J. Appl. Probab. 1–19 (2024)
*doi:*10.1017/jpr.2024.52

THE DUTCH DRAW: CONSTRUCTING A UNIVERSAL BASELINE FOR BINARY CLASSIFICATION PROBLEMS

ETIENNE VAN DE BIJL,[*] [**]
JAN KLEIN,[*] [***] AND
JORIS PRIES,[*] [****] *Centrum Wiskunde & Informatica*
SANDJAI BHULAI,[*****] [******] AND
MARK HOOGENDOORN,[*******] *Vrije Universiteit Amsterdam*
ROB VAN DER MEI,[*] [*****] [********] *Vrije Universiteit Amsterdam and Centrum Wiskunde & Informatica*

Abstract

Novel prediction methods should always be compared to a baseline to determine their performance. Without this frame of reference, the performance score of a model is basically meaningless. What does it mean when a model achieves an F_1 of 0.8 on a test set? A proper baseline is, therefore, required to evaluate the 'goodness' of a performance score. Comparing results with the latest state-of-the-art model is usually insightful. However, being state-of-the-art is dynamic, as newer models are continuously developed. Contrary to an advanced model, it is also possible to use a simple dummy classifier. However, the latter model could be beaten too easily, making the comparison less valuable. Furthermore, most existing baselines are stochastic and need to be computed repeatedly to get a reliable expected performance, which could be computationally expensive. We present a universal baseline method for all *binary classification* models, named the *Dutch Draw* (DD). This approach weighs simple classifiers and determines the best classifier to use as a baseline. Theoretically, we derive the DD baseline for many commonly used evaluation measures and show that in most situations it reduces to (almost) always predicting either zero or one. Summarizing, the DD baseline is *general*, as it is applicable to any binary classification problem; *simple*, as it can be quickly determined without training or parameter tuning; and *informative*, as insightful conclusions can be drawn from the results. The DD baseline serves two purposes. First, it is a robust and universal baseline that enables comparisons across research papers. Second, it provides a sanity check during the prediction model's development process. When a model does not outperform the DD baseline, it is a major warning sign.

Received 31 March 2022; accepted 13 August 2024.

[*] Postal address: Centrum Wiskunde & Informatica, Department of Stochastics, Science Park 123, 1098 XG, Amsterdam, Netherlands.
[**] Email address: evdb@cwi.nl
[***] Email address: jan_g_klein@outlook.com
[****] Email address: jorispries@gmail.com
[*****] Postal address: Department of Mathematics, Vrije Universiteit Amsterdam, De Boelelaan 1111, 1081 HV, Amsterdam, Netherlands.
[******] Email address: s.bhulai@vu.nl
[*******] Postal address: Department of Computer Science, De Boelelaan 1111, 1081 HV, Amsterdam, Netherlands. Email: m.hoogendoorn@vu.nl
[********] Email address: mei@cwi.nl

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

Keywords: Baseline; benchmark; evaluation; supervised learning; binary classification

2020 Mathematics Subject Classification: Primary 62C12
Secondary 62K86, 68Q87, 62H30, 62H15

1. Introduction

A typical data science project can be crudely simplified into the following steps: (i) comprehending the problem context, (ii) understanding the data, (iii) preparing the data, (iv) modeling, (v) evaluating the model, and (vi) deploying the model [19]. Before deploying a new model, it should be tested to determine whether it meets certain predefined outcome criteria. A baseline plays an essential role in this evaluation, as it gives an indication of the actual performance of a model.

However, which method should be selected to construct a baseline? A good baseline is desirable, but what explicitly makes a baseline 'good'? Comparing results with the latest state-of-the-art model is usually insightful. However, being state-of-the-art is dynamic as newer models are continuously developed. The reproducibility of such a model is also often a problem because code is not published or large amounts of computational resources are required to retrain the model. Furthermore, most existing baselines are stochastic and need to be computed repeatedly to get a reliable expected performance, which could be computationally expensive. These aspects make comparing older results with newer research hard or even impossible. Nevertheless, it is important to stress that the comparison with a state-of-the-art model still has merit. However, we are pleading for an *additional* universal baseline that can be computed quickly (without the need for training) and can make it possible to compare results across research domains and papers. With that aim in mind, we outline three principal properties that any universal baseline construction method should have: *generality*, *simplicity*, and *informativeness*.

Generality In research, a new model is commonly compared to a limited number of existing models used in the same field. Although these are usually carefully selected, they are still subjectively chosen. Take binary classification, in which the objective is to label each observation as either zero or one. Here, we could already select a decision tree [10], random forest [5], variants of naive Bayes [18], k-nearest neighbors [1], support vector machine [16], neural network [17], or logistic regression model [15] to evaluate the performance. These models are often trained specifically for a problem instance with parameters tuned for optimal performance in that specific case. Hence, these methods are not general. We could not take a decision tree used for determining bankruptcy [10] and use it as a baseline for a pathological voice detection problem [11]; at least structural adaptations and retraining are necessary. A good standard baseline should be applicable to all binary classification problems, irrespective of the domain.

Simplicity A universal baseline should not be too complex. Unfortunately, it is hard to determine whether a baseline is too complex for a measure. Essentially, two components are critical in our view: *computational time* and *explainability*. For practical applications, the baseline should be determined relatively quickly. For example, training a neural network many times to generate an average baseline or optimizing the parameters of a certain model could take too much valuable time. Secondly, if a baseline is very complex, it can be harder to draw meaningful conclusions. Is this ingeniously complicated baseline expected to outperform a new model, or is it exactly what we would expect? This leads to the last property of a good standard baseline.

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

Informativeness A baseline should also be informative. When a method achieves a score higher or lower than the baseline, clear conclusions need to be drawn. Is it obvious that the baseline should be beaten? Consider the athletic event *high jump*, where an athlete needs to jump over a bar at a specific height. If the bar is set too low, anyone can jump over it. If the bar is too high, no one makes it. Neither situation gives us additional information to distinguish a professional athlete from a regular amateur. The bar should be placed at a height where the professional could obviously beat it, but the amateur cannot. Drawing from this analogy, a baseline should be beaten by any developed model. If not, this should be considered a major warning sign.

Our research focuses on finding a general, simple, and informative baseline for *binary classification* problems. Although we focus on these types of problems, the three properties should also hold for constructing baselines in other supervised learning problems, such as multiclass classification and regression. Two methods that immediately come to mind are *dummy classifiers* and *optimal threshold classifiers*. They could be ideal candidates for our additional universal baseline.

Dummy classifier A dummy classifier is a *non-learning* model that makes predictions following a simple set of rules. For example, always predicting the *most frequent* class label, or predicting each class with some probability. A dummy classifier is simple and general but not always informative. The information gained by performing better than a simple dummy classifier can be zero. With the plethora of dummy classifiers, the selection of one of those classifiers is also arbitrary and questionable.

Optimal threshold classifier For a large family of binary performance measures, [7] determined that the optimal classifier consists of a sign function with a threshold tailored to each specific measure. To determine the optimal classifier, it is necessary to know or approximate $\mathbb{P}(Y=1 \mid X=x)$, which is the probability that the binary label Y is '1' given the features $X=x$. [8] had a similar approach, but only focused on the F_1 score. The conditional probabilities need to be learned from training data. However, this leads to arbitrary selections, as a model is necessary to approximate these probabilities. It is a clever approach, but unfortunately there is no clear-cut best approximation model for different research domains. If the approximation model is not accurate, the optimal classifier is based on wrong information, which makes it hard to draw meaningful conclusions from this approach.

