Journal of Classification
https://doi.org/10.1007/s00357-025-09510-9

The Dutch Scaler Performance Indicator: How Much Did My Model Actually Learn?

Etienne Pieter van de Bijl[1] · Jan Gerard Klein[1] · Joris Pries[1] · Sandjai Bhulai[2] · Robert Douwe van der Mei[1,2]

Accepted: 13 March 2025

Abstract
Evaluation metrics provide a means for quantifying and comparing performances of supervised learning models, but drawing meaningful conclusions from acquired scores requires a contextual framework. Our paper addresses this by introducing the Dutch scaler (DS), a novel performance indicator for binary classification models. It quantifies a model's learning by contextualizing empirical metric scores with a baseline (Dutch draw) and a new instrument (Dutch oracle) representing the prediction quality of an "optimal" classifier. The DS performance indicator expresses the relative contribution of these components to obtain a model's score, specifying the actual learning quality. We derived closed-form expressions to map metric scores to DS scores for common evaluation metrics and categorized them by their functional form and second derivative. The DS enhances the assessment of classifiers and facilitates a framework to compare prediction quality differences between models with varying metric scores.

Keywords Performance indicator · Performance metrics · Binary classification · Machine learning · Supervised learning · Evaluation

✉ Etienne Pieter van de Bijl
evdb@cwi.nl

Jan Gerard Klein
jgklein92@gmail.com

Joris Pries
jorispries@gmail.com

Sandjai Bhulai
s.bhulai@vu.nl

Robert Douwe van der Mei
mei@cwi.nl

[1] Department of Stochastics, Centrum Wiskunde & Informatica, Science Park 123, Amsterdam 1098 XG, North-Holland, the Netherlands

[2] Faculty of Mathematics, Vrije Universiteit Amsterdam, De Boelelaan 1111, Amsterdam 1081 HV, North-Holland, the Netherlands

Published online: 12 April 2025

1 Introduction

"*How much did my model actually learn?*" is a fundamental question that should be answered during the development of any statistical or machine-learning model. But how do we define learning in this context? A formal definition is given in Mitchell (1997) who states that a machine learns by utilizing experience (data) to improve its performance on a certain task. If we look at classification problems, the task of a model is to learn to map input data to labels. More precisely, the goal is to approximate the mapping function so well that the classifier makes accurate predictions for newly acquired input data. Checking whether this goal is met before a classifier is deployed is crucial, but how can one do so?

The performance of a classifier and the quality of its predictions are commonly expressed in performance metrics scores, such as the F_β score and accuracy (Hossin & Sulaiman, 2015; Mohri et al., 2018). For decades, various domains have proposed evaluation metrics for classification problems highlighting one or more aspects of the confusion matrix and expressing a model's performance in a single value. Earlier research endeavors have been dedicated to studying and comparing performance metrics. For instance, Sokolova and Lapalme (2009) analyzed the invariance of performance metrics by manipulating individual and total counts within the confusion matrix. Several studies provide overviews of created metrics and identify relationships, properties, and dependencies between them (Canbek et al., 2022b, 2021). Others compared metrics by analyzing their characteristics analytically or experimentally (Powers, 2020; Ferri et al., 2009). Meanwhile, Brzezinski et al. (2018) present a comprehensive list of ten desirable properties for performance metrics, though under varying class distribution scenarios. In more recent work, Gösgens et al. (2021) identify three essential properties that performance metrics can potentially exhibit: monotonicity, distance, and a constant baseline. Unfortunately, no single metric simultaneously fulfills all three of these properties. Some argue specific metrics should be preferred over others when evaluating a binary classifier on a certain problem (Chicco & Jurman, 2020; Delgado & Tibau, 2019; Luque et al., 2019). However, no metric/measure is objectively "the best" for all situations, so multiple metrics, and thus aspects, should be considered when assessing the quality of a classifier. Furthermore, obtained performance metric scores should be averaged over multiple (test) datasets or random seeds, as relying on a single instance could be misleading. After all, even a broken clock can be right twice a day. Moreover, considering these scores alone is insufficient to draw meaningful conclusions without a frame of reference. To illustrate this, consider the following example. Suppose a classifier obtains an average F_1 score of 0.9 and its corresponding baseline is 0.89. How would we quantify how much the model has learned when evaluated on this selected performance metric? In addition, what if, due to data quality issues such as data noise and/or the stochastic nature of a selected classifier, an expected F_1 score of the highest metric score of 1.0 on a test dataset is unattainable?

Performance indicators are higher-order performance metrics incorporating the notion of performance bounds in quantifying a model's prediction quality. They facilitate the further comprehension and comparison of performance metrics (Canbek et al., 2022a). Conceptually, it is a metric that provides insight into a goal or a predetermined baseline (Texel, 2013). Based on the available literature, only one study proposes an indicator called the accuracy barrier (ACCBAR Canbek et al., 2017). This indicator assesses whether the performance of a classifier is close to that of a dummy classifier (e.g., labeling only positive or negative). The ACCBAR contextualizes the performance metric accuracy by subtracting a baseline (the null error rate, NER, or the no information rate, NIR) from obtained evaluation scores. This approach, however, has limitations as it focuses merely on the relative performance compared

to the baseline, but it does not quantify the model's absolute prediction quality. Moreover, it is worth considering why this indicator function does not incorporate the upper bound. In conclusion, while the ACCBAR can contextualize metric scores, it raises questions about the need to explore performance indicators further to provide a more sophisticated assessment of classification model performance.

This paper proposes a novel performance indicator for binary classification problems called the Dutch scaler (DS). The DS quantifies the prediction quality of a classifier through a derived metric referred to as the DS performance indicator (DSPI). To contextualize metric scores, the DS employs two key components: the Dutch draw (DD) baseline, which is an input-independent baseline approach indicating the expected metric score of a stochastic model that does not learn from data (as discussed in van de Bijl et al., 2024; Pries et al., 2023), and the Dutch oracle (DO). This method represents and marks the prediction quality of an "optimal" classifier in the context of selected performance metrics. These components serve as reference points for determining the DSPI. The DS allows us to perform comparative analyses of classifiers on performance metrics as we harmonize evaluation scores of diverse metrics into a uniform reference framework. The DS is (1) applicable for many commonly used performance metrics in binary classification problems, (2) reproducible and simple, (3) contextualizing using relevant performance thresholds, and (4) crucial to better assess performance metric scores. The DS is a valuable addition to the existing body of literature, providing an indispensable framework for improving the interpretation of realized performance metric scores. In conclusion, the DS is an essential instrument for the data science toolbox.

Our contributions are as follows: (1) we introduce the DS and show how it quantifies how much a model learned by integrating the DD baseline and the upper-performance value as performance bounds, (2) we provide closed-form expressions, mathematical properties, and parameters specifications of the DS for a set of performance metrics to convert metric scores into DSPI scores and summarize them in several tables, (3) we visually show the functions mapping metric scores to DSPI scores and categorize them on their concavity, (4) we demonstrate how the DS can be used in practice to compare classifiers and contrast the DS with the ACCBAR, and (5) we made the DS available in a Python package (van de Bijl, 2023).

The organization of this paper is as follows: Section 2 provides preliminaries for binary classification and briefly discusses the DD baseline. Sections 3 and 4 introduce the DO and DS, respectively. Section 5 gives a concavity analysis on the DS metric score transformations. Section 6 provides a comparative study between DSPI scores and ACCBAR scores and shows how the DSPI can be used to compare classifiers when selecting multiple metrics. Section 7 concludes with a discussion and future research directions.

2 Preliminaries

In this section, we provide the mathematical notations for binary classification problems, discuss how prediction quality is quantified using evaluation metrics, and define how one can achieve performance optimality. We then elaborate on the DD baseline and provide some necessary mathematical notations for constructing it.

2.1 Binary Classification

The task in a binary classification problem is to let a classifier learn the relationship between input data and a binary output vector. We consider a set of $M \in \mathbb{N}^+$ instances of the form

$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_M, y_M)\}$, where, without loss of generality, $\mathbf{x}_i \in \mathbb{R}^d$ is a d dimensional feature space and $y_i \in \{0, 1\}$ is its corresponding label. Observations with response value 1 are "positive," while those with 0 are "negative." We denote the whole dataset as $(\mathbf{X}, \mathbf{y})$ with $\mathbf{X} := (\mathbf{x}_1 \ldots \mathbf{x}_M)^\top \in \mathbb{R}^{M \times d}$ and $\mathbf{y} := (y_1, y_2, \ldots, y_M)$. Let P denote the number of positive instances and N denote the number of negative instances. By definition, $P + N = M$.

2.2 Performance Quantification

By comparing the labels of a set of predicted instances, denoted by $\hat{\mathbf{y}}$, with their actual labels, four base measures can be constructed: the number of true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP). Let us denote $z : \mathbf{y} \times \hat{\mathbf{y}} \mapsto \mathbb{N}^4$ as the function to derive those four base measures. We define $\hat{P} = \text{TP} + \text{FP}$ as the number of predicted positive instances and $\hat{N} = \text{TN} + \text{FN}$ as the number of the predicted negatives.

Performance metrics are derived from the before-mentioned base measures. Let us define a performance metric as $\mu : \mathbb{N}^4 \mapsto \mathbb{R}$, mapping a function of the four base measures to a performance metric score. A metric does not necessarily require all base measures as input values, so only those used are included in the functional notation throughout this research. Let us denote a composition function deriving the performance metric score directly from the predicted and the actual label vectors as $w = \mu \circ z : \mathbf{y} \times \hat{\mathbf{y}} \mapsto \mathbb{R}$. In this research, we consider the same performance metrics as stated in van de Bijl et al. (2024), and their definitions and co-domains can be found there.

