INVESTIGATING CALIBRATED CLASSIFICATION SCORES THROUGH THE LENS OF INTERPRETABILITY

ALIREZA TORABIAN

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTERS OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING & COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

AUGUST 2023

Abstract

Calibration is a frequently invoked concept when useful label probability estimates are required on top of classification accuracy. A calibrated model is a scoring function whose scores correctly reflect underlying label probabilities. Calibration in itself however does not imply classification accuracy, nor human interpretable estimates, nor is it straightforward to verify calibration from finite data. There is a plethora of evaluation metrics (and loss functions) that each assess a specific aspect of a calibration model. In this work, we initiate an axiomatic study of the notion of calibration and evaluation measures for calibration. We catalogue desirable properties of calibration models as well as evaluation metrics and analyze their feasibility and correspondences. We complement this analysis with an empirical evaluation, comparing two metrics and comparing common calibration methods to employing a simple, interpretable decision tree.

Acknowledgements

I wish to extend my gratitude towards my supervisor, Professor Ruth Urner, for developing the central idea for this work and providing constant financial support throughout the conduct of this research. I am grateful to all members of my examining committee, Dr. Aijun An, Dr. Kevin McGregor and Dr. Shahin Kamali, for their time and their helpful comments on my thesis.

Table of Contents

List of Tables

List of Figures

Abbreviations

Table 1: Abbreviations list

Abbreviation	**Full form**
SVM	Support Vector Machines
PS	Platt Scaling
IR	Isotonic Regression
PCT	Probability Calibration Tree
DT	Decision Tree
CE	Calibration Error
TCE	Theoretical Calibration Error
PC	Probabilistic Count
MSE	Mean Square Error
RMSE	Root Mean Square Error
AUC	Area Under the Curve

Continued on next page

Table 1 – continued from previous page

Abbreviation	Full form
ECE	Expected Calibration Error
PDE	Probability Deviation Error
KNN	K-Nearest Neighbors
BFSL	Breadth First Search Leaf

Chapter 1

Introduction

The field of machine learning has a profound impact on various sectors, from healthcare to finance, and from natural language processing to computer vision. One of the critical elements contributing to its success is the ability to make reliable and accurate predictions. However, a model's accuracy, though important, is not always sufficient in many applications. Equally important is an accurate estimate of the confidence with which these predictions can be made, a concept often referred to as model calibration.

Model calibration is crucial in scenarios where not just the prediction but the uncertainty associated with the prediction is significant. In medical diagnosis, for example, predicting a disease with a certain probability is important, but so is the reliability of that probability estimate. An incorrectly calibrated model could give a misleading impression of certainty, with potentially serious consequences. Hence, having a well-calibrated model, i.e., a model whose predicted probabilities reflect the true likelihood of an event, is critical.

This thesis aims to provide an in-depth study of the notion of calibration in machine learning models. We develop a formal framework of requirements for calibration, explore various calibration methods and metrics, and empirically evaluate the performance of these methods on real-world and synthetic datasets. Our analysis is in particular driven by a requirement of human interpretability. As the notion of calibration is specifically geared towards providing valid uncertainty estimates to a human user, the interpretability of the calibration model itself seems imperative.

In Chapter 2, we develop our formal framework of requirements for calibrated predictors within a standard setting of statistical learning. Starting from the definition of calibration, we formulate a list of other requirements, such as classification accuracy, approximating the distribution's regression function, interpretability, and monotonicity with respect to the regression function. We postulate that such requirements are often implicitly assumed or hoped to be implied by a calibrated model, but rarely made explicit and formal. We then start our analysis by formally examining the interplay of these properties. That is, we analyze which of these properties imply each other or are independent of each other respectively. This part highlights the challenges in their mutual fulfilment. The chapter proceeds with also presenting approximate (or probabilistic) versions of the above properties and a similar analysis of the relation between these.

In Chapter three, we turn our attention to exploring different commonly used calibration methods and evaluation metrics for calibration. We review a variety of methods for calibration, such as Isotonic Regression (IR), Platt Scaling (PS), and Probability Calibration Trees (PCT).

Since we are in particular interested in facilitating the interpretability of a calibration model, we propose to compare these to the performance of a simple, pure decision tree for calibration (where each leaf is assigned a proportion of labels rather than a single predicted label). We then also review metrics that are commonly used to evaluate calibrated predictors. Since quality of calibration can generally not be directly estimated from finite samples, a variety of surrogate metrics are employed for this purpose. We describe the characteristics, workings, and potential limitations of each reviewed metric. Furthermore, we introduce a new evaluation metric called Probability Deviation Error (PDE). We also compare different approaches for computing calibration metrics based on samples, including whether labels and scores are incorporated individually or averaged over bins, and review options of binning methods. Our aim in this chapter is to provide a comprehensive overview that will set the foundation for the experiments carried out in the subsequent chapter.

In the fourth and final chapter, we describe the experiments conducted as part of our study. Three distinct experimental analyses have been designed to address various aspects of machine learning calibration. In our first experiment, we contrast two calibration metrics, Expected Calibration Error (ECE) and our proposed metric, PDE, examining the impacts of varying binning strategies. The second experiment involves a comparative analysis of the various calibration techniques on real-world datasets. The final experiment explores the trade-off we observed between being an accurate classifier and generating calibrated probabilities.

The contributions of this work can be succinctly summarized as follows:

Chapter 2 This work offers a formalization and analysis of requirements from calibrated predictors. This exposes desirable properties that we believe are often implicitly presumed, and initiates a formal analysis of the relation between these properties. The anaylsis particularly highlights the importance of human interpretability in the context of calibration.

Chapter 3 The thesis provides a comprehensive review and critique of a variety commonly used methods for obtaining calibrated predictors and methods to evaluate calibrated predictors. We here propose a new metric, the Probability Deviation Error (PDE), to evaluate the quality of calibration.

Chapter 4 We provide an empirical evaluation of three aspects of interest: comparing two evaluation metrics (PDE and ECE), comparing several calibration methods, in particular tree based methods in light of the interpretability requirement (PCT and a simple decision tree) and we explore the trade-off between accuracy and calibration.

We hope that this thesis will contribute to the ongoing discourse on model calibration, exploring its nuances and importance in the development of trustworthy and robust machine learning models. We hope to shed light on the intricacies of calibration and motivate further research in this critical area.

1.1 Literature Review

The work in this thesis naturally builds in large parts on previously established notions, methods and metrics. Since several parts of this thesis, in particular Chapter 3, focus on reviewing and analysing existing approaches (and thus literature discussion is provided there), we focus here on providing a brief overview of the main aspects.

The calibration models utilized in this study, namely **Platt Scaling**[1], **Isotonic Regression**[2], and **Probability Calibration Trees (PCTs)**[3], are well-established in the field of machine learning. Each model, with its distinct strengths and limitations, offers a unique perspective on the calibration problem and has greatly contributed to the development of our research. We delve into the details of these models, their individual characteristics, procedures, and potential limitations in Chapter 3 "Overview on Calibration Methods and Metrics".

The calibration metrics used for our evaluation purposes have been similarly inspired by past research. **Classification loss** is one of the most basic yet effective measures of a model's predictive accuracy. **Expected Calibration Error (ECE)** has become a standard in the machine learning calibration literature [4]. **Root Mean Square Error (RMSE)** is a universally accepted measure to quantify prediction errors.

In response to identified limitations and challenges in existing metrics, we introduce a new metric, the **Probability Deviation Error (PDE)**, aiming to provide a more robust measure of calibration performance.

Our thesis acknowledges and builds upon the contributions made in these works. By

understanding their work and its implications, we have been able to develop our research further, and hope to foster a more profound and systematic understanding of aspects of calibration in machine learning.

Chapter 2

Formal Framework for Calibration

2.1 Formal Setup

Binary Classification We consider the standard setup of statistical learning for classification: We let X denote a feature space and Y a label space, by default we consider a binary classification setting, that is $Y = \{0, 1\}$. The data generation is modelled as a distribution D over $X \times Y$. We use D_X to denote the marginal of D over X. We use $\text{supp}(\cdot)$ to denote the support of distribution; and with slight abuse of notation, for a data generating distribution D over $X \times Y$ we will often write $\text{supp}(D)$ to refer to the support of the marginal D_X. Further, we let $\eta_D : X \to [0, 1]$ denote the *regression function* of the data-generating distribution D:

$$\eta_D(x) = \mathbb{P}_{(x',y)\sim D}[y = 1 | x' = x]$$

A *predictor* (or *forecaster*) is a function $f : X \to \mathbb{R}$ that assigns every feature vector a real valued score. We use $\mathcal{F} = \mathbb{R}^X$ to denote the set of all (measurable) predictors and $\mathcal{C} = \{0, 1\}^X$

to denote the set of all classifiers (note that $\mathcal{C} \subseteq \mathcal{F}$). Given a data generating distribution D, we let $\text{range}_D(f)$ denote the *effective range* of the predictor: $\text{range}_D(f) \coloneqq \{f(x) | x \in \text{supp}(D)\}$. In this study, when referring to the *cells* generated by predictor f, we are specifically discussing the subsets in domain X that are being induced by the function $\text{range}_D(f)$ (the preimages of $\text{range}_D(f)$).

A *classifier* is a function $h : X \to Y$ that assigns every feature vector a class label. For binary classification, it is common to obtain a classifier by thresholding some predictor. Given a predictor $f : X \to \mathbb{R}$, we define the classifier induced by f with threshold θ as

$$f_\theta(x) = \mathbb{1}\left[f(x) \geqslant \theta\right]$$

where $\mathbb{1}[\cdot]$ denotes the indicator function.

Predictors f are evaluated by means of a *loss function* $\ell : \mathcal{F} \times X \times Y$, where $\ell(f, x, y)$ indicates how bad the prediction $f(x)$ is given that label y is observed. The goal is to derive a predictor with low *expected loss*

$$\mathcal{L}_D(f) = \mathbb{E}_{(x,y)\sim D}[\ell(f, x, y)]$$

over the data-generating process D. The *binary loss* is the standard evaluation metric for classifiers

$$\ell^{0/1}(h, x, y) = \mathbb{1}\left[h(x) \neq y\right].$$

A classifier that attains the minimal achievable binary loss over a distribution is referred to as a *Bayes classifier*, and its loss as the *Bayes error* of the distribution D. We will let $\text{opt}_D^{0/1}$ denote the Bayes error of distribution D.

Calibration In many applications, it is desirable to not only achieve low classification loss (that is, high classification accuracy), but to have a forecaster that accurately reflects *probabilities* of the label events. The notion of *calibration* reflects such a property; namely, that the predicted value $f(x)$ accurately reflects the probability of seeing label 1 among all instances that are given value $f(x)$.

Definition 1. *A predictor $f : X \to [0, 1]$ is* perfectly calibrated *if the following holds $\forall x \in X$:*

$$f(x) = \mathbb{E}_{(x',y')\sim D}[y' \mid f(x') = f(x)]$$

Since the above property can rarely be expected to hold precisely for a learned predictor, the following notion is used to measure the "distance" from being calibrated for a predictor. This is the most common notion used as an error for calibration [5]–[8].

Definition 2. *[7] For distribution D over $X \times Y$, and $p \in \mathbb{N}$, the L_p* norm calibration error *of predictor $f : X \to [0, 1]$, is given by:*

$$\mathrm{CE}_{p,D}(f) = \Big(\mathop{\mathbb{E}}_{(x,y)\sim D}[|f(x) - \mathop{\mathbb{E}}_{(x',y')\sim D}[y' \mid f(x') = f(x)]|^p] \Big)^{1/p}$$

We note that a predictor is perfectly calibrated if and only if the calibration error is zero (for any p). However, observe that the calibration error is not a loss function: the above definition involves the term $\mathbb{E}_{(x',y')\sim D}[y' \mid f(x') = f(x)]$ being compared to the predictor's output $f(x)$. The calibration error can thus not be evaluated or directly estimated from finite samples from the distribution. Therefore, a variety of alternative evaluation metrics are typically applied to gauge the degree of calibration of a forecaster that are mentioned in Section 3.2. For calibration, there usually are initial scores from some model and using an

additional calibrator model we try to map the scores such that the new scores are calibrated. The initial scores are generated using a base predictor $f_B : X \to \mathbb{R}$ that assigns each feature vector to an initial real valued score.

2.2 Desiderata From Calibration

We now list some formal requirements for predictors that are aimed to be calibrated. The goal here is to make motivations that are often implicit, explicit and formal. While various works in the literature highlight different implicit requirements, there is no structured investigation into the expectations from calibration models. The first, obvious, requirement in the list below, refers to the notion of calibration as defined above (Definitions 1 and 2). However, such a predictor should often have additional qualities that are not subsumed (or implied) by the notion of calibration itself. For example, it is still desirable (if not imperative) that the predictor allows to be thresholded into a classifier with high accuracy. Moreover, the hope behind calibration is often that the predictor f will actually be a good representation of data-generating distribution's regression function η. Neither of these two requirements is implied by the notion of calibration (see below for a formal elaboration on this).

Furthermore, calibration is a notion that is inherently aimed at aiding human interpretation. The intent of providing a probability estimate rather than merely the classification label that is most likely for a given instance, is to provide a human user with a better estimate of the certainty of the label. However, we would argue that, for this estimate to be meaningful to a human user, *the user needs to have a notion of the pool of instances that received this*

estimate. That is, if a model, that say is certified to be calibrated, outputs $f(x) = 0.7$, the human user needs a notion of the set $f^{-1}(0.7) = \{x \in X | f(x) = 0.7\}$, among which the user is now promised that 70% of instances will have label 1. Since calibration does not imply in this case that 70% of instances with this exact feature vector x will have label 1, the mere statement $f(x) = 0.7$ (even with guarantee of the predictor being calibrated) does not provide insight into the data generating process.

Another property that would allow for a degree of human interpretability is point-wise monotonicity with respect to the regression function. For a user, it might be easier to understand that among two instances x and x', the first instance x is more likely to have a positive label 1 than the second instance x' based on observing that $f(x) > f(x')$. However, this type of pairwise comparison is valid only if the predictor is point-wise monotonic with respect to the data-generating distribution's regression function.

The following list summarizes our desiderata from a calibrated model:

Formal requirements We let $f : X \to \mathbb{R}$ denote some predictor and D be a data generating distribution over $X \times Y$. The following are desirable properties for f:

1. **Calibration**: Ideally, f is perfectly calibrated (see Definition 1). That is:

$$\forall r \in \text{range}_D(f) \ : \ \mathbb{E}_{(x,y)\sim D}[y \mid f(x) = r] = r.$$

2. **Classification accuracy**: Thresholding on f yields a good classifier. That is:

$$\exists \theta \in \mathbb{R} \ : \ \mathcal{L}_D^{0/1}(f_\theta) \text{ is close to } \text{opt}_D^{0/1}.$$

We say that f *yields an optimal classifier*, if there exists a threshold θ for which f_θ attains the Bayes error $\text{opt}_D^{0/1}$ (note that this does not necessarily imply zero classification loss).

3. **Approximating the regression function**: The predictor f is close to the regression function η_D, that is:

$$f(x) \text{ is close to } \eta_D(x)$$

for all $x \in \text{supp}(D)$. We say f *is a perfect approximation of the regression function* if $f(x) = \eta_D(x)$ holds for all $x \in \text{supp}(D)$.

4. **Interpretability**: The sets $\{f^{-1}(r) \mid r \in \text{range}_D(f)\}$ are meaningful to a human user, where $f^{-1}(r)$ denotes the pre-image of value r:

$$f^{-1}(r) := \{x \in X \mid f(x) = r\}.$$

The challenge with measuring interpretability lies in its inherent subjectivity and the lack of a universally accepted definition.

One approach for evaluating the interpretability of a model involves assessing the cells induced by predictor f. One aspect to assess using the cells is considering the number of cells induced by f, denoted as $|\text{range}_D(f)|$. A model is considered more interpretable if it produces a finite and relatively small effective range.

5. **Monotonicity**: Predictor f generates probability estimates that are monotonic with respect to the regression function η_D, that is:

$$(\eta_D(x_i) - \eta_D(x_j)) \cdot (f(x_i) - f(x_j)) \geqslant 0$$

for all $x_i, x_j \in \text{supp}(D)$. If equality holds only when $\eta_D(x_i) = \eta_D(x_j)$, we call f strictly monotonic with respect to η_D.

We will start by analyzing these strictly phrased properties. In Section 2.3, we will investigate the relaxations of these properties.

2.2.1 Interplay of Strict Properties

We start our analysis by investigating relationships, implications and compatibilities between the above desiderata. This initial analysis employs deterministic versions of these requirements: a predictor satisfies a certain property (such as perfect calibration or perfect approximation of η) or not.

At first glance, it might appear as if calibration is a stronger requirement than the existence of a threshold for accurate classification. However, it is not difficult to see that calibration is actually a property that is independent of accuracy. A predictor can be perfectly calibrated while effectively useless for classification. And conversely a predictor can be highly accurate while not being calibrated at all.

Observation 1. *Calibration does not imply optimal classification accuracy and optimal classification accuracy does not imply calibration.*

Proof. Consider a one-dimensional feature space, $X = \mathbb{R}$, and a distribution D that has marginal mass distributed uniformly on two points -1 and $+1$ with a deterministic regression function $\eta_D(x) = \mathbb{1}[x \geqslant 0]$. Now the constant predictor $f(x) = 0.5$ is perfectly calibrated, but any threshold $\theta \in \mathbb{R}$ will result in classification loss 0.5. On the other hand, a predictor g

with $g(x) = 0.5 - \epsilon$ for $x < 0$ and $g(x) = 0.5 + \epsilon$ for $x \geqslant 0$ for any $\epsilon > 0$ admits a threshold (namely $\theta = 0.5$) such that the resulting classifier g_θ has classification loss 0 while not being calibrated at all. □

Of course, the regression function η_D is always a predictor (albeit usually an unknown one) that is both perfectly calibrated and optimally accurate (by definition, with threshold $\theta = 0.5$). However, for most cases (that is, for distributions where the regression function is not too simple) it is not the only predictor that enjoys these two qualities. This then means that these two properties together (calibration and optimal classification accuracy) do not imply that the regression function η_D is well approximated.

Theorem 2. *The regression function η_D is not the only predictor satisfying both perfect calibration and optimal classification accuracy if and only if one of the sets $(\text{range}_D(\eta_D) \cap [0, 0.5))$ and $(\text{range}_D(\eta_D) \cap [0.5, 1])$ has size at least 2 (that is, if and only if a Bayes optimal predictor outputs both labels and the effective range of η_D has size at least 3; or a Bayes optimal predictor outputs only one label and the effective range of η_D has size at least 2).*

Proof. Let's assume that one of the sets $\text{range}_D(\eta_D) \cap [0, 0.5)$ and $\text{range}_D(\eta_D) \cap [0.5, 1]$ has size at least 2. Without loss of generality we can assume that there are at least two values smaller than 0.5 in the effective range. That is, there exist $\eta_1, \eta_2 \in \text{range}_D(\eta_D)$, with $\eta_1, \eta_2 < 0.5$ and $\eta_1 \neq \eta_2$. Let's denote regions where the regression function takes on these values by $X_1 = \eta_D^{-1}(\eta_1)$, and $X_2 = \eta_D^{-1}(\eta_2)$. Now consider the predictor

$$f(x) = \begin{cases} \mathbb{E}_{(x',y)\sim D}[y \mid x' \in (X_1 \cup X_2)] & \text{if } x \in (X_1 \cup X_2) \\ \eta_D(x) & \text{if } x \notin (X_1 \cup X_2). \end{cases}$$

By construction, this predictor, thresholded at 0.5 has the same classification loss as η_D (that is the Bayes loss) while being different from η_D.

