigma_{\bar{\beta}}$ is given by

$$\Sigma_{\bar{\beta}} = C\Sigma_{\bar{b}}C^T \tag{71}$$

$$C = A^{-1} - \frac{A^{-1}\mathbf{1}\mathbf{1}^T A^{-1}}{\mathbf{1}^T A^{-1}\mathbf{1}}. \tag{72}$$

That suggests that the unbiased KernelSHAP $\bar{\beta}_n$ has a variance that reduces at a rate of $O(\frac{1}{n})$.

A.2.3 Convergence Detection

Even though the original KernelSHAP $\hat{\beta}_n$ is difficult to analyze theoretically, I. Covert and Lee (2020) show empirically that its bias is negligible and its variance is lower than the unbiased KernelSHAP $\bar{\beta}_n$. That suggests that the $O(\frac{1}{n})$ rate should also hold for $\hat{\beta}_n$ in practice. This property is difficult to prove, but it can be used for convergence detection in practice.

The KernelSHAP's covariance, for both $\Sigma_{\hat{\beta}_n}$ and $\Sigma_{\bar{\beta}_n}$, can be empirically estimated using an online algorithm. Specifically, for any n, we can select an intermediate value $m << n$, and calculate multiple independent estimates $\hat{\beta}_m$ or $\bar{\beta}_m$ while running the sampling algorithm, then the covariance can be estimated as

$$\hat{\Sigma}_{\hat{\beta}_n} = \frac{m}{n}\text{Cov}(\hat{\beta}_m) \quad \text{or} \quad \hat{\Sigma}_{\bar{\beta}_n} = \frac{m}{n}\text{Cov}(\bar{\beta}_m).$$

Finally, the algorithm can be considered converged when the largest standard deviation is a sufficiently small proportion t (e.g., $t = 0.01$) of the range in the estimated Shapley values.

That is, for the original KernelSHAP, we stop at step n when

$$\max_i \sqrt{\frac{(\hat{\Sigma}_{\hat{\beta}_n})_{ii}}{n}} < t \left(\max_i (\hat{\beta}_n)_i - \min_i (\hat{\beta}_n)_i \right).$$

A.3 Projected Stochastic Gradient Algorithm

Similar to all other linear regression problems, the objective function in (15) can also be solved using the *stochastic gradient descent (SGD)* algorithm. Moreover, the learning algorithm has to be implemented within certain convex sets caused by the constraints. More details are given in the following.

A.3.1 Projected SGD Algorithm

To avoid a large gradient norm, we assume that the model outputs and the norm of Shapley values are both upper bounded, i.e., for all $z \in p(Z)$, $|v(z)| \leq C_1$ and $||\phi(u)|| \leq C_2$ with $C_1, C_2 > 0$ which can be large enough. Let's denote function $h(\beta, Z) = v(\mathbf{0}) + Z^T\beta - v(Z)$, and convex sets $K_1 = \{\beta : \mathbf{1}^T\beta = v(\mathbf{1}) - v(\mathbf{0})\}$, $K_2 = \{\beta : ||\beta|| \leq C_2\}$, and $K = K_1 \cap K_2$. The objective is to find a unique solution β^* on K to minimize $\mathbb{E}_{p(Z)} h(\beta, Z)$. To solve it, the projected SGD algorithm, at each iteration t, follows the steps:

$$\begin{aligned} z_t &\sim p(Z), \\ \beta_t &= \text{Proj}_K \left(\beta_{t-1} - \gamma \nabla h(\beta_{t-1}, z_t) \right), \end{aligned}$$

where Proj_K is the orthogonal projection on K, and γ is the learning rate. As shown in (Simon & Vincent, 2020), Dykstra's algorithm can be used to find the orthogonal projection Proj_K.

To ensure a constant upper bound of the stochastic gradient norm $\mathbb{E}[||\nabla h(\beta_t, Z)||^2]$,

Simon and Vincent (2020) suggests that, instead of $p(Z)$, we can sample coalition z_t from a distribution $q(Z)$, which is defined as

$$q(Z) \propto p(Z)\sqrt{|Z|}(C_1 + C_2\sqrt{|Z|}),$$

where $|Z|$ is the size of the coalition. Then, using *importance sampling rate*, the projected SGD algorithm becomes:

$$\begin{aligned} z_t &\sim q(Z), \\ \beta_t &= \text{Proj}_K\left(\beta_{t-1} - \gamma\frac{p(z_t)}{q(z_t)}\nabla h(\beta_{t-1}, z_t)\right). \end{aligned}$$

With this change, we can find an upper bound for the stochastic gradient norm:

$$\mathbb{E}_{q(Z)}\left[||\frac{p(Z)}{q(Z)}\nabla h(\beta_t, Z)||^2\right] \leq B^2 = 4\left[\sum_{l=1}^{d-1}\frac{d-1}{\sqrt{l}(d-l)}(C_1 + C_2\sqrt{l})\right].$$

A.3.2 Convergence Rate

As analyzed by Simon and Vincent (2020), the convergence rate of the projected SGD algorithm depends on the chosen learning rate. In details, denoting T as the total number of iterations, and $\mu = 1 - \frac{1}{d}$, then,

- with an inverse decreasing learning rate $\gamma_t = \frac{2}{\mu(t+1)}$ and $\bar{\beta}_T = \frac{2}{(T+1)(T+2)}\sum_{t=0}^{T}(t+1)\beta_t$, the convergence rate is

$$\mathbb{E}\left[||\bar{\beta}_T - \beta^*||^2\right] \leq \frac{4B^2}{\mu^2 T} = O(\frac{1}{T});$$

- with a square root decreasing learning rate $\gamma_t = \frac{2C_2}{B\sqrt{t}}$ and $\bar{\beta}_T = \frac{1}{T}\sum_{t=0}^{T}\beta_t$, the

convergence rate is

$$\mathbb{E}\left[||\bar{\beta}_T - \beta^*||^2\right] \leq \frac{4BC_2}{\mu\sqrt{T}} = O(\frac{1}{\sqrt{T}});$$

- with a constant learning rate $\gamma < \frac{1}{\mu} = \frac{d}{d-1}$, the convergence rate is

$$\mathbb{E}\left[||\beta_T - \beta^*||^2\right] \leq (1-\gamma\mu)^T||\beta_0 - \beta^*||^2 + \frac{\gamma B^2}{\mu} = O(\rho^T) + O(\gamma),$$

where $\rho = 1 - \gamma\mu$.