2.3 Performance Optimality

The task in a classification problem is to find the "optimal" classifier, which makes accurate predictions. In Pries et al. (2023), three important factors are stated to specify "optimality": (1) as binary classifiers can be stochastic, we typically examine their expected performance; (2) the "optimal" classifier is the best in the set of considered classifiers; and (3) this classifier should be the "best" for a specific dataset, but all permutations of this dataset should be considered to prevent a coincidental perfectly prediction by a deterministic classifier. "Best" and "optimal" here depend on the desire to maximize or minimize one or more selected performance metrics. For example, accuracy is a metric we would naturally like to maximize, while the false discovery rate is preferably minimized. Assuming that the selected metrics are naturally maximized, this paper provides derivations only for maximization. For naturally minimized metrics, one can follow the procedure below by substituting "maximizing" with "minimizing" or simply multiplying the metric with "-1".

A binary classifier is defined as a function $h : \chi \times \mathbb{R} \to \{0, 1\}$ that maps feature values to zero or one, where the second input is used to capture the randomness nature in a stochastic model, often referred to as the random seed (Pries et al., 2023). This classifier can only label one instance at a time, so to classify multiple instances simultaneously, we define $h_M : \chi^M \times \mathbb{R} \to \{0, 1\}^M$ as the function that predicts $M \in \mathbb{N}^+$ instances. Any single instance classifier h can be extended to predict M instances simultaneously by applying this classifier for each instance individually. Let us define H_M as the set of all possible classifiers of the form h_M.

The objective in searching for the "best" classifier is to identify the so-called the average-permutation-optimal classifier for a selected performance metric. This means we try to find the classifier that optimizes the expected performance score, accounting for the variations due to

data permutations and classifier randomness. Let us denote π as a function that permutes data in all possible permutations, as specified in Pries et al. (2023). If we selected a performance metric μ, the average-permutation expectation for a single classifier h_M can be derived as follows:

$$\mu_{h_M} := \mathbb{E}_\pi[\mathbb{E}_{r\in\mathbb{R}}[w(h_M(\mathbf{X}_\pi, r), \mathbf{y}_\pi)]],$$

where w is the composition function of μ, as described in Section 2.2. To find the average-permutation-optimal classifier, we have to search for the classifier with the highest average-permutation expectation in the set of all possible classifiers H_M, or mathematically $h_M^{\max} \in \arg\max_{h_M\in H_M} \mu_{h_M}$. We define the average-permutation-optimal expectation score for performance metric μ as follows:

$$\Omega_\mu := \mathbb{E}_\pi[\mathbb{E}_{r\in\mathbb{R}}[w(h_M^{\max}(\mathbf{X}_\pi, r), \mathbf{y}_\pi)]].$$

The average-permutation expectation cannot always be determined due to the generalization error, but we can look at the empirical estimator: $\bar{\mu}$. Suppose we have a random classifier h'_M, and we apply it K times on a dataset with different seeds. The empirical estimator can be determined by

$$\bar{\mu} := \frac{\sum_{r=1}^{K} w(h'_M(\mathbf{X}, r), \mathbf{y})}{K}.$$

This empirical value is the score we want to contextualize with the performance bounds in our performance indicator. The difference between Ω_μ and $\overline{\mu}$ is also known as the generalization error. The generalization error is, unfortunately, inevitable. The probably approximately correct (PAC) learning framework offers theorems that estimate the number of instances needed to determine the likelihood that the error remains within a specified bound. While this framework can guide the selection of K, there is no definitive rule of thumb; generally, a higher K tends to yield better results. The following sections discuss the two critical components for deriving essential performance bounds: a baseline and the performance score of the "optimal" classifier.

2.4 Dutch Draw Baseline

The selected method for deriving a binary classification baseline for a given metric is the DD baseline (van de Bijl et al., 2024). We selected this method as it is the best binary classifier that does not learn from data (input-independent), as proven in Pries et al. (2023). A baseline is established by deriving the "optimal" DD classifier, which optimizes the expected value of the selected performance metric. A DD classifier predicts M instances by randomly assigning the value "1" to $d = \lfloor M\theta \rceil$ instances and the value "0" to the remaining $M - d$ instances. Here, $\lfloor x \rceil$ denotes rounding x to the nearest integer (e.g., 1.4 rounds to 1 and 2.6 rounds to 3). The parameter $\theta \in [0, 1]$ specifies the proportion of M predicted as positive. Since this random classifier depends solely on θ, it is considered data-independent and does not learn.

Let us provide a mathematical formulation for the DD classifier. Let $S := \{0, 1\}^M$ denote the set of all possible binary vectors to predict M instances simultaneously. We can decompose this set S into disjoint sets such that all vectors contain the same number of ones in each set. Let us thus define $S_k := \{s \in S | \sum_{i=1}^{M} s_i = k\}$ with $k \in \{0, 1, \ldots, M\}$. It holds that $\cup_{k=0}^{M} S_k = S$. A DD classifier, denoted by $\sigma_\theta : \mathbb{N} \mapsto \{0, 1\}^M$, randomly draws one vector from one of these decomposed sets (S_k) and is mathematically defined as $\sigma_\theta(M) \sim \text{Uniform}(S_{\lfloor M\theta \rceil})$ where $\text{Uniform}(A)$ is defined as the uniform distribution over a set A and $\lfloor M\theta \rceil$ specifies the number

of positively predicted instances ($\hat{P}$) for this classifier. The distributions of the base measures, denoted by $\text{TN}_\theta^{\text{DD}}$, $\text{TP}_\theta^{\text{DD}}$, $\text{FN}_\theta^{\text{DD}}$, and $\text{FP}_\theta^{\text{DD}}$, are directly determined by σ_θ and all follow a hypergeometric distribution with parameters depending on M, P, and $\lfloor M\theta \rceil$.

It can be observed in this mathematical notation that multiple values of θ can result in the same classifier. For example, suppose $M = 10$ and $\theta_1 = 0.1$ and $\theta_2 = 0.11$, then $\sigma_{\theta_1}(M) = \sigma_{\theta_2}(M)$. Let us, therefore, introduce another variable $\theta^* := \frac{\lfloor M\theta \rceil}{M}$ as the discretized version of θ. This θ^* reduces the search space for finding the set of θ, leading to the optimal DD. Furthermore, we define

$$\Theta^* := \left\{ \frac{\lfloor M\theta \rceil}{M} : \theta \in [0, 1] \right\} = \left\{ 0, \frac{1}{M}, \ldots, \frac{M-1}{M}, 1 \right\},$$

as the set of all unique values that θ^* can obtain for all $\theta \in [0, 1]$. There are, however, some limitations to $\hat{P}$ or $\hat{N}$. For example, we require $\hat{P} > 0$ to have a defined precision score. Let us, therefore, introduce Θ_μ^* as the set of possible θ^* values respecting the limitations of the evaluation metric μ. The DD baseline is the expectation of the optimal DD classifier for a selected performance metric μ. Let us define θ_{opt}^* as the θ^* leading to this optimum. In mathematical terms, we try to find

$$\theta_{\text{opt}}^* \in \Theta_{\text{opt}}^* := \underset{\theta^* \in \Theta_\mu^*}{\arg\max} \left\{ \mathbb{E}\left[w(\sigma_{\theta^*}(M), \mathbf{y}) \right] \right\}.$$

The DD baselines and corresponding θ_{opt}^* values for all selected evaluation metrics μ can be found in van de Bijl et al. (2024).

3 Dutch Oracle

Deriving an exact upper bound on the expected performance of supervised learning models is often challenging for several reasons. First, its performance is bounded by the quality and size of the used data (Jain et al., 2020). The existence of noise in labels or input data, class imbalance, sample bias, and outliers illustrate why classifiers cannot always make perfect predictions (Gupta et al., 2021). Secondly, there does not exist one unique model that achieves the highest possible performance score for all problems as described in the no-free-lunch theorem (Wolpert & Macready, 1997). Therefore, we would have to search the set of all possible classifiers to find the model achieving the "best" score for a specific problem. Thirdly, there are an infinite number of possible classifiers. Still, we can only gather empirical results for a finite number of classifiers, so it is always possible that we did not consider a model that approximates the target function better. Upper bounds should represent expected performances, while with empirical results, we can only estimate the expectations.

Fortunately, we can approximate the performance of an "optimal" classifier using an abstraction from the active learning domain called oracle (Kuncheva et al., 2003). In general, an oracle is an entity that knows the correct answer to all questions (Raykar et al., 2009). In a classification problem, this would be a model that always predicts the correct label. But, as mentioned in Raykar et al. (2009), even an oracle sometimes makes mistakes or is not always correct, introducing the notion of an imperfect oracle. This imperfect oracle can approximate the expected performance score of the optimal model by finding the right balance of correct and incorrect predictions.