Conversely, let's assume that there exists a predictor f that is perfectly calibrated and achieves Bayes loss, but is not identical to the regression function η_D (meaning the functions differ with positive probability with respect to D_X). Since $\mathcal{L}_D^{0/1}(f) = \text{opt}_D^{0/1}$, the sets $f^{-1}([0, 0.5)) \cap \text{supp}(D)$ and $\eta_D^{-1}([0, 0.5)) \cap \text{supp}(D)$ must be identical and the sets $f^{-1}([0.5, 1]) \cap \text{supp}(D)$ and $\eta_D^{-1}([0.5, 1]) \cap \text{supp}(D)$ must be identical. Now if η_D was constant on both of these sets, then the only way for f to be calibrated would be to also take on the same constant value. Thus, if f differs from η_D in the support of D_X while being calibrated, it must be the case that η_D is not constant on at least one of $\eta_D^{-1}([0.5, 1]) \cap \text{supp}(D)$ or $\eta_D^{-1}([0, 0.5)) \cap \text{supp}(D)$, which implies that at least one of the sets $\text{range}_D(\eta_D) \cap [0, 0.5)$ and $\text{range}_D(\eta_D) \cap [0.5, 1]$ has size at least 2. □

Corollary 3. *Perfect calibration and optimal classification accuracy together do not imply perfect approximation of η_D.*

Rather than calibration, it turns out that strict monotonicity is a property that is closely related to both optimal classification accuracy and approximation of the regression function.

Observation 4. *Strict Monotonicity implies optimal classification accuracy.*

Proof. Consider a predictor f and assume that f satisfies strict monotonicity. Using the threshold 0.5 on the regression function η_D, we can split the set $\text{supp}(D)$ into two disjoint subsets $X_- \coloneqq \{x \in \text{supp}(D) \ : \ \eta_D(x) < 0.5\}$ and $X_+ = \{x \in \text{supp}(D) : \eta_D(x) \geqslant 0.5\}$. Let

$f_{X_-} \coloneqq \{f(x) : x \in X_-\}$ and $f_{X_+} \coloneqq \{f(x) : x \in X_+\}$. For any x_i from X_- and any x_j from X_+, $f(x_i) < f(x_j)$ since $\eta_D(x_i) < \eta_D(x_j)$ and f is strictly monotonic. This shows that any member of f_{X_-} is smaller than any member of f_{X_+}. Therefore, $\inf(f_{X_+}) >= \sup(f_{X_-})$. The threshold $(\inf(f_{X_+}) + \sup(f_{X_-}))/2$ is one of the thresholds on f to achieve the Bayes classification error. □

Theorem 5. *A predictor f is a perfect approximation of the regression function η_D if and only if it is perfectly calibrated and strictly monotonic.*

Proof. For the first direction, let's assume that f is a perfect approximation of the regression function η_D. In this case we have $f(x) = \eta_D(x)$ for all $x \in \text{supp}(D)$. So for any such x_i and x_j, $f(x_i) - f(x_j)$ is of the same sign as $\eta_D(x_i) - \eta_D(x_j)$. Hence, f is strictly monotonic. Using the threshold 0.5 on predictor f, we then obtain a Bayes classifier. Further, for any r in $\text{range}_D(f)$,

$$\mathop{\mathbb{E}}_{(x,y)\sim D}[y \mid f(x) = r] \;=\; \mathop{\mathbb{E}}_{(x,y)\sim D}[y \mid \eta_D(x) = r] \;=\; r$$

which shows that f is calibrated. Now we will argue that the only predictor that satisfies perfect calibration and strict monotonicity is the regression function η_D. By way of contradiction, let's assume a calibrated and strictly monotonic f is not identical to the regression function, that is, there exists an $x' \in \text{supp}(D)$ with $f(x') \neq \eta_D(x')$. Let $S \coloneqq \{x \in \text{supp}(D) : f(x) = f(x')\}$. Since f is strictly monotonic, we have $\eta_D(x) = \eta_D(x')$ for all $x \in S$. Therefore,

$$\mathop{\mathbb{E}}_{(x,y)\sim D}[y \mid f(x) = f(x')] = \mathop{\mathbb{E}}_{(x,y)\sim D}[y \mid x \in S] = \eta_D(x').$$

Since $\eta_D(x') \neq f(x')$, $\mathbb{E}_{(x,y)\sim D}[y \mid f(x) = f(x')] \neq f(x')$, which shows that f cannot be calibrated.

□

Corollary 6. *Neither calibration nor strict monotonicity alone implies the perfect approximation of* η_D.

In Figure 2.1, we have summarized the relationship between the properties outlined in our theorems.

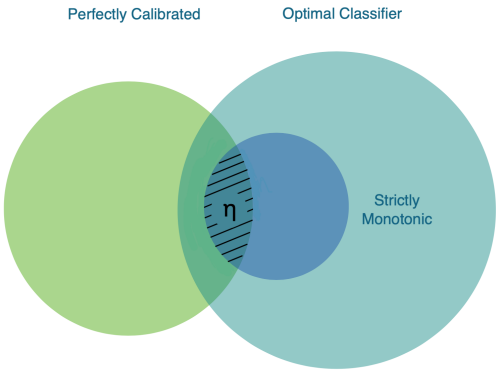


Figure 2.1: Interplay of calibration properties. The intersection of strictly monotonic and perfectly calibrated predictors only contains the regression function of the distribution or η.

2.3 Quantifying Violations

For a predictor f that is learned from finite samples of the data generating process, it is unlikely to fulfill the properties precisely. Thus, we now introduce relaxed, probabilistic

versions of our five desirable properties, or measures of how much the properties are violated and will then see how these relaxations affect the interplay of the properties.

1. **Calibration**: Degree of calibration is measured by the *calibration error* (see also Definition 2):

$$\mathrm{CE}_{p,D}(f) = \Big(\mathop{\mathbb{E}}_{(x,y)\sim D}[|f(x) - \mathop{\mathbb{E}}_{(x',y')\sim D}[y' \mid f(x') = f(x)]|^p] \Big)^{1/p}$$

2. **Classification accuracy**: Thresholding f at some θ yields classifier f_θ. The quality of classification is measured by the standard classification loss:

$$\mathcal{L}_D^{0/1}(f_\theta) = \mathbb{E}_{(x,y)\sim D}\mathbb{1}\left[f_\theta(x) \neq y\right]$$

   Another metric that is widely used to measure the accuracy of classifiers is mean squared error (MSE):

$$\mathrm{MSE}_D(f) = \mathbb{E}_{(x,y)\sim D}[(y - f(x))^2]$$

3. **Approximating the regression function**: To assess whether the predictor f is effectively approximating the regression function, we define ϵ-close and (ϵ, δ)-close approximations.

   Definition 3. *For distribution D over $X \times Y$ with the regression function η_D, predictor $f : X \to \mathbb{R}$ is ϵ-close to the regression function η_D if:*

$$\forall x \in \mathrm{supp}(D), |f(x) - \eta_D(x)| \leqslant \epsilon.$$

Definition 4. *For distribution D over $X \times Y$ with the regression function η_D, predictor $f : X \to \mathbb{R}$ is an (ϵ, δ)-close approximation for the regression function η_D if:*

$$\mathbb{P}_{x \sim D_X}[|f(x) - \eta_D(x)| \leqslant \epsilon] \geqslant 1 - \delta.$$

Predictor f is a *perfect approximation* of the regression function η_D if and only if it is $(0, 0)$-close approximation for η_D.

4. **Interpretability**: We introduce a novel measure, the *probabilistic count*, as a measure that quantifies the size of a predictor's range, taking into consideration the underlying distribution of the data. A lower probabilistic count indicates a model that is more interpretable.

   Definition 5. *For distribution D over $X \times Y$, the probabilistic count of predictor $f : X \to \mathbb{R}$ is defined as:*

$$\mathrm{PC}_D(f) = \frac{1}{\mathbb{P}_{x,x' \sim D_X}[f(x) = f(x')]}.$$

5. **Monotonicity**: Kendall's τ (tau) coefficient is introduced to measure the monotonicity of finite samples [9]. For any set of samples $(x_1, y_1), ..., (x_n, y_n)$, any pair of samples (x_i, y_i) and (x_j, y_j), where $i < j$, are discordant if $(x_i - x_j).(y_i - y_j) < 0$. Kendall's tau coefficient is defined as:

$$\tau = 1 - \frac{2 \times \text{number of discordant pairs}}{\binom{n}{2}}$$

Kendall's τ coefficient is in the range $-1 \leqslant \tau \leqslant 1$. $\tau = 1$ represents perfect agreement between the ranking of two variables, and $\tau = -1$ represents perfect disagreement, i.e. one ranking is the reverse of the other. This coefficient can be used to measure monotonicity, but it doesn't consider the ties to measure strict monotonicity. According to this coefficient, we introduce probabilistic Kendall's tau to measure the monotonicity on two random variables, which are $\eta_D(x)$ and $f(x)$ in our case.

Definition 6. *Probabilistic Kendall's tau for predictor $f : X \to \mathbb{R}$ with respect to the regression function η_D on the distribution D over $X \times Y$ is defined as:*

$$\mathrm{KT}_D(f) = 1 - 2 \times \mathbb{P}_{x,x' \sim D_X}[(\eta_D(x) - \eta_D(x')) \cdot (f(x) - f(x')) < 0 \mid x \neq x']$$

In the next sections, we undertake a thorough exploration of the individual characteristics of the violation measures we've introduced. Each of these measures quantifies specific aspects of deviation from our predictor's ideal performance. Understanding these characteristics is not only key to interpreting the interplay among these measures but also sheds light on potential trade-offs. Furthermore, this knowledge equips us to handle violations more effectively and refine our learning strategies for optimal performance.

2.3.1 Exploring the Probabilistic Count (PC)

In this section, we delve into the characteristics of our newly introduced measure, the probabilistic count. Initially, we establish a connection between the probabilistic count and

the true size of the effective range. Subsequently, we elaborate more on this metric using some examples.

Theorem 7. *For distribution D over $X \times Y$ and predictor $f : X \to \mathbb{R}$,*

$$\mathrm{PC}_D(f) \leqslant |\mathrm{range}_D(f)|,$$

and $\mathrm{PC}_D(f) = |\mathrm{range}_D(f)|$ if and only if all cells generated by f have the same probability weight.

Proof. Assume that f generates n cells from $\mathrm{supp}(D)$ with values r_1, r_2, ..., r_n (i.e., $|\mathrm{range}_D(f)| = n$), and $p_i \coloneqq \mathbb{P}_{x\sim D_X}[f(x) = r_i]$.

$$\begin{aligned}
\frac{1}{\mathrm{PC}_D(f)} &= \mathbb{P}_{x,x'\sim D_X}[f(x) = f(x')] \\
&= \sum_{i=1}^{n} \big[\, \mathbb{P}_{x,x'\sim D_X}[f(x) = f(x')|f(x) = r_i] \\
&\qquad * \mathbb{P}_{x,x'\sim D_X}[f(x) = r_i]\big] \\
&= \sum_{i=1}^{n} \big[\, \mathbb{P}_{x'\sim D_X}[f(x') = r_i] \\
&\qquad * \mathbb{P}_{x\sim D_X}[f(x) = r_i]\big] \\
&= \sum_{i=1}^{n} p_i^2.
\end{aligned}$$

We define two vectors with size n as $u = (p_1, ..., p_n)$ and $v = (1, ..., 1)$. Using Cauchy-Schwarz inequality,

$$|\langle u, v\rangle|^2 \leqslant |\langle u, u\rangle|.|\langle v, v\rangle|,$$

and the equality holds if and only if u and v are parallel. According to this inequality,

$$|\langle u, v\rangle|^2 = (\sum_{i=1}^{n} p_i)^2 \leqslant (\sum_{i=1}^{n} p_i^2) * n$$
$$\Rightarrow \sum_{i=1}^{n} p_i^2 \geqslant \frac{1}{n}$$

So, $\mathrm{PC}_D(f) \leqslant n$. The equality holds if and only if $(p_1, ..., p_n)$ and $(1, ..., 1)$ are parallel which will be the case that all p_is are equal. □

Metric probabilistic count depends on the number of cells and their probability weights. Smaller $\mathrm{PC}_D(f)$ indicates a more interpretable predictor. In the following, we present a series of examples of this metric. In each example, the cells created in the range of f from $\mathrm{supp}(D)$ are visualized in a bar. The length of each partition represents its probability in the distribution D:

- In the case that all cells have the same probability, $\mathrm{PC}_D(f)$ is the number of cells. Figure 2.2 shows n cells with $1/n$ weight. PC_D of this predictor is n.

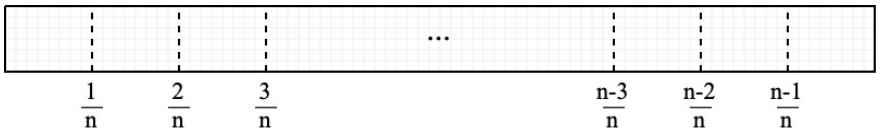


Figure 2.2: n cells with the same weight

- Cells with small weights do not have much effect on the probabilistic count. In Figure 2.3, while we have 9 cells, $\mathrm{PC}_D(f)$ is close to 4. This shows that PC has more consideration

for the weighted cells. $\mathrm{PC}_D(f) = \frac{1}{\frac{1}{4}^2 * 3 + \frac{19}{80}^2 + \frac{1}{80}^2 * 5} \approx 4.08$.

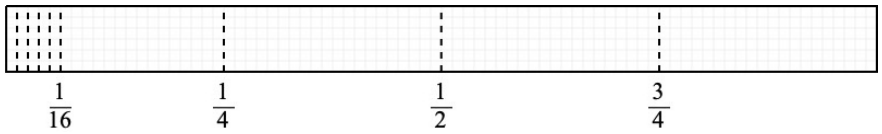


Figure 2.3: 9 cells including 5 small cells

- In Figure 2.4, there are cells with different weights. $\mathrm{PC}_D(f) = \frac{1}{\frac{1}{4}^2 * 2 + \frac{1}{2}^2} \approx 2.66$.

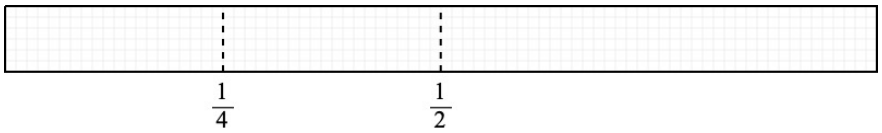


Figure 2.4: Three cells with different weights

- When we have three equal cells, as we discussed, the probabilistic count is 3. In Figure 2.5, we have split the third partition into ten small cells with the same weights. $\mathrm{PC}_D(f) = \frac{1}{\frac{1}{3}^2 * 2 + \frac{1}{30}^2 * 10} \approx 4.28$.

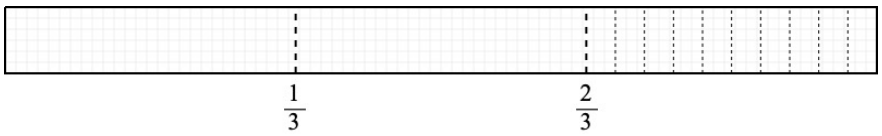


Figure 2.5: 12 cells including 10 cells distributed on a third

- In Figure 2.6, we have split each of the second and the third cells into ten small cells with the same weights. $\mathrm{PC}_D(f) = \frac{1}{\frac{1}{3}^2 + \frac{1}{30}^2 * 20} = 7.5$.

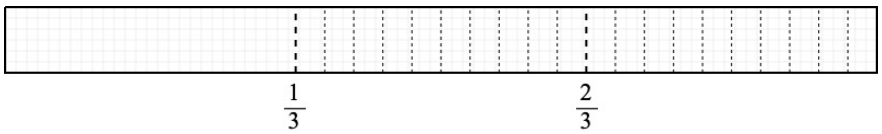


Figure 2.6: 21 cells including 20 cells distributed on two thirds

PC is not monotonic with the actual number of cells. The function in Figure 2.3 generates 9 cells while its PC is 4.28. If we compare this predictor with a function that generates 5 balanced cells, which leads to PC equals to 5, we can show that the predictor in Figure 2.3 has less PC while it has more cells.

2.3.2 Analysis of Cell Merging and Averaging

In this section, we explore two intuitive actions that contribute significantly to improving model interpretability: cell merging and score averaging. Originating from the decision tree methodology, these actions simplify the structure of the model, making its predictions more understandable and traceable. Firstly, we delve into the merging of two cells, thereby reducing model complexity. Secondly, we discuss replacing a cell's score with the label's average. Through the analysis of these modifications, we aim to demonstrate that a higher degree of interpretability can be achieved without compromising the model's predictive power.

In Table 2.1, the results of this section are summarized.

	$\mathrm{CE}_{p,D}$	$\mathcal{L}_D^{0/1}$	MSE_D	PC_D	KT_D
Cell merging along with averaging scores	$\downarrow=$ (Thm. 9)	$\uparrow\downarrow=$ (Obs. 11)	$\uparrow\downarrow=$ (Obs. 11)	$\downarrow=$ [1] (Thm. 8)	$\uparrow\downarrow=$ (Obs. 12)
Average label assigning	$\downarrow=$ (Thm. 13)	$\downarrow=$ (Thm. 13)	$\downarrow=$ (Thm. 13)	$\downarrow=$ (Thm. 13)	$\uparrow\downarrow=$ (Obs. 14)

Table 2.1: Implications of cell merging and score averaging on the properties of a predictor. The arrows and equality signs represent the possible outcomes for each property through each actions. The first action involves merging multiple cells into a single entity and assigning it a score derived from the weighted average of the original cells' scores. The subsequent action replaces each cell's scores with the true average labels of each respective cell.

The interpretability of a predictor is measured based on several criteria, one of which is the size of its effective range. When two cells are combined, the size of the effective range decreases. In the following section, we provide further details on how joining cells affects other aspects of a predictor.

The subsequent theorem explores the effects of merging cells on the probabilistic count of the predictor. Due to the fact that the probabilistic count is not influenced by the scores assigned to the regions, the score of the new cell derived from merging cells does not impact the result. As expected based on the measure's definition, when two regions are combined, the resulting probabilistic count is decreased.

Theorem 8. *For distribution D over $X \times Y$ and predictor $f : X \to \mathbb{R}$, $\forall r_1, r_2 \in \mathrm{range}_D(f)$,*

[1] The conclusion for PC_D still holds for any new cell's score according to Theorem 8.

$\forall r \in \mathbb{R}$ and predictor $g : X \to \mathbb{R}$ defined as follows:

$$g(x) \coloneqq \begin{cases} r & \text{if } f(x) = r_1 \text{ or } r_2 \\ f(x) & \text{otherwise,} \end{cases}$$

then,

$$\mathrm{PC}_D(g) \leqslant \mathrm{PC}_D(f).$$

Proof.

$$\begin{aligned} \frac{1}{\mathrm{PC}_D(g)} - \frac{1}{\mathrm{PC}_D(f)} &= \mathbb{P}_{x,x' \sim D_X}[g(x) = g(x')] - \mathbb{P}_{x,x' \sim D_X}[f(x) = f(x')] \\ &= \mathbb{E}_{x,x' \sim D_X}[\mathbb{1}\left[g(x) = g(x')\right] - \mathbb{1}\left[f(x) = f(x')\right]] \end{aligned}$$

For any $x, x' \in \mathrm{supp}(D)$, $\mathbb{1}\left[g(x) = g(x')\right]$ and $\mathbb{1}\left[f(x) = f(x')\right]$ have the same value (either both are 1 or 0), except when both x and x' belongs to cells with f value equals to r_1 or r_2 or r, since these are the only different cells between f and g. For all x and x' belongs to these cells, $g(x) = g(x') = r$ and $\mathbb{1}\left[g(x) = g(x')\right] = 1$. Therefore, for any x and x' in $\mathrm{supp}(D)$, the term $\mathbb{1}\left[g(x) = g(x')\right] - \mathbb{1}\left[f(x) = f(x')\right] \geqslant 0$. So, $\mathrm{PC}_D(g) \leqslant \mathrm{PC}_D(f)$. □

In the next theorem of this section, we investigate a property of L_p norm calibration error. We show that if we join two cells in the partition generated by the range of a predictor and replace their score with the weighted average of their scores, the new L_p norm calibration error will be less than or equal to the calibration error of the original predictor. Differing from the previous scenario, in this case, the new region's score influence the calibration error.