Both the dummy and optimal threshold classifiers have their strengths and weaknesses. This paper introduces a novel baseline approach called the *Dutch Draw* (DD), which eliminates these weaknesses while keeping the strengths. The DD can be seen as a dummy classifier on steroids. Instead of arbitrarily choosing a dummy classifier, we mathematically derive which classifier has the best expected performance from a family of classifiers. Also, this expected performance can be directly determined, making it very fast to obtain the baseline. The DD baseline is:

- applicable to any binary classification problem;
- reproducible;
- simple;
- parameter-free;

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

- more informative than any single dummy baseline;
- an explainable minimal requirement for any new model.

This makes the DD an ideal candidate for a universal baseline in binary classification.
Our contributions are as follows:

- We introduce the DD and explain why this method produces a universal baseline that is general, simple, and informative for any binary classification problem.
- We provide the mathematical properties of the DD for many evaluation measures, and summarize them in several tables.
- We demonstrate the usefulness of the DD baseline and how it can be used in practice to identify when models should definitely be reconsidered.
- We have made the DD available in a Python package (https://github.com/joris-pries/DutchDraw).

2. Preliminaries

Before formulating the DD, we need to introduce the necessary notation and simultaneously provide elementary information on binary classification. This is required to explain how binary models are evaluated. Then, we discuss how evaluation measures are constructed for binary classification and examine the most commonly used ones.

2.1. Binary classification

The goal of *binary classification* is to learn (from a dataset) the relationship between the input variables and the binary output variable. When the dataset consists of $M \in \mathbb{N}_{>0}$ observations, let $\mathcal{M} := \{1, \dots, M\}$ be the set of observation indices. Each instance, denoted by $\mathbf{x}_i$, has $K \in \mathbb{N}_{>0}$ explanatory feature values. These features can be categorical or numerical. Without loss of generality, we assume that $\mathbf{x}_i \in \mathbb{R}^K$ for all $i \in \mathcal{M}$. Moreover, each observation has a corresponding output value $y_i \in \{0, 1\}$. Now, let $\mathbf{X} := [\mathbf{x}_1 \ \cdots \ \mathbf{x}_M]^\top \in \mathbb{R}^{M \times K}$ denote the matrix with all observations and their explanatory feature values, and let $\mathbf{y} = (y_1, \dots, y_M) \in \{0, 1\}^M$ be the response vector. The complete dataset is then represented by $(\mathbf{X}, \mathbf{y})$. We call the observations with response value 1 'positive', while the observations with response value 0 are 'negative'. Let P denote the number of positives and N the number of negatives. Note that, by definition, $P + N = M$ must hold.

2.2. Evaluation measures

An *evaluation measure* quantifies the prediction performance of a trained model. We categorize the evaluation measures into two groups: *base measures* and *performance metrics* [3]. Since there are two possible values for both the predicted and the true classes in binary classification, there are four base measures: the number of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). Performance metrics are a function of one or more of those four base measures. To shorten the notation, let $\hat{P} := \text{TP} + \text{FP}$ and $\hat{N} := \text{TN} + \text{FN}$ denote the number of positively and negatively predicted instances, respectively.

All the base measures and performance metrics considered are shown in Table 1 along with their abbreviations, possible alternative names, definitions, and corresponding codomains. The

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

TABLE 1: Definitions and codomains of evaluation measures.

Measure	Definition	Codomain
True positives	TP	$\mathbb{N}_0$
True negatives	TN	$\mathbb{N}_0$
False negatives	FN	$\mathbb{N}_0$
False positives	FP	$\mathbb{N}_0$
True positive rate, recall, sensitivity	$\mathrm{TPR} = \mathrm{TP}/\mathrm{P}$	$[0, 1]$
True negative rate, specificity, selectivity	$\mathrm{TNR} = \mathrm{TN}/\mathrm{N}$	$[0, 1]$
Positive predictive value, precision	$\mathrm{PPV} = \mathrm{TP}/\hat{P}$	$[0, 1]$
Negative predictive value	$\mathrm{NPV} = \mathrm{TN}/\hat{N}$	$[0, 1]$
F_β score	$F_\beta = \dfrac{(1+\beta^2)}{(1/\mathrm{PPV}) + (\beta^2/\mathrm{TPR})}$	$[0, 1]$
Youden's J statistic/index, (bookmaker) informedness	$\mathrm{J} = \mathrm{TPR} + \mathrm{TNR} - 1$	$[-1, 1]$
Markedness	$\mathrm{MK} = \mathrm{PPV} + \mathrm{NPV} - 1$	$[-1, 1]$
Accuracy	$\mathrm{Acc} = (\mathrm{TP} + \mathrm{TN})/M$	$[0, 1]$
Balanced accuracy	$\mathrm{BAcc} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR})$	$[0, 1]$
Matthews correlation coefficient	$\mathrm{MCC} = \dfrac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{\hat{P} \cdot \hat{N} \cdot \mathrm{P} \cdot \mathrm{N}}}$	$[-1, 1]$
Cohen's kappa	$\kappa = \dfrac{\mathrm{P_o} - \mathrm{P_e}}{1 - \mathrm{P_e}}$, with $\mathrm{P_o} = \mathrm{Acc},\ \mathrm{P_e} = (\hat{P} \cdot P + \hat{N} \cdot N)/M^2$	$[-1, 1]$
Fowlkes–Mallows index, G-mean 1	$\mathrm{FM} = \sqrt{\mathrm{TPR} \cdot \mathrm{PPV}}$	$[0, 1]$
G-mean 2	$\mathrm{G}^{(2)} = \sqrt{\mathrm{TPR} \cdot \mathrm{TNR}}$	$[0, 1]$
Prevalence threshold	$\mathrm{PT} = \dfrac{\sqrt{\mathrm{TPR} \cdot (1 - \mathrm{TNR})} - (1 - \mathrm{TNR})}{\mathrm{TPR} - (1 - \mathrm{TNR})}$	$[0, 1]$
Threat score, critical success index	$\mathrm{TS} = \dfrac{\mathrm{TP}}{P + \mathrm{FP}}$	$[0, 1]$

codomains show in what set the measure can theoretically take values (without considering the exact values of P, N, $\hat{P}$ and $\hat{N}$). In Section 3, the case-specific codomains are provided when we discuss the evaluation measures in more detail. Some performance metrics, such as the false negative rate (FNR) and false discovery rate (FDR), are omitted, as they can easily be derived from the results of other measures. Finally, note that the list is not exhaustive but contains most of the commonly used evaluation measures.

2.2.1. *Ill-defined measures.* Not every evaluation measure is well-defined. Often, the problem occurs due to division by zero. For example, the *true positive rate* (TPR) defined as TPR =

TABLE 2: Assumptions on domains P, N, $\hat{P}$ and $\hat{N}$. Some measures are not defined if P, N, $\hat{P}$, or $\hat{N}$ is equal to zero. These domain requirements are therefore necessary ($M > 0$ always).