We propose a novel (im)perfect oracle classifier called the DO. The DO can be seen as a proxy for optimal model performance, establishing the theoretical performance limit for any chosen metric. The imperfect oracle enables the incorporation of upper limits when

determining prediction quality scores, providing more nuanced evaluations of a classifier's performance. The DO approximates optimal model performance by balancing correct predictions with occasional errors. The DO makes an incorrect prediction for each instance with a probability $\rho \in [0, 1]$. As such, each base measure can be represented as the outcome of a binomial trial. Thus, we get

$$\begin{aligned} \mathrm{TP}_\rho^{\mathrm{DO}} &\sim B(P, (1-\rho)), \quad \mathrm{FN}_\rho^{\mathrm{DO}} \sim B(P, \rho), \\ \mathrm{TN}_\rho^{\mathrm{DO}} &\sim B(N, (1-\rho)), \quad \mathrm{FP}_\rho^{\mathrm{DO}} \sim B(N, \rho). \end{aligned}$$

Independently of the actual label of the instance, the DO will make a mistake with this probability ρ. Deriving expectations of these base measures with the DO gives us the following:

$$\begin{aligned} \mathbb{E}\left[\mathrm{TP}_\rho^{\mathrm{DO}}\right] &= P(1-\rho), \quad \mathbb{E}\left[\mathrm{FN}_\rho^{\mathrm{DO}}\right] = P\rho, \\ \mathbb{E}\left[\mathrm{TN}_\rho^{\mathrm{DO}}\right] &= N(1-\rho), \quad \mathbb{E}\left[\mathrm{FP}_\rho^{\mathrm{DO}}\right] = N\rho. \end{aligned}$$

Exploring the extremes of the prediction quality generated by the DO proves to be insightful. Choosing $\rho = 0$ results in a flawless DO, which we call the perfect predictor (PP). The PP, a classical approach, accurately predicts the label for a given instance, making it suitable for approximating the maximum performance score when a model consistently achieves perfect predictions. Conversely, opting for $\rho = 1$ leads to a DO prone to mere errors, labeled the terrible predictor. Instances presented to this oracle consistently receive incorrect labels. However, since the objective is to estimate the score of the optimal model, this oracle lacks practical value, with any baseline model expected to outperform it.

Let us outline two ways to determine ρ. First, it can be determined through one expert straightforward estimation, providing an approximate but reasonably accurate assessment of the upper limit. Although not as precise as rigorous quantitative analysis, the user-driven estimation method is quick and practical, making it a valuable tool for setting the upper boundary, especially when more rigorous data or analysis is not readily available. Second, a more quantitative approach involves using theoretically derived upper bounds on classifier performance. One such approach is determining the Bayes error rate (Ishida et al., 2023; Antos et al., 1999), representing the minimum error rate that any classifier can theoretically achieve. Once the Bayes error is known, the corresponding expected metric score for this "optimal" model can be calculated. For instance, if the Bayes error rate for accuracy is 0.15, then the expected accuracy of the "optimal" classifier would be 0.85. The corresponding ρ to align the DO under the DS is 0.15, but in Section 4.6 and specifically Table 4, we will give exact formulas to translate other theoretical "optimal" model scores to ρ. Still, depending on the problem instance and the theoretical approach chosen, other bounds—such as those derived from PAC learning theory—can provide quantitative values to determine the upper limit of an "optimal" classifier's performance.

4 Dutch Scaler

This section describes the DS and how it quantifies how much a classifier has learned. First, we discuss the essential characteristics an indicator should possess to achieve this purpose. Second, we mathematically define the DS and show under which conditions our indicator satisfies these desirable properties.

4.1 Desirable Properties of an Indicator

Before we discuss our proposed performance indicator, let us outline the two key characteristics it should possess.

Normalization and Range [0,1] If a classifier's metric score surpasses the baseline of a selected performance metric and does not exceed the expected score of the "optimal" model, then the indicator score should fall within the range [0, 1]. In this context, "0" signifies the baseline performance, while "1" represents the upper limit of the expected performance. These bounds should be based on expectations rather than empirical averages, as the latter lacks theoretical grounding and may be harder to interpret or extend to new situations. On the one hand, if a classifier's performance falls short of the baseline, the indicator score should be negative. This negative score signals that the classifier underperforms in comparison to the baseline. Identical to a negative R^2 in a regression context, its numerical value lacks practical significance because it indicates that the regressor's performance is inferior to predicting the mean. On the other hand, if the empirical metric score is above the expectation of an "optimal" model, indicating a generalization error, the derived score will be above 1. An interpretation of this indicator score can be either (1) the expectation of the "optimal" model is incorrect or (2) the empirical estimation of the expected metric score is deviant from the actual expectation of the corresponding model.

The overly-optimistic empirical acquired score introduces the generalization error in the latter situation. In either scenario, the values should be carefully re-evaluated.

Strictly Increasing with Performance Indicator scores must exhibit a strict positive monotone relationship with the realized scores. As a metric score increases, its corresponding indicator score should also consistently rise. Failing to demonstrate this relationship would be counterproductive, as it would mean that striving for better performance does not yield higher indicator scores.

These properties contribute to form a clearly defined, easily interpretable performance indicator that provides valuable insights for evaluating and comparing classifier performance.

4.2 Definition and Objective

The DS quantitatively contextualizes metrics scores by redefining the computation of the base measures to determine how much a model has learned. The DS base measures are derived using a convex combination of the expectations of the DD baseline and DO base measures. The mathematical definition of each DS base measure is expressed as follows:

$$\begin{aligned}
\mathrm{TP}_\alpha &:= \alpha\mathbb{E}\left[\mathrm{TP}_\rho^{\mathrm{DO}}\right] + (1-\alpha)\mathbb{E}\left[\mathrm{TP}_{\theta^*_{\mathrm{opt}}}^{\mathrm{DD}}\right], \\
\mathrm{TN}_\alpha &:= \alpha\mathbb{E}\left[\mathrm{TN}_\rho^{\mathrm{DO}}\right] + (1-\alpha)\mathbb{E}\left[\mathrm{TN}_{\theta^*_{\mathrm{opt}}}^{\mathrm{DD}}\right], \\
\mathrm{FN}_\alpha &:= \alpha\mathbb{E}\left[\mathrm{FN}_\rho^{\mathrm{DO}}\right] + (1-\alpha)\mathbb{E}\left[\mathrm{FN}_{\theta^*_{\mathrm{opt}}}^{\mathrm{DD}}\right], \\
\mathrm{FP}_\alpha &:= \alpha\mathbb{E}\left[\mathrm{FP}_\rho^{\mathrm{DO}}\right] + (1-\alpha)\mathbb{E}\left[\mathrm{FP}_{\theta^*_{\mathrm{opt}}}^{\mathrm{DD}}\right].
\end{aligned}$$

Here, $\alpha \in [0, 1]$ is the DSPI. The DSPI controls the weight assigned to the DO and DD baseline base measures expectations. Metric scores (μ) are derived from one or more of the four base measures, as discussed in Section 2.2. Similarly, metrics under the

DS can be computed using the same transformation, and we denote its computation as $\mu_\alpha := \mu(\text{TP}_\alpha, \text{TN}_\alpha, \text{FP}_\alpha, \text{FN}_\alpha)$.

The objective of the DS is to find the DSPI score (α) that results in the corresponding acquired metric score. Mathematically, find α such that $\mu_\alpha = \overline{\mu}$, where $\overline{\mu}$ is a realized performance metric score described in Section 2.3. The intuition behind this approach is that we are interested in how much contribution of the DO and the DD baseline is required to achieve an identical empirical metric score. The underlying question is *"How much learning is needed to achieve the same score?"*. There are metrics, such as the true positive rate, where the DD baseline directly results in the "optimal" model's expected metric score. The highest possible score can be achieved for these metrics without learning from the data. We focus on metrics where the DD baseline does not result in the "optimal" model score and where a model's performance lies between the DD baseline and the DO. Mathematical reverse engineering is needed to derive the required α, which we will show in the following sections.

4.3 Substitutions and Properties

Having established the definitions of the base measures under the DS, we can now formulate the performance metrics using these base measures. Let us substitute the expectations of the DD baseline and the DO in the definition of the DS as stated in Sections 2.4 and 3. From van de Bijl et al. (2024), it follows that $\mathbb{E}\left[\text{TP}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = P\theta^*_{\text{opt}}$, $\mathbb{E}\left[\text{TN}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = N(1-\theta^*_{\text{opt}})$, $\mathbb{E}\left[\text{FN}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = P - \mathbb{E}\left[\text{TP}^{\text{DD}}_{\theta^*_{\text{opt}}}\right]$, and $\mathbb{E}\left[\text{FP}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = N - \mathbb{E}\left[\text{TN}^{\text{DD}}_{\theta^*_{\text{opt}}}\right]$. By substituting them into these expectations in the definition of the DS base measures, we get the following:

$$\text{TP}_\alpha := \alpha\mathbb{E}\left[\text{TP}^{\text{DO}}_\rho\right] + (1-\alpha)\mathbb{E}\left[\text{TP}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = \alpha P(1-\rho-\theta^*_{\text{opt}}) + P\theta^*_{\text{opt}}, \tag{1}$$

$$\text{TN}_\alpha := \alpha\mathbb{E}\left[\text{TN}^{\text{DO}}_\rho\right] + (1-\alpha)\mathbb{E}\left[\text{TN}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = \alpha N(\theta^*_{\text{opt}}-\rho) + N(1-\theta^*_{\text{opt}}), \tag{2}$$

$$\text{FN}_\alpha := \alpha\mathbb{E}\left[\text{FN}^{\text{DO}}_\rho\right] + (1-\alpha)\mathbb{E}\left[\text{FN}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = \alpha P(\rho-1+\theta^*_{\text{opt}}) + P(1-\theta^*_{\text{opt}}), \tag{3}$$

$$\text{FP}_\alpha := \alpha\mathbb{E}\left[\text{FP}^{\text{DO}}_\rho\right] + (1-\alpha)\mathbb{E}\left[\text{FP}^{\text{DD}}_{\theta^*_{\text{opt}}}\right] = \alpha N(\rho-\theta^*_{\text{opt}}) + N\theta^*_{\text{opt}}. \tag{4}$$