Theorem 9. *For distribution D over $X \times Y$ and predictor $f : X \to \mathbb{R}$, $\forall r_1, r_2 \in \text{range}_D(f)$ and predictor $g : X \to \mathbb{R}$ defined as follows:*

$$g(x) := \begin{cases} r & \text{if } f(x) = r_1 \text{ or } r_2 \\ f(x) & \text{otherwise,} \end{cases}$$

where $r = \frac{r_1 * \mathbb{P}_{x \sim D_X}[f(x)=r_1] + r_2 * \mathbb{P}_{x \sim D_X}[f(x)=r_2]}{\mathbb{P}_{x \sim D_X}[f(x)=r_1] + \mathbb{P}_{x \sim D_X}[f(x)=r_2]}$, *we have*

$$\text{CE}_{p,D}(g) \leqslant \text{CE}_{p,D}(f).$$

Proof. First, we need to discuss a lemma that is integral to the proof of the theorem.

Lemma 10. *L_p norm calibration error of predictor f can be rewritten as follows using the regression function* [2]*:*

$$\text{CE}_{p,D}(f) = \left(\mathbb{E}_{x \sim D_X}[|\, \mathbb{E}_{x' \sim D_X}[f(x') - \eta_D(x') \mid f(x') = f(x)]|^p] \right)^{1/p}$$

Proof.

$$\begin{aligned} \forall x \in \text{supp}(D),\ & |f(x) - \mathbb{E}_{(x',y') \sim D}[y' \mid f(x') = f(x)]| \\ &= |f(x) - \mathbb{E}_{x' \sim D_X}[\eta_D(x') \mid f(x') = f(x)]| \\ &= |\, \mathbb{E}_{x' \sim D_X}[f(x) - \eta_D(x') \mid f(x') = f(x)]| \\ &= |\, \mathbb{E}_{x' \sim D_X}[f(x') - \eta_D(x') \mid f(x') = f(x)]| \end{aligned}$$

Since the expectation is conditioned on any x' such that $f(x') = f(x)$, we could replace $f(x)$ with $f(x')$ in the last line. □

[2] D_X is the marginal distribution of D over X.

With the foundation provided by the preceding lemma, we can now turn to the proof of the theorem.

If $r_1 = r_2$ then f and g are the same functions, and $\mathrm{CE}_{p,D}(f) = \mathrm{CE}_{p,D}(g)$. Otherwise,

$$\begin{aligned}\mathrm{CE}_{p,D}(f)^p - \mathrm{CE}_{p,D}(g)^p &= \mathbb{E}_{x\sim D_X}[|\,\mathbb{E}_{x'\sim D_X}[f(x') - \eta_D(x') \mid f(x') = f(x)]|^p] \\ &\quad - \mathbb{E}_{x\sim D_X}[|\,\mathbb{E}_{x'\sim D_X}[g(x') - \eta_D(x') \mid g(x') = g(x)]|^p]\end{aligned}$$

(according to Lemma 10)

$$\begin{aligned}&= \mathbb{E}_{x\sim D_X}[|\,\mathbb{E}_{x'\sim D_X}[f(x') - \eta_D(x') \mid f(x') = f(x)]|^p \\ &\quad - |\,\mathbb{E}_{x'\sim D_X}[g(x') - \eta_D(x') \mid g(x') = g(x)]|^p]\end{aligned}$$

The cells that f and g have different expectations in the previous term are the ones with $f(x)$ equals to r_1 or r_2 or r. All members in these three cells are in the same cell in the range of g with $g(x) = r$. So,

$$\begin{aligned}\mathrm{CE}_{p,D}(f)^p - \mathrm{CE}_{p,D}(g)^p &= |\,\mathbb{E}_{x'\sim D_X}[f(x') - \eta_D(x') \mid f(x') = r_1]|^p . \mathbb{P}_{x\sim D_X}[f(x) = r_1] \\ &\quad + |\,\mathbb{E}_{x'\sim D_X}[f(x') - \eta_D(x') \mid f(x') = r_2]|^p . \mathbb{P}_{x\sim D_X}[f(x) = r_2] \\ &\quad + |\,\mathbb{E}_{x'\sim D_X}[f(x') - \eta_D(x') \mid f(x') = r]|^p . \mathbb{P}_{x\sim D_X}[f(x) = r] \\ &\quad - |\,\mathbb{E}_{x'\sim D_X}[g(x') - \eta_D(x') \mid g(x') = r]|^p . \mathbb{P}_{x\sim D_X}[g(x) = r]\end{aligned}$$

For the rest of the proof, we use the following notations:

$$e_1 := \mathbb{E}_{x' \sim D_X}[f(x') - \eta_D(x') \mid f(x') = r_1]$$

$$e_2 := \mathbb{E}_{x' \sim D_X}[f(x') - \eta_D(x') \mid f(x') = r_2]$$

$$e_3 := \mathbb{E}_{x' \sim D_X}[f(x') - \eta_D(x') \mid f(x') = r]$$

$$e' := \mathbb{E}_{x' \sim D_X}[g(x') - \eta_D(x') \mid g(x') = r]$$

$$w1 := \mathbb{P}_{x \sim D_X}[f(x) = r_1]$$

$$w2 := \mathbb{P}_{x \sim D_X}[f(x) = r_2]$$

$$w3 := \mathbb{P}_{x \sim D_X}[f(x) = r]$$

$$w' := \mathbb{P}_{x \sim D_X}[g(x) = r]$$

Since $g(x) = r$ if and only if $f(x) = r_1$ or r_2 or r, $w' = w_1 + w_2 + w_3$. Then we have,

$$\begin{aligned} \mathrm{CE}_{p,D}(f)^p - \mathrm{CE}_{p,D}(g)^p = |e_1|^p.w_1 + |e_2|^p.w_2 + |e_3|^p.w_3 \\ - |e'|^p.(w_1 + w_2 + w_3) \end{aligned} \tag{2.1}$$

Now we use the law of total expectation to rewrite e' as e_1, e_2, and e_3:

$$
\begin{aligned}
e' = \big[& \mathbb{E}_{x' \sim D_X}[g(x') - \eta_D(x') \mid g(x') = r, f(x') = r_1].w_1 \\
& + \mathbb{E}_{x' \sim D_X}[g(x') - \eta_D(x') \mid g(x') = r, f(x') = r_2].w_2 \\
& + \mathbb{E}_{x' \sim D_X}[g(x') - \eta_D(x') \mid g(x') = r, f(x') = r].w_3 \big] \\
& /(w_1 + w_2 + w_3) \\
= \big[& (r - \mathbb{E}_{x' \sim D_X}[\eta_D(x') \mid f(x') = r_1]).w_1 \\
& + (r - \mathbb{E}_{x' \sim D_X}[\eta_D(x') \mid f(x') = r_2]).w_2 \\
& + (r - \mathbb{E}_{x' \sim D_X}[\eta_D(x') \mid f(x') = r]).w_3 \big] \\
& /(w_1 + w_2 + w_3) \\
= \big[& (r + e_1 - r_1).w_1 + (r + e_2 - r_2).w_2 + (e_3).w_3 \big] \\
& /(w_1 + w_2 + w_3) \\
= \big[& e_1.w_1 + e_2.w_2 + e_3.w_3 \big]/(w_1 + w_2 + w_3)
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow |e'|^p &= \big|\frac{e_1.w_1 + e_2.w_2 + e_3.w_3}{w_1 + w_2 + w_3}\big|^p \\
&\leqslant \big[\frac{|e_1|.w_1 + |e_2|.w_2 + |e_3|.w_3}{w_1 + w_2 + w_3}\big]^p \\
&= \big[|e_1|.w'_1 + |e_2|.w'_2 + |e_3|.w'_3\big]^p,
\end{aligned}
$$

in which $w'_i = w_i/(w_1 + w_2 + w_3)$. So, $w'_1 + w'_2 + w'_3 = 1$. Function $|\ .\ |^p$ is a convex function for any $p \in \mathbb{N}$. Therefore, according to the Jensen's inequality [10]:

$$\left[|e_1|.w'_1 + |e_2|.w'_2 + |e_3|.w'_3\right]^p \leqslant |e_1|^p.w'_1 + |e_2|^p.w'_2 + |e_3|^p.w'_3$$

$$\Rightarrow \quad |e'|^p \leqslant \frac{|e_1|^p.w_1 + |e_2|^p.w_2 + |e_3|^p.w_3}{w_1 + w_2 + w_3}$$

$$\Rightarrow \quad \mathrm{CE}_{p,D}(f)^p \geqslant \mathrm{CE}_{p,D}(g)^p \text{(using Equation 2.1)}$$

$$\Rightarrow \quad \mathrm{CE}_{p,D}(f) \geqslant \mathrm{CE}_{p,D}(g).$$

□

The following two observations demonstrate the impact of merging cells on both the classification loss and the probabilistic Kendall's tau.

Observation 11. *Joining regions generated by predictor $f : X \to \mathbb{R}$, and replacing the new score of the region with the weighted average scores of the joint regions may lead to improvement, weakening or no effect on the classification loss and mean squared error.*

Proof. Consider a predictive model f that generates scores of 0.4 and 0.8 for two of its regions with the same weight. Merging these regions results in a new region with an average score of 0.6. Assume that we're using a threshold $\theta = 0.5$ to classify points. In this scenario, the labels for any samples deriving from the second region remain unchanged, while the labels assigned to points in the first region shift from 0 to 1. The classifier's performance is dependent on the regression function.

Let's consider the case where the majority of the points in the second region are given a probability higher than 0.5 from the regression function. In this case, the Bayes optimal classifier would assign a label of 1 to most of this region, which is the same as their new

label assigned by the classifier. This results in improved classifier performance and a lower classification loss.

Conversely, if the Bayes classifier assigns a label of 0 to the majority of points in this region, merging the regions actually weakens the model. Consequently, the new classification loss increases.

To demonstrate potential outcomes for mean squared error, let's consider a scenario where the predictor f assigns a value of 0 to cell a and a value of 1 to cell b. These cells have equal weights. Upon merging them, their combined score becomes 0.5. When all true labels in cell a are 0 and all labels in cell b are 1, the mean squared error increases from 0 to 0.25. Conversely, if the labels in cell a are 1 while those in cell b are 0, the mean squared error decreases from 1 to 0.25. □

Observation 12. *Joining regions generated by predictor $f : X \to \mathbb{R}$, and replacing the new score of the region with the weighted average scores of the joint regions may lead to improvement, weakening or no effect on monotonicity of the predictor and the probabilistic Kendall's tau.*

Proof. Consider a predictive function f that assigns scores r_1 and r_2 to two distinct regions of equal weight, with $r_1 < r_2$. If these regions were to be joined, the resulting region would have a score $\bar{r} = \frac{r_1+r_2}{2}$. Suppose also that f assigns scores a and b to two other regions such that $r_1 < a < \bar{r} < b < r_2$. Assume a regression function η_D assigns identical probabilities to the points within each of the regions defined by r_1, r_2, a, and b. For ease of notation, let's denote the probabilities assigned to r_1, r_2, a, and b as $\eta_D(r_1)$, $\eta_D(r_2)$, $\eta_D(a)$, and $\eta_D(b)$

respectively. Consider the following scenarios:

- The case where the probability relationship is as follows:

$$\eta_D(r_1) < \eta_D(a) < \eta_D(b) < \eta_D(r_2)$$

  In this case, the initial predictor f produces scores that exhibit monotonic behavior across these four regions. However, upon merging the regions with scores r_1 and r_2, monotonicity would be disrupted between the newly formed region and the regions defined by scores a and b.

- The case where

$$\eta_D(a) < \eta_D(r_1) < \eta_D(r_2) < \eta_D(b)$$

  Here, the score relationship between the regions r_1 and a, as well as between r_2 and b, would be non-monotonic. However, if the regions with scores r_1 and r_2 were to be merged, the score of the newly formed region would fall between the regions with scores a and b. This realignment would restore monotonicity in relation to the regression function η_D.

□

An interesting characteristic emerges when the scores for each region are substituted with the average of the true labels within that region. The following theorem explores the impact of this substitution on the performance of the predictor in terms of classification, calibration, and interpretability, demonstrating how this action improves them. If only a single score is

assigned to each region, the most optimal score proves to be the average of labels within each respective region. This true label average is typically not available to a user

Theorem 13. *For distribution D over $X \times Y$ and any predictor $f : X \to \mathbb{R}$ that generates a finite number values over X, consider dividing X into bins b_i based on the outputs of f. By replacing the predicted values for each bin with the average of the true labels in that bin, the calibration error becomes zero and the mean squared error and probabilistic count improve. The classification loss also improves if threshold $\theta = 0.5$ is used. Such a predictor is defined as $\bar{f}_D : X \to [0, 1]$ where*

$$\bar{f}_D(x) \coloneqq \mathbb{E}_{(x',y')\sim D}[y' \mid f(x) = f(x')].$$

Then,

$$\mathrm{CE}_{p,D}(\bar{f}_D) = 0,$$

$$\mathrm{MSE}_D(\bar{f}_D) \leqslant \mathrm{MSE}_D(f),$$

$$\mathcal{L}_D^{0/1}((\bar{f}_D)_\theta) \leqslant \mathcal{L}_D^{0/1}(f_\theta),$$

$$\mathrm{PC}_D(\bar{f}_D) \leqslant \mathrm{PC}_D(f),$$

where $\theta = 0.5$.

Proof. Assume that f generates n different outputs over X, i.e., $\{f(x)|x \in \mathrm{supp}(D)\} = \{s_1^f, s_2^f, ..., s_n^f\}$. X is divided into n bins as follows:

$$\forall i \in [1,n], b_i^f := \{x \in X : f(x) = s_i^f\}.$$

So, for any predictor $f : X \to \mathbb{R}$,

$$\begin{aligned}
\text{MSE}_D(f) &= \mathbb{E}_{(x,y)\sim D}[(y - f(x))^2] \\
&= \sum_{i\in[1,n]} \mathbb{E}_{(x,y)\sim D}[(y - f(x))^2 \mid x \in b_i^f] * \mathbb{P}_{x\sim D_X}[x \in b_i^f] \\
&= \sum_{i\in[1,n]} \mathbb{E}_{(x,y)\sim D}[(y - s_i^f)^2 \mid x \in b_i^f] * \mathbb{P}_{x\sim D_X}[x \in b_i^f] \\
&= \sum_{i\in[1,n]} (\mathbb{E}_{(x,y)\sim D}[y^2 \mid x \in b_i^f] - 2s_i^f \, \mathbb{E}_{(x,y)\sim D}[y \mid x \in b_i^f] \\
&\qquad + (s_i^f)^2) * \mathbb{P}_{x\sim D_X}[x \in b_i^f]
\end{aligned}$$

Using $\bar{y}_i^f$ as $\mathbb{E}_{(x,y)\sim D}[y \mid x \in b_i^f]$ for any $i \in [1,n]$ and the identity $\mathbb{E}[X^2] = \mathbb{V}ar(X) + (\mathbb{E}[X])^2$, we rewrite the expression as:

$$\begin{aligned}
\text{MSE}_D(f) &= \sum_{i\in[1,n]} (\mathbb{V}ar_{(x,y)\sim D}[y \mid x \in b_i^f] + (\bar{y}_i^f)^2 - 2s_i^f \bar{y}_i^f + (s_i^f)^2) * \mathbb{P}_{x\sim D_X}[x \in b_i^f] \\
&= \sum_{i\in[1,n]} (\mathbb{V}ar_{(x,y)\sim D}[y \mid x \in b_i^f] + (\bar{y}_i^f - s_i^f)^2) * \mathbb{P}_{x\sim D_X}[x \in b_i^f]
\end{aligned}$$

The predictor $\bar{f}_D$ generates different scores than f while the bins they generate are the same, i.e., any two elements from $\text{supp}(D)$ that belongs to the same bin in the range of f are in the same bin in the range of $\bar{f}_D$ as well. So, $\forall i \in [1,n], b_i^{\bar{f}_D} = b_i^f, \bar{y}_i^{\bar{f}_D} = \bar{y}_i^f$, and $\mathbb{V}ar_{(x,y)\sim D}[y \mid x \in b_i^{\bar{f}_D}] = \mathbb{V}ar_{(x,y)\sim D}[y \mid x \in b_i^f]$. Also, according to the definition of $\bar{f}_D$, $\forall i \in [1,n], s_i^{\bar{f}_D} = \bar{y}_i^{\bar{f}_D}$. So,

$$\begin{aligned}\mathrm{MSE}_D(f) - \mathrm{MSE}_D(\bar{f}_D) &= \sum_{i\in[1,n]} ((\bar{y}_i^f - s_i^f)^2 - (\bar{y}_i^{\bar{f}_D} - s_i^{\bar{f}_D})^2) * \mathbb{P}_{x\sim D_X}[x \in b_i^f] \\ &= \sum_{i\in[1,n]} (\bar{y}_i^f - s_i^f)^2 * \mathbb{P}_{x\sim D_X}[x \in b_i^f] \geqslant 0\end{aligned}$$

According to the definition of $\bar{f}_D$ and the fact that the bins generated by f and $\bar{f}_D$ are the same, $\forall x \in \mathrm{supp}(D)$, $\bar{f}_D(x) = \mathbb{E}_{(x',y')\sim D}[y' \mid \bar{f}_D(x) = \bar{f}_D(x')]$. According to this expression, we rewrite the calibration error for $\bar{f}_D$ as follows:

$$\begin{aligned}\mathrm{CE}_{p,D}(\bar{f}_D) &= \left(\mathbb{E}_{(x,y)\sim D}[|\bar{f}_D(x) - \mathbb{E}_{(x',y')\sim D}[y' \mid \bar{f}_D(x') = \bar{f}_D(x)]|^p]\right)^{1/p} \\ &= \Big(\mathbb{E}_{(x,y)\sim D}[|\, \mathbb{E}_{(x',y')\sim D}[y' \mid \bar{f}_D(x) = \bar{f}_D(x')] \\ &\qquad - \mathbb{E}_{(x',y')\sim D}[y' \mid \bar{f}_D(x') = \bar{f}_D(x)]|^p]\Big)^{1/p} \\ &= 0.\end{aligned}$$

Now we rewrite the classification loss of predictor f using the n bins:

$$\begin{aligned}\mathcal{L}_D^{0/1}(f_\theta) &= \mathbb{E}_{(x,y)\sim D}\mathbb{1}\left[f_\theta(x) \neq y\right] \\ &= \sum_{i\in[1,n]} \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[f_\theta(x) \neq y\right] \mid x \in b_i^f] * \mathbb{P}_{x\sim D_X}[x \in b_i^f] \\ &= \sum_{i\in[1,n]} \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[\mathbb{1}\left[f(x) \geqslant \theta\right] \neq y\right] \mid x \in b_i^f] * \mathbb{P}_{x\sim D_X}[x \in b_i^f].\end{aligned}$$

Let's consider the expectation part of this expression for one arbitrary bin:

$$\begin{aligned}\mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[\mathbb{1}\left[f(x) \geqslant \theta\right] \neq y \mid x \in b_i^f\right] \\ = \mathbb{E}_{x\sim D_X}\Big[\mathbb{1}\left[f(x) < \theta\right] * \eta_D(x) \\ + \mathbb{1}\left[f(x) \geqslant \theta\right] * (1 - \eta_D(x)) \mid x \in b_i^f\Big] \end{aligned} \tag{2.2}$$

The classification loss is changed on a bin if and only if the label assigned to that bin (with a threshold, $\theta = 0.5$) changes. The label on the whole bin is identical and is either 0 or 1. Without loss of generality let's suppose the bin b_i^f is labelled 0 under f and is reassigned a label of 1 under $\bar{f}D$. Consequently, $s_i^f < 0.5$ and $\mathbb{E}_{(x,y)\sim D}[y \mid x \in b_i^f] \geqslant 0.5$, which means $\mathbb{E}_{x\sim D_X}[\eta_D(x) \mid x \in b_i^f] \geqslant 0.5$. Now we rewrite equation 2.2 for both predictors f and $\bar{f}_D$:

$$\mathbb{E}_{(x,y)\sim D}[\mathbb{1}\Big[\mathbb{1}\left[f(x) \geqslant \theta\right] \neq y\Big] \mid x \in b_i^f] = \mathbb{E}_{x\sim D_X}[\eta_D(x) \mid x \in b_i^f]$$

And for $\bar{f}_D$:

$$\mathbb{E}_{(x,y)\sim D}[\mathbb{1}\Big[\mathbb{1}\left[\bar{f}_D(x) \geqslant \theta\right] \neq y\Big] \mid x \in b_i^f] = \mathbb{E}_{x\sim D_X}[1 - \eta_D(x) \mid x \in b_i^f]$$

Since $\mathbb{E}_{x\sim D_X}[\eta_D(x) \mid x \in b_i^f] \geqslant 0.5$:

$\mathbb{E}_{x\sim D_X}[1 - \eta_D(x) \mid x \in b_i^f] \leqslant \mathbb{E}_{x\sim D_X}[\eta_D(x) \mid x \in b_i^f]$

$\implies \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\Big[\mathbb{1}\left[f(x) \geqslant \theta\right] \neq y\Big] \mid x \in b_i^f] \geqslant \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\Big[\mathbb{1}\left[\bar{f}_D(x) \geqslant \theta\right] \neq y\Big] \mid x \in b_i^f].$

This means that the classification loss on any arbitrary bin for predictor f is greater than or equal to the loss for $\bar{f}_D(x)$, completing our proof that predictor $\bar{f}_D(x)$ leads to an improved classification loss.