Appendix B

Gaussian Removal Function

If we assume that the input features follow a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ, i.e., $X \sim \mathcal{N}(\mu, \Sigma)$, then an arbitrary feature sub-vector also follows a multivariate Gaussian $X_S \sim \mathcal{N}(\mu_S, \Sigma_{SS})$. In more details, given a subset $S \in \mathcal{P}([d])$, we can write $p(X) = p(X_S, X_{\bar{S}}) \approx \mathcal{N}(\mu, \Sigma)$ with $\mu = (\mu_S, \mu_{\bar{S}})^T$ and $\Sigma = \begin{bmatrix} \Sigma_{SS} & \Sigma_{S\bar{S}} \\ \Sigma_{\bar{S}S} & \Sigma_{\bar{S}\bar{S}} \end{bmatrix}$, and $p(X_S) \approx \mathcal{N}(\mu_S, \Sigma_{SS})$. Then, the conditional distribution $p(X_{\bar{S}}|X_S = x_S)$ can be computed as

$$p(X_{\bar{S}}|X_S = x_S) = \frac{p(X_S = x_S, X_{\bar{S}})}{p(X_s = x_s)} \approx \mathcal{N}(\mu_{\bar{S}|S}, \Sigma_{\bar{S}|S}) \tag{73}$$

with

$$\mu_{\bar{S}|S} = \mu_{\bar{S}} + \Sigma_{\bar{S}S}\Sigma_{SS}^{-1}(x_S - \mu_S) \tag{74}$$

and

$$\Sigma_{\bar{S}|S} = \Sigma_{\bar{S}\bar{S}} - \Sigma_{\bar{S}S}\Sigma_{SS}^{-1}\Sigma_{S\bar{S}}. \tag{75}$$

The mean vector μ and covariance Σ can be estimated by the samples from the explaining set $\mathcal{D}_p(X)$. Once we have the above multivariate Gaussian estimate of $p(X_{\bar{S}}|X_S = x_S)$,

we can sample from it, and empirically estimate f_S as

$$f_S(x_S) \approx \mathbb{E}_{x'_{\bar{S}} \sim \mathcal{N}(\mu_{\bar{S}|S}, \Sigma_{\bar{S}|S})}[f(x_S, x'_{\bar{S}})].$$

Appendix C

Additional Examples of Outperformance Score

C.1 Outperformance Score of MCC

Different from f1_score which cares more about the performance of the positive class, Matthews Correlation Coefficient (MCC) is a performance metric trying to balance both classes. The formula of MCC can be written as

$$MCC(\alpha, \beta, \pi) = \frac{1 - \alpha - \beta}{\sqrt{(1 - \alpha + \frac{\pi}{1-\pi}\beta)(1 - \beta + \frac{\pi}{1-\pi}\alpha)}}. \tag{76}$$

It is complicated to compute the closed-form formula of the outperformance function of MCC, but it can be approximated by the trapezoidal rule or Monte Carlo algorithm. MCC has a symmetric domain on $[-1, 1]$, and its baseline is always 0, which is achieved by the trivial majority classifier (TMC) that always predicts the majority class. Because of the symmetric domain, the TMC always returns an outperformance score of 0 for MCC, i.e.,