By definition, we have $0 \leq \text{TP}_\alpha, \text{FN}_\alpha \leq P$, and $0 \leq \text{TN}_\alpha, \text{FP}_\alpha \leq N$. As mentioned before, $\text{TP}_\alpha + \text{FN}_\alpha = P$ and $\text{TN}_\alpha + \text{FP}_\alpha = N$. A convenient mathematical expression in the derivation of closed-form expressions of performance metrics under the DS is the number of positively/negatively predicted instances, which is given by

$$\hat{P} = \text{TP}_\alpha + \text{FP}_\alpha = \alpha(\rho(N-P) + P - M\theta^*_{\text{opt}}) + M\theta^*_{\text{opt}}, \tag{5}$$

$$\hat{N} = \text{TN}_\alpha + \text{FN}_\alpha = \alpha(\rho(P-N) + M\theta^*_{\text{opt}} - P) + M(1-\theta^*_{\text{opt}}). \tag{6}$$

These derivations help to rewrite metric definitions in terms of the DS parameters. Table 1 shows the DS definition for the selected performance metrics. They are functions of α, P, M, θ^*_{opt}, and ρ. The values of M and P are inherent to the specific dataset employed, while ρ is a user-estimated parameter. We will elaborate on this later, but since multiple values of θ^*_{opt} can lead to the DD baseline, it is essential to establish criteria for selecting the appropriate θ^*_{opt} for the DS. In the following sections, we will elaborate on the selection of θ^*_{opt}, describe the mathematical reverse engineering of α, and show the restrictions on the user-estimated parameter ρ to satisfy all required properties defined for an indicator. Still, first, we will look at the range of the DS.

Table 1 Performance metrics substituted with DS base measures

Metric	DS Abbr.	Definition under the DS
Positive predicted value	PPV_α	$\frac{\alpha P(1-\rho-\theta^*_{\text{opt}})+P\theta^*_{\text{opt}}}{\alpha(\rho(N-P)+P-M\theta^*_{\text{opt}})+M\theta^*_{\text{opt}}}$
Negative predicted value	NPV_α	$\frac{\alpha N(\theta^*_{\text{opt}}-\rho)+N(1-\theta^*_{\text{opt}})}{\alpha(\rho(P-N)+M\theta^*_{\text{opt}}-P)+M(1-\theta^*_{\text{opt}})}$
F_β score	F^β_α	$\frac{(1+\beta^2)(\alpha P(1-\rho-\theta^*_{\text{opt}})+P\theta^*_{\text{opt}})}{\alpha(\rho(N-P)+P-M\theta^*_{\text{opt}})+M\theta^*_{\text{opt}}+P\beta^2}$
Youden's J statistic/index	J_α	$\alpha(1-2\rho)$
Markedness	MK_α	$\alpha(1-2\rho)\frac{PN}{\hat{P}\hat{N}}$
Accuracy	Acc_α	$\frac{\alpha P(1-\rho-\theta^*_{\text{opt}})+P\theta^*_{\text{opt}}+\alpha N(\theta^*_{\text{opt}}-\rho)+N(1-\theta^*_{\text{opt}})}{M}$
Balanced accuracy	BAcc_α	$\alpha(1-2\rho)\frac{1}{2}+\frac{1}{2}$
Matthews' correlation coefficient	MCC_α	$\alpha(1-2\rho)\sqrt{\frac{PN}{\hat{P}\hat{N}}}$
Cohen's kappa	κ_α	$\alpha(1-2\rho)\frac{2PN}{\hat{P}N+\hat{N}P}$
Fowlkes-Mallow index	FM_α	$\frac{\alpha P(1-\rho-\theta^*_{\text{opt}})+P\theta^*_{\text{opt}}}{\sqrt{P(\alpha(\rho(N-P)+P-M\theta^*_{\text{opt}})+M\theta^*_{\text{opt}})}}$
Threat score	TS_α	$\frac{\alpha P(1-\rho-\theta^*_{\text{opt}})+P\theta^*_{\text{opt}}}{P+\alpha N(\rho-\theta^*_{\text{opt}})+N\theta^*_{\text{opt}}}$
G-mean 2	$\text{G}^{(2)}_\alpha$	$\sqrt{(\alpha(1-\rho+\theta^*_{\text{opt}})+\theta^*_{\text{opt}})(\alpha(\theta^*_{\text{opt}}-\rho)+(1-\theta^*_{\text{opt}})}$

4.4 Range

The range of the metric scores (μ_α) the DS can achieve is determined by the selected metric and its specific definition. Let us formulate two theorems describing the value of μ_α at $\alpha \in \{0, 1\}$ to check whether and under which conditions the previously described desirable properties hold.

4.4.1 Baseline Bound

Our first desirable property of an indicator describes that an indicator score of 0 should indicate the expectation score of a baseline model. For our indicator, we selected the DD baseline as our baseline approach. The following theorem indicates the two properties metric μ_α should possess such that it holds that the DS with $\alpha = 0$ equals the DD baseline.

Theorem 4.1 *If μ_α is strictly increasing in $\alpha \in [0, 1]$ and μ_0 is linear in $TP^{DD}_{\theta^*_{opt}}$, then its minimum is $\mathbb{E}\left[\mu_\alpha\left(TP^{DD}_{\theta^*_{opt}}\right)\right]$, which is the DD baseline.*

Proof It holds that μ_α is strictly increasing in α, so the minimum of μ_α is achieved for $\alpha = 0$, i.e.,

$$\begin{aligned}\min_{\alpha\in[0,1]} \mu_\alpha &= \mu_{\alpha=0}(\text{TP}_{\alpha=0}, \text{TN}_{\alpha=0}) \\ &\overset{Eqs.\ (1),\ (2)}{=} \mu_{\alpha=0}\left(0\mathbb{E}\left[\text{TP}^{\text{DO}}_\rho\right] + 1\mathbb{E}\left[\text{TP}^{\text{DD}}_{\theta^*_{\text{opt}}}\right], 0\mathbb{E}\left[\text{TN}^{\text{DO}}_\rho\right] + 1\mathbb{E}\left[\text{TN}^{\text{DD}}_{\theta^*_{\text{opt}}}\right]\right)\end{aligned}$$

$$= \mu_{\alpha=0}\left(\mathbb{E}\left[\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right], \mathbb{E}\left[\mathrm{TN}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right]\right).$$

Hence, the minimum $\mu_{\alpha=0}$ depends only on the DD. Under the DD, all four base measures can be written as linear functions of $\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}$. We can drop the $\mathrm{TN}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}$ argument in μ_α. Let $\mu^*_{\alpha=0}$ be this measure, i.e., $\mu^*_{\alpha=0}\left(\mathbb{E}\left[\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right]\right) := \mu_{\alpha=0}\left(\mathbb{E}\left[\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right], N - \lfloor M\theta^*_{\mathrm{opt}}\rceil + \mathbb{E}\left[\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right]\right)$. By the linearity of the expectation operator, we have

$$\mu_{\alpha=0}\left(\mathbb{E}\left[\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right], \mathbb{E}\left[\mathrm{TN}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right]\right) = \mu^*_{\alpha=0}\left(\mathbb{E}\left[\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right]\right) = \mathbb{E}\left[\mu^*_{\alpha=0}\left(\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right)\right].$$

The right-hand side is precisely the definition of the DD baseline for the evaluation measures μ_α. □

It immediately follows from Theorem 4.1 that $\mathbb{E}\left[\mu_{\alpha=0}\left(\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}, \mathrm{TN}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right)\right] \leq \mu_\alpha(\mathrm{TP}_\alpha, \mathrm{TN}_\alpha)$ under the specified conditions. Some metrics, i.e., TS_α and $\mathrm{G}^{(2)}_\alpha$, do not satisfy the linearity property in $\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}$ under μ_0, so Theorem 4.1 does not necessarily hold for them. It is, however, possible that an equality does occur. For TS_α, inequality occurs only in cases where $P = 1$ and $\theta^*_{\mathrm{opt}} < 1$. Since no closed-form expression is currently known for determining θ^*_{opt} for $\mathrm{G}^{(2)}_\alpha$ resulting in the DD baseline, identifying when $\mu_{\alpha=0}$ equals the DD baseline remains an open question. In the case of inequality, additional considerations or adjustments are required to construct indicator scores satisfying the desirable properties of an indicator as described in Section 4.1.

4.4.2 Optimal Model Bound

Using the following theorem, let us now examine the alignment of the DS with the expected performance score of the "optimal" classifier.