When the initial scores are replaced with the average of true labels, it is possible for two regions to have equal new scores. In such cases, these regions are being merged, and as stated in Theorem 8, it follows that $\mathrm{PC}_D(\bar{f}_D) \leqslant \mathrm{PC}_D(f)$.

□

Observation 14. *Substituting the scores for each region in the range of predictor $f : X \to \mathbb{R}$ with their respective average true labels may lead to improvement, weakening or no effect on the monotonicity of the predictor and the probabilistic Kendall's tau.*

Proof. Let's take into account a predictor f that generates only two distinct scores, denoted as r_a and r_b. In other words, only two regions exist within the range of the function f. We denote the average true label of these regions as $\bar{y}_a$ and $\bar{y}_b$. Let's assume $r_a < r_b$. Consider the following scenarios:

- Let's consider a case that each region a and b contain four distinct domain points, and the values of regression function for the samples in region a are 0.1, 0.1, 0.1, and 1 and 0.2 for all four points in region b. In this case, $\bar{y}_a = 0.325$ and $\bar{y}_b = 0.2$. The probabilistic Kendall's tau for predictor f and the new predictor after the substitution, denoted as $\bar{f}_D$, are:

$$\mathrm{KT}_D(f) = 1 - 2 * \frac{8}{56} = \frac{5}{7}, \qquad \mathrm{KT}_D(\bar{f}_D) = 1 - 2 * \frac{24}{56} = \frac{1}{7}.$$

  In this case, substituting the scores with the average true labels has weakened the monotonicity of the predictor.

- Consider the same case as the previous one, the only difference being that the regression function values for samples in region a are now 0.1, 0.3, 0.3, and 1. In this case the probabilistic Kendall's tau has been improved as follows:

$$\mathrm{KT}_D(f) = 1 - 2 * \frac{24}{56} = \frac{1}{7}, \qquad \mathrm{KT}_D(\bar{f}_D) = 1 - 2 * \frac{8}{56} = \frac{5}{7}.$$

□

2.3.3 Effects of Approximating Regression Function

This section offers guarantees regarding the performance of a predictor that is approximating the regression function. The performance of predictors being discussed here is regarding the calibration error and classification error. In the subsequent proposition and theorem, we establish an upper bound for the L_p norm calibration error, assuming the predictor is sufficiently proximate to the regression function. As the predictor deviates more from the regression function, the established upper bound expands correspondingly.

Proposition 15. *For distribution D over $X \times Y$ and any predictor $f : X \to \mathbb{R}$, if f is ϵ-close to the regression function, then $\forall p \in \mathbb{N}$, $\mathrm{CE}_{p,D}(f) \leqslant \epsilon$.*

Proof. For any r in $\mathrm{range}_D(f)$, consider the set $f^{-1}(r) \cap \mathrm{supp}(D)$ that is the preimage of value r from $\mathrm{supp}(D)$. Then,

$$\forall x \in f^{-1}(r) \cap \mathrm{supp}(D),\ r - \epsilon \leqslant \eta_D(x) \leqslant r + \epsilon \ \ (\text{since } |f(x) - \eta_D(x)| \leqslant \epsilon)$$

$$\Rightarrow r - \epsilon \leqslant \mathbb{E}_{(x,y)\sim D}[y \mid f(x) = r] \leqslant r + \epsilon$$

$$\Rightarrow |r - \mathbb{E}_{(x,y)\sim D}[y \mid f(x) = r]| \leqslant \epsilon$$

Since this inequality holds for any r in $\text{range}_D(f)$, $\text{CE}_{p,D}(f) \leqslant \epsilon$. □

Theorem 16. *For distribution D over $X \times Y$ and any predictor $f : X \to [0, b]$, $\forall \epsilon \in [0, 0.5]$ and $\forall \delta \in [0, 1]$ if the predictor f is (ϵ, δ)-close approximation for the regression function, then $\forall p \in \mathbb{N}$,*

$$\text{CE}_{p,D}(f) \leqslant ((1-\delta)\epsilon^p + \delta b^p)^{1/p}.$$

Since the predictor f is usually supposed to generate probabilities to approximate the regression function, we may assume that its range is $[0, 1]$. In this case,

$$\text{CE}_{p,D}(f) \leqslant ((1-\delta)\epsilon^p + \delta)^{1/p}.$$

Proof. With probability of at least $1 - \delta$ over x, $|f(x) - \eta_D(x)| \leqslant \epsilon$. Since the range of f and η are $[0, b]$ and $[0, 1]$ respectively, the maximum difference between them is b. Therefore:

$$\mathbb{E}_{x' \sim D_X}[|f(x') - \eta_D(x')|] \leqslant (1-\delta)\epsilon + \delta b.$$

We can generalize this statement for L_p norm. For any $p \in \mathbb{N}$,

$$\mathbb{E}_{x' \sim D_X}[|f(x') - \eta_D(x')|^p] \leqslant (1-\delta)\epsilon^p + \delta b^p.$$

According to the law of total expectation:

$$\mathbb{E}_{x' \sim D_X}[|f(x') - \eta_D(x')|^p] = \mathbb{E}_{x \sim D_X}[\mathbb{E}_{x' \sim D_X}[|f(x') - \eta_D(x')|^p \mid f(x) = f(x')]].$$

Since $|f(x') - \eta_D(x')|$ is always non-negative and $p \in \mathbb{N}$, we can use Jensen's inequality and move the exponent:

$$\mathbb{E}_{x \sim D_X}[\mathbb{E}_{x' \sim D_X}[|f(x') - \eta_D(x')|^p \mid f(x) = f(x')]] \geqslant$$

$$\mathbb{E}_{x \sim D_X}[\mathbb{E}_{x' \sim D_X}[|f(x') - \eta_D(x')| \mid f(x) = f(x')]^p]$$

Then we can rearrange and move the absolute sign:

$$\mathbb{E}_{x\sim D_X}[\mathbb{E}_{x'\sim D_X}[|f(x') - \eta_D(x')| \mid f(x) = f(x')]^p] \geqslant$$

$$\mathbb{E}_{x\sim D_X}[|\,\mathbb{E}_{x'\sim D_X}[f(x') - \eta_D(x') \mid f(x) = f(x')]|^p]$$

This implies that

$$\mathbb{E}_{x\sim D_X}[|\,\mathbb{E}_{x'\sim D_X}[f(x') - \eta_D(x') \mid f(x) = f(x')]|^p] \leqslant (1-\delta)\epsilon^p + \delta b^p.$$

According to Lemma 10

$$\mathrm{CE}_{p,D}(f) \leqslant ((1-\delta)\epsilon^p + \delta b^p)^{1/p}.$$

□

In the following proposition and theorem, we prove a lower bound for the classification accuracy of a predictor, using 0.5 as the decision threshold, given that the predictor aligns sufficiently closely with the regression function. Proposition 17 and Theorem 18 illustrate that the accuracy of the classifier improves when the regression function yields probabilities that are more deterministic, meaning they are closer to either 0 or 1.

Proposition 17. *For distribution D over $X \times Y$ and any predictor $f : X \to \mathbb{R}$, $\forall \epsilon \in [0, 0.5]$ if f is ϵ-close to the regression function, then for $\theta = 0.5$,*

$$\mathcal{L}_D^{0/1}(f_\theta) - \mathrm{opt}_D^{0/1} \leqslant 2\epsilon \cdot \mathbb{P}_{x\sim D_X}[\eta_D(x) \in [0.5 - \epsilon, 0.5 + \epsilon)],$$

Proof. Let's partition $\text{supp}(D)$ into the following subspaces:

$$\begin{aligned} X_- &:= \{x \in \text{supp}(D) : \eta_D(x) < 0.5 - \epsilon\}, \\ X_{\epsilon_-} &:= \{x \in \text{supp}(D) : 0.5 - \epsilon \leqslant \eta_D(x) < 0.5\}, \\ X_{\epsilon_+} &:= \{x \in \text{supp}(D) : 0.5 \leqslant \eta_D(x) < 0.5 + \epsilon\}, \\ X_+ &:= \{x \in \text{supp}(D) : 0.5 + \epsilon \leqslant \eta_D(x)\}. \end{aligned}$$

We define risk and probability for any subspace of X as follows: $\forall A \subseteq X$,

$$\begin{aligned} R_{g,\theta}(A) &= \mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\left[g_\theta(x) \neq y\right] | x \in A\right], \\ P(A) &= \mathbb{P}_{x\sim D_X}[x \in A]. \end{aligned}$$

Now, let's reformulate the classification loss based on the definitions provided earlier:

$$\mathcal{L}_D^{0/1}(g_\theta) = R_{g,\theta}(X_-).P(X_-) + R_{g,\theta}(X_+).P(X_+) + R_{g,\theta}(X_{\epsilon_-}).P(X_{\epsilon_-}) + R_{g,\theta}(X_{\epsilon_+}).P(X_{\epsilon_+}).$$

Since f is ϵ-close to the regression function η, $\forall x \in \text{supp}(D) : \eta_D(x) - \epsilon \leqslant f(x) \leqslant \eta_D(x) + \epsilon$. So,

$$\begin{aligned} &\forall x \in X_-, f(x) < 0.5 \\ &\forall x \in X_+, f(x) \geqslant 0.5. \end{aligned}$$

If we use $\theta = 0.5$ as the threshold on f to classify, any x in subspaces X_- and X_+ would have the same label as the Bayes classifier. Therefore, $R_{f,0.5}(X_-) = R_{\eta_D,0.5}(X_-)$ and $R_{f,0.5}(X_+) = R_{\eta_D,0.5}(X_+)$.

On classifying items from sets X_{ϵ_-} and X_{ϵ_+}, the worst case is when all items from X_{ϵ_-} are classified as 1, and all items from X_{ϵ_+} are classified as 0. So,

$$R_{f,0.5}(X_{\epsilon_-}) \leqslant 1 - R_{\eta_D,0.5}(X_{\epsilon_-}),$$

$$R_{f,0.5}(X_{\epsilon_+}) \leqslant 1 - R_{\eta_D,0.5}(X_{\epsilon_+}).$$

According to the definition of X_{ϵ_-} and X_{ϵ_+},

$$0.5 - \epsilon \leqslant R_{\eta_D,0.5}(X_{\epsilon_-}) \leqslant 0.5,$$

$$0.5 - \epsilon \leqslant R_{\eta_D,0.5}(X_{\epsilon_+}) \leqslant 0.5. \tag{2.3}$$

$$\begin{aligned}
\mathcal{L}_D^{0/1}(f_\theta) - \text{opt}_D^{0/1} &= R_{f,0.5}(X_-).P(X_-) + R_{f,0.5}(X_+).P(X_+) \\
&\quad + R_{f,0.5}(X_{\epsilon_-}).P(X_{\epsilon_-}) + R_{f,0.5}(X_{\epsilon_+}).P(X_{\epsilon_+}) \\
&\quad - R_{\eta_D,0.5}(X_-).P(X_-) - R_{\eta_D,0.5}(X_+).P(X_+) \\
&\quad - R_{\eta_D,0.5}(X_{\epsilon_-}).P(X_{\epsilon_-}) - R_{\eta_D,0.5}(X_{\epsilon_+}).P(X_{\epsilon_+})) \\
&= R_{f,0.5}(X_{\epsilon_-}).P(X_{\epsilon_-}) + R_{f,0.5}(X_{\epsilon_+}).P(X_{\epsilon_+}) \\
&\quad - R_{\eta_D,0.5}(X_{\epsilon_-}).P(X_{\epsilon_-}) - R_{\eta_D,0.5}(X_{\epsilon_+}).P(X_{\epsilon_+}) \\
&\leqslant (1 - R_{\eta_D,0.5}(X_{\epsilon_-})).P(X_{\epsilon_-}) + (1 - R_{\eta_D,0.5}(X_{\epsilon_+})).P(X_{\epsilon_+}) \\
&\quad - R_{\eta_D,0.5}(X_{\epsilon_-}).P(X_{\epsilon_-}) - R_{\eta_D,0.5}(X_{\epsilon_+}).P(X_{\epsilon_+}) \\
&\leqslant 2.\epsilon.(P(X_{\epsilon_-}) + P(X_{\epsilon_+})) \quad \text{(according to equation 2.3)} \\
&= 2\epsilon.\,\mathbb{P}_{x\sim D_X}[\eta_D(x) \in [0.5 - \epsilon, 0.5 + \epsilon)].
\end{aligned}$$

□

Theorem 18. *For distribution D over $X \times Y$ and any predictor $f : X \to \mathbb{R}$, $\forall \epsilon \in [0, 0.5]$ and $\forall \delta \in [0, 1]$ if f is (ϵ, δ)-close approximation for the regression function, then for $\theta = 0.5$,*

$$\mathcal{L}_D^{0/1}(f_\theta) - \text{opt}_D^{0/1} \leqslant 2\epsilon \cdot p_\epsilon(1 - \delta) + \delta,$$

where p_ϵ is defined as $\mathbb{P}_{x \sim D_X}[\eta_D(x) \in [0.5 - \epsilon, 0.5 + \epsilon) \big| |f(x) - \eta_D(x)| \leqslant \epsilon]$.

Proof. First, we define the following subspaces:

$$X_{close} \coloneqq \{x \in \text{supp}(D) : |f(x) - \eta_D(x)| \leqslant \epsilon\},$$

$$X_{far} \coloneqq \{x \in \text{supp}(D) : |f(x) - \eta_D(x)| > \epsilon\}.$$

With probability of at least $1 - \delta$ over x, $x \in X_{close}$. Using the subspaces defined earlier, the following equation is provided. h_{Bayes} is the Bayes optimal classifer that is defined as $h_{\text{Bayes}}(x) = \mathbb{1}\left[\eta_D(x) \geqslant 0.5\right]$.

$$\begin{aligned}
\mathcal{L}_D^{0/1}(f_\theta) - \text{opt}_D^{0/1} &= \mathbb{E}_{(x,y)\sim D}\, \mathbb{1}\left[f_{0.5}(x) \neq y\right] - \mathbb{E}_{(x,y)\sim D}\, \mathbb{1}\left[h_{\text{Bayes}}(x) \neq y\right] \\
&= (\mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[f_{0.5}(x) \neq y\right] \ \big| x \in X_{far}] \\
&\quad - \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[h_{\text{Bayes}}(x) \neq y\right] \ \big| x \in X_{far}]).\, \mathbb{P}_{x\sim D_X}[x \in X_{far}] \\
&\quad + (\mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[f_{0.5}(x) \neq y\right] \big| x \in X_{close}] \\
&\quad - \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[h_{\text{Bayes}}(x) \neq y\right] \ \big| x \in X_{close}]).\, \mathbb{P}_{x\sim D_X}[x \in X_{close}] \qquad (2.4)
\end{aligned}$$

The subspace X_{close} includes items with ϵ-close f to η, which makes us able to use Proposition

17 here. So,

$$\begin{aligned}
&\mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[h_{\text{Bayes}}(x) = y\right] \ | x \in X_{close}] - \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[f_{0.5}(x) = y\right] \ | x \in X_{close}] \\
&\leqslant 2\,\mathbb{P}_{x\sim D_X}[\eta_D(x) \in [0.5-\epsilon, 0.5+\epsilon) | x \in X_{close}] \\
&= 2p_\epsilon
\end{aligned}$$

For the subspace X_{far}, considering the maximum difference between the classification loss of two predictors, the following upper bound is provided:

$$\mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[f_{0.5}(x) \neq y\right] | x \in X_{far}] - \mathbb{E}_{(x,y)\sim D}[\mathbb{1}\left[h_{\text{Bayes}}(x) \neq y\right] | x \in X_{far}] \leqslant 1$$

Now we rewrite the equation 2.4:

$$\begin{aligned}
\mathcal{L}_D^{0/1}(f_\theta) - \text{opt}_D^{0/1} &\leqslant 2\epsilon.p_\epsilon.\,\mathbb{P}_{x\sim D_X}[x \in X_{close}] + 1.\,\mathbb{P}_{x\sim D_X}[x \in X_{far}] \\
&\leqslant 2\epsilon.p_\epsilon(1-\delta) + \delta.
\end{aligned}$$

□

Chapter 3

Overview on Calibration Methods and Metrics

In this chapter, we delve into the intricacies of calibration methods and evaluation metrics that are instrumental in the field of machine learning. We commence our exploration with the presentation of several established calibration methods. Each method is explicated in detail, focusing on their individual characteristics, procedures, and potential limitations. We also propose employing a decision tree for calibration, a new approach in our research context, given that decision trees are inherently interpretable models.

The subsequent section presents an overview of several evaluation metrics. These are essential tools employed to measure the performance of the calibration methods. We provide a comprehensive understanding of each metric, their computations, and implications on the calibration models. A critique of these metrics underlines the need for innovative metrics. This

revelation of challenges in existing metrics propels us to introduce a new metric, Probability Deviation Error (PDE), designed to effectively address these issues and provide a more robust measurement of calibration performance.

The chapter culminates with a critical discussion on two contrasting approaches for computing calibration metrics in machine learning models, the individual or point-wise labels and scores, and the binning method averages. We weigh the merits and demerits of each method while introducing three unique approaches to define regions in the evaluation process: Uniform-mass, K-nearest neighbor (KNN), and Breadth First Search Leaf (BFSL) which is a novel partitioning methodology that we have developed.

By comparing these methods, we gain a deeper understanding of how different approaches can affect the evaluation of calibration models. This comprehensive overview sets the foundation for the following chapters, in which we delve into the experiments.

3.1 Calibration Methods

This section will provide a summary of several commonly used probability calibration methods, namely Platt scaling [1], isotonic regression [2], and histogram binning [11]. We review a model named scaling-binning [7], which is a combination of Platt scaling and histogram binning. Additionally, we delve into a tree-based methodology, named the probability calibration tree [3]. Most of the existing calibration methods rely on the existence of a previously trained base model. At the end of this section, we discuss how we use pure decision trees as calibrators for the first time. Using decision trees, no base model is required to be trained.

3.1.1 Platt Scaling

Platt Scaling (PS) has been proposed as a method to transform Support vector machine (SVM) predictions into calibrated probabilities [1]. A sigmoid function is fitted with logistic regression to map the log-odds scores generated by the base model to the calibrated probabilities. When training the logistic regression model, the scores produced by the SVM model serve as its inputs. Since the true probabilities are not provided in the training dataset, labels are used as the target of the logistic regressing model. The logistic regression model is represented as

$$\frac{1}{1+\epsilon^{-\beta_0+\beta_1 h(x)}}$$

where $h(x)$ is the initial SVM model, and β_0 and β_1 are the parameters of the regression model. The regression model is trained to generate labels y.

To avoid overfitting, an independent set is used to train the calibrator model. Although Platt scaling was originally introduced to calibrate SVM models, it has been shown that it works well with boosted models and naive Bayes classifiers [12]. Platt scaling cannot directly be used for multi-class problems unless using techniques such as the one-vs-rest (OvR) and the one-vs-one (OvO) methods. Using these methods, a multi-class classification is split into multiple binary classification problems. In OvR, one binary classification model is created for each class in the original problem to predict the membership in that class. Therefore, to train the binary classification model for a class, all samples in that class are considered positive and the others are negative. In OvO, there would be a binary classification model for each pair of classes in the original problem. Each time, only the members of two classes

are considered, and a model is trained to predict the membership of samples among those two classes.