$$\text{OPS}_{MCC}(0; \pi) = Pr\{MCC < 0 | P = \pi\} = 0.5.$$

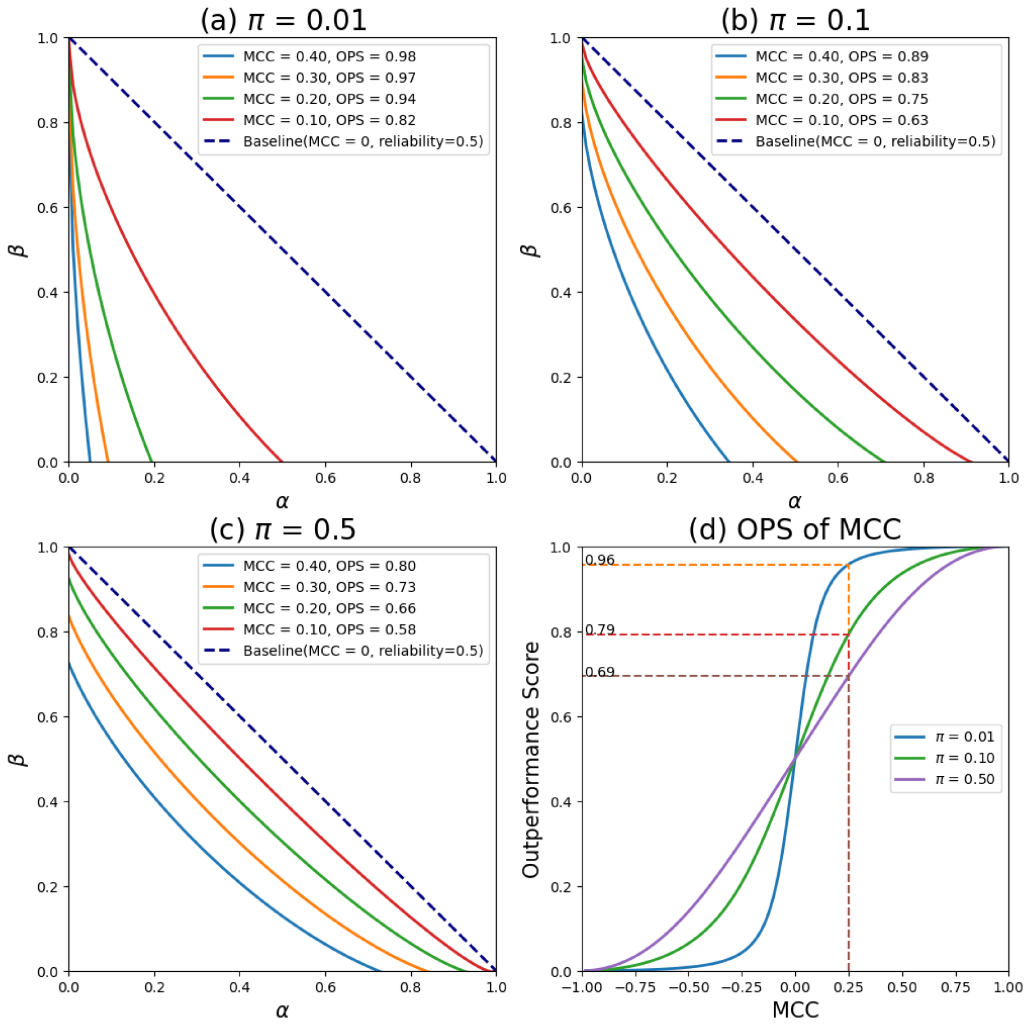
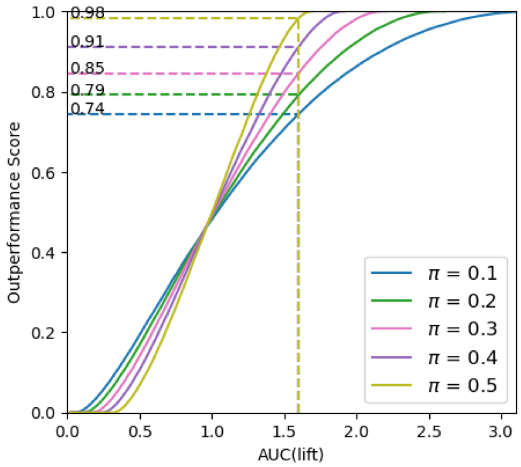
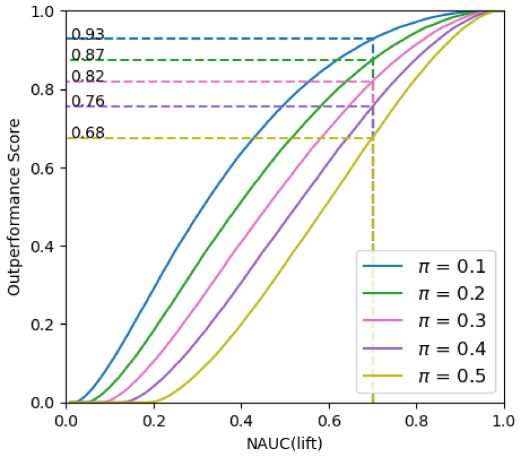


Figure C.1: Geometric representation of OPS_{MCC} when (a) $\pi = 0.01$; (b) $\pi = 0.1$; (c) $\pi = 0.5$. And (d) plots the outperformance score of MCC for different π.

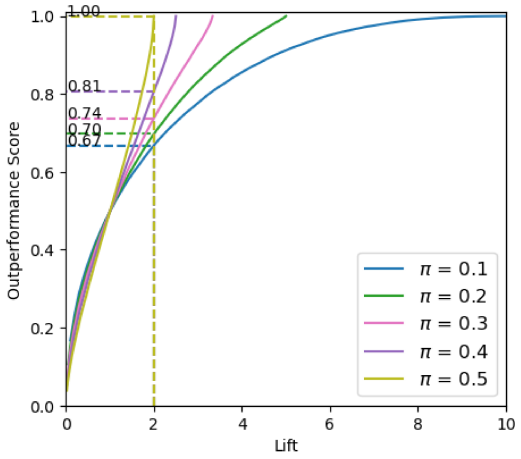
Figure C.1 (a), (b), and (c) show the geometric representation of OPS_{MCC} when $\pi = 0.01, 0.1, 0.5$, respectively. Given $P = \pi$, and a value $\mu \in (0, 1)$ (here we only care about the performances above the baseline), the unit square of (α, β) is divided into two parts by the curve $MCC(\alpha, \beta, \pi) = \mu$, and $\text{OPS}_{MCC}(\mu; \pi)$ is the area of right part. Also, Figure C.1 (d) demonstrates that, for the same positive MCC, e.g., $MCC = 0.25$, the outperformance score increases as π decreases.



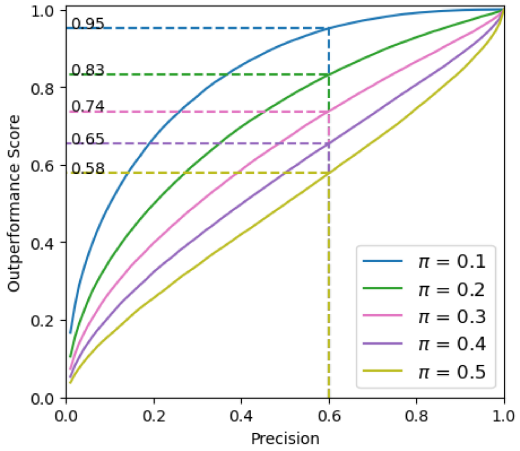
(a) OPS of AUC(lift)



(b) OPS of NAUC(lift)



(c) OPS of Lift | Percentage=0.1



(d) OPS of Precision | Percentage=0.1

Figure C.2: The OPS functions of lift curve: (a) AUC, (B) NAUC, (C) Lift|Percentage=0.1, and (D) Precision|Percentage=0.1, conditional on different imbalance rates.