	Domain requirement			
Measure	P	N	$\hat{P}$	$\hat{N}$
TP, TN, FN, FP, Acc, κ	—	—	—	—
TPR, TS	>0	—	—	—
TNR	—	>0	—	—
PPV	—	—	>0	—
NPV	—	—	—	>0
F_β, FM	>0	—	>0	—
J, BAcc, $G^{(2)}$	>0	>0	—	—
MK	—	—	>0	>0
MCC	>0	>0	>0	>0

TP/P cannot be calculated whenever $P = 0$. Therefore, we have made assumptions for the allowed values of P, N, $\hat{P}$, and $\hat{N}$. These are shown in Table 2. One exception is the *prevalence threshold* (PT) [2], where the denominator is zero if TPR is equal to the false positive rate (defined as $\mathrm{FPR} = \mathrm{FP}/N$). Depending on the classifier, this situation could occur regularly. Therefore, PT is omitted throughout the rest of this research.

3. Dutch Draw

In this section, we introduce the DD framework and discuss how this method is able to provide a universal baseline for any evaluation measure. This baseline is general, simple, and informative, which is crucial for a good baseline, as we explained in Section 1. First, we provide the family of DD classifiers and thereafter explain how the optimal classifier generates the baseline.

3.1. Dutch Draw classifiers

Our research aims to provide a universal baseline for any evaluation measure in binary classification. The DD baseline comes from choosing the best DD classifier. Before discussing what 'best' entails, we have to define the DD classifier in general. This classifier generates the predictions for observations by outputting a vector of M random binary values. It is described in words as:

$$\sigma_\theta(M) := \{\text{Take a random subset of } \mathcal{M} \text{ of size } \lfloor M \cdot \theta \rceil \text{ without replacement;} \\ \text{assign 1 to these observations and 0 to the remaining}\}.$$

Here, $\lfloor \cdot \rceil$ is the function that rounds its argument to the nearest integer. The parameter $\theta \in [0, 1]$ controls the percentage of observations predicted as positive. The mathematical definition of σ_θ is $\sigma_\theta(M) := (\mathbf{1}_E(i))_{i\in\mathcal{M}}$ with $E \subseteq \mathcal{M}$ uniformly drawn such that $|E| = \lfloor M \cdot \theta \rceil$, with $(\mathbf{1}_E(i))_{i\in\mathcal{M}}$ the vector with ones in the positions in E and zeros elsewhere. Note that a classifier σ_θ does not learn from the features in the data, just as a dummy classifier. The set of all DD classifiers $\{\sigma_\theta : \theta \in [0, 1]\}$ is the complete family of models that classify a random sample of any size as positive.

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

Given a DD classifier, the number of predicted positives $\hat{P}$ depends on θ and is given by $\hat{P}_\theta := \lfloor M \cdot \theta \rceil$; the number of predicted negatives is $\hat{N}_\theta := M - \lfloor M \cdot \theta \rceil$. Specifically, these two numbers are integers; thus, different values of θ can lead to the same value of $\hat{P}_\theta$. Therefore, we introduce the parameter $\theta^* := \lfloor M \cdot \theta \rceil / M$ as the discretized version of θ. Furthermore, we define

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

as the set of all unique values that θ^* can obtain for all $\theta \in [0, 1]$.

Next, we derive mathematical properties of the DD classifier for every evaluation measure in Table 1 (except PT). Note that the DD is stochastic; thus, we examine the *distribution* of the evaluation measure. Furthermore, we also determine the *range* and *expectation* of a DD classifier.

3.1.1. *Distribution.* The distributions of the base measures (see Section 2.2) are directly determined by σ_θ. Consider, for example, TP, the number of positive observations that are also predicted to be positive. In a dataset of M observations with P labeled positive, $\lfloor M \cdot \theta \rceil$ random observations are predicted as positive in the DD approach. This implies that TP_θ is hypergeometrically distributed with parameters M, P, and $\lfloor M \cdot \theta \rceil$, as the classifier randomly draws $\lfloor M \cdot \theta \rceil$ samples without replacement from a population of size M, where P samples are labeled positive. Thus,

$$\mathbb{P}(\mathrm{TP}_\theta = s) = \begin{cases} \dfrac{\binom{P}{s} \cdot \binom{M-P}{\lfloor M\cdot\theta \rceil - s}}{\binom{M}{\lfloor M\cdot\theta \rceil}} & \text{if } s \in \mathcal{D}(\mathrm{TP}_\theta), \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{D}(\mathrm{TP}_\theta)$ is the domain of TP_θ. The definition of this domain is given in (5).

The other three base measures are also hypergeometrically distributed, following similar reasoning. This leads to:

$$\begin{aligned} \mathrm{TP}_\theta &\sim \mathrm{Hypergeometric}(M, P, \lfloor M \cdot \theta \rceil), \\ \mathrm{FP}_\theta &\sim \mathrm{Hypergeometric}(M, N, \lfloor M \cdot \theta \rceil), \\ \mathrm{FN}_\theta &\sim \mathrm{Hypergeometric}(M, P, M - \lfloor M \cdot \theta \rceil), \\ \mathrm{TN}_\theta &\sim \mathrm{Hypergeometric}(M, N, M - \lfloor M \cdot \theta \rceil). \end{aligned}$$

Note that these random variables are not independent. In fact, they can all be written in terms of TP_θ. This is a crucial effect of the DD approach, as it reduces the formulations to only a function of a single variable. Consequently, most evaluation measures can be written as a linear combination of only TP_θ. With only one random variable, theoretical derivations and optimal classifiers can be determined. As mentioned, $\mathrm{TP}_\theta + \mathrm{FN}_\theta = P$ and $\mathrm{TN}_\theta + \mathrm{FP}_\theta = N = M - P$, and we also have $\mathrm{TP}_\theta + \mathrm{FP}_\theta = \lfloor M \cdot \theta \rceil$, because this denotes the total number of positively predicted observations. These three identities are linear in TP_θ. Thus, each base measure can be written in the form $X_\theta(a, b) := a \cdot \mathrm{TP}_\theta + b$ with $a, b \in \mathbb{R}$. Additionally, let $f_{X_\theta}(a, b)$ be the probability distribution of $X_\theta(a, b)$. Then, by combining the identities, we get

$$\mathrm{TP}_\theta = \mathrm{TP}_\theta, \tag{1}$$

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

$$\mathrm{FP}_\theta = \hat{P}_\theta - \mathrm{TP}_\theta, \tag{2}$$

$$\mathrm{FN}_\theta = P - \mathrm{TP}_\theta, \tag{3}$$

$$\mathrm{TN}_\theta = N - \hat{P}_\theta + \mathrm{TP}_\theta, \tag{4}$$

with $\hat{P}_\theta := \lfloor M \cdot \theta \rceil$.