Theorem 4.2 *If μ_α is strictly increasing in $\alpha \in [0, 1]$, then the maximum of μ_α is equal to $\mu_{\alpha=1}(\mathbb{E}\left[\mathrm{TP}^{\mathrm{DO}}_\rho\right], \mathbb{E}\left[\mathrm{TN}^{\mathrm{DO}}_\rho\right])$.*

Proof μ_α is strictly increasing, and hence, the maximum of μ_α is achieved for $\alpha = 1$, i.e.,

$$\begin{aligned}\max_{\alpha\in[0,1]} \mu_\alpha &= \mu_{\alpha=1}(\mathrm{TP}_{\alpha=1}, \mathrm{TN}_{\alpha=1}) \\ &\overset{Eqs.\ (1),\ (2)}{=} \mu_{\alpha=1}\left(1\mathbb{E}\left[\mathrm{TP}^{\mathrm{DO}}_\rho\right] + 0\mathbb{E}\left[\mathrm{TP}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right], 1\mathbb{E}\left[\mathrm{TN}^{\mathrm{DO}}_\rho\right] + 0\mathbb{E}\left[\mathrm{TN}^{\mathrm{DD}}_{\theta^*_{\mathrm{opt}}}\right]\right) \\ &= \mu_{\alpha=1}(\mathbb{E}\left[\mathrm{TP}^{\mathrm{DO}}_\rho\right], \mathbb{E}\left[\mathrm{TN}^{\mathrm{DO}}_\rho\right]).\end{aligned}$$

□

More specifically,

$$\max_{\alpha\in[0,1]} \mu_\alpha = \mu_{\alpha=1}(P(1-\rho), N(1-\rho)).$$

Table 2 gives the boundaries of a set of selected performance metrics we would naturally like to maximize. Since the considered performance metrics, except for TS_α and $\mathrm{G}^{(2)}_\alpha$, exhibit DD baseline linearity in $\mathrm{TP}^{\mathrm{DD}}_\theta$, it can be inferred that the lower bound of the DS is equivalent to the DD baseline. Notably, the alignment of TS_α remains accurate despite its non-linearity in $\mathrm{TP}^{\mathrm{DD}}_\theta$. To ensure that $\mu_0 < \mu_1$, certain constraints on the parameter ρ are necessary, which we will specify in Section 4.6. When the realized performance score is between these bounds, we should search for the resulting indicator score, which we will discuss in the next section.

Table 2 Boundaries of DS performance metrics

Metric	DD baseline Θ^*_{opt}	Baseline bound (μ_0)	Optimal model bound (μ_1)
PPV_α	$\Theta^* \setminus \{0\}$	$\frac{P}{M}$	$\frac{P(1-\rho)}{\rho(N-P)+P}$
NPV_α	$\Theta^* \setminus \{1\}$	$\frac{N}{M}$	$\frac{N(1-\rho)}{\rho(P-N)+N}$
F^β_α	$\{1\}$	$\frac{(1+\beta^2)P}{M+P\beta^2}$	$\frac{(1+\beta^2)P(1-\rho)}{\rho(N-P)+P(1+\beta^2)}$
J_α	Θ^*	0	$1-2\rho$
MK_α	$\Theta^* \setminus \{0, 1\}$	0	$\frac{NP(1-2\rho)}{-\rho^2(P-N)^2+\rho(P-N)^2+NP}$
Acc_α	if $P = N \rightarrow \Theta^*$ else $\rightarrow \{\lvert P < \frac{M}{2}\rvert\}$	$\frac{\max\{P,N\}}{M}$	$1-\rho$
BAcc_α	Θ^*	0.5	$1-\rho$
MCC_α	$\Theta^* \setminus \{0, 1\}$	0	$\frac{\sqrt{PN}(1-2\rho)}{\sqrt{(\rho(N-P)+P)(\rho(P-N)+N)}}$
κ_α	if $P = M \rightarrow \Theta^* \setminus \{1\}$ else $\rightarrow \Theta^*$	0	$\frac{2PN(1-2\rho)}{\rho(N-P)^2+2PN}$
FM_α	$\{1\}$	$\sqrt{\frac{P}{M}}$	$\frac{\sqrt{P}(1-\rho)}{\sqrt{P(1-\rho)+N\rho}}$
TS_α	if $P = 1 \rightarrow \Theta^* \setminus \{0\}$ else $\rightarrow \{1\}$	$\frac{P\theta^*_{\text{opt}}}{P+N\theta^*_{\text{opt}}}$	$\frac{P(1-\rho)}{\rho N+P}$
$\text{G}^{(2)}_\alpha$	n/a	$\sqrt{\theta^*_{\text{opt}}(1-\theta^*_{\text{opt}})}$	$1-\rho$

4.5 Selecting θ^*_{opt} and Reverse Engineering α

The definitions of the performance metrics under the DS provided in Table 1 show that some metrics (e.g., J_α and BAcc_α) are independent of θ^*_{opt}, while other metrics (e.g., FM_α, F^β_α, or Acc_α) have a unique corresponding DD baseline optimizer (θ^*_{opt}) as shown in Table 2, and there are metrics for which multiple θ^*_{opt} result in the DD baseline ($|\Theta^*_{\text{opt}}| > 1$). To avoid ambiguity regarding the choice of θ^*_{opt} when there are multiple options, we outline the selection criteria that should guide it. If one has obtained an empirical performance metric score $\overline{\mu}$, then θ^*_{opt} should be selected so that the least required indicator rate α to achieve this. The intuition behind minimizing α is that we want to maximize the utility of the DD baseline. If a higher DD baseline can be achieved without learning by selecting another θ^*_{opt}, then this θ^*_{opt} should be selected. If we can achieve a higher score without learning, the DD baseline difference should not be considered as learning. The DD baseline parameter θ^*_{opt} should thus be selected to minimize α under the restriction that the constructed score μ_α equals the empirical gathered performance metric score $\overline{\mu}$.

Reverse engineering α with the selection criteria of θ^*_{opt} to obtain a metric score requires solving the following optimization problem for a selected performance metric μ:

$$\begin{aligned} \min \quad & \alpha, \\ \text{s.t.} \quad & \overline{\mu} = \mu_\alpha(\text{TP}_\alpha, \text{TN}_\alpha, \text{FP}_\alpha, \text{FN}_\alpha), \\ & \theta^*_{\text{opt}} \in \Theta^*_{\text{opt}}, \\ & \alpha \in [0, 1]. \end{aligned}$$

The solution to this problem specifies the weight α required to achieve the same realized performance score. Let us notate μ_α as a function $s = \mu_\alpha(\alpha, P, M, \theta^*_{\text{opt}}, \rho)$. Given that we are interested in α, we denote μ_α^{-1} as the inverse function of μ_α to $s : \alpha = \mu_\alpha^{-1}(s, P, M, \theta^*_{\text{opt}}, \rho)$. We search for the θ^*_{opt} by taking the derivative of μ_α^{-1} to θ^*_{opt} and then see what θ^*_{opt} results in the lowest α. Table 3 shows the DD strategies leading to the DD baseline and which θ^*_{opt} results in the lowest α for a subset of the selected performance metrics. Results for μ_α substituted with θ^*_{opt} for TS are in the Supplementary Material. The second column shows the θ^*_{opt} leading to the lowest $\alpha \in [0, 1]$, and the third column gives the DS definitions using these unique values, and we can exclude θ^*_{opt} from each function. Some metrics are independent of θ^*_{opt}, so multiple θ^*_{opt} lead to the same indicator score.

4.6 Limitations for ρ

Ensuring an accurate approximation of the performance score of an "optimal" classifier by the DO is crucial for two primary reasons. First, it establishes the target benchmark when searching for the "optimal" model. Setting an incorrect target might result in prematurely concluding the search upon discovering a suboptimal model, or conversely, the search may persist indefinitely, even if finding the truly optimal model proves unattainable. Second, any inaccuracies in this approximation can potentially compromise our proposed indicator's desired properties, as detailed in Section 4.1.

We must set ρ at a correct value to guarantee that the DO performs better than the DD baseline. This can be accomplished by ensuring that μ_α exhibits a strictly increasing trend to α. Table 4 shows the derivative of each performance metric in α and specifies which ρ should be selected to satisfy both properties described in Section 4.1. Results for ρ in FM, MK, and MCC are in the Supplementary Material. For most performance metrics, $\rho < 0.5$ implies having a strictly increasing μ_α in α, while there are metrics where the limitation of ρ depends on P and M (and β for the F_β score). The last column specifies how ρ can be determined

Table 3 Optimal θ^*_{opt} selection for performance metrics

Metric	$\arg\min_{\theta^*_{\text{opt}}}\{\mu_\alpha^{-1}\}$	μ_α with θ^*_{opt} substituted
PPV_α	$\{\frac{1}{M}\}$	$\frac{P}{M}\frac{\alpha(M(1-\rho)-1)+1}{\alpha(\rho(N-P)+P-1)+1}$
NPV_α	$\{\frac{M-1}{M}\}$	$\frac{N}{M}\frac{\alpha(M(1-\rho)-1)+1}{\alpha(\rho(P-N)+N-1)+1}$
Acc_α	if $P = N \rightarrow \Theta^*$ else $\rightarrow \{[P < \frac{M}{2}]\}$	$\frac{\alpha(\min\{P,N\}-M\rho)+\max\{P,N\}}{M}$
BAcc_α	Θ^*	$\alpha(1-2\rho)\frac{1}{2}+\frac{1}{2}$
J_α	Θ^*	$\alpha(1-2\rho)$
FM_α	$\{1\}$	$\sqrt{P}\frac{1-\alpha\rho}{\sqrt{\alpha(\rho(N-P)-N)+M}}$
F_α^β	$\{1\}$	$P(1+\beta^2)\frac{1-\alpha\rho}{\alpha(\rho(N-P)-N)+M+P\beta^2}$
κ_α	if $P = N \rightarrow \Theta^*$ else $\rightarrow \{[P < \frac{M}{2}]\}$	$\frac{2PN\alpha(1-2\rho)}{\alpha(\rho(N-P)^2-(\min\{P,N\})^2+PN)+M\min\{P,N\}}$
TS_α	if $P = 1$ and $\text{TS}_\alpha = \frac{1}{N} \rightarrow \Theta^* \setminus \{0\}$ if $P = 1$ and $\text{TS}_\alpha > \frac{1}{N} \rightarrow \{\frac{1}{M}\}$ else $\rightarrow \{1\}$	See Suppl. (A.8)