C.2 Outperformance Score of Lift Curve

As discussed in Section 5.2.4, the lift curve illustrates the trade-off between $lift = \frac{precision}{\pi}$ (x-axis) and percentage (y-axis). For the lift curve, since $NAUC = \frac{AUC}{1-\log \pi}$, by the linear property 6.1, AUC and NAUC share the same outperformance score, i.e., $\text{OPS}_{AUC(lift)}(\mu; \pi) = \text{OPS}_{NAUC(lift)}(\frac{\mu}{1-\log \pi}; \pi)$. As illustrated in Figure C.2a and Figure C.2b, the distributions of AUC(lift) and NAUC(lift) are both sensitive to imbalance rate π. However, for a given value of AUC (e.g., 1.6), the outperformance score decreases as π

decreases, while for a given value of NAUC (e.g., 0.7), the outperformance score increases as π decreases.

For a specific point (u, v) on the lift curve, by the linear property, at $percentage = u$, the outperformance score of $lift = v$ and $precision = \pi v$ are equivalent. Figures C.2c and C.2d demonstrate the OPS functions of lift and precision, respectively, at percentage=0.1, conditioned on various values of π. We observe that for a fixed lift value (e.g., 2), the outperformance score decreases as π decreases, while for a fixed precision value (e.g., 0.6), the outperformance score increases as π decreases.

Appendix D

Experimental Results on Loan Default Dataset for Section 5.7

Loan Default Dataset This dataset contains 255,347 loan holders, each characterized by 17 features related to their personal profiles and loan information. The objective is to predict, based on these features, which individuals are at the highest risk of defaulting on their loans. Similarly, three groups of samples are generated as test sets:

- (General group) This test set is randomly sampled from the original dataset, with 11.24% of individuals labeled as having loan default.
- (Low-income group) This test set is randomly sampled from loan holders whose income is lower than 35K, with around 20% of them having loan default.
- (Aid-center group) This test set is independently sampled in each class, with a customized proportion of 30% of individuals having loan default. This group can be presumed to represent loan holders visiting a location where a financial aid center is situated.

Table D.1: Summary of the test sets for loan default dataset.

	General Group	Low-income Group	Aid-center Group
Sample Size	10,000	10,108	10,063
Class Imblance Rates	0.112	0.203	0.3

Table D.2: The classifier's prediction performance on Loan Default Dataset. ($t = 0.19$).

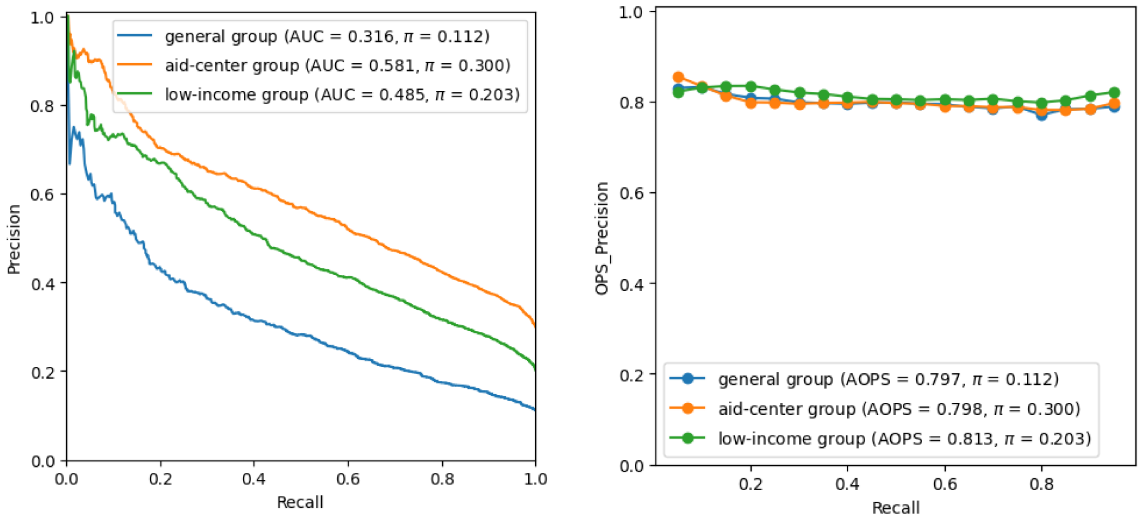
Test Sets	**f1**	**OPS(f1)**	**MCC**	**OPS(MCC)**
General Group ($\pi = 0.112$)	0.361	0.825	0.268	0.798
Low-income Group ($\pi = 0.203$)	0.475	0.806	0.316	0.787
Aid-center Group ($\pi = 0.3$)	0.514	0.735	0.344	0.78

Table D.3: The classifier's risk detection performance on the Loan Default Dataset.

	General Group ($\pi = 0.112$)	**Low-income Group** ($\pi = 0.203$)	**Aid-center Group** ($\pi = 0.3$)
Overall risk detection performance			
AUC(PRC)	0.316	0.485	0.581
OPS(AUC)	0.808	0.838	0.832
AOPS(Precision)	0.797	0.813	0.798
Risk detection performance at recall=0.9			
Precision	0.151	0.278	0.376
OPS(Precision)	0.784	0.784	0.813

Table D.4: The classifier's recommendation performance on the Loan Default Dataset.