3.1.2 Isotonic Regression

Isotonic Regression (IR) aims to acquire a piecewise constant function approximating the mapping from the base model's scores $f_B(.)$ to the calibrated probabilities [2]. The goal is to find a mapping such that the calibrated scores are monotonically increasing (isotonic) with respect to the initial scores from the base model. In this method, the initial scores serve merely to sequence the samples. Using the initial scores, a sequence of samples is provided. First we initial a mapping $f_0 : \mathbb{R} \to \mathbb{R}$ using dataset D on this sequence:

$$f_0(s) := \frac{\sum_{(x_i,y_i)\in D} y_i \mathbb{1}\left[f_B(x_i) = r\right]}{\sum_{(x_i,y_i)\in D} \mathbb{1}\left[f_B(x_i) = r\right]}, \forall r \in \mathbb{R}.$$

For each value of r we have a block of samples. For a new sample that there is no training sample with the same score (i.e., $\sum_{(x_i,y_i)\in D} \mathbb{1}\left[f_B(x_i) = r\right] = 0$), the average of the samples' labels with the closest scores are returned. If the current mapping f_k is already isotonic, it will be returned as the calibrator. Otherwise, the mapping is updated to f_{k+1} as follows. In this case, since the scores are not monotonic, there must be two consecutive blocks of points such that $f_k(r_i) > f_k(r_j)$ while $r_i < r_j$. These two blocks are taken as pair-adjacent violators, and their value is replaced by their average. So,

$$f_{k+1}(r_i) = f_{k+1}(r_j) = (f_k(r_i) + f_k(r_j))/2$$

Therefore, the blocks become conjoined and are considered as a single block. This process

is repeated until there are no violators. Similar to Platt scaling, isotonic regression cannot be used for multi-class classification directly and a separate set must be used for calibrator training to avoid overfitting on the base model.

3.1.3 Histogram Binning

In the histogram binning methodology as presented by [11], samples are arranged based on the scores produced by the base model $f_B(.)$. These arranged samples are then partitioned into B distinct bins. The calibrated probability for each bin is computed as the proportion of positive samples within it. When a new sample is categorized into a bin, the probability associated with that particular bin is assigned as the score for the sample. It necessitates the use of multi-class strategies like one-vs-rest and one-vs-one for problems of that nature. Moreover, it does not ensure strict monotonicity in relation to the base model scores.

3.1.4 Scaling-Binning

The method discussed in [7] utilizes the principles of Platt scaling and histogram binning. The method is implemented in three steps. Initially, similar to Platt scaling, a regression function is trained to map the scores from the base model to actual labels. Next, the output from the regression function is used to partition the space into B bins, each containing uniform-mass. Finally, the calibrated probability is determined by considering the mean output of the regression for the samples within each bin. To prevent overfitting, independent datasets are employed at each stage. As this method utilizes binning and averaging, similar

to histogram binning, it fails to maintain strict monotonicity in relation to the scores from the base model. For multi-class problems, similar strategies as used in histogram binning and Platt scaling are required, such as one-vs-rest and one-vs-one.

3.1.5 Probability Calibration Tree

Probability Calibration Trees (PCT) is a logistic tree based model to map the generated scores by a classifier into calibrated scores [3]. To train this calibrator, not only the scores and labels but also the sample features are considered. First, a simple decision tree is trained according to the features and labels without using the scores. The stopping criteria for the decision tree is a minimum for the number of samples in the node. Then, at each node a regression model is fitted using logit-boost algorithm to map the scores to their calibrated probabilities. For a new sample, first the decision tree is followed using the sample's features until a leaf is reached. Then using the regression model and the sample's score the calibrated probability is computed. To train the regression models in each leaf, root mean square error (RMSE) loss is used. After training the decision tree and regression models in the nodes using a different set of training samples, the leaf is pruned using cost-complexity pruning to minimize the cost RMSE $+ \alpha|T|$, in which $|T|$ is the number of nodes and α is the pruning factor. The optimal value for α is determined using cross-validation method. Since decision trees and logit-boost are usable for multi-class problems, PCT can be directly used for these problems. Training the probability calibration trees is done using an independent set.

3.1.6 Decision Tree

Regarding the interoperability of decision trees, we are motivated to use these models as calibration methods for the first time. This novel use of decision trees for calibration expands the existing repertoire of calibration techniques, and serves as a unique contribution to the field of machine learning calibration. Here, we have removed the regression models on nodes of PCT model to have a pure decision tree. The decision tree is trained in the same way as PCT, and then the average label of samples on each node is assigned as the generated score by the node. The theorems in Section indicates that the true average of labels in each cell achieves the most calibrated scores and the least classification and mean squared error. We use the average of labels over the training samples on each node as the estimation for its true label average. In the last step, we do the same cost-complexity pruning step as PCT using the cost $\text{RMSE} + \alpha|T|$. As demonstrated in Theorem 13, this approach of assigning the average of labels within each node or region as the generated score optimizes performance in classification and calibration.

3.2 Evaluation Metrics

Evaluating calibration models necessitates the use of appropriate metrics. However, since the true probabilities from the underlying distribution are usually inaccessible, a straightforward metric to assess calibration is non-existent. Here, we present and discuss several evaluation metrics used in this research, including Root mean square error (RMSE)[13], Classification

loss[14], Area under the ROC curve (AUC)[15], Area under the validity curve (AUC_V)[16], and Expected calibration error (ECE)[4]. Furthermore, we introduce a novel calibration metric called Probability Deviation Error (PDE). We let x_i and y_i denote the features and label of a single sample respectively, in which $y_i \in \{0, 1\}$. $D^{(n)}$ denotes the collection of n samples that are used for the evaluation process:

$$D^{(n)} := ((x_1, y_1), (x_2, y_2), ..., (x_n, y_n))$$

3.2.1 Root Mean Square Error (RMSE)

$$\mathcal{L}_n^{\text{RMSE}}(f) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2}$$

This metric is being widely used for classification problems [13]. Using this metric in optimizing a model may result to outputs that are also calibrated. However, $RMSE$ cannot evaluate calibration accurately. Consider the case that all samples are correctly assigned with their true probability. RMSE for this model is high since it compares the model's outputs with the labels rather than their true probability. Consider the case that the distribution generates labels 0 and 1 with the same probability for any member of the domain. Model $f(x) = 1/2$ generates correct probabilities and it is calibrated, while RMSE for this model is $1/2$.

3.2.2 Classification Loss

$$\mathcal{L}_n^{0/1}(f) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}(y_i)^2 & \text{if } f(x_i) \leqslant 1/2\\ (1-y_i)^2 & \text{otherwise,}\end{cases}$$

which is the empirical classification error when we use 1/2 as the threshold on the predicted probabilities [14]. This metric does not care if the generated probabilities are calibrated as long as they are in the right position relative to the threshold. Also, it may output a high error for a calibrated model since it evaluates the classification correctness, e.g., for the same calibrated model mentioned in RMSE part over the same distribution, the classification error is 1/2.

3.2.3 Area Under the ROC Curve (AUC)

Using different thresholds in the range $[0, 1]$ on the predicted probabilities, samples are classified and true positive rate (TPR) and false positive rate (FPR) are computed.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

Using the values for TPR and FPR on different thresholds, we would have a curve whose area is the metric AUC [15]. If our model generates wrong scores but in the correct order, AUC assigns a good score to the model, e.g., AUC for the model $f(z) = \mathbb{E}_{(x,y)\sim D}[y \mid x = z]/100$ is 1.0 while the scores are not close at all.

3.2.4 Area Under the Validity Curve (AUC_V)

Using a threshold on L_1 norm calibration error we would have the following function:

$$V(\epsilon) = \mathbb{P}_{(x,y)\sim D}[|f(x) - \mathbb{E}_{(x',y')\sim D}[y' \mid f(x') = f(x)]| \leqslant \epsilon]$$

This function provides a curve named *validity curve* whose integral is considered as metric AUC_V [16]. The relation between the area under this curve and the L_1 *norm calibration error* for any predictor $f : X \to [0, 1]$ has been shown as follows [16]:

$$\begin{aligned}
\mathrm{CE}_1(f) &= \mathop{\mathbb{E}}_{(x,y)\sim D}[|f(x) - \mathop{\mathbb{E}}_{(x',y')\sim D}[y' \mid f(x') = f(x)]|] \\
&= \int_0^1 \mathop{\mathbb{P}}_{(x,y)\sim D}[\, |f(x) - \mathop{\mathbb{E}}_{(x',y')\sim D}[y' \mid f(x') = f(x)]| > \epsilon]d\epsilon \\
&= 1 - \int_0^1 \mathop{\mathbb{P}}_{(x,y)\sim D}[|f(x) - \mathop{\mathbb{E}}_{(x',y')\sim D}[y' \mid f(x') = f(x)]| \leqslant \epsilon]d\epsilon \\
&= 1 - \int_0^1 V(\epsilon)d\epsilon \\
&= 1 - \mathrm{AUC}_V(f)
\end{aligned} \tag{3.1}$$

For a set of n samples $D^{(n)}$, AUC_V is estimated as the area under the following function [16]:

$$\hat{V}(\epsilon) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[|f(x_i) - \hat{\mathbb{E}}_{(x,y)\sim D}[y|f(x) = f(x_i)]| \leqslant \epsilon\right]$$

where $\hat{\mathbb{E}}_{(x,y)\sim D}[y \mid f(x) = p]| := \frac{\sum_{i=1}^{n} y_i \mathbb{1}\left[f(x_i)=p\right]}{\sum_{i=1}^{n} \mathbb{1}\left[f(x_i)=p\right]}$.

Since the empirical expectation counts the number of samples with the sample score and in the experiments we have finite data, we may not have an accurate count and need an appropriate binning method. The authors have applied equal-mass binning and utilized the average of scores and labels. However, this approach led to a challenge where they changed the generated scores and used the average scores within each bin.

K-Nearest Neighbor Based AUC_V Estimation

We introduce a different methodology to partition a finite number of samples and use them to estimate the metric AUC_V that is based on the K-nearest neighbor (KNN) algorithm. We began by identifying the 10 nearest neighbors for each score and averaged their labels to establish the empirical expectation for the label associated with that particular score. We use 10 as the parameter for KNN as the *log* of the size for the datasets.

Definition 7. *K-nearest neighbor based* AUC_V *estimation over* n *samples* $D^{(n)}$ *for predictor* $f : X \to [0, 1]$ *is the area under the produced validity curve by the following function using different values for* ϵ*:*

$$\hat{V}_{\mathrm{KNN}}(\epsilon) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[|f(x_i) - \hat{\mathbb{E}}_{\mathrm{KNN}(x,y)\sim D}[y|f(x) = f(x_i)]| \leqslant \epsilon\right],$$

where the empirical expectation estimation is

$$\hat{\mathbb{E}}_{\mathrm{KNN}(x,y)\sim D}[y \mid f(x) = p]| := \frac{\sum_{i=1}^{n} y_i \mathbb{1}\left[f(x_i) \in \text{k-nn}(p)\right]}{k}$$

In the empirical expectation estimation, k *is the number of neighbors used in KNN, and* k-nn(p) *is the set of* k *samples with the closest* f *scores to* p*.*

In our methodology, we take into account the scores of each individual, instead of relying on the average scores over different bins. The benefit of our approach is to avoid considering a modified version of the model rather than the actual outcomes. We have used this estimation rather than the equal-mass binning estimation in our experiments. We denote the KNN based AUC_V estimation by $\mathrm{AUC}_{V,\mathrm{KNN}}$.

3.2.5 Expected Calibration Error (ECE)

Criterion expected calibration error (ECE) compares the average predicted scores and the average of true labels with respect to a set of bins $b_1, b_2, ..., b_B$ [4]. The weight, denoted as w_i, of bin b_i represents the fraction of samples contained within this particular bin. The binning method that ECE is initially introduced with is uniform-mass binning. In our research, we have introduced a new technique for binning that is also used with ECE. Therefore, since the binning is not necessarily uniform-mass, weights $w_1, w_2, ..., w_B$ may not be equal. The L_p norm ECE is formally defined as follows:

$$\mathrm{ECE}_p := \big(\frac{1}{\sum_{i=1}^{B} w_i} \sum_{i=1}^{B} w_i * \mathrm{PCE}(b_i)^p\big)^{1/p}.$$

PCE or partition calibration error is the difference between the average of values generated by f and the average of labels in a bin that is computed as

$$\mathrm{PCE}(b_i) := \frac{|\sum_{j=1}^{n}(f(x_j) - y_j)\mathbb{1}\left[x_j \in b_i\right]|}{\sum_{j=1}^{n}\mathbb{1}\left[x_j \in b_i\right]}$$

When a pre-defined partitioning is not available, the produced scores are sorted and allocated into a fixed number (B) of equally weighted bins. This criterion is referred to as $\mathrm{ECE}_{B,p}$. As the bins have uniform weights, the criterion can be formulated as:

$$\mathrm{ECE}_{B,p} := \big(\frac{1}{B} \sum_{i=1}^{B} \mathrm{PCE}(b_i)^p\big)^{1/p}.$$

While ECE is widely used to evaluate calibration models ([17]–[19]), there are some problems with this metric that we dicuss in Section 3.3. Since this criterion uses binning, the

result of ECE involves a binning bias. ECE is very sensitive to the number of bins. Using too few bins can result in an inaccurate evaluation, and using too many bins can result in overfitting and a noisy evaluation.

3.2.6 Probability Deviation Error (PDE)

One major concern regarding the ECE criterion is that it relies on calculating the average of scores within each bin. However, if there happen to be both under-estimated and over-estimated scores within the same bin, the ECE fails to detect this discrepancy (Example 1). To address this issue, we propose a new metric called the probability deviation error (PDE). The PDE compares the point-wise scores with the average label in each bin, thereby fixing the problem associated with the ECE. On predefined bins $b_1, b_2, ..., b_B$ with weights $w_1, w_2, ..., w_B$, the L_p norm PDE is defined as follows.

$$\text{PDE}_p := \big(\frac{1}{\sum_{i=1}^{B} w_i} \sum_{i=1}^{B} w_i * \text{PPD}(b_i)^p\big)^{1/p}.$$

PPD or partition probability deviation is the average difference between point-wise scores generated by f in a bin and their average label that is

$$\text{PPD}(b_i) := \frac{\sum_{j=1}^{n} |f(x_j) - \hat{y}_i| \mathbb{1}\left[x_j \in b_i\right]}{\sum_{j=1}^{n} \mathbb{1}\left[x_j \in b_i\right]}$$

where $\hat{y}_i = \frac{y_j \sum_{j=1}^{n} \mathbb{1}\left[x_j \in b_i\right]}{\sum_{j=1}^{n} \mathbb{1}\left[x_j \in b_i\right]}$.

In the case that there is no pre-defined partitioning, the generated scores are sorted and divided into a fixed number (B) of uniform mass bins. In this case, we denote the criterion

as $\mathrm{PDE}_{B,p}$. Since the weights of the bins are uniform, the criterion can be expressed as

$$\mathrm{PDE}_{B,p} := \big(\frac{1}{B}\sum_{i=1}^{B}\mathrm{PPD}(b_i)^p\big)^{1/p}.$$

3.3 Critiquing Expected Calibration Error

The expected calibration error is a popular metric that has been extensively employed in calibration studies [17]–[19]. The ECE method, employs the average scores within each bin, rather than examining individual scores. This approach makes the metric less accurate. A primary limitation of this method is that when a bin contains both overestimated and underestimated scores, they negate each other, resulting in the ECE not catching any error for them. To illustrate this issue, consider the following example:

Example 1. *In order to compute the expected calibration error, we need to partition the samples. Upon dividing the scores into bins, one particular bin contains several samples, with half of them having a score of 0.35 and the other half having a score of 0.65. Assume that from the distribution, the regression function* $\eta(x) = 0.5$ *for all* x *in this bin. In this situation, the expected average label for the bin would also be* 0.5*. Consequently, when calculating the error using the ECE method for this bin, the result would be* $|\frac{0.35+0.65}{2} - 0.5| = 0$*, indicating no error. In contrast, the probability deviation error method takes individual scores into account. For the same bin as in example, the error calculated using the PDE method would be* $\frac{|0.35-0.5|+|0.65-0.5|}{2} = 0.15$*, which demonstrates zero bias, the bias is defined in Definition 9. The PDE approach offers a more accurate representation of the correct calibration by*

considering the individual scores in each bin, avoiding the cancelation effect experienced in the ECE method.

3.4 Discussion on Individual vs. Binning Metrics

This section compares two approaches for computing calibration metrics in machine learning models: individual or point-wise labels and scores, and binning method averages. When using the individual approach, the score or label assigned to each individual sample is determined independently of the other samples in the same bin. On the other hand, when using a binning approach, a specific binning method is employed, and the average score or label of all samples within a bin/region is used as a representation of the overall outcome or label for that bin.

In contrast to classification problems, the optimum goal in calibration is to achieve scores close to the regression function η. Since the regression function η is not accessible, an estimation of η is used to evaluate the calibration models. This makes defining regions an essential step in evaluating calibration models. To define the regions, there are different approaches that are explained below.

3.4.1 Uniform-Mass Binning

Using this method, all created regions contain the same number of samples. First, the samples are arranged in order of their respective scores. Then, they are separated into the required number of bins. The samples that have close scores will be in the same region. Uniform-mass binning is one of the most popular binning methods and it is the method ECE is initially

introduced with [4].

3.4.2 Binning via KNN

To the best of our knowledge, this is the first time that K-Nearest Neighbor (KNN) method has been used to define the regions for a calibration metric. In this approach, for each sample, a different region is defined. The region for each sample is the K closest samples with respect to their scores. The same as uniform-mass binning, the samples with similar scores will be in each other's region.

3.4.3 BFSL Binning

We have introduced Breadth First Search Leaf (BFSL) binning approach that is applicable to any tree-based calibration models. There are several calibration models that are using a tree structure ([3], [20]). Using BFSL, we take advantage of the structure of the calibration models to define the regions. The main motivation of this approach is to provide a more interpretable and explainable way of partitioning the predicted probabilities as interpretability is one of the desired properties in calibration framework. First, by performing a breadth first search starting from the root of the original tree, the shortest sub-tree with B leaves is extracted, in which B is the required number of bins. The leaves of this sub-tree are the regions generated by this method. In this method, samples that follow similar paths in the tree are in the same region. The number of bins can be chosen based on the specific requirements of the analysis, and the resulting bins are expected to capture the properties of the model's behavior. The

benefits of this approach include the ability to interpret regions due to the utilization of the tree's structure. Furthermore, these regions rely on the characteristics of the samples, not just their scores. In some of the experiments, we have used the leaves generated by the original tree in the calibration model without performing breadth first search. This partitioning allows us to evaluate the model using the structure of the tree without additional pruning.

In evaluation, we don't have access to the regression function η and only a finite number of samples are available. As a result, when we want to obtain an expected value for η for a given sample, we need to calculate the average of labels over the other available samples that are similar to the given one. However, when it comes to scores, there is no real benefit in averaging them and replacing the actual outcome with their mean value since we have the individual score for each sample. The metrics that consider output from the bins rather than the individual scores are actually assessing a tweaked version of the model. It may not truly represent what the original model is like.

Table 3.1 presents an analysis of the evaluation metrics we have employed, with a focus on how they impact the scores of calibrated models or the ground truth labels based on binning. To illustrate, let us consider the case of $\mathrm{AUC}_{V,\mathrm{KNN}}$. Here, the ground truth labels of each sample are replaced by the average of other samples' labels in the K-nearest neighbor of the score, but the score assigned to each sample remains unchanged.

Metric	Generated score	Ground truth
RMSE	-	-
Classification loss	-	-
AUC	-	-
$\text{AUC}_{V,\text{KNN}}$	-	KNN
ECE	Uniform-mass /BFSL	Uniform-mass /BFSL
PDE	-	Uniform-mass /BFSL

Table 3.1: The table displays metrics and indicates whether they have modified the scores generated by models or the ground truth labels based on a binning on the samples. We denote the situations where the score or label is not altered based on the other samples by using a dash ("-"). In the other cases, the average of the score or label over the regions generated by the mentioned approach is considered rather than the individual score or label assigned to the sample.