Table 4 Limitations on ρ for ensuring indicator properties

Metric	Derivative μ_α in α	$\arg\max_\rho s.t. \frac{\partial \mu_\alpha}{\partial \alpha} > 0$	$\rho = \mu_1^{-1}(\Omega_{\mu_\alpha})$
Acc_α	$\frac{\min\{P,N\}}{M} - \rho$	$\min\{\frac{N}{M}, \frac{P}{M}\}$	$1 - \Omega_{\text{Acc}}$
BAcc_α	$\frac{1}{2} - \rho$	0.5	$1 - \Omega_{\text{BAcc}}$
J_α	$1 - 2\rho$	0.5	$\frac{1}{2} - \frac{1}{2}\Omega_J$
PPV_α	$\frac{PN}{M} \frac{1-2\rho}{(\alpha(\rho(N-P)+P-1)+1)^2}$	0.5	$\frac{P(1-\Omega_{\text{PPV}})}{\Omega_{\text{PPV}}(N-P)+P}$
NPV_α	$\frac{PN}{M} \frac{1-2\rho}{(\alpha(\rho(P-N)+N-1)+1)^2}$	0.5	$\frac{N(1-\Omega_{\text{NPV}})}{\Omega_{\text{NPV}}(P-N)+N}$
F_α^β	$\frac{P(1+\beta^2)(N(1-2\rho)-\rho P\beta^2)}{(\alpha(\rho(N-P)-N)+M+P\beta^2)^2}$	$\frac{N}{2N+P\beta^2}$	$\frac{(1+\beta^2)P(1-\Omega_{\text{F}^\beta})}{\Omega_{F_\alpha^\beta}(N-P)+P(1+\beta^2)}$
κ_α	$\frac{(1-2\rho)2PNM\min\{P,N\}}{(\alpha(\rho(N-P)^2+PN-(\min\{P,N\})^2)+M\min\{P,N\})^2}$	0.5	$\frac{2PN(1-\Omega_\kappa)}{\Omega_\kappa(N-P)^2+4PN}$
FM_α	$\sqrt{P}\frac{\alpha\rho^2 P+\alpha N\rho(1-\rho)+N(1-\rho)+\rho(P-2M)}{2(M-\alpha(N(1-\rho)+P\rho))^{\frac{3}{2}}}$	$\frac{N}{3N+P}$	See Suppl. (A.11)
MK_α	$(1-2\rho)PN\frac{M^2\theta^*_{\text{opt}}+\hat{P}^2-2\theta^*_{\text{opt}}M\hat{P}}{(\hat{P}\hat{N})^2}$	0.5	See Suppl. (A.13)
MCC_α	$(1-2\rho)\sqrt{PN}M\frac{\hat{P}(1-\theta^*_{\text{opt}})+\hat{N}\theta^*_{\text{opt}}}{(\hat{P}\hat{N})^{\frac{3}{2}}}$	0.5	See Suppl. (A.14)
TS_α	$* \to M\frac{2N-\rho(M+2N)}{(M+N-\alpha N(1-M\rho))^2}$ $** \to P\frac{N-\rho(M+N)}{(M-N\alpha(1-\rho))^2}$	$\frac{N}{M+N}$	$\frac{P(1-\Omega_{\text{TS}})}{\Omega_{\text{TS}}N+P}$
$\text{G}_\alpha^{(2)}$	N.A.	0.5	$1 - \Omega_{\text{G}^{(2)}}$

$* = \{P = 1 \wedge \frac{1}{N(1-2\rho)} \leq \alpha \leq 1\}$, $** = \{P > 1\}$ or $\{P = 1 \wedge 0 \leq \alpha \leq \frac{1}{N(1-2\rho)}\}$

when the user knows the expected performance metric score of the "optimal" model. This information is valuable if the user wants to estimate ρ on empirical estimates. The derivative of μ_α should be strictly increasing in α, and ρ should be smaller than the inverse DS function at the expected performance score of the "optimal" classifier. In the next section, we will elaborate on categorizing performance metrics based on their second derivative.

5 Concavity Analysis of the DSPI

In the previous section, we demonstrated how the DS enables the transformation of metric scores into a uniform interval with reference points. We also examined the conditions under which the DS meets the two desirable properties of an indicator. However, the second property, a positive monotonic relationship between realized scores and the DSPI, raises questions about the concavity of the transformation, which is still unaddressed. We conduct a concavity analysis of the DSPI to explore this aspect further. Figure 1 shows three visualizations of the effect of α on μ_α for the selected metrics. Plot (a) shows that some performance metric combinations have identical starting points with the same DD baseline, but their concavity differs. In plot (b), we transformed each metric with a simple min-max transformation to the same starting and ending point to emphasize concavity differences. Plot (c) shows the effect of increasing ρ on the concavity of the performance metrics. The graphs collectively illustrate how the concavity of various evaluation metrics behaves under different transformations and

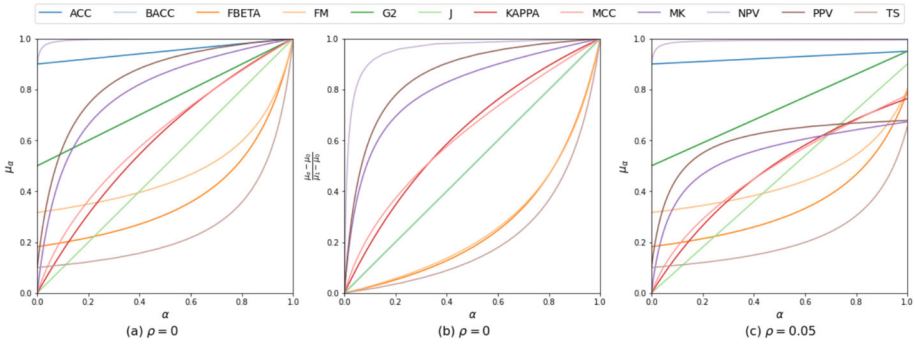

Fig. 1 Visualizations of DSPI for $M = 100$ and $P = 10$

parameter adjustments. It can be observed that the impact of increasing ρ is that the slope of the curves and the "optimal" classifier bound decreases. The progression in these graphs highlights how the metrics' concavity is sensitive to the DS standardization and parameter variation.

Figure 1 indicates concavity differences between the metrics. The critical question remains: what are the implications of the specific form of concavity, and do metrics consistently exhibit the same type of concavity? Let us use the following definition to categorize metrics under the DS on their second derivative.

Definition 1 (from Adams and Essex 2010) Suppose $f(x)$ is a function twice continuously differentiable on an interval I; then it is concave up, concave down, or linear, depending on whether for all $x \in I$ it holds that $\frac{\partial^2 f(x)}{\partial x^2} > 0$, $\frac{\partial^2 f(x)}{\partial x^2} < 0$, or $\frac{\partial^2 f(x)}{\partial x^2} = 0$, respectively.

A concave-up relationship between performance metric scores and their corresponding DSPI value exhibits a pattern that mirrors the essence of the Pareto principle (although not precisely): a disproportionate big DSPI increment is achieved for a relatively low metric score. Achieving an increment in DSPI becomes harder when increasing the performance score. For concave-down, it is exactly the other way around: a disproportionate large metric score is required for a relatively small increment in the DSPI. This concavity analysis helps determine how much better one model is over another when outperforming it in the performance metric score.

Table 5 shows the second derivative of each metric under the DS and indicates whether metrics are concave up/down or linear. The second derivative shows the curvature of each performance metric under the DS. All first derivatives are non-decreasing/positive, but the concavity of the performance metrics depends on the second derivative. The second derivative for the MK_α and MCC_α are derived, but whether the curvature is concave up/down or linear depends on the corresponding θ^*_{opt}. Accuracy, balanced accuracy, and the J statistic are always linear metrics in α. This means that the relative performance metric increase, starting from the DD baseline and ending at the DO, is one-to-one with an increment in how much a model has learned. There are only some situations where learning occurs linearly for precision, the negative predictive value, and the kappa. With this categorization of the performance metric, we can better interpret the relationship between increasing the performance metric score and the achieved quantification of how much a classifier has learned.

Table 5 Derivative analysis of the DS

Metric	Second derivative μ_α to α	Conc. Up?	Conc. Down?	Linear?
Acc_α	0			✓
BAcc_α	0			✓
J_α	0			✓
PPV_α	$\frac{\partial \text{PPV}_\alpha}{\partial \alpha} \frac{-2(\rho(N-P)+P-1)}{\hat{P}}$		✓	$P = N = 1, P = 1 \wedge \rho = 0$
NPV_α	$\frac{\partial \text{NPV}_\alpha}{\partial \alpha} \frac{-2(\rho(P-N)+N-1)}{\hat{N}}$		✓	$P = N = 1, N = 1 \wedge \rho = 0$
F_α^β	$\frac{\partial F_\alpha^\beta}{\partial \alpha} \frac{2(N(1-\rho)+P\rho)}{P\beta^2+\hat{P}}$	✓		
κ_α	$\frac{\partial \kappa_\alpha}{\partial \alpha} \frac{-2(\rho(N-P)^2+PN-(\min\{P,N\})^2)}{\hat{P}N+\hat{N}P}$		✓	$P = N$
TS_α	$\frac{\partial \text{TS}_\alpha}{\partial \alpha} \frac{2N(\theta^*_{\text{opt}}-\rho)}{\text{FP}_\alpha+P}$	✓		
FM_α	$\frac{\partial \text{FM}_\alpha}{\partial \alpha} \frac{N(1-\rho)+\rho P}{2\hat{P}}$	✓		

6 Dutch Scaler in Practice

Now that all the DSPI's theoretical properties have been discussed, it is time to demonstrate its value in practice. First, we compare the DSPI with the ACCBAR and show their differences. Second, we demonstrate how the DS provides a means to compare classifiers on multiple selected metrics.