Example 1. (*Distribution of F_β score.*) To illustrate how the probability function $f_{X_\theta}(a, b)$ can directly be derived, we consider the F_β score [4]. It is the weighted harmonic average between the true positive rate (TPR_θ) and the positive predictive value (PPV_θ). The latter two performance metrics are discussed extensively in the Supplementary Material. The F_β score balances predicting the actual positive observations correctly (TPR_θ) and being cautious in predicting observations as positive (PPV_θ). The factor $\beta > 0$ indicates how much more TPR_θ is weighted compared to PPV_θ. The F_β score is commonly defined as

$$\mathrm{F}_\theta^{(\beta)} = \frac{1+\beta^2}{(1/\mathrm{PPV}_\theta) + (\beta^2/\mathrm{TPR}_\theta)}.$$

By substituting PPV_θ and TPR_θ by their definitions (see Table 1) and using (1) and (2), we get

$$\mathrm{F}_\theta^{(\beta)} = \frac{(1+\beta^2)\mathrm{TP}_\theta}{\beta^2 \cdot P + \lfloor M \cdot \theta \rceil}.$$

Since PPV_θ is only defined when $\hat{P}_\theta = \lfloor M \cdot \theta \rceil > 0$ and TPR_θ is only defined when $P > 0$, for $\mathrm{F}_\theta^{(\beta)}$ we need that both these restrictions hold. The definition of $\mathrm{F}_\theta^{(\beta)}$ is linear in TP_θ and can therefore be formulated as

$$\mathrm{F}_\theta^{(\beta)} = X_\theta\left(\frac{1+\beta^2}{\beta^2 \cdot P + \lfloor M \cdot \theta \rceil}, 0\right).$$

3.1.2. *Range.* The values that $X_\theta(a, b)$ can attain depend on a, b, and the domain of TP_θ. Without restriction, the maximum number that TP_θ can be is P. Then, all positive observations are also predicted to be positive. However, when θ is small enough that $\lfloor M \cdot \theta \rceil < P$, then only $\lfloor M \cdot \theta \rceil$ observations are predicted as positive. Consequently, TP_θ can only reach the value $\lfloor M \cdot \theta \rceil$ in this case. Hence, in general, the upper bound of the domain of TP_θ is $\min\{P, \lfloor M \cdot \theta \rceil\}$. The same reasoning holds for the lower bound: when θ is small enough, the minimum number of TP_θ is 0 since all positive observations can be predicted as negative. However, when θ gets large enough, positive observations have to be predicted positive even if all $M - P$ negative observations are predicted positive. Thus, in general, the lower bound of the domain is $\max\{0, \lfloor M \cdot \theta \rceil - (M - P)\}$. Now, let $\mathcal{D}(\mathrm{TP}_\theta)$ be the domain of TP_θ, then

$$\mathcal{D}(\mathrm{TP}_\theta) := \{i \in \mathbb{N}_0 \colon \max\{0, \lfloor M \cdot \theta \rceil - (M-P)\} \leq i \leq \min\{P, \lfloor M \cdot \theta \rceil\}\}. \tag{5}$$

Consequently, the range of $X_\theta(a, b)$ is given by

$$\mathcal{R}(X_\theta(a, b)) := \{a \cdot i + b\}_{i \in \mathcal{D}(\mathrm{TP}_\theta)}. \tag{6}$$

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

3.1.3. *Expectation.* The introduction of $X_\theta(a, b)$ allows us to write its expected value in terms of a and b. This statistic is required to calculate the actual baseline. Since TP_θ has a Hypergeometric(M, P, $\lfloor M \cdot \theta \rceil$) distribution, its expected value is known and given by

$$\mathbb{E}[\mathrm{TP}_\theta] = \frac{\lfloor M \cdot \theta \rceil}{M} \cdot P.$$

Next, we obtain the following general definition for the expectation of $X_\theta(a, b)$:

$$\mathbb{E}[X_\theta(a, b)] = a \cdot \mathbb{E}[\mathrm{TP}_\theta] + b = a \cdot \frac{\lfloor M \cdot \theta \rceil}{M} \cdot P + b. \tag{7}$$

This rule is consistently used to determine the expectation for each measure.

Example 2. (*Expectation of F_β score.*) To demonstrate how the expectation is calculated for a performance metric, we again consider $\mathrm{F}_\theta^{(\beta)}$. It is linear in TP_θ with $a = (1 + \beta^2)/(\beta^2 \cdot P + \lfloor M \cdot \theta \rceil)$ and $b = 0$, and so its expectation is given by

$$\begin{aligned}\mathbb{E}\big[\mathrm{F}_\theta^{(\beta)}\big] = \mathbb{E}\left[X_\theta\left(\frac{1+\beta^2}{\beta^2 \cdot P + \lfloor M \cdot \theta \rceil}, 0\right)\right] &\overset{(7)}{=} \frac{1+\beta^2}{\beta^2 \cdot P + \lfloor M \cdot \theta \rceil} \cdot \mathbb{E}[\mathrm{TP}_\theta] + 0 \\ &= \frac{\lfloor M \cdot \theta \rceil \cdot P \cdot (1+\beta^2)}{M \cdot (\beta^2 \cdot P + \lfloor M \cdot \theta \rceil)} \\ &= \frac{(1+\beta^2) \cdot P \cdot \theta^*}{\beta^2 \cdot P + M \cdot \theta^*}.\end{aligned}$$

A full overview of the distribution and mean of all the base and performance metrics considered is given in Table 3. The Supplementary Material provides all the calculations to derive the corresponding distributions and expectations.

3.2. Optimal Dutch Draw classifier

Next, we discuss how the DD baseline will ultimately be derived. To do so, an overview is presented in Figure 1. Starting with the definition of the DD classifiers in Section 3.1 and determining their expectations for commonly used measures (see Table 3), we are now able to identify the *optimal* DD classifier. Given a performance metric and dataset, the optimal DD classifier is found by optimizing the associated expectation for $\theta \in [0, 1]$. It is natural to assume that the considered evaluation measures/metrics are maximized. Thus, metrics that are usually minimized are omitted from this paper. The procedure can be followed for these metrics by replacing maximizing with minimizing.

3.2.1. *Dutch Draw baseline.* The optimal DD classifiers and the corresponding DD baseline can be found in Table 4. For many performance metrics, it is optimal to predict all instances as positive or all of them as negative. In some cases, this is not allowed due to ill-defined measures. Then, it is often optimal to only predict one sample differently. For several other metrics, almost all parameter values give the optimal baseline. Next, we give an example to illustrate how the results of Table 4 are derived.

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

TABLE 3: Properties of performance metrics for a DD classifier. Expectation and distribution of each performance metric for a DD classifier σ_θ with $\theta^* = \lfloor M \cdot \theta \rceil / M$.

		Distribution $f_{X_\theta}(a, b)$	
Measure	Expectation	a	b
TP	$\theta^* \cdot P$	1	0
TN	$(1-\theta^*)(M-P)$	1	$M-P-M\cdot\theta^*$
FN	$(1-\theta^*)P$	-1	P
FP	$\theta^*(M-P)$	-1	$M\cdot\theta^*$
TPR	θ^*	$1/P$	0
TNR	$1-\theta^*$	$\dfrac{1}{M-P}$	$1-\dfrac{M\cdot\theta^*}{M-P}$
PPV	$\dfrac{P}{M}$	$\dfrac{1}{M\cdot\theta^*}$	0
NPV	$1-\dfrac{P}{M}$	$\dfrac{1}{M(1-\theta^*)}$	$1-\dfrac{P}{M(1-\theta^*)}$
F_β	$\dfrac{(1+\beta^2)\theta^*\cdot P}{\beta^2\cdot P+M\cdot\theta^*}$	$\dfrac{1+\beta^2}{\beta^2\cdot P+M\cdot\theta^*}$	0
J	0	$\dfrac{M}{P(M-P)}$	$-\dfrac{M\cdot\theta^*}{M-P}$
MK	0	$\dfrac{1}{M\cdot\theta^*(1-\theta^*)}$	$-\dfrac{P}{M(1-\theta^*)}$
Acc	$\dfrac{(1-\theta^*)(M-P)+\theta^*\cdot P}{M}$	$\dfrac{2}{M}$	$1-\theta^*-\dfrac{P}{M}$
BAcc	$\dfrac{1}{2}$	$\dfrac{M}{2P(M-P)}$	$\dfrac{1}{2}-\dfrac{M\cdot\theta^*}{2(M-P)}$
MCC	0	$\dfrac{1}{\sqrt{P(M-P)\theta^*(1-\theta^*)}}$	$-\dfrac{\sqrt{P\cdot\theta^*}}{\sqrt{(M-P)(1-\theta^*)}}$
κ	0	$\dfrac{2}{P(1-\theta^*)+(M-P)\theta^*}$	$-\dfrac{2\theta^*\cdot P}{P(1-\theta^*)+(M-P)\theta^*}$
FM	$\sqrt{\dfrac{\theta^*\cdot P}{M}}$	$\dfrac{1}{\sqrt{P\cdot M\cdot\theta^*}}$	0
$G^{(2)}$	—	Nonlinear in TP_θ	Nonlinear in TP_θ
TS	—	Nonlinear in TP_θ	Nonlinear in TP_θ