Chapter 4

Experiments

In our research, we have conducted three experimental analyses, each addressing a distinct aspect in machine learning calibration. The initial experiment was designed with the goal of contrasting two calibration metrics, ECE and PDE. This study further delved into the impacts of varying binning strategies, contrasting between BFSL and uniform-mass binning. To evaluate the precision of these metrics, their outcomes were compared with a different metric known as Theoretical Calibration Error (TCE). This metric is contingent upon the values of the regression function. As the regression function is not directly accessible, we chose the generation and usage of synthetic datasets for this purpose.

The objective of our second experiment was to conduct a comparative analysis of various calibration techniques utilizing real world datasets. The calibration models under assessment included Platt scaling, isotonic regression, probability calibration trees, and decision trees. We analyzed various dimensions of these models employing a range of metrics. The metrics

incorporated in this phase include both calibration and classification measures. We utilized RMSE, classification loss, AUC, ECE combined with uniform-mass binning, and PDE on the leaves of the calibration models' trees as the metrics for this evaluation.

The final experiment aims to analyse a tradeoff we have observed between being an accurate classifier and generating calibrated probabilities. We train decision tree models of varying complexities, denoted by their sizes, on both synthetic and real world datasets. The models thus trained are assessed using PDE as a calibration measure and classification loss as an evaluation of their classification proficiency.

In the ensuing section, we outline the structure and design of our experimental setup. Subsequent sections are dedicated to detailing each of our individual experiments and the associated analysis.

4.1 Experiment Setup

4.1.1 Real World Datasets

We have used a range of datasets for binary classification and multi-class classification problems. All 36 datasets used in our experiments are from UCI [21]. These datasets are summarized in Table 4.1.

Dataset	Instances	Attributes	Classes
audiology	226	69	24
bank-marketing	41188	19	2
bankruptcy	10503	64	2
car-evaluation	1728	6	4
cervical-cancer	858	32	2
colposcopy	287	62	2
credit-rating	690	15	2
cylinder-bands	512	39	2
german-credit	1000	20	2
hand-postures	78095	39	2
htru2	17898	8	2
iris	150	4	3
kr-vs-kp	3196	36	2
mfeat-factors	2000	216	10
mfeat-fourier	2000	76	10
mfeat-karhunen	2000	64	10
mfeat-morph	2000	6	10
mfeat-pixel	2000	240	10
mice-protein	1080	80	8
new-thyroid	215	5	3
news-popularity	39644	59	2
nursery	12960	8	5
optdigits	5620	64	10
page-blocks	5473	10	5
pendigits	10992	16	10
phishing	1353	10	3
pima-diabetes	768	8	2
segment	2310	20	7
shuttle	58000	9	7
sick	3772	29	2
spambase	4601	57	2
taiwan-credit	30000	23	2
tic-tac-toe	958	9	2
vote	435	16	2
vowel	990	14	10
yeast	1484	8	10

Table 4.1: Real world datasets used in our experiments.

4.1.2 Synthetic Datasets

For certain aspects of our analysis, we require data with known underlying distributions, which isn't always achievable with real world data. To meet this need, we created two synthetic dataset generators.

The first synthetic dataset generators is designed for a binary classification problem and incorporates three distinct features, hence we've named it *Synthetic-3*, referring to the three generated features. To generate this dataset, we use a Gaussian mixture model with ten components to create the initial attributes for each sample. The true probability for each data point is then calculated using a polynomial function of the features followed by a sigmoid, representing the regression function of the underlying distribution for this dataset. This generates a probability value that represents the ground truth for each sample. Lastly, we assign binary labels to the samples using a random process, where the selection probability is weighted according to the computed true probabilities. This procedure ensures that the resulting dataset aligns with a known distribution, providing us with a valuable resource for our subsequent analysis.

The regression function for Synthetic-3 is constructed in such a way that approximately half of the components correspond to the first class, while the remaining components represent the other class. This ensures a balanced dataset. Throughout the entire generated dataset, the average value of the regression function ranged between 50% and 55% over different experiments, further confirming that the dataset is balanced. The distribution of 1000 samples generated by Synthetic-3 is shown in Figure 4.1.

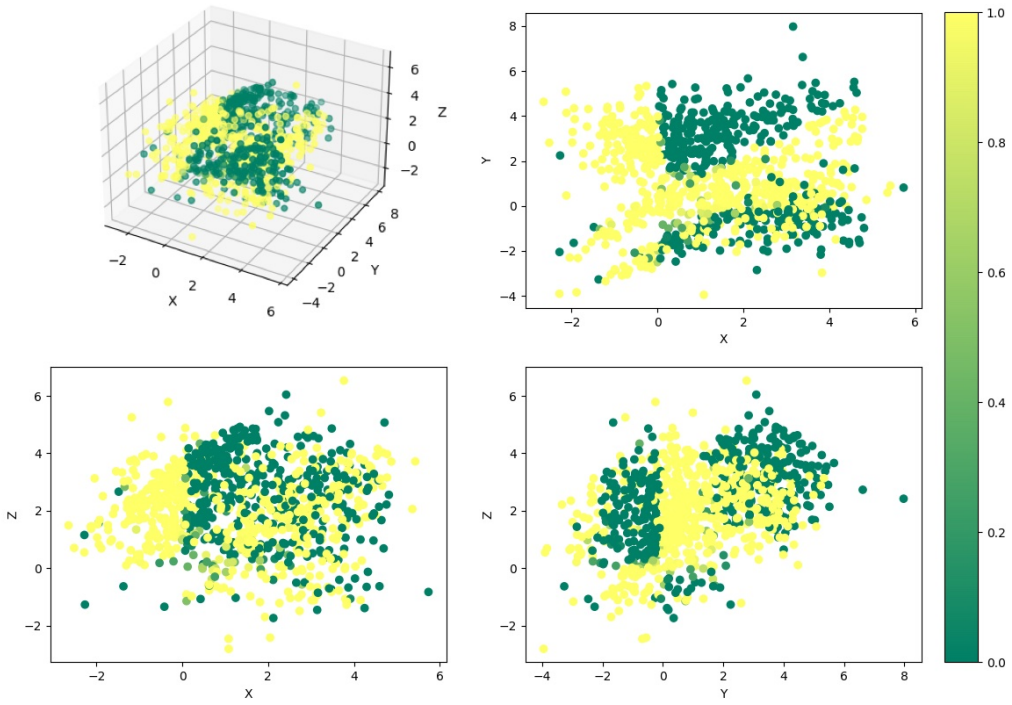


Figure 4.1: 1000 samples are generated using Synthetic-3. The colors indicate the likelihood of each sample being labeled as class one.

We've also created a second synthetic dataset with the goal of enhancing the robustness of our analysis and allow us to further validate our findings. This new dataset, namely *Synthetic-5*, consists of five features and is generated using the Gaussian mixture distribution with 14 components. In comparison to its predecessor, Synthetic-3, this dataset is considerably more complex due to the greater number of features and the intricacy of the polynomial function used for the regressing function, which contains a higher number of sub-expressions. The average value of the regressing function ranged between 56% and 62%.

4.1.3 Calibration Metric Bias

In order to effectively assess the proficiency and accuracy of calibration metrics, we compare their outcomes with a benchmark known as the Theoretical Calibration Error (TCE) [20]. The TCE is a theoretical construct designed to measure the error with respect to the regression function which is the optimum goal for estimation in machine learning calibration. If we have scores close to the regression function, the model is calibrated and optimal classifier as we have shown in Section . The TCE is defined as follows:

Definition 8. *Theoretical Calibration Error (TCE) for predictor $f : X \to [0, 1]$ over n samples $D^{(n)}$ with respect to the distribution D over $X \times Y$ is defined as ([20]):*

$$\begin{aligned}\mathrm{TCE}(D, D^{(n)}, f) &\coloneqq \frac{1}{n}\sum_{i=1}^{n}(|f(x_i) - \mathop{\mathbb{E}}_{(x,y)\sim D}[y \mid x = x_i]|) \\ &= \frac{1}{n}\sum_{i=1}^{n}(|f(x_i) - \eta_D(x_i)|),\end{aligned}$$

in which η_D is the regression function on distribution D.

The TCE serves as a foundation to define the concept of bias in the context of calibration metrics. Bias, in this case, refers to the deviation between the outcomes of the calibration metrics and the TCE. It is mathematically defined as follows:

Definition 9. *The bias of a calibration metric μ for predictor $f : X \to [0, 1]$ over n samples $D^{(n)}$ with respect to the distribution D over $X \times Y$ is defined as ([20]):*

$$\mathrm{Bias}(D, D^{(n)}, \mu) \coloneqq \frac{1}{m}\sum_{i=1}^{m}|\mu(D^{(n)}, f) - \mathrm{TCE}(D, D^{(n)}, f)|,$$

where m is the number of experiments to estimate the bias, which is 10 in our analysis.

This definition of bias offers a powerful tool for the comparative analysis of various calibration metrics. By measuring the extent to which each metric deviates from the TCE, we can effectively rank them based on their accuracy and reliability. This helps identify the metrics that most closely estimate the error in calibration. Both TCE and bias are predicated on the regression function. However, in practical scenarios involving real world datasets, this regression function is typically inaccessible. To overcome this issue and facilitate our analyses involving TCE and bias, we have employed synthetic datasets described in Section 4.1.2.

4.2 Calibration Metrics Analysis

Following our critique of the metric expected calibration error in Section 3.3, now we analyse it and compare the bias in Definition 9 of metrics ECE and PDE to discuss their proficiency in capturing the error of models in generating calibrated probabilities. Another aspect of the metrics that is examined in the analysis is their approach to partition the sample space. In this step, we have compared BFSL and unifrom-mass binning using the bias. To have access to the regression function and thus the bias, we have used synthetic datasets mentioned in Section 4.1.2.

In this phase of our analysis, we assess the metrics ECE and PDE by comparing their scores to the theoretical calibration error, defined in Definition 8, which is the main factor of concern in calibration [20], and computing their bias with respect to the Definition 9.

During this phase, we aim to analyze another aspect of the metrics, which is the partitioning

method. Previous research has primarily relied on the uniform-mass binning method, which is often used in conjunction with ECE. However, we proposed a new partitioning method that is based on the nodes of a tree-based calibration model. This method is called *breadth first search leaf (BFSL)* and is defined in Section 3.4.3. It is important to note the nodes selected through BFSL process are only used to partition and create the regions and the scores are still being generated by the original tree and its leaves. In this experiment, the cost-complexity post-pruning step has been eliminated because the tree grown by PCT falls short of the minimum requirement of having at least B nodes in the tree.

For this experiment, we have used the synthetic datasets described in Section 4.1.2 as we require the regression function to measure the bias of the metrics. In each of our experiments, we generate a total of 50,000 samples. The base model is trained on 45% of these samples, while the calibrator utilizes the remaining 55%. For each experiment, required number of test samples are generated following the same procedure in addition to the previously generated samples.

The settings for the experiment are as follows:

- The metric used for the evaluation of the model is either PDE or ECE.
- The partitioning required for the evaluation metric is done using either uniform-mass or BFSL binning.
- The experiments are done on a PCT model and a decision tree.
- Both Synthetic-3 and Synthetic-5 procedures are used.

- Each setting is evaluated in 10 experiments.

The results of this experiment on datasets generated by Synthetic-3 and Synthetic-5 are presented in Figure 4.4 and Figure 4.5, respectively.

Our analysis indicates that, in the context of evaluating calibration models, using PDE as the metric and BFSL binning using the calibration tree is the most effective strategy. It is important to note that a significant number of bins are required to optimize this process, with 64 bins proving to be ideal in our case. Interestingly, even when using ECE as a metric, BFSL binning results in lower bias compared to uniform-mass binning. If a calibration tree isn't available and uniform-mass binning is applied for partitioning, PDE still outperforms ECE.

However, it's important to clarify that BFSL partitioning is not universally ideal. It requires a tree-based calibrator which isn't always possible. Moreover, while the number of bins must be sufficiently high for this partitioning method to be effective, it can pose challenges when sample sizes are relatively small. The other requirement for this method is that the calibrator tree should have at least as many leaves as the number of required bins which makes this method applicable only on sufficiently complex trees.

4.3 Evaluating Calibration Methods

We have evaluated models Platt scaling, isotonic regression, probability calibration tree, and decision tree using different measures to evaluate them in terms of classification, monotonicity, and calibration. These experiments are done using all real world datasets mentioned in

Method	RMSE	Classification loss	AUC	$\text{ECE}_{B=32,p=1}$	$\text{PDE}_{p=1}$	$\text{AUC}_{V,\text{KNN}}$
PS	21	22	27	7	-	5
IR	25	24	28	14	-	9
PCT	**31**	**31**	**32**	**26**	19	**15**
DT	24	24	19	21	**36**	7

Table 4.2: The calibration methods are assessed and compared using different type of metrics. The numbers are the winning times of each model based on each metric. The numbers in each column do not add up to the number of datasets as multiple models may have won simultaneously. The pre-fixed bins for metric PDE are the leaves generated by PCT and DT. The number of neighbors used in $\text{AUC}_{V,\text{KNN}}$ are ten.

the experiment setup. Evaluation is done in 10 iterations, and each time the samples are randomly arranged. Training, calibration, and test sets include 40.5%, 49.5% and 10% of samples, respectively. Training of the tree-based calibration models used in the experiments involve a cost-complexity post-pruning step. To find the optimal cost-complexity parameter, a 5-fold cross-validation is done on the calibration set. After this step, the whole calibration set is used to train the calibration model and the model is pruned using the found optimal factor.

The results are shown in Table 4.2. To evaluate the models using ECE, 32 bins are generated by sorting the scores. For PDE, the generated bins by the predictor are used which are the leaves of the tree in model PCT and DT. Since there are no regions generated from

models PS and IR, there is no PDE score for these models. By utilizing the leaves of tree models as the bins in evaluation, we are able to take into account the interpretability of the models. For both ECE and PDE, L1-norm is reported.

We have further assessed the calibration models using the area under the validity curve or AUC_V. An estimation of AUC_V was implemented using the K-nearest neighbours approach, in particular, $\text{AUC}_{V,\text{KNN}}$ with 10 neighbours was applied. The validity curve was derived for each model and dataset by leveraging the KNN method. The comparison results of the area under the validity curve for each dataset between the models are illustrated in Table 4.2. The averaged validity curves across all datasets are visualized in Figure 4.2 for each respective model. In Figure 4.3, we have shown these curves over a few of datasets used in this experiment.

For the models PS, IR, PCT, and DT, the $\text{AUC}_{V,\text{KNN}}$ values obtained are 0.933, 0.939, 0.943, and 0.937, respectively. This evaluation showcases a marginally superior performance from the PCT model. However, the observed differences in scores are not significant as it has been shown in Figure 4.2 and Figure 4.3, thereby these minor differences prevent us from making a conclusive evaluation of the models based solely on this metric.

As demonstrated in Section 4.2, the Probability Deviation Error (PDE) metric exhibits the least bias, particularly on the nodes of the calibration tree, when measuring the calibration error of a model. Given this finding, a comparison of the models using the results in Table 4.2 suggests that decision trees perform significantly better in generating calibrated probabilities, without any losses against the other models across all datasets.

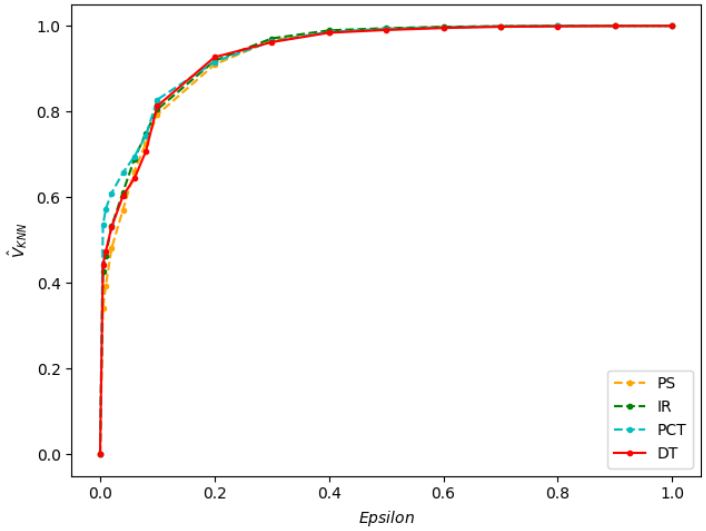
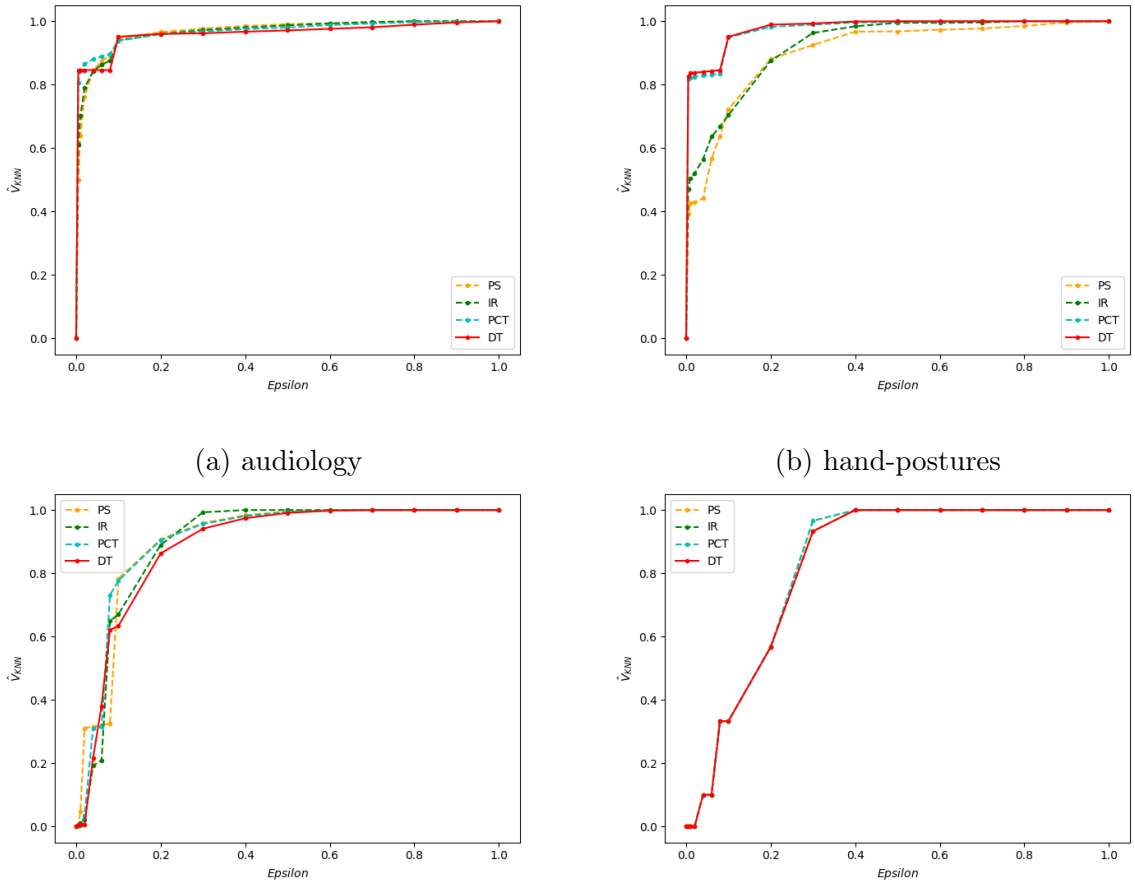


Figure 4.2: Averaged validity curves for the calibration models across all datasets. $\hat{V}_{\text{KNN}}$ represents the KNN estimation of the validity curve defined in Definition 7.

However, when considering classification performance, Probability Calibration Trees (PCTs) appear to be superior, yielding the least errors in most instances. This observation implies that while PCTs may not be the most effective in calibration, they serve as excellent classifiers. These results suggest that decision trees should be considered as one of the top options. Their ability to deliver high-quality calibrated probabilities while they are interpretable underscores their importance and potential in calibration-focused machine learning tasks.



(a) audiology

(b) hand-postures

(c) bank-marketing

(d) news-popularity

Figure 4.3: Validity curves for the calibration models over a few of datasets. $\hat{V}_{\text{KNN}}$ represents the KNN estimation of the validity curve defined in Definition 7.