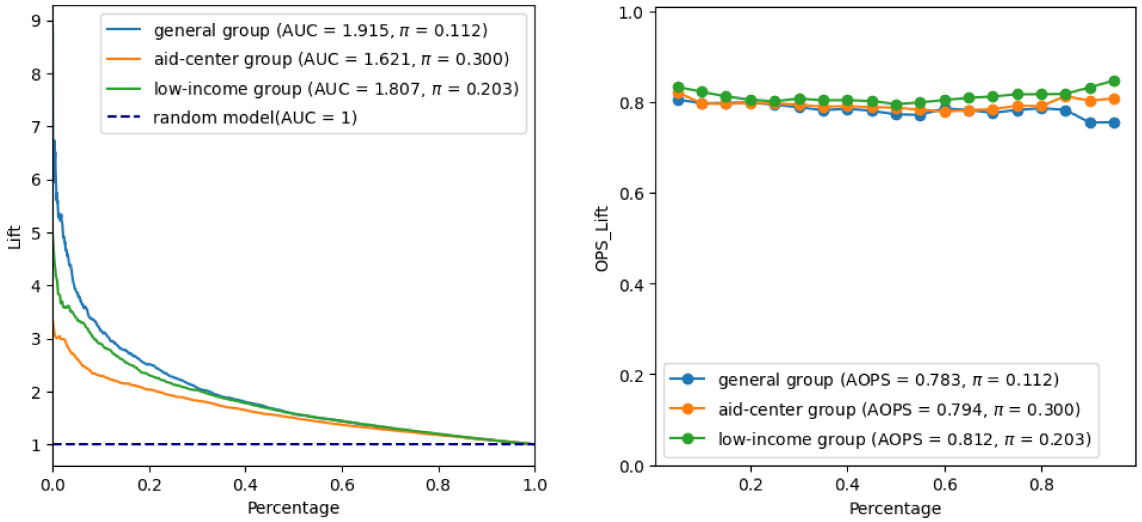
	General Group ($\pi = 0.112$)	**Low-income Group** ($\pi = 0.203$)	**Aid-center Group** ($\pi = 0.3$)
Overall recommendation performance			
AUC(Lift curve)	1.915	1.807	1.621
NAUC(Lift curve)	0.601	0.696	0.736
OPS(AUC or NAUC)	0.849	0.869	0.857
AOPS(Lift or Precision)	0.783	0.812	0.794
Recommendation performance at K=500			
Lift	3.843	3.387	2.627
Precision	0.432	0.686	0.788
OPS(Lift or Precision)	0.805	0.832	0.821



(a) PRC

(b) OPRC

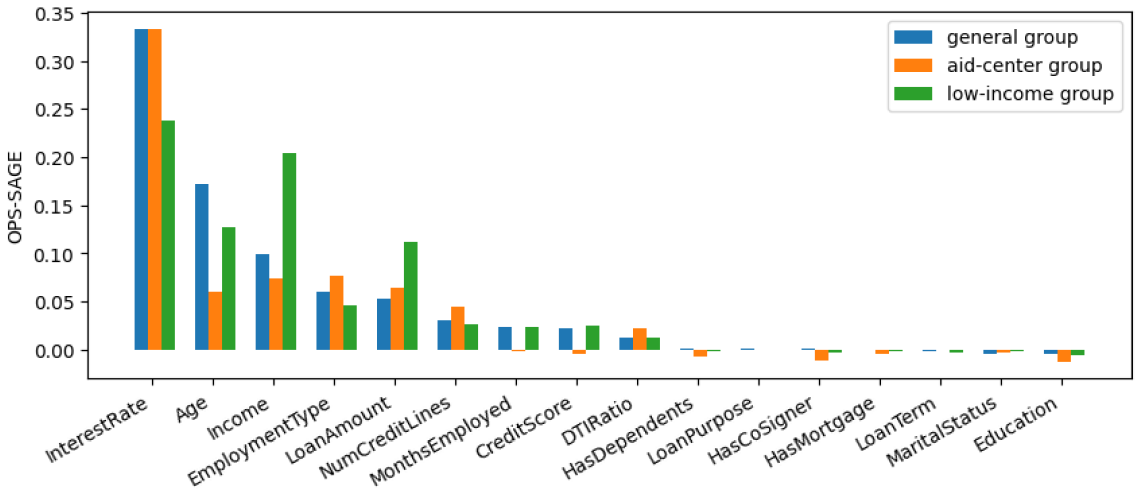
Figure D.1: Visualize the classifier's overall risk detection performance with (a) Precision-Recall curve (PRC), and (b) OPS(Precision)-Recall curve (OPRC), on the Loan Default Dataset.



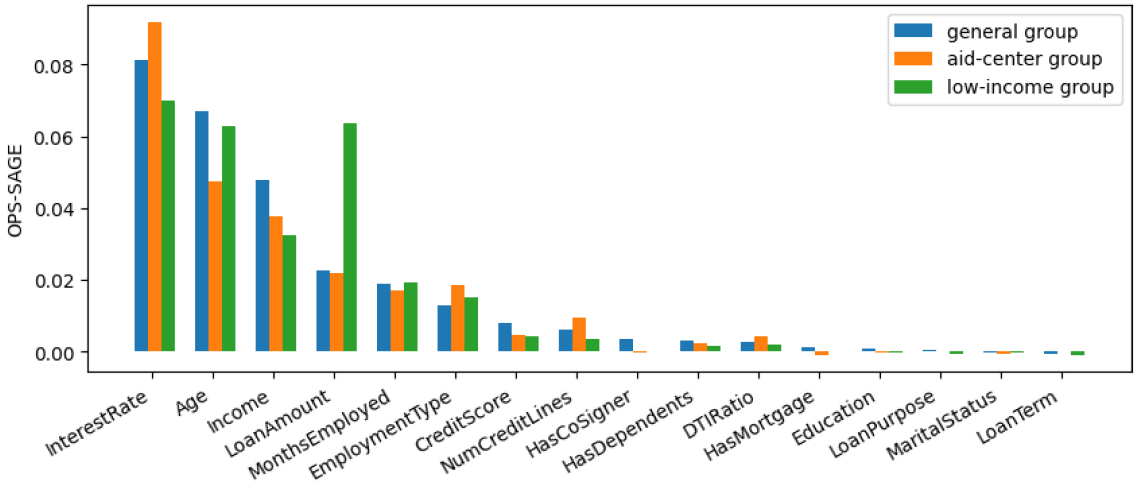
(a) Lift curve

(b) OPS(lift) curve

Figure D.2: Visualize the classifier's overall recommendation performance with (a) lift curve, and (b) OPS(lift) curve, on the Loan Default Dataset.