Example 3. (*DD baseline for the F_β score.*) To determine the DD baseline, the extreme values of the expectation $\mathbb{E}\big[F_\theta^{(\beta)}\big]$ need to be identified. To do this, examine the function $f\colon [0, 1] \to [0, 1]$ defined as

$$f(t) = \frac{(1+\beta^2)\cdot P \cdot t}{\beta^2 \cdot P + M \cdot t}.$$

TABLE 4: DD baseline. For many performance metrics, the maximum expected score of all allowed DD classifiers is determined, which is the DD baseline. In this table, the baselines and the optimizing parameters are given. '—' denotes that no closed-form expression was found.

Measure	$\max\{\mathbb{E}\}$	$\Theta^{\star}_{\max} := \arg\max\{\mathbb{E}\}$
TPR	1	$\{1\}$
TNR	1	$\{0\}$
PPV	P/M	$\Theta^* \setminus \{0\}$
NPV	$1-(P/M)$	$\Theta^* \setminus \{1\}$
F_β	$\dfrac{(1+\beta^2)\cdot P}{\beta^2\cdot P+M}$	$\{1\}$
J	0	Θ^*
MK	0	$\Theta^* \setminus \{0, 1\}$
Acc	$\max\{P/M, 1-(P/M)\}$	$\{[P<M/2]\}$[a]
BAcc	$\frac{1}{2}$	Θ^*
MCC	0	$\Theta^* \setminus \{0, 1\}$
κ	0	Θ^*[b]
FM	$\sqrt{P/M}$	$\{1\}$
$\mathrm{G}^{(2)}$	—	—
TS	P/M	$\{1\}$[c]

[a] If $P=M/2$, then Θ^*. Note that Iverson brackets are used to simplify the notation.
[b] If $P=M$, then $\Theta^* \setminus \{1\}$.
[c] If $P=1$, then $\Theta^* \setminus \{0\}$.

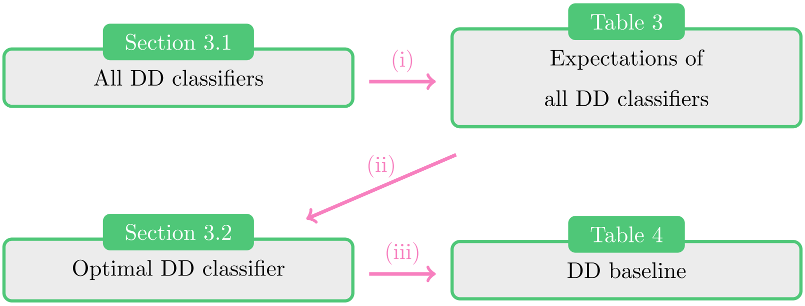

FIGURE 1: Road to the DD baseline. This is an overview of how the DD baseline is determined. (i) All expectations are derived. (ii) The expectation is maximized. (iii) The performance of the best DD classifier is the DD baseline.

The relationship between f and $\mathbb{E}[\mathrm{F}_\theta^{(\beta)}]$ is given as $f(\lfloor M\cdot\theta\rceil/M)=\mathbb{E}[\mathrm{F}_\theta^{(\beta)}]$. To find the extreme values, we have to look at the derivative of f,

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t}=\frac{\beta^2(1+\beta^2)\cdot P^2}{(\beta^2\cdot P+M\cdot t)^2}.$$

It is strictly positive for all t in its domain. Thus, f is strictly increasing in t. This means $\mathbb{E}[\mathrm{F}_\theta^{(\beta)}]$ is non-decreasing in θ and also in θ^*, because the term $\theta^*=\lfloor M\cdot\theta\rceil/M$ is non-decreasing

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

in θ. Hence, the maximum expectation of $\mathrm{F}_\theta^{(\beta)}$ is

$$\max_{\theta\in[1/(2M),1]}\left(\mathbb{E}\left[\mathrm{F}_\theta^{(\beta)}\right]\right)=\max_{\theta\in[1/(2M),1]}\left(\frac{(1+\beta^2)\cdot P\cdot\lfloor M\cdot\theta\rceil}{M\cdot(\beta^2\cdot P+\lfloor M\cdot\theta\rceil)}\right)=\frac{(1+\beta^2)\cdot P}{\beta^2\cdot P+M}.$$

Note that $\lfloor M\cdot\theta\rceil>0$ is a restriction for $\mathrm{F}_\theta^{(\beta)}$; hence, the maximum is taken over the interval $[1/(2M), 1]$. Furthermore, the optimization value $\theta_{\max}$ is given by

$$\theta_{\max}\in\operatorname*{arg\,max}_{\theta\in[1/(2M),1]}\left(\mathbb{E}\left[\mathrm{F}_\theta^{(\beta)}\right]\right)=\operatorname*{arg\,max}_{\theta\in[1/(2M),1]}\left(\frac{\lfloor M\cdot\theta\rceil}{\beta^2\cdot P+\lfloor M\cdot\theta\rceil}\right)=\left[1-\frac{1}{2M},1\right].$$

Following this reasoning, the discrete form $\theta^*_{\max}$ is given by

$$\theta^*_{\max}\in\operatorname*{arg\,max}_{\theta^*\in\Theta^*\setminus\{0\}}\left\{\mathbb{E}\left[\mathrm{F}_{\theta^*}^{(\beta)}\right]\right\}=\operatorname*{arg\,max}_{\theta^*\in\Theta^*\setminus\{0\}}\left\{\frac{\theta^*}{\beta^2\cdot P+M\cdot\theta^*}\right\}=\{1\}.$$

This implies that predicting everything positive yields the largest $\mathbb{E}\left[\mathrm{F}_\theta^{(\beta)}\right]$.

3.2.2. *Selecting performance metrics.* The expectations of the DD given in Table 4 and the codomains of each performance metric given in Table 1 indicate that DD baseline values are identical to the expected score of the 'optimal' model for some performance metrics, like TPR/TNR. Evaluating the performance of a classifier on one of these metrics will always give an unsatisfying result, as input-dependent classifiers can only underperform or match the DD baseline. In addition, evaluating a classifier on only one of the other metrics cannot give a holistic view of the performance of a classifier, as each metric weights the base measures differently. There is no metric 'objectively' better than all other metrics. Multiple performance metrics should be checked to evaluate a classifier properly. The expectations given in Table 4 serve as a lookup table to validate the performance of a classifier from a holistic perspective.