4.4 Calibration-Classification Tradeoff

The experiment begins with an evaluation of decision tree models trained on multiple datasets. These models varied in complexity, as characterized by the size of the tree, i.e., the number of leaves it contains. For each dataset, decision trees of different sizes were trained and subsequently evaluated using two key metrics: Probability deviation error (PDE) and

classification loss. We allocated half of each dataset for training the decision tree, with the remaining portion serving as the test set. The datasets we have used are the 36 UCI datasets described in Section 4.1.1 and two synthetic datasets with 50,000 samples generated using the procedures explained in Section 4.1.2.

PDE, a calibration metric, was employed to assess the quality of the predicted probabilities produced by the models. It offers a measure of the divergence between the predicted and actual class probabilities. On the other hand, classification loss, a performance metric, was used to evaluate the model's proficiency in correctly classifying instances. It quantifies the discrepancy between predicted and actual class labels.

Additionally, we incorporated the cost-complexity pruned decision tree to contrast its performance against the models trained in this experiment.

The results observed from the experiment were insightful. Figure 4.6 presents the findings of this experiment for a select group of datasets. Firstly, we found that PDE consistently increased as the size of the decision tree increased across all datasets. This trend indicates that as the decision trees grew in complexity, the calibration quality decreased. In other words, the models' predicted probabilities became less representative of the true class probabilities as the decision trees became larger. The deductions proved in Section supports the results here as we have shown by merging the cells induced by predictor, we will improve the calibration, and by decreasing the size of the decision trees we are doing the same action.

Contrary to the PDE trend, classification loss generally decreased as the size of the decision tree increased. This indicates that more complex trees were typically more successful

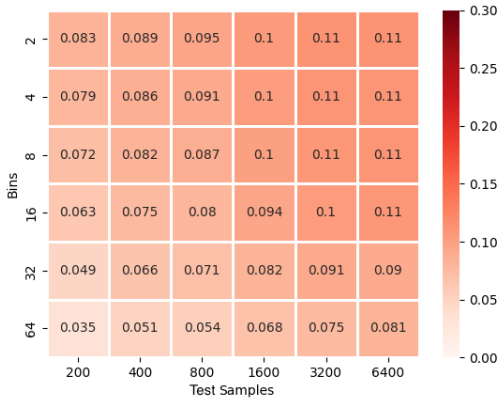
in their classification tasks. This characteristic persists until the decision tree reaches a size that leads to overfitting on the data. The specific tree size at which this occurs varies across datasets, dependent on their unique characteristics. Figures 4.6a and 4.6c represent this characteristic. However, it's worth noting that this was not an absolute trend, as exceptions have been observed in a few of datasets, a case in point being the dataset represented in Figure 4.6e.

The performance of cost-complexity pruned decision trees compared to pre-pruned trees trained in this experiment was examined using a combination of PDE and classification loss metrics. The outcomes varied considerably across the different datasets. In 11 datasets, such as those represented in Figures 4.6c and 4.6d, the performance of post-pruned trees was observed to be weaker. Conversely, in 9 datasets, like those represented in Figures 4.6a and 4.6b, post-pruned trees exhibited superior performance. In the remaining 18 datasets, such as those shown in Figures 4.6e and 4.6f, the performance of the post-pruned trees remained largely unchanged. These findings suggest that while the effectiveness of cost-complexity pruning can vary based on the unique characteristics of each dataset, its overall impact seems to be quite subtle.

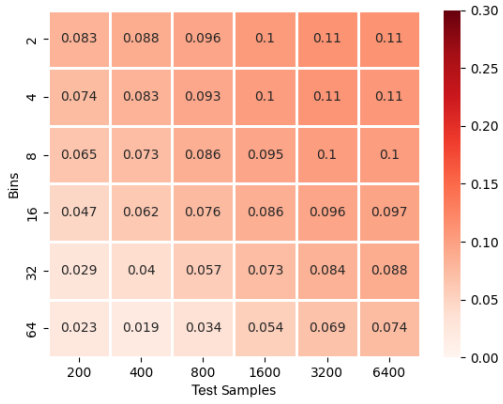
In conclusion, our experiment provides evidence supporting the existence of a tradeoff between calibration and classification performance in decision tree models. As decision trees increase in size and complexity, they generally become better at classification as long as they are not overfitted but worse in terms of calibration. It should be noted, however, that this relationship is not universal and can be influenced by dataset-specific factors. Additionally,

The performance of a post-pruned tree is nearly identical to that of a decision tree of the same size, and this may be contingent on the distinct characteristics of the data.

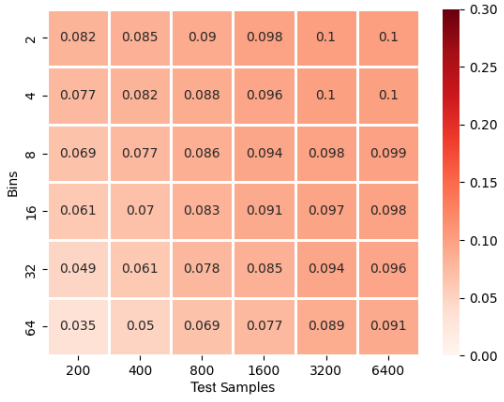
The comprehensive results from all datasets utilized in this experiment can be found in the Appendix, specifically in section A.1.



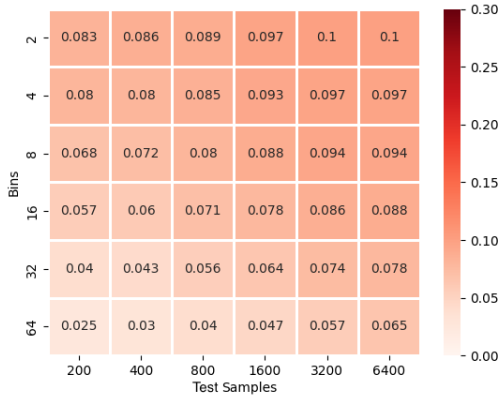
(a) Bias in uniform-mass using ECE for a DT



(b) Bias in BFSL using ECE for a DT

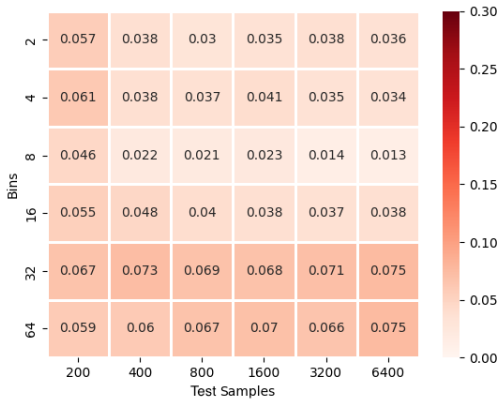


(c) Bias in uniform-mass using ECE for a PCT

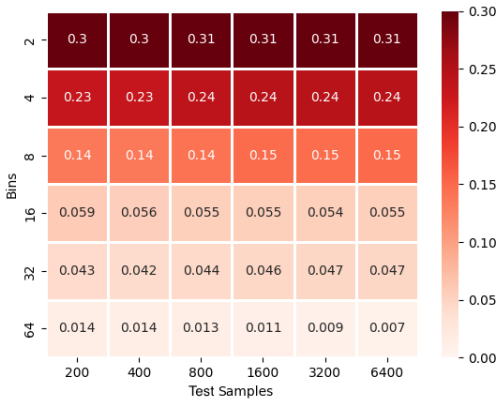


(d) Bias in BFSL using ECE for a PCT

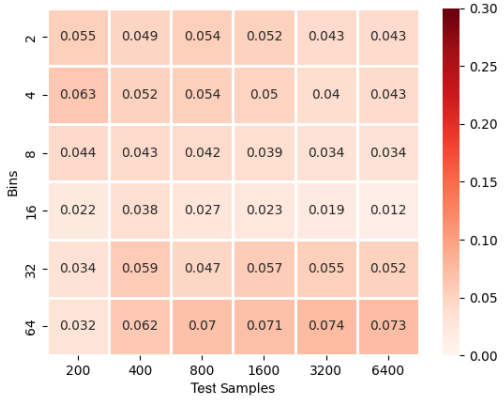
Figure 4.4: Continued on next page



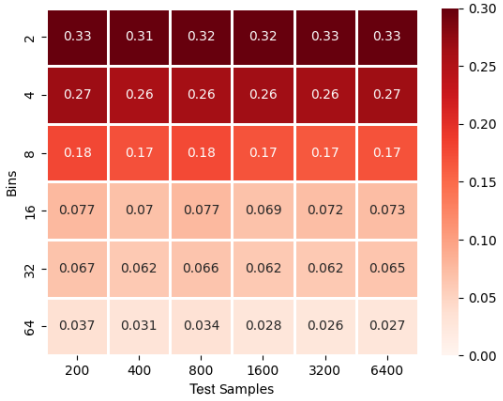
(e) Bias in uniform-mass using PDE for a DT



(f) Bias in BFSL using PDE for a DT

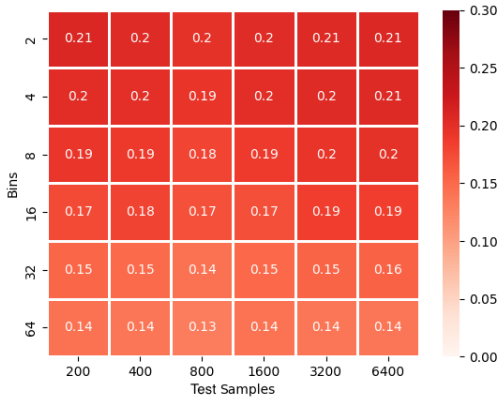


(g) Bias in uniform-mass using PDE for a PCT

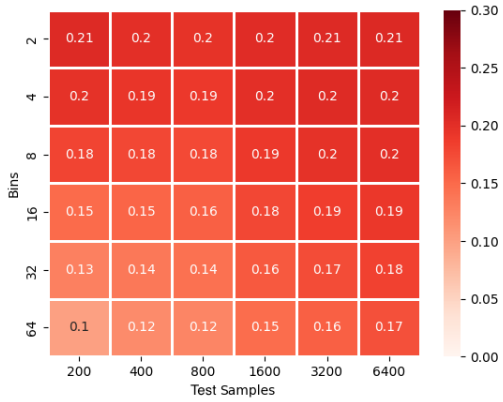


(h) Bias in BFSL using PDE for a PCT

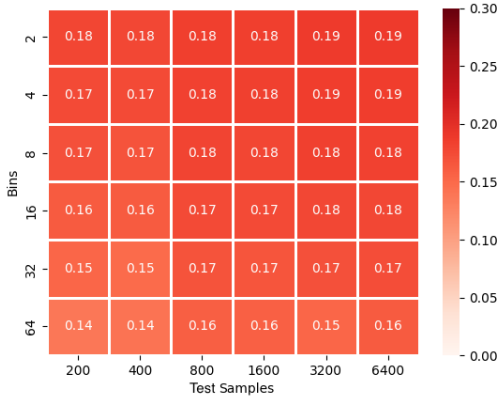
Figure 4.4: Evaluating the bias in various calibration metrics, in conjunction with two partitioning techniques, applied to a decision tree and a probabilistic calibration tree. Models were trained on 50,000 samples generated using procedure Synthetic-3.



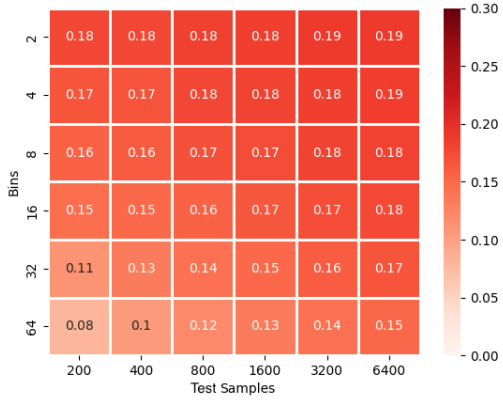
(a) Bias in uniform-mass using ECE for a DT



(b) Bias in BFSL using ECE for a DT

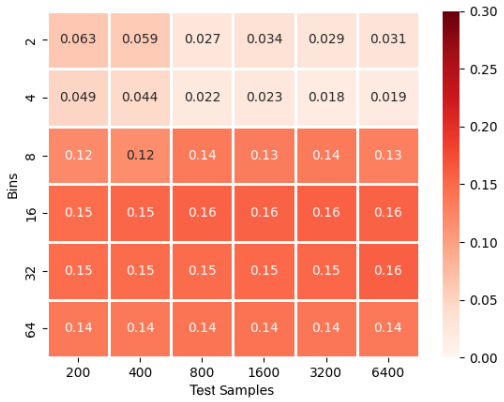


(c) Bias in uniform-mass using ECE for a PCT

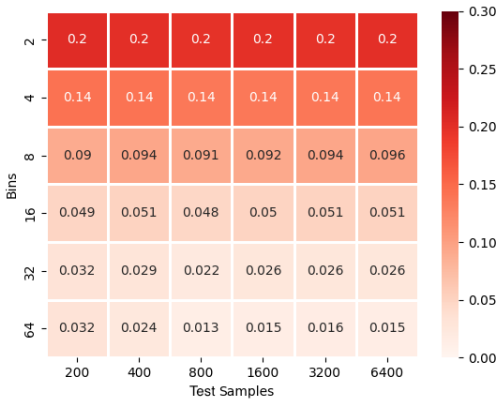


(d) Bias in BFSL using ECE for a PCT

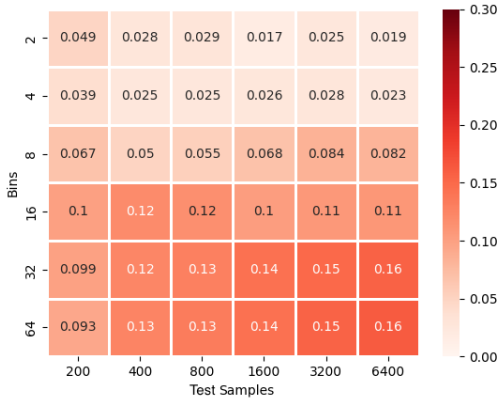
Figure 4.5: Continued on next page



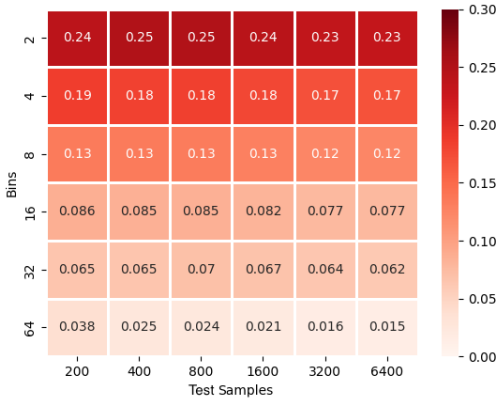
(e) Bias in uniform-mass using PDE for a DT



(f) Bias in BFSL using PDE for a DT

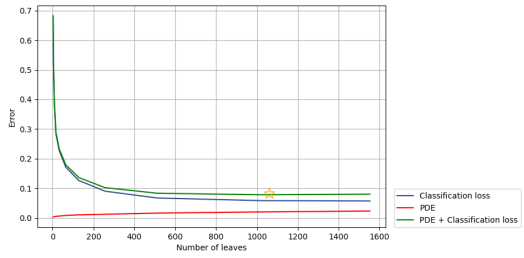


(g) Bias in uniform-mass using PDE for a PCT

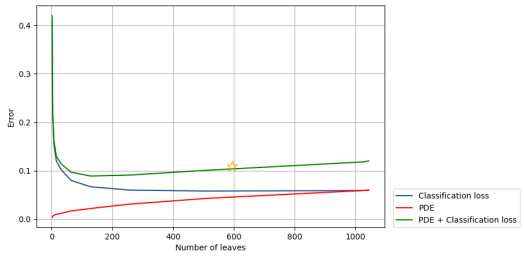


(h) Bias in BFSL using PDE for a PCT

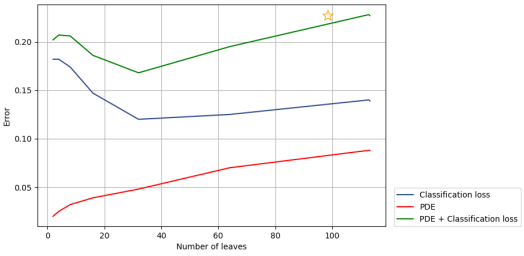
Figure 4.5: Evaluating the bias in various calibration metrics, in conjunction with two partitioning techniques, applied to a decision tree and a probabilistic calibration tree. Models were trained on 50,000 samples generated using procedure Synthetic-5.



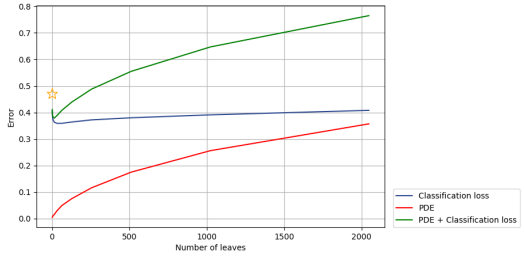
(a) hand-postures



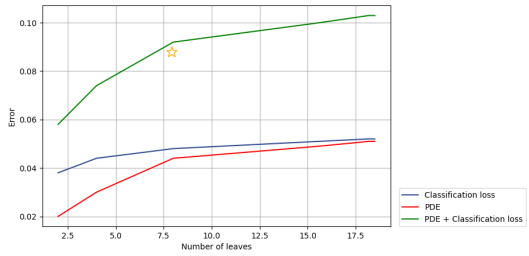
(b) Synthetic-5 generator with 50,000 samples



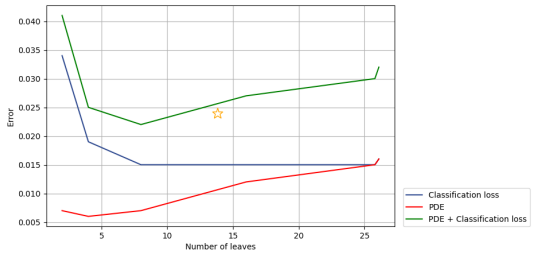
(c) phishing



(d) news-popularity



(e) vote



(f) sick

Figure 4.6: Evaluating decision trees of varying sizes across multiple datasets using probability deviation error (PDE) and classification loss. The star symbol (☆) denotes the combined total of PDE and classification loss for the decision tree that has undergone cost-complexity post-pruning, also signifying its size.

Bibliography

[1] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classif.*, vol. 10, 2000.

[2] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2002, pp. 694–699.

[3] T. Leathart, E. Frank, G. Holmes, and B. Pfahringer, "Probability calibration trees," *CoRR*, 2018.

[4] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, *Binary classifier calibration: Non-parametric approach*, 2014.

[5] K. Nguyen and B. O'Connor, "Posterior calibration and exploratory analysis for natural language processing models," *CoRR*, vol. abs/1508.05154, 2015. arXiv: `1508.05154`.

[6] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," *CoRR*, vol. abs/1812.04606, 2018.

[7] A. Kumar, P. S. Liang, and T. Ma, "Verified uncertainty calibration," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019.

[8] V. Kuleshov and P. S. Liang, "Calibrated structured prediction," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015.

[9] L. Puka, "Kendall's tau," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 713–715.

[10] J. L. W. V. Jensen, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta Mathematica*, vol. 30, no. none, pp. 175–193, 1906. DOI: 10.1007/BF02418571. [Online]. Available: https://doi.org/10.1007/BF02418571.

[11] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," *ICML*, 2001.

[12] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning*, Association for Computing Machinery, 2005.

[13] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006, ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207006000239.

[14] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modelling under imbalanced distributions," *ArXiv*, vol. abs/1505.01658, 2015. [Online]. Available: `https://api.semanticscholar.org/CorpusID:16780476`.

[15] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997, ISSN: 0031-3203.

[16] C. Gupta and A. Ramdas, "Distribution-free calibration guarantees for histogram binning without sample splitting," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, PMLR, 2021, pp. 3942–3952.

[17] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI Press, 2015, pp. 2901–2907.

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, *On calibration of modern neural networks*, 2017. arXiv: `1706.04599 [cs.LG]`.