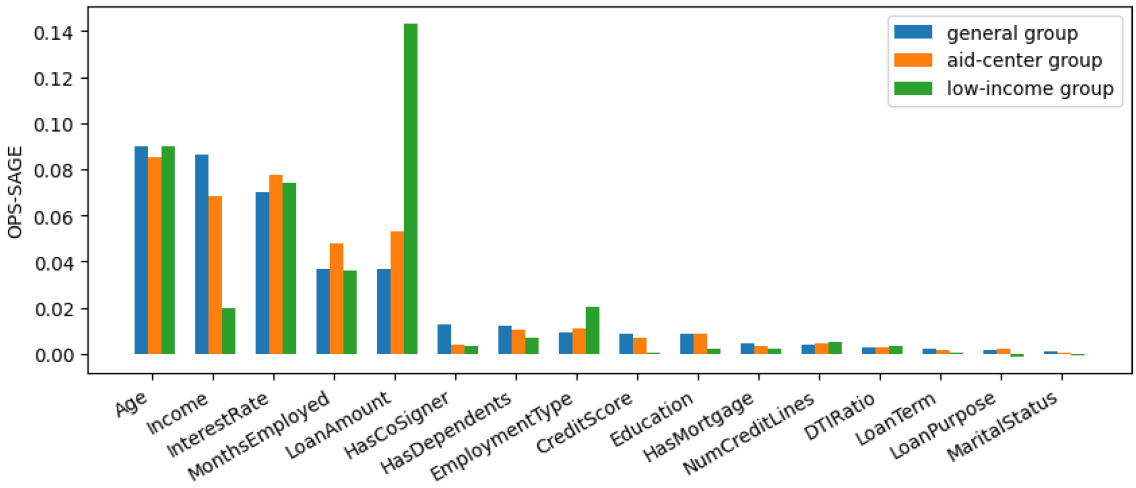


(a) OPS-SAGE w.r.t. f1_score given $t = 0.19$

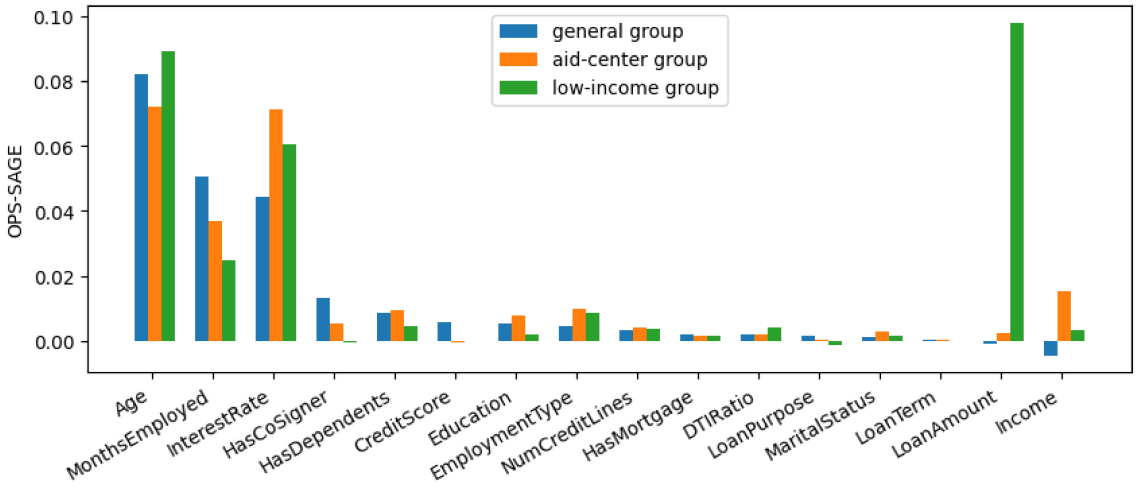


(b) OPS-SAGE w.r.t. MCC given $t = 0.19$

Figure D.3: The classifier's OPS-SAGE w.r.t. (a) f1_score, and (b) MCC, given $t = 0.19$, on the Loan Default Dataset.