3.2.3. *Non-linear performance metrics.* We have shown that the DD baseline is straightforward for performance metrics that can be written in *linear* terms of TP_θ. However, there are performance metrics, such as $\mathrm{G}_\theta^{(2)}$, where this is impossible. This could make it hard to derive a closed-form expression for the maximum expectation. Previously, we saw in Table 4 that $\theta^*=0$ or $\theta^*=1$ was often optimal. Examining $\mathrm{G}_\theta^{(2)}$ more closely shows that simply selecting $\theta^*=0$ or $\theta^*=1$ would result in the worst possible score. To show that the optimal parameter is less straightforward in this case, we show the optimal $\theta^*_{\max}$ in Figure 2 for a fixed M and increasing P. This shows that $\theta=0.5$ is not always optimal. The optimal value significantly differs when $P\ll N$ or $P\gg N$. We believe that the following reasoning can explain this: Observe that $\mathrm{G}^{(2)}=\sqrt{\mathrm{TPR}\cdot\mathrm{TNR}}$ is zero when either TPR or TNR is zero, which is the minimum score. When there are few positive labels, it must be prevented that all these samples are falsely predicted negative, which is why $\theta^*_{\max}$ is increased. The reverse holds when there are only a few negative samples. The DD baseline can still be derived for non-linear performance metrics by determining the expectations of *all* DD classifiers. However, future research can greatly improve this (see Section 5).

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

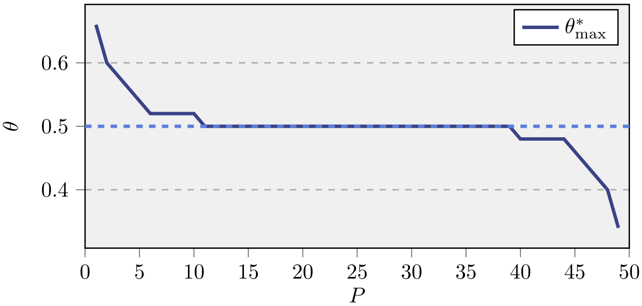

FIGURE 2: Non-trivial θ^*_{max} for $\text{G}^{(2)}_\theta$. For each $P \in \{1, \ldots, 49\}$, the optimal θ^*_{max} is derived for the performance metric $\text{G}^{(2)}_\theta$ with a dataset consisting of P positive and $50 - P$ negative samples. This shows that the optimal value is not straightforward.

4. Dutch Draw in practice

Now that we have established how to derive the DD baseline, it is time to see how the DD could be used in practice.

Example 4. (*Cleveland heart disease*.) A dataset (*Cleveland heart disease*) was provided to predict whether patients have heart disease using several feature values [6]. We randomly split the dataset into a training (90%) and a test set (10%) and chose the F_1 measure to evaluate how well a model performs. The DD baseline, derived using the M and P of the test data, immediately provided a performance reference (0.735) for any model. We trained two common machine learning algorithms (*decision tree* and *k-nearest neighbors*) with default parameters in *scikit-learn* [13]. To estimate the expected performance, we averaged the results obtained from 10 different random seeds for each stochastic model. They achieved an average score of 0.727 and 0.710, respectively, which are worse than the DD baseline. This is a major warning sign. Although the decision tree performed better than k-nearest neighbors, it should still not be used. Thus, we decided to train three other models (*logistic regression*, *random forest*, and *Gaussian naive Bayes*), which ended up performing better than the baseline. Finally, we selected the logistic regression model in practice, as this model achieved the highest score *and* beat the DD baseline. Table 5 shows an overview of the performance of these five models and the baseline on a set of selected performance metrics.

Example 5. (*ImageNet*.) The previous example illustrates how the DD baseline provides insights into the performance of a set of standard machine-learning models. Nevertheless, tasks involving higher-dimensional data necessitate more advanced model architectures. Image-related classification tasks serve as a prevalent example of such complex problems, and we aim to demonstrate in this example how the DD baseline imparts valuable insights for these models tackling more demanding challenges. The *ImageNet Large Scale Visual Recognition Challenge 2012* (ILSVRC2012) offers a significant dataset comprised of over one million images across a thousand diverse object categories [14]. This challenge involves developing a multiclass classification model that accurately predicts which object (such as a sock, volleyball, vase, etc.) is visible in each image. We selected a range of state-of-the-art vision models, including DenseNet121, GoogLeNet, ResNet18, ResNet50, VGG11, and VGG19, to tackle this classification problem. These models were implemented and pre-trained in PyTorch [12]

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

TABLE 5: Comparing performance with the DD baseline. Five standard machine learning algorithms (decision tree (DT), k-nearest neighbors (KNN), logistic regression (LR), random forest (RF), and Gaussian naive Bayes (GNB)) were tested on the Cleveland Heart Disease dataset for many commonly used performance metrics. The average results on the test data using 10 random seeds for the stochastic models are compared with the DD baseline (DDB, first column). The bold scores highlight the situations where a model achieved a score inferior to the DD baseline.

Metric	DDB	DT	KNN	LR	RF	GNB
Acc	0.581	0.710	0.710	0.903	0.871	0.839
BAcc	0.500	0.718	0.729	0.906	0.868	0.840
F_1	0.735	**0.727**	**0.710**	0.914	0.889	0.857
FM	0.762	**0.730**	**0.719**	0.915	0.889	0.857
$G^{(2)}$	0.500	0.716	0.719	0.906	0.867	0.840
J	0.000	0.436	0.457	0.812	0.735	0.679
κ	0.000	0.422	0.434	0.803	0.735	0.672
MCC	0.000	0.430	0.457	0.805	0.735	0.674
MK	0.000	0.425	0.457	0.798	0.735	0.668
NPV	0.419	0.625	0.611	0.857	0.846	0.786
PPV	0.581	0.800	0.846	0.941	0.889	0.882
TS	0.581	**0.571**	**0.550**	0.842	0.800	0.750

TABLE 6: Comparative accuracy scores across categories. For each vision model, the category-wise accuracy is determined on the validation dataset of ILSVRC2012. The DD baseline is identical for each category, as there are 50 images in the data (0.999). For each vision model, the number of categories with scores inferior to the DD baseline is presented.

Model	# Underperforming categories
DenseNet121	49
GoogLeNet	97
ResNet18	94
ResNet50	43
VGG11	112
VGG19	63

on the training dataset of the ImageNet dataset. Typically, the performance of these models is assessed based on top-one or top-five accuracy rates. However, our interest lies in identifying the categories where these models do not perform as well in testing the underlying performance. The test data labels are not publicly available, but we can assess these models using the provided validation dataset, which contains 50 images per category. The outputs of the selected vision models on this dataset are vectors of probabilities representing the likelihood of the input image belonging to each of the 1000 classes. The model assigns the class label with the highest likelihood to the instance. We calculated the four base measures for each category i by categorizing all instances of label i as 'positive' and all other labels as 'negative' in both the predicted and actual data. There are 50 images per category in the validation dataset, so the DD baseline, independently of the selected evaluation metric, is identical for each category ($P = 50$, $M = 50\,000$). Table 6 shows the number of categories having an

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

inferior *Accuracy* score when compared to its corresponding DD baseline (0.999). As the DD baseline takes class imbalance into account, accuracy scores can be interpreted for this binary classification problem even though this metric is not robust against imbalance. It is crucial to explore categories that fail the sanity check, whether this stems from issues with the data or the models themselves. There are, in total, 24 categories where none of the selected vision models actually outperform the DD baseline, indicating underlying issues with these categories. Enhancing the performance in these weaker categories would contribute to the overall efficacy of these multiclass classification models.