[19] J. Błasiok, P. Gopalan, L. Hu, and P. Nakkiran, "A unifying theory of distance from calibration," *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 2022.

[20] S. Huang, Y. Wang, L. Mou, *et al.*, "Mbct: Tree-based feature-aware binning for individual uncertainty calibration," Association for Computing Machinery, 2022, pp. 2236–2246.

[21] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: `http://archive.ics.uci.edu/ml`.

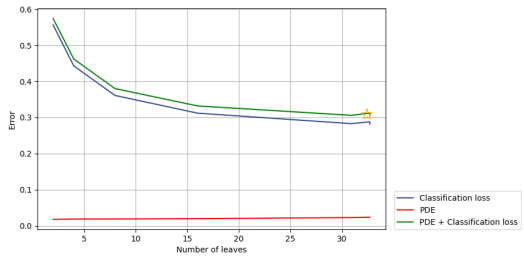
Appendix A

Supplementary Materials

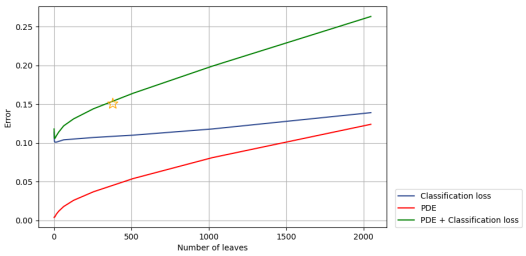
In this chapter of appendix, we present additional materials that supplement the primary content of our thesis.

A.1 Calibration-Classification Tradeoff Extended Results

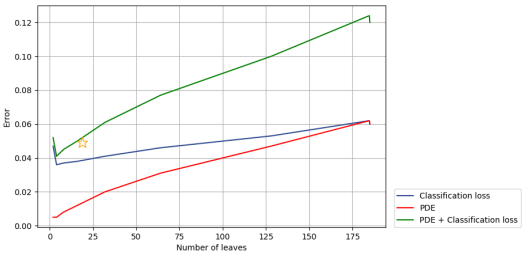
This section presents the results from the experiment conducted as detailed in Section 4.4. These findings, derived from analysis across all the datasets used in our study, are concisely presented in Figure A.1.



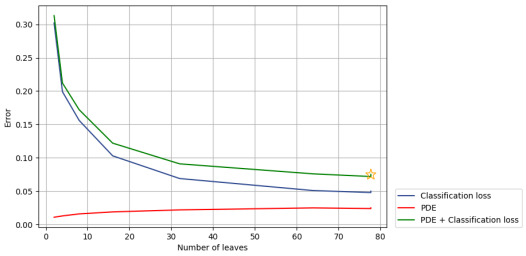
audiology



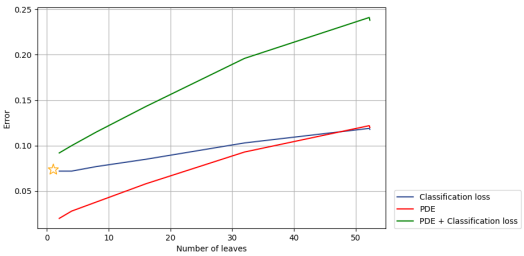
bank-marketing



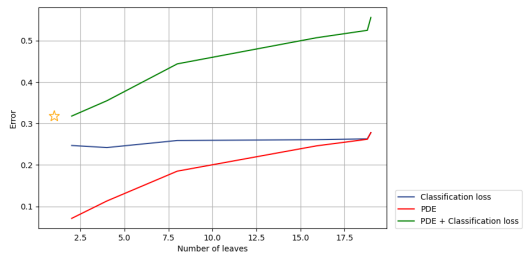
bankruptcy



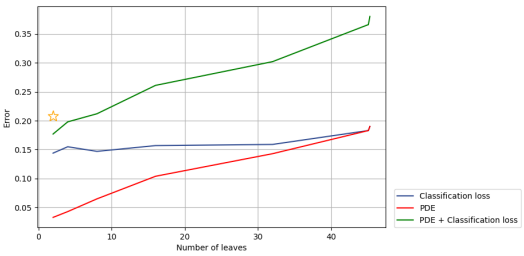
car-evaluation



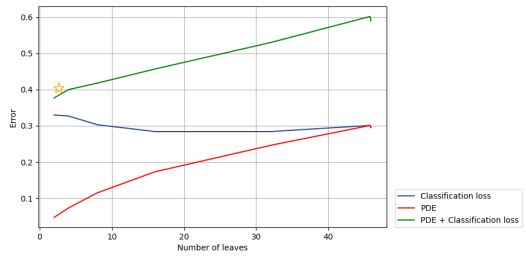
cervical-cancer



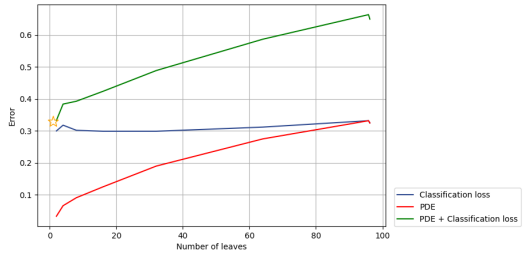
colposcopy



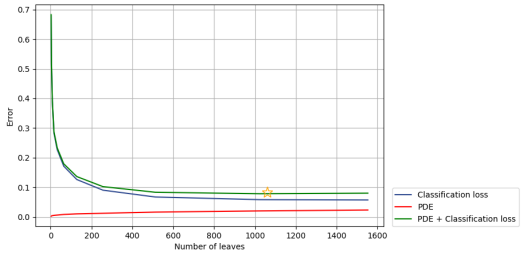
credit-rating



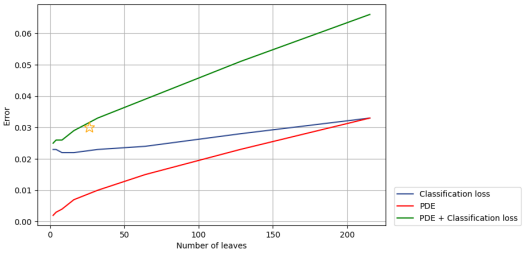
cylinder-bands



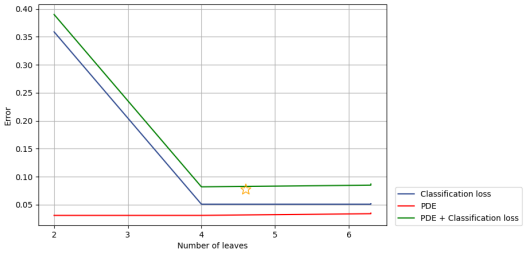
german-credit



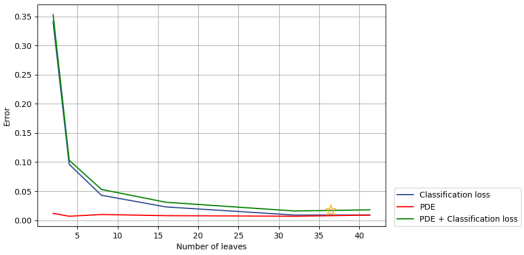
hand-postures



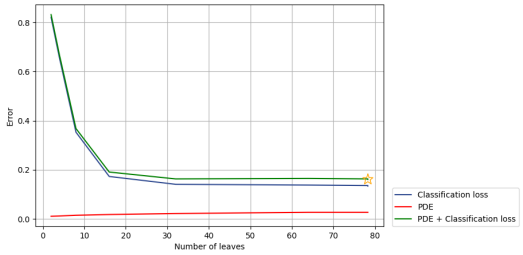
htru2



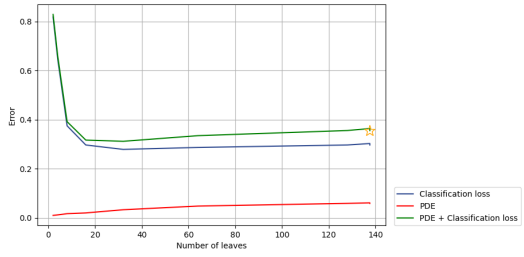
iris



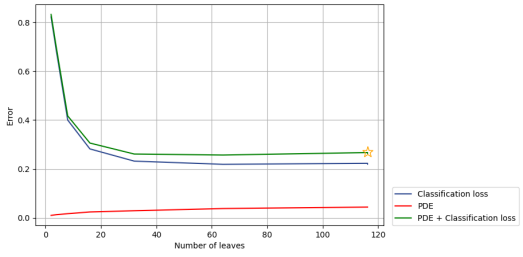
kr-vs-kp



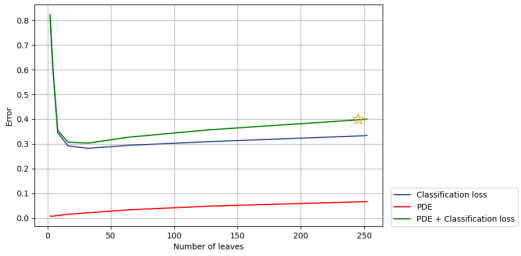
mfeat-factors



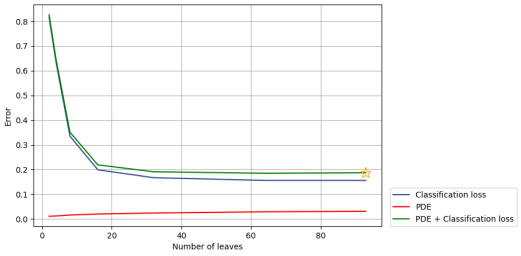
mfeat-fourier



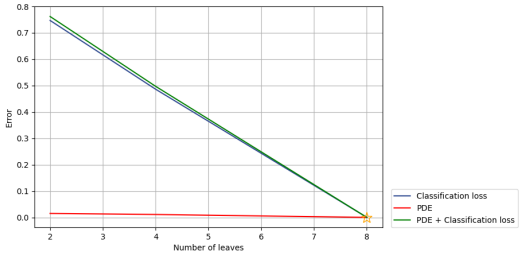
mfeat-karhunen



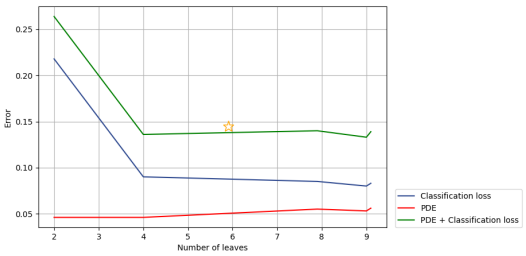
mfeat-morph



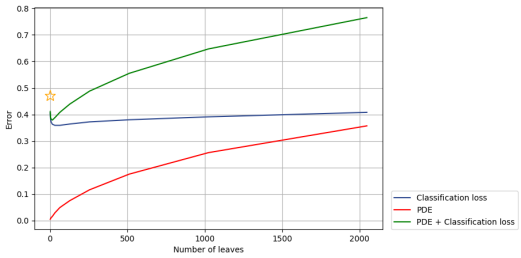
mfeat-pixel



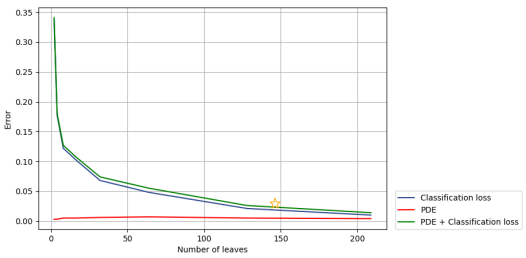
mice-protein



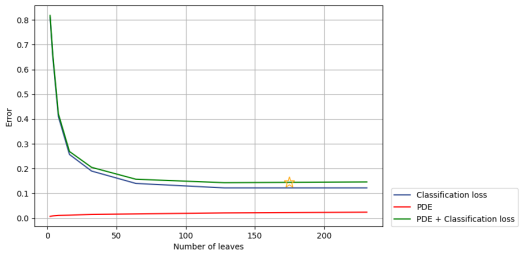
new-thyroid



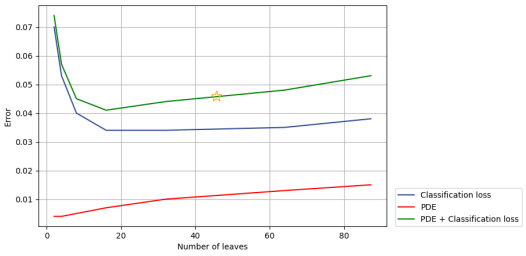
news-popularity



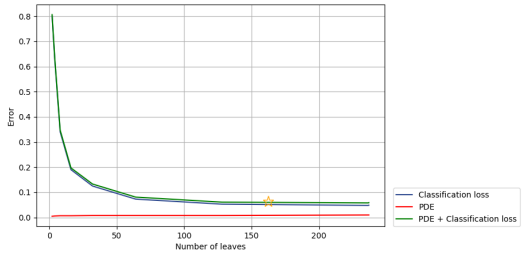
nursery



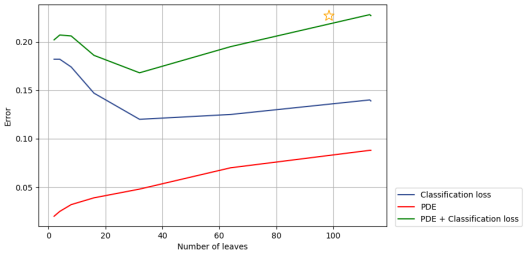
optdigits



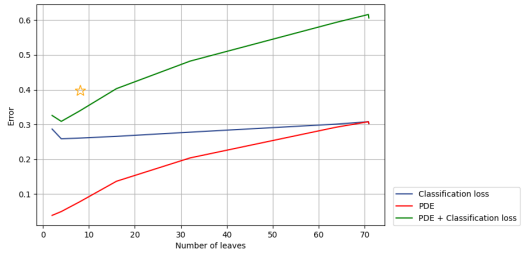
page-blocks



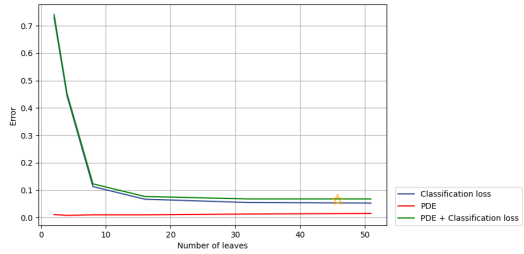
pendigits



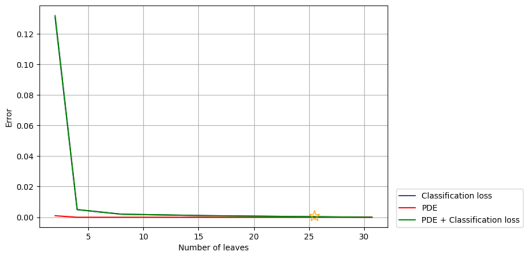
phishing



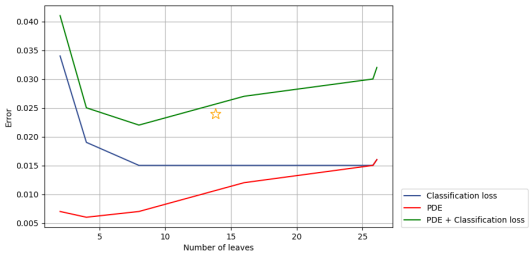
pima-diabetes



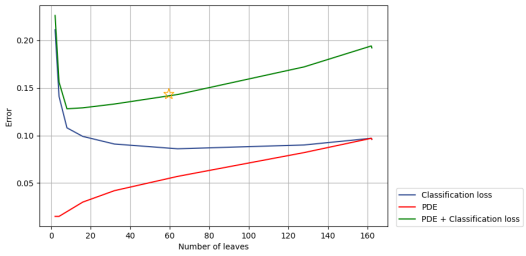
segment



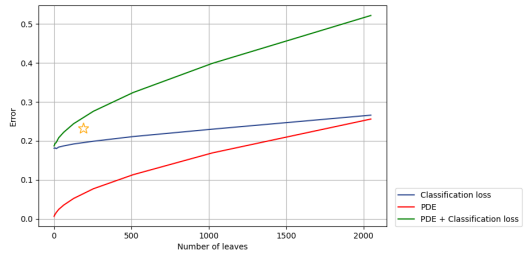
shuttle



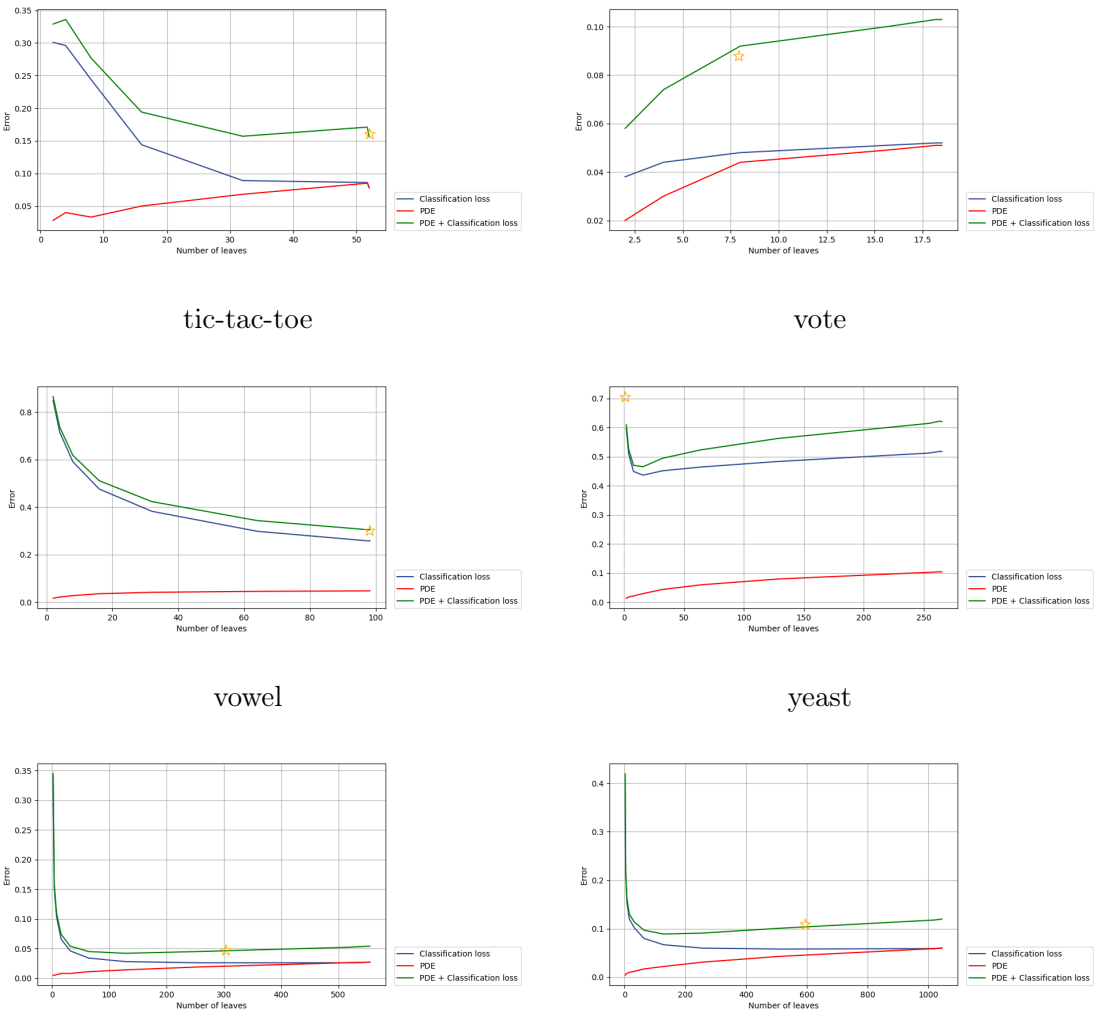
sick



spambase



taiwan-credit



tic-tac-toe vote

vowel yeast

Synthetic-3 generator with 50,000 samples Synthetic-5 generator with 50,000 samples

Figure A.1: Evaluating decision trees of varying sizes across multiple datasets using probability deviation error (PDE) and classification loss. The star symbol (☆) denotes the combined total of PDE and classification loss for the decision tree that has undergone cost-complexity post-pruning, also signifying its size.