6.1 DSPI Versus ACCBAR

Let us compare the DSPI with the values produced by the indicator ACCBAR. We used the same 28 studies to gather DSPI scores as were obtained for the ACCBAR (Canbek et al., 2020). The ACCBAR is derived by subtracting the baseline from the performance metric score. The resulting DSPI, ACCBAR, and scaled ACCBAR scores for the metric accuracy are shown in Fig. 2. As shown in Table 5, accuracy is a measure for which learning is linearly in the increment of the performance metric scores when transformed by the DS. Therefore, when we min-max scale the ACCBAR scores with the baseline and the upper-performance metric score of the optimal model as reference points, it can be observed that these scaled scores are identical to the DSPI scores. The absolute values differ, but when scaled, they are identical. This means that for performance metrics that are linear in α, ACCBAR and DSPI assessments are identical.

Differences between the scaled ACCBAR and DSPI arrive when we consider performance metrics that, under the DS transformation, are not linear in α. Figure 3 shows computed ACCBAR, scaled ACCBAR, and DSPI scores for the F_1 score. Unlike Fig. 2, only seven studies reported the F_1 score of their method, allowing us to compute the ACCBAR and DSPI on the F_1 scores only for these studies. For this metric, it can be observed that the DSPI is higher than the scaled ACCBAR in all cases. This means the DSPI assesses that the model has learned more than the quantification computed with the ACCBAR. The F_β performance metric under the DS is a concave-up function, so the learning quantity assessed by the DS will always be higher for these kinds of metrics. When performance metrics are concave-down in the DS, the quantitative assessment of the prediction performance of a

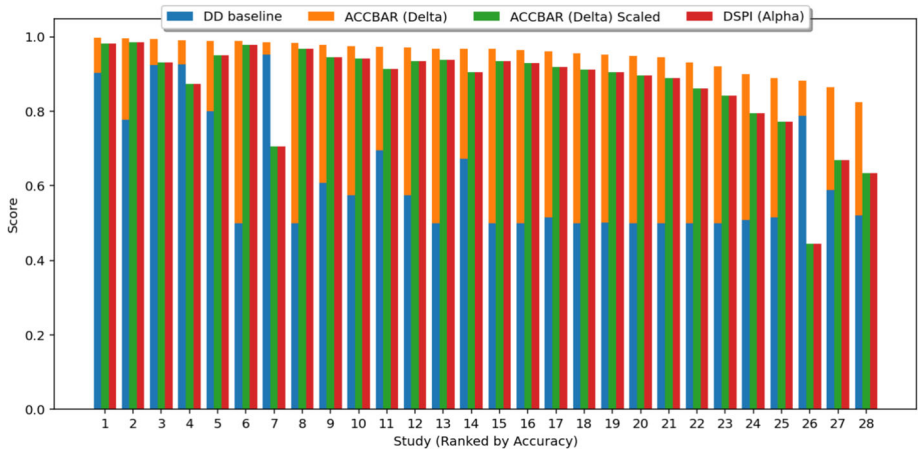

Fig. 2 Performance indicator scores of DSPI and ACCBAR on accuracy for 28 different studies

model will be lower than the ACCBAR will indicate. In general, the quality assessment of the DSPI, when compared to the scaled ACCBAR, will be more conservative when the metrics exhibit concave-down behavior. At the same time, a concave-up trend will result in a more favorable evaluation. Still, the DSPI scores have more practical meaning as they consider absolute prediction quality in contrast to the relative baseline performance quantification by the ACCBAR.

6.2 Comparing Classifiers with the DSPI

As previously discussed, individual metrics often emphasize distinct aspects of the confusion matrix, making direct comparisons unfounded. For instance, a recall of 0.9 and a precision

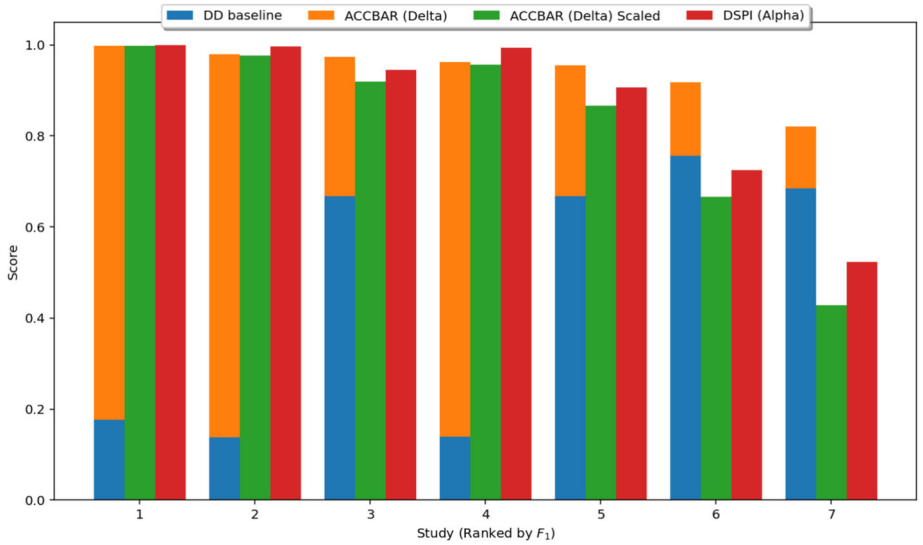

Fig. 3 Performance indicator scores of DSPI and ACCBAR on the F_1 score for seven studies

of 0.3 are meaningful when interpreted together but cannot be directly compared. The DSPI addresses this challenge by converting metrics into a unified scale, allowing for more coherent evaluations across classifiers. When applying this transformation, it makes more sense to average the standardized scores. This enables a more accurate and holistic assessment of a classifier's effectiveness. To demonstrate this, let us consider the results of a binary classification study. In Sidey-Gibbons and Sidey-Gibbons (2019), three machine learning classifiers were used on the Breast Cancer Wisconsin Diagnostic Data Set, an often studied classification problem (Wolberg et al., 1993). The task here is to predict whether tumors are benign or malignant.

Table 6 shows the base measure scores of these classifiers when evaluated on a test dataset, the corresponding metric scores, derived DSPI, and scaled ACCBAR scores. Three machine learning models, generalized linear models (GLM), support vector machines (SVM), and artificial neural networks (ANN), were trained to classify $M = 227$ tumor instances into benign ($N = 150$) or malignant ($P = 77$). DSPI scores are more conservative in their prediction quality assessment than acquired metric scores, especially for PPV and NPV. One exception to this rule is the threat score, where the DSPI gives higher scores. Metrics concave-up under the DS are metrics where the DSPI gives higher scores than scaled ACCBAR scores, while metrics that are linear or concave-down under the DS will always have DSPI scores that are equal or more conservative. The scaled ACCBAR scores provide a performance indication of a classifier but fail to make valid comparisons between scores of different metrics. Looking at the average DSPI scores and metrics scores, it is valid to average the DSPI scores as they belong to the same uniform framework. A remark here is that it would only be appropriate to average metrics if they are all either naturally maximized or minimized. With these averaged DSPI scores, we can get a better holistic performance assessment.

Table 6 Comparison of metric (μ), DSPI (μ_α), and scaled ACCBAR (Δ) scores

Metric	GLM			SVM			ANN		
	$\overline{\mu}$	μ_α	Δ	$\overline{\mu}$	μ_α	Δ	$\overline{\mu}$	μ_α	Δ
TN	148			146			148		
FN	10			5			11		
FP	2			4			2		
TP	67			72			66		
PPV	0.971	0.221	0.956	0.947	0.130	0.920	0.971	0.218	0.955
NPV	0.937	0.028	0.813	0.967	0.058	0.902	0.931	0.025	0.796
ACC	0.947	0.844	0.844	0.960	0.883	0.883	0.943	0.831	0.831
BACC	0.928	0.857	0.857	0.954	0.908	0.908	0.922	0.844	0.844
FBETA	0.918	0.908	0.833	0.941	0.936	0.881	0.910	0.899	0.818
MCC	0.882	0.842	0.882	0.911	0.882	0.911	0.872	0.829	0.872
J	0.857	0.857	0.857	0.908	0.908	0.908	0.844	0.844	0.844
MK	0.908	0.806	0.908	0.914	0.821	0.914	0.901	0.792	0.901
KAPPA	0.879	0.846	0.879	0.911	0.886	0.911	0.869	0.833	0.869
FM	0.919	0.906	0.806	0.941	0.934	0.859	0.912	0.896	0.790
TS	0.848	0.908	0.770	0.889	0.936	0.832	0.835	0.899	0.751
Average	0.909	0.729	0.855	0.931	0.753	0.894	0.901	0.719	0.843

7 Discussion and Conclusion

In this research, we introduced the DS, a performance indicator to quantify how much a model actually has learned. Our indicator is an essential quantifier to interpret performance metric scores and assess the prediction quality of classifiers. Classifiers can be compared under the DS as it transforms metrics to a uniform reference framework, and we can determine how much better one model is. We showed how the DSPI scores can be derived for a set of commonly used performance metrics. We proposed two essential properties that an indicator should possess and show under which conditions a DS metric satisfies them to have a valid interpretation. If performance metrics under the DD baseline are linear in $\text{TP}_\theta^{\text{DD}}$, it is shown that the DD baseline equals the lower bound of the DS. If not, the quantification mechanism to contextualize performance metric scores will not (necessarily) work. For example, the DD baseline for the G-mean 2 score could be below the lower bound of the DS, making the alignment incorrect.