Example 6. (*CelebA*.) In the previous example, we delved into a high-dimensional multiclass classification problem, illustrating how the DD baseline aids in the improvement of models tackling this problem. Now, we focus on another realm of high-dimensional challenges, specifically those involving multiple binary classification tasks, the so-called multitask learning problem, namely the image-related task of detecting facial attributes. The problem is to detect these attributes in images, like wearing glasses or a cap. The *CelebA* dataset can be used for this task and generative image creation [9]. The dataset consists of 100 000+ images of 10 177 identities. Each image is annotated with 40 facial attributes. To the best of our knowledge, no publicly available trained model is yet available. Fortunately, we can use pre-trained models of other datasets, like the ImageNet dataset, and replace the last layer of each network with a layer making predictions for each binary facial attribute. The approach is commonly known as transfer learning. We selected the same vision models as for the ImageNet dataset. We trained this last layer on the training dataset of CelebA based on the weighted cross entropy loss that corrects for imbalance in the training data and the standard Adam optimizer in batches of 100 images. The labels for the test dataset comprising 19 962 images are publicly available, so the trained vision models were evaluated on this data. Table 7 shows the results of the vision models evaluated on the test dataset of the CelebA dataset. The DD baseline helps identify attributes where the score of the transferred state-of-the-art vision models was inferior. In our evaluation, we selected the F_1 score to assess each model's performance, as the data has an attribute-dependent class imbalance. Still, similar results can be acquired when selecting another evaluation metric. Remarkably, the performance of all the vision models was inferior for the attributes *Big_Lips*, *Narrow_Eyes*, *Oval_Face*, *Pointy_Nose*, and *Wearing_Necklace*. Like our statement for the ImageNet dataset, data- or model-related issues can cause this underperformance. Again, the DD baseline helps identify features where the score of classification models is inferior.

5. Discussion and conclusion

In this research, we have proposed a new baseline methodology called the *Dutch Draw*. The DD baseline is:

(i) applicable to any binary classification problem;

(ii) reproducible;

(iii) simple;

(iv) parameter-free;

(v) more informative than any single dummy baseline;

(vi) an explainable minimal requirement for any new model.

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

TABLE 7: F_1 scores for vision models on the CelebA test dataset. After retraining the last layer of each pre-trained vision mode, the following F_1 scores are obtained. The DD baseline, DenseNet121, GoogLeNet, ResNet18, ResNet50 are abbreviated as DDB, DN121, GN, RN18, and RN50, respectively. We selected the attributes where at least one of the vision models underperformed compared to the DD baseline. We highlight the performance metric scores in bold, indicating a score inferior to the DD baseline.

Attribute	DDB	DN121	GN	RN18	RN50	VGG11	VGG19
Arched_Eyebrows	0.443	0.476	0.497	**0.432**	**0.383**	0.519	0.446
Bags_Under_Eyes	0.337	0.444	0.431	0.401	0.491	0.350	**0.131**
Big_Lips	0.493	**0.209**	**0.315**	**0.137**	**0.227**	**0.131**	**0.124**
Brown_Hair	0.305	0.416	0.454	0.524	0.548	0.388	**0.255**
Bushy_Eyebrows	0.229	0.348	**0.210**	0.346	0.373	**0.202**	0.282
Double_Chin	0.087	0.325	0.317	0.267	0.220	0.127	**0.071**
Mouth_Slightly_Open	0.662	0.790	0.710	0.765	0.767	**0.655**	**0.643**
Mustache	0.074	0.119	0.228	0.259	0.142	**0.079**	**0.058**
Narrow_Eyes	0.259	**0.201**	**0.081**	**0.122**	**0.112**	**0.018**	**0.016**
Oval_Face	0.456	**0.354**	**0.239**	**0.432**	**0.398**	**0.330**	**0.383**
Pale_Skin	0.081	0.369	0.357	0.432	0.459	0.082	**0.055**
Pointy_Nose	0.444	**0.257**	**0.218**	**0.387**	**0.330**	**0.208**	**0.282**
Receding_Hairline	0.156	0.226	0.305	0.368	0.190	**0.142**	**0.060**
Rosy_Cheeks	0.134	0.430	0.333	0.350	0.438	**0.103**	**0.117**
Straight_Hair	0.347	0.384	**0.321**	0.399	0.540	0.383	**0.234**
Wearing_Earrings	0.343	0.536	0.412	0.456	0.479	0.372	**0.262**
Wearing_Necklace	0.242	**0.051**	**0.036**	**0.037**	**0.109**	**0.038**	**0.018**

We have shown that for the most commonly used measures, the DD baseline can be theoretically determined (see Table 4). When the baseline cannot be derived directly, it can quickly be identified by computation. For most performance metrics, the DD baseline reduces to one of the following three cases:

(i) always predicting positive or negative;

(ii) always predicting positive or negative, except for one instance;

(iii) any DD classifier, except maybe for $\theta^* = 0$ or $\theta^* = 1$.

However, there are exceptions to these three cases, as was shown with the $\mathrm{G}_\theta^{(2)}$. This shows that the DD is not always reduced to one of the three previously mentioned cases and does not always give straightforward results.

By introducing the DD baseline, we have simplified and improved the evaluation process of new binary classification methods. We consider it a minimum requirement for any novel model to at least beat the DD baseline. When this does not happen, the question is raised of how much a new method has even learned from the data since the DD baseline is derived from dummy classifiers. When the novel model has beaten the DD baseline, it should still be compared to a state-of-the-art method in that domain to obtain additional insights. In Section 4, we have shown how the DD should be used in practice. In Example 4, we showed that commonly used approaches, such as k-nearest neighbors and a decision tree, can underperform.

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

An advantage of the DD is that the expected performance can be directly derived, as opposed to other (stochastic) models, where the expected value can only be estimated through (many) experiments. Examples 5 and 6 discussed two image-related tasks where the DD baseline can identify task-related underperformance of vision models. The DD baseline can identify problematic categories in multiclass problems or specific tasks in multitask classification problems. The goal of these experiments is to show that benchmarking classification models is essential. The insights obtained from the DD baseline provide valuable perspectives that empower us to enhance and refine models. Hence, using the Dutch Draw as a general, simple, and informative baseline should be the new gold standard in any binary model evaluation process.

5.1. Further research

Our baseline is a stepping stone for further research, where multiple avenues should be explored. We discuss five possible research directions.

First, a wide range of other input-independent classifiers should be explored. Does the DD baseline outperform these methods, and can this be proven? If so, it would provide a solid argument to use the DD baseline over any other input-independent baseline in future applications, which is why this research question is important.

Second, we can now determine whether a binary classification model performs better than a universal baseline. However, we do not yet know *by how much* it performs better (or worse). For example, let the baseline have a score of 0.5 and a new model a score of 0.9. How much better is the latter score? It could be that a tiny bit of extra information quickly pushes the score from 0.5 to 0.9. Or, it is possible that a model needs a lot of information to understand the intricacies of the problem, making it very difficult to reach a score of 0.9. Thus, it is necessary to quantify how hard it is to reach any score. Also, when another model is added that achieves a score of 0.91, can the difference in the performance of these models be quantified? Is it only a slightly better model, or is it a leap forward?