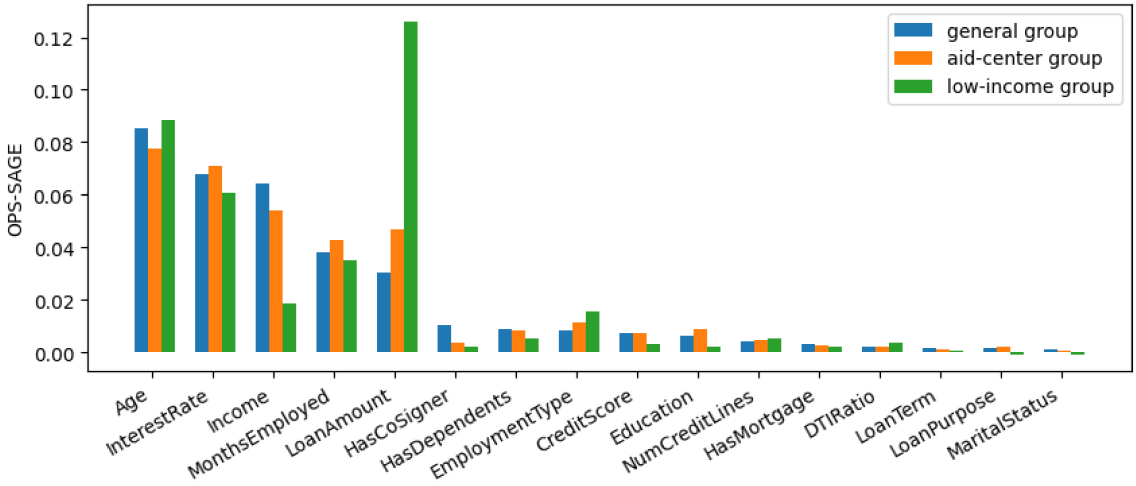


(a) OPS-SAGE w.r.t. OPS(AUC) of PRC

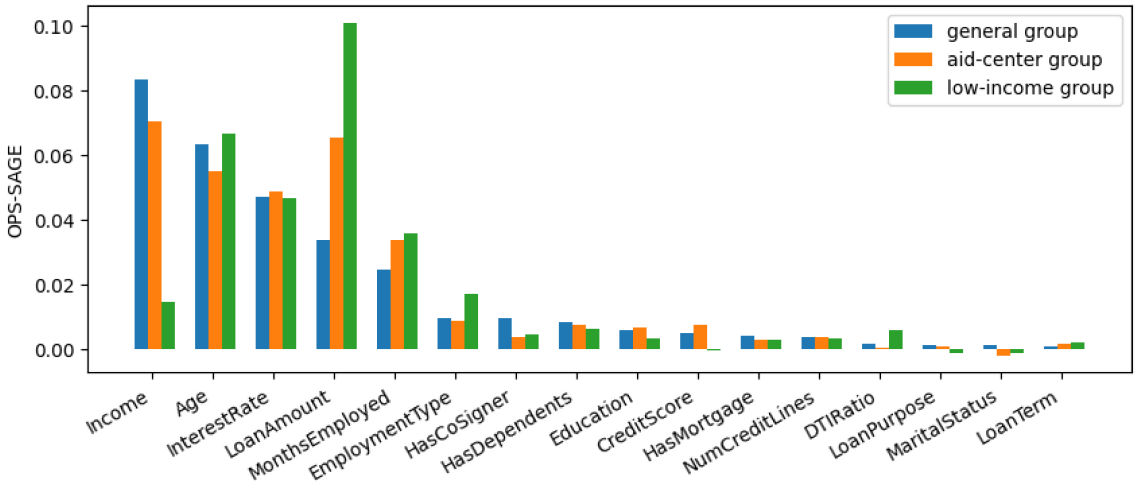


(b) OPS-SAGE w.r.t. OPS(Precision) given Recall=0.9

Figure D.4: The classifier's OPS-SAGE w.r.t. (a) OPS(AUC) of PRC, and (b) OPS(Precision) given Recall=0.9, on the Loan Default Dataset.



(a) OPS-SAGE w.r.t. OPS(AUC) of Lift curve



(b) OPS-SAGE w.r.t. OPS(lift) given K=500

Figure D.5: The classifier's OPS-SAGE w.r.t. (a) OPS(AUC) of Lift curve, and (b) OPS(Lift or Precision) given K=500, on the Loan Default Dataset.

References

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

Alatrany, A. S., Khan, W., Hussain, A., Kolivand, H., & Al-Jumeily, D. (2024). An explainable machine learning approach for alzheimer's disease classification. *Scientific Reports*, *14*(1), 2637.

Arrieta, A. B., Rodríguez, N. D., Ser, J. D., Bennetot, A., Tabik, S., Barbado, A., ... Herrera, F. (2019). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Inf. Fusion*, *58*, 82-115. Retrieved from https://api.semanticscholar.org/CorpusID:204824113

Becker, B., & Kohavi, R. (1996). *Adult*. UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5XW20)

Belghazi, M., Oquab, M., & Lopez-Paz, D. (2019). Learning about an exponential amount of conditional distributions. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2019/file/5a0c828364dbf6dd406139dab7b25398-Paper.pdf

Chen, H., Covert, I. C., Lundberg, S. M., & Lee, S.-I. (2023). Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence*, 1–12.

Chen, H., Janizek, J. D., Lundberg, S., & Lee, S.-I. (2020). True to the model or true to the data? *arXiv preprint arXiv:2006.16234*.

Chicco, D., Tötsch, N., & Jurman, G. (2021). The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining*, *14*, 1–22.

Covert, I., & Lee, S.-I. (2020). Improving kernelshap: Practical shapley value estimation via linear regression. *arXiv preprint arXiv:2012.01536*.

Covert, I., Lundberg, S. M., & Lee, S.-I. (2020). Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, *33*, 17212–17223.

Covert, I. C., Lundberg, S., & Lee, S.-I. (2021). Explaining by removing: A unified framework for model explanation. *The Journal of Machine Learning Research*, *22*(1), 9477–9566.

Datta, A., Sen, S., & Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 ieee symposium on security and privacy (sp)* (pp. 598–617).

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition* (Vol. 31). Springer Science & Business Media.

Fanaee-T, H. (2013). *Bike Sharing Dataset.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5W894)

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Fisher, A., Rudin, C., & Dominici, F. (2019). 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.

Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., & Feige, I. (2020). Shapley

explainability on the data manifold. *arXiv preprint arXiv:2006.01272*.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning* (pp. 1321–1330).

Hart, S. (1989). Shapley value. In J. Eatwell, M. Milgate, & P. Newman (Eds.), *Game theory* (pp. 210–216). London: Palgrave Macmillan UK. Retrieved from https://doi.org/10.1007/978-1-349-20181-5_25 doi: 10.1007/978-1-349-20181-5_25

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, *5*(2), 1.