We proposed the DO as an imperfect oracle classifier to approximate the expected performance metric score of an "optimal" model. This classifier makes a correct prediction with probability $1 - \rho$ independently of the instance. This ρ is a user-defined parameter. This novel classifier is independent of data and does not look at correlations between instances, so further research can propose better methods to improve this approximation. A better approximation could improve the quantification of how much a model has learned. Fortunately, the DS is a flexible framework in which the DO can easily be replaced with a more sophisticated model that approximates the performance of the "optimal" model.

With our uniform reference framework, we categorized performance metrics on their concavity under the DS and showed how some metrics are more similar than others. The DS translates performance metrics scores into the same frame of reference to compare performance metrics based on their curvature of the second derivative. The first derivative is always positive as this is our desirable property, but this is not always true for the second derivative. With visualizations, we have shown the impact of increasing the performance metric score on the actual acquired prediction quality improvement. Performance metrics that are concave up, down, or linearly increasing indicate whether or not it is relatively easy/hard to increase the performance acquired by the model.

Although the primary focus of this research lies in quantifying the learning progress of binary classification models, the concept can be expanded to encompass models of various classifier types or those addressing different problem domains. For instance, one could examine multi-classification classifiers or explore regression problems. To integrate the DS framework into these problems, we first need to define the expected performance of both a baseline and an "optimal" model. In this context, the baseline model is input-independent, representing the expected score of a model that has not been learned. This approach allows us to determine the metric score for a non-learning model. The key question then becomes how to adjust the baseline measures in a multiclass classification problem or the output data points in a regression problem in the direction of the "optimal" model to match the model's score. By doing so, we can quantify the extent to which models in these tasks have learned. In essence, developing indicators in these contexts enhances interpretability and improves the ability to assess the overall quality of models.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00357-025-09510-9.

Acknowledgements The authors wish to thank the anonymous referees for their valuable comments, which have significantly improved the readability and quality of the paper.

Data and Code Availability The ACCBAR dataset can be found from https://data.mendeley.com/datasets/5c442vbjzg/3. The source code for reproducing the experimental results in this work can be found in https://github.com/etiennevandebijl/Dutch-Scaler.

Declarations

Ethical Approval Not applicable

Conflict of Interest The authors declare no competing interests.

References

Adams, R. A., & Essex, C. (2010). Calculus: A complete course, 7th ed edn. Pearson Canada, Toronto. OCLC: 749131682.

Antos, A., Devroye, L., & Gyorfi, L. (1999). Lower bounds for Bayes error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,21*(7), 643–645. https://doi.org/10.1109/34.777375. Accessed 02 Sept 2024.

Brzezinski, D., Stefanowski, J., Susmaga, R., & Szczech, I. (2018). Visual-based analysis of classification measures and their properties for class imbalanced problems. *Information Sciences,462*, 242–261. https://doi.org/10.1016/j.ins.2018.06.020. Accessed 04 May 2023.

Canbek, G., Sagiroglu, S., Temizel, T. T., & Baykal, N. (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. In *2017 International conference on computer science and engineering (UBMK)* (pp. 821–826). IEEE, Antalya. https://doi.org/10.1109/UBMK.2017.8093539. http://ieeexplore.ieee.org/document/8093539/. Accessed 2024-04-05.

Canbek, G., Taskaya-Temizel, T., & Sagiroglu, S. (2020). Binary-classificationperformance evaluation reporting survey data with the findings. Mendeley. https://doi.org/10.17632/5C442VBJZG.3. https://data.mendeley.com/datasets/5c442vbjzg/3. Accessed 29 Jan 2024.

Canbek, G., Taskaya-Temizel, T., & Sagiroglu, S. (2021). BenchMetrics: A systematic benchmarking method for binary classification performance metrics. *Neural Computing and Applications,33*(21), 14623–14650. https://doi.org/10.1007/s00521-021-06103-6. Accessed 04 May 2023.

Canbek, G., Temizel-Taskaya, T., & Sagiroglu, S. (2022a). Accuracy barrier (ACCBAR): A novel performance indicator for binary classification. In *2022 15th International conference on information security and cryptography (ISCTURKEY)* (pp. 92–97). https://doi.org/10.1109/ISCTURKEY56345.2022.9931888

Canbek, G., Taskaya-Temizel, T., & Sagiroglu, S. (2022). PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics. *SN Computer Science,4*(1), 13. https://doi.org/10.1007/s42979-022-01409-1. Accessed 04 May 2023.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics,21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7. Accessed 11 May 2023.

Delgado, R., & Tibau, X.-A. (2019). Why Cohen's Kappa should be avoided as performance measure in classification. *PLOS ONE,14*(9), 0222916. https://doi.org/10.1371/journal.pone.0222916. Publisher: Public Library of Science. Accessed 12 May 2023.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters,30*(1), 27–38. https://doi.org/10.1016/j.patrec.2008.08.010. Accessed 04 May 2023.

Gösgens, M., Zhiyanov, A., Tikhonov, A., & Prokhorenkova, L. (2021). Good classification measures and how to find them. In *Advances in neural information processing systems*, (vol. 21,

pp. 17136–17147). Type: Conference paper. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85129054764&partnerID=40&md5=755934440b2394000bb3e6544c930dd7

Gupta, N., Mujumdar, S., Patel, H., Masuda, S., Panwar, N., Bandyopadhyay, S., Mehta, S., Guttula, S., Afzal, S., Sharma Mittal, R., & Munigala, V. (2021). Data quality for machine learning tasks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, (pp. 4040–4041). ACM, Virtual Event Singapore. https://doi.org/10.1145/3447548.3470817. https://dl.acm.org/doi/10.1145/3447548.3470817. Accessed 06 Jun 2023.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process,5*(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201. Accessed 16 May 2023.

Ishida, T., Yamane, I., Charoenphakdee, N., Niu, G., & Sugiyama, M. (2023). Is the performance of my deep network too good to be true? A direct approach to estimating the bayes error in binary classification. In *The eleventh international conference on learning representations*. https://openreview.net/forum?id=FZdJQgy05rz

Jain, A., Patel, H., Nagalapatti, L., Gupta, N., Mehta, S., Guttula, S., Mujumdar, S., Afzal, S., Sharma Mittal, R., & Munigala, V. (2020). Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM IGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3561–3562). ACM, Virtual Event CA USA. https://doi.org/10.1145/3394486.3406477. https://dl.acm.org/doi/10.1145/3394486.3406477. Accessed 06 Jun 2023.

Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., & Duin, R. P. W. (2003). Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications,6*(1), 22–31. https://doi.org/10.1007/s10044-002-0173-7. Accessed 06 Jun 2023.

Luque, A., Carrasco, A., Martín, A., & De Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition,91*, 216–231. https://doi.org/10.1016/j.patcog.2019.02.023. Accessed 29 Jan 2024.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill, New York: McGraw-Hill series in computer science.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning, second (edition). Adaptive computation and machine learning: The MIT Press, Cambridge, Massachusetts.

Powers, D. M. W. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv. arXiv:2010.16061 [cs, stat]. http://arxiv.org/abs/2010.16061. Accessed 04 May 2023.

Pries, J., van de Bijl, E. P., Klein, J. G., Bhulai, S., & van der Mei, R. D. (2023). The optimal input-independent baseline for binary classification: The Dutch Draw. *Statistica Neerlandica,12297*. https://doi.org/10.1111/stan.12297. Accessed 03 May 2023.

Raykar, V. C., Yu, S., Zhao, L. H., Jerebko, A., Florin, C., Valadez, G. H., Bogoni, L., & Moy, L. (2009). Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual international conference on machine learning* (pp. 889–896). ACM, Montreal Quebec Canada. https://doi.org/10.1145/1553374.1553488. https://dl.acm.org/doi/10.1145/1553374.1553488. Accessed 04 May 2023.

Sidey-Gibbons, J. A. M., & Sidey-Gibbons, C. J. (2019). Machine learning in medicine: A practical introduction. *BMC Medical Research Methodology,19*(1), 64. https://doi.org/10.1186/s12874-019-0681-4. Accessed 27 Aug 2024.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management,45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002. Accessed 04 May 2023.

Texel, P. P. (2013). Measure, metric, and indicator: An object-oriented approach for consistent terminology. In *2013 Proceedings of IEEE Southeastcon* (pp. 1–5). IEEE, Jacksonville, FL, USA. https://doi.org/10.1109/SECON.2013.6567438. http://ieeexplore.ieee.org/document/6567438/. Accessed 04 May 2023.

van de Bijl, E. P. (2023). Dutch Scaler. https://github.com/etiennevandebijl/Dutch-Scaler

van de Bijl, E. P., Klein, J. G., Pries, J., Bhulai, S., Hoogendoorn, M., & van der Mei, R. D. (2024). The Dutch Draw: Constructing a universal baseline for binary classification problems. *Journal of Applied Probability,1–19*,. https://doi.org/10.1017/jpr.2024.52. Accessed 30 Jan 2025.

Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1993). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B. https://archive.ics.uci.edu/dataset/17. Accessed 27 Aug 2024.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation,1*(1), 67–82. https://doi.org/10.1109/4235.585893. Accessed 06 Jun 2023.