Third, our DD baseline could be used to construct new standardized evaluation measures from their original versions. The advantage of these new measures would be that the interpretation of their scores is independent of the number of positive and negative observations in the dataset. In other words, the DD baseline would already be incorporated in the new measure, so comparing a score to the baseline is no longer necessary. The DD baseline can be used to scale a measure in many ways. Let Δ_{max} and Δ_{min} denote the maximum and minimum DD baseline, respectively. As an example, a measure μ with range $[\mu_{\text{min}}, \mu_{\text{max}}]$ that needs to be maximized can be rescaled by

$$\mu_{\text{rescaled}} = \begin{cases} -1 & \text{if } \mu \leq \Delta_{\text{min}}, \\ \dfrac{\mu - \Delta_{\text{max}}}{\Delta_{\text{max}} - \Delta_{\text{min}}} & \text{if } \Delta_{\text{min}} \leq \mu \leq \Delta_{\text{max}}, \\ \dfrac{\mu - \Delta_{\text{max}}}{\mu_{\text{max}} - \Delta_{\text{max}}} & \text{otherwise.} \end{cases}$$

Everything below the lowest DD baseline (Δ_{min}) gets value -1, because every Dutch Draw classifier performs better. This should be a major warning sign. A score between Δ_{min} and Δ_{max} is rescaled to $[-1, 0]$. This value indicates that the performance is still worse than the best DD baseline. All scores above Δ_{max} are scaled to [0,1]. In this case, the performance of a classifier outperforms the best DD baseline.

Fourth, another natural extension would be to drop the binary assumption and consider multiclass classification. This is more complicated than it seems because not every multiclass

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

evaluation measure follows automatically from its binary counterpart. However, we expect that for most multiclass measures, it is again optimal always to predict a single specific class.

Fifth, for some (less straightforward) performance metrics, the DD baseline is derived by examining the expectations of *all* DD classifiers. Thus, faster techniques should be developed for large applications. Insights could greatly improve the computation time. For example, we conjecture for $\mathrm{G}_{\theta}^{(2)}$ that $\theta^*_{\max} \in \left[0, \frac{1}{2}\right]$ when $P > N$, and $\theta^*_{\max} \in \left[\frac{1}{2}, 1\right]$, when $P < N$. This already reduces the search domain by half. Proving convexity could also make it easier to derive the optimal value. Decreasing the computation time could be essential for some large applications and should be investigated.

Finally, we have published the code for the DD so the reader can easily implement the baseline in their binary classification problems.

Acknowledgements

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

Funding Information

There are no funding bodies to thank relating to the creation of this article.

Competing Interests

There were no competing interests to declare during this article's preparation or publication process.

Data

All data used in this research is cited in the appropriate sections.

Code availability

The Dutch Draw code can be found at https://github.com/joris-pries/DutchDraw.

Authors' contributions (Contributor Roles Taxonomy (CRediT))

- Etienne van de Bijl: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration.
- Jan Klein: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration.
- Joris Pries: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration.
- Sandjai Bhulai: Conceptualization, Validation, Writing - Review & Editing, Supervision.

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press

- Mark Hoogendoorn: Conceptualization, Validation, Writing - Review & Editing, Supervision.
- Rob van der Mei: Conceptualization, Validation, Writing - Review & Editing, Supervision.

Supplementary Material

The supplementary material for this article can be found at https://doi.org/10.1017/jpr.2024.52.

References

[1] Araújo, R. de A., Oliveira, A. L. and Meira, S. (2017). A morphological neural network for binary classification problems. *Eng. Appl. Artif. Intel.* **65**, 12–28. https://doi.org/10.1016/j.engappai.2017.07.014

[2] Balayla, J. (2020). Prevalence threshold (ϕe) and the geometry of screening curves. *PLoS ONE* **15**, e0240215. https://doi.org/10.1371/journal.pone.0240215

[3] Canbek, G., Sagiroglu, S., Temizel, T. T. and Baykal, N. (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. In *Proc. 2017 Int. Conf. Computer Science and Engineering (UBMK)*, IEEE. https://doi.org/10.1109/ubmk.2017.8093539

[4] Chinchor, N. (1992). MUC-4 evaluation metrics. In *Proc. 4th Conf. Message Understanding*, Association for Computational Linguistics, pp. 22–29. https://doi.org/10.3115/1072064.1072067

[5] Couronné, R., Probst, P. and Boulesteix, A.-L. (2018). Random forest versus logistic regression: A large-scale benchmark experiment. *BMC Bioinformatics* **19**, 270. https://doi.org/10.1186/s12859-018-2264-5

[6] Janosi, A., Steinbrunn, W., Pfisterer, M. and Detrano, R. (1988). *Heart Disease*. UCI Machine Learning Repository, available at https://doi.org/10.24432/C52P4X

[7] Koyejo, O., Natarajan, N., Ravikumar, P. and Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. In *Proc. 27th Int. Conf. Neural Information Processing Systems*, Vol. 2. MIT Press, Cambridge, MA, pp. 2744–2752.

[8] Lipton, Z. C., Elkan, C. and Naryanaswamy, B. (2014). Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases*, eds T. Calders, F. Esposito, E. Hüllermeier and R. Meo, Springer, Berlin, pp. 225–239.

[9] Liu, Z., Luo, P., Wang, X. and Tang, X. (2015). Deep learning face attributes in the wild. In *Proc. Int. Conf. Computer Vision*, pp. 3730–3738.

[10] Min, J. H. and Jeong, C. (2009). A binary classification method for bankruptcy prediction. *Expert Syst. Appl.* **36**, 5256–5263. https://doi.org/10.1016/j.eswa.2008.06.073

[11] Muhammad, G. and Melhem, M. (2014). Pathological voice detection and binary classification using MPEG-7 audio features. *Biomed. Sig. Proc. Control* 11, 1–9. https://doi.org/10.1016/j.bspc.2014.02.001

[12] Paszke A. *et al.* (2019). PyTorch: An imperative style, high-performance deep learning library. *In Advances in Neural Information Processing Systems*, Vol. **32**, Curran Associates, Inc., Red Hook, NY, pp. 8024–8035.

[13] Pedregosa, F. *et al.* (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830.

[14] Russakovsky, O. *et al.* (2015). ImageNet large scale visual recognition challenge. *Int. J. Computer Vision* **115**, 211–225. https://doi.org/10.1007/s11263-015-0816-y

[15] Sergioli, G., Giuntini, R. and Freytes, H. (2019). A new quantum approach to binary classification. *PLoS ONE* **14**, e0216224. https://doi.org/10.1371/journal.pone.0216224

[16] Shahraki, H. R., Pourahmad, S. and Zare, N. (2017). *k* important neighbors: A novel approach to binary classification in high dimensional data. *BioMed Research International* **2017**, 7560807. https://doi.org/10.1155/2017/7560807

[17] Sundarkumar, G. G. and Ravi, V. (2013). Malware detection by text and data mining. In *Proc. 2013 IEEE Int. Conf. Computational Intelligence and Computing Research*, IEEE, pp. 1–6. https://doi.org/10.1109/iccic.2013.6724229

[18] Wang, S. and Manning, C. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proc. 50th Ann. Meeting Association for Computational Linguistics*, Vol. 2, pp. 90–94. https://www.aclweb.org/anthology/P12-2018

[19] Wirth, R. and Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Proc. 4th Int. Conf. Practical Applications of Knowledge Discovery and Data Mining*. https://cs.unibo.it/montesi/CBD/Beatriz/10.1.1.198.5133.pdf

https://doi.org/10.1017/jpr.2024.52 Published online by Cambridge University Press