Huang, X., & Marques-Silva, J. (2023). The inadequacy of shapley values for explainability. *arXiv preprint arXiv:2302.08160*.

inversion, Paik, J., Maggie, mrbhbs, Randazzo, S., & tnat1031. (2020). *Mechanisms of action (moa) prediction.* Kaggle. (https://kaggle.com/competitions/lish-moa)

Janizek, J. D., Dincer, A. B., Celik, S., Chen, H., Chen, W., Naxerova, K., & Lee, S.-I. (2021). Uncovering expression signatures of synergistic drug response using an ensemble of explainable ai models. *BioRxiv*, 2021–10.

Janzing, D., Minorics, L., & Blöbaum, P. (2020). Feature relevance quantification in explainable ai: A causal problem. In *International conference on artificial intelligence and statistics* (pp. 2907–2916).

Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.

Jethani, N., Sudarshan, M., Covert, I. C., Lee, S.-I., & Ranganath, R. (2022). FastSHAP: Real-time shapley value estimation. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=Zq2G_VTV53T

Koyejo, O. O., Natarajan, N., Ravikumar, P. K., & Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. *Advances in Neural Information Processing sSstems*, *27*.

Kumar, I., Scheidegger, C., Venkatasubramanian, S., & Friedler, S. (2021). Shapley residuals: Quantifying the limits of the shapley value for explanations. *Advances in Neural Information Processing Systems*, *34*, 26598–26608.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., & Friedler, S. (2020). Problems with shapley-value-based explanations as feature importance measures. In *International conference on machine learning* (pp. 5491–5500).

Kırboğa, K., & Kucuksille, E. U. (2023, 08). Identifying cardiovascular disease risk factors in adults with explainable artificial intelligence. *Anatolian journal of cardiology*, *27*. doi: 10.14744/AnatolJCardiol.2023.3214

Lin, C., Covert, I., & Lee, S.-I. (2024). On the robustness of removal-based feature attributions. *Advances in Neural Information Processing Systems*, *36*.

Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied stochastic models in business and industry*, *17*(4), 319–330.

Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, *17*(2), 145–151.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., ... Lee, S.-I. (2020). From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, *2*(1), 2522-5839.

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, *91*, 216–231.

Manning, C. D. (2009). *An introduction to information retrieval*. Cambridge university press.

Mase, M., Owen, A. B., & Seiler, B. B. (2019). Explaining black box decisions by shapley cohort refinement. *ArXiv*, *abs/1911.00467*.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, *267*, 1–38.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Naeini, M. P., Cooper, G., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 29).

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on machine learning* (pp. 625–632).

NIKHIL. (2019). *Loan default prediction dataset.* Kaggle. (https://www.kaggle.com/datasets/nikhil1e9/loan-default)

Novakovsky, G., Dexter, N., Libbrecht, M. W., Wasserman, W. W., & Mostafavi, S. (2023). Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, *24*(2), 125–137.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Petsiuk, V., Das, A., & Saenko, K. (2018). Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.

Pontius, R., & Si, K. (2014, 06). The total operating characteristic to measure diagnostic ability for multiple thresholds. *International Journal of Geographical Information Science*, *28*, 570-583. doi: 10.1080/13658816.2013.862623

Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., & Petitcolas, F. (2021). Explainable machine learning for fraud detection. *Computer*, *54*(10), 49–59.

Qiu, W., Chen, H., Dincer, A. B., Lundberg, S., Kaeberlein, M., & Lee, S.-I. (2022). Interpretable machine learning prediction of all-cause mortality. *Communications Medicine*, *2*(1), 125.

Raghavan, V., Bollmann, P., & Jung, G. S. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems (TOIS)*, *7*(3), 205–229.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " 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).

Roth, A. E. (1988). *The shapley value: Essays in honor of lloyd s. shapley*. Cambridge University Press.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3), e0118432.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Shapley, L. S. (1953). A value for n-person games. *Contribution to the Theory of Games*, *2*.

Simon, G., & Vincent, T. (2020). A projected stochastic gradient algorithm for estimating

shapley value applied in attribute importance. In *Machine learning and knowledge extraction: 4th ifip tc 5, tc 12, wg 8.4, wg 8.9, wg 12.9 international cross-domain conference, cd-make 2020, dublin, ireland, august 25–28, 2020, proceedings 4* (pp. 97–115).

Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the aaai/acm conference on ai, ethics, and society* (pp. 180–186).

Snider, B., Patel, B., & McBean, E. (2021). Insights into co-morbidity and other risk factors related to covid-19 within ontario, canada. *Frontiers in Artificial Intelligence*, *4*, 684609.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, *41*, 647–665.

Štrumbelj, E., Kononenko, I., & Šikonja, M. R. (2009). Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, *68*(10), 886–904.

Sundararajan, M., & Najmi, A. (2020). The many shapley values for model explanation. In *International conference on machine learning* (pp. 9269–9278).

Taufiq, M. F., Blöbaum, P., & Minorics, L. (2023). Manifold restricted interventional shapley values. In *International conference on artificial intelligence and statistics* (pp. 5079–5106).

Teboul, A. (2022). *Heart disease health indicators datase.* Kaggle. (https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset)

Watson, D. (2022). Rational shapley values. In *Proceedings of the 2022 acm conference on fairness, accountability, and transparency* (pp. 1083–1094).

Xu, Q., & Liang, Y. (2001). Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, *56*, 1-11. Retrieved from https://api.semanticscholar.org/CorpusID:15797313

Yagin, F. H., Cicek, I. B., Alkhateeb, A., Yagin, B., Colak, C., Azzeh, M., & Akbulut, S. (2023). Explainable artificial intelligence model for identifying covid-19 gene biomarkers. *Computers in Biology and Medicine*, *154*, 106619.

Yeh, C.-K., Lee, K.-Y., Liu, F., & Ravikumar, P. (2022). Threading the needle of on and off-manifold value functions for shapley explanations. In *International conference on artificial intelligence and statistics* (pp. 1485–1502).