



BenchMetrics Prob: benchmarking of probabilistic error/loss performance evaluation instruments for binary classification problems

Gürol Canbek¹

Received: 5 October 2022 / Accepted: 21 March 2023 / Published online: 19 April 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Probabilistic error/loss performance evaluation instruments that are originally used for regression and time series forecasting are also applied in some binary-class or multi-class classifiers, such as artificial neural networks. This study aims to systematically assess probabilistic instruments for binary classification performance evaluation using a proposed two-stage benchmarking method called BenchMetrics Prob. The method employs five criteria and fourteen simulation cases based on hypothetical classifiers on synthetic datasets. The goal is to reveal specific weaknesses of performance instruments and to identify the most robust instrument in binary classification problems. The BenchMetrics Prob method was tested on 31 instrument/instrument variants, and the results have identified four instruments as the most robust in a binary classification context: Sum Squared Error (*SSE*), Mean Squared Error (*MSE*), Root Mean Squared Error (*RMSE*, as the variant of *MSE*), and Mean Absolute Error (*MAE*). As *SSE* has lower interpretability due to its $[0, \infty)$ range, *MAE* in $[0, 1]$ is the most convenient and robust probabilistic metric for generic purposes. In classification problems where large errors are more important than small errors, *RMSE* may be a better choice. Additionally, the results showed that instrument variants with summarization functions other than mean (e.g., median and geometric mean), *LogLoss*, and the error instruments with relative/percentage/symmetric-percentage subtypes for regression, such as Mean Absolute Percentage Error (*MAPE*), Symmetric *MAPE* (*sMAPE*), and Mean Relative Absolute Error (*MRAE*), were less robust and should be avoided. These findings suggest that researchers should employ robust probabilistic metrics when measuring and reporting performance in binary classification problems.

Keywords Performance measures · Probabilistic error/loss · Squared error · Binary classification · Regression · Time series forecasting

1 Introduction

Performance evaluation instruments are used to assess, compare, and publish a classifier's categorization or labeling success in training, validation, and test datasets. Several instruments have been proposed and used in the literature. Researchers rely on existing practices and prefer one or a few conventional instruments for performance evaluation [1]. Apart from well-known confusion-matrix-derived performance evaluation instruments (e.g., accuracy and *F1*), several probabilistic error/loss instruments have been defined and used for different machine learning algorithms in various applications in the literature [1, 2]. Those error instruments,

such as Mean Absolute Error (*MAE*, also known as “Mean Absolute Deviation” (*MAD*), “L1 loss” or “L1-norm”), *MSE* (Mean Squared Error, also known as “Brier score”, “L2 loss” or “L2-norm”), and Root Mean Square Error (*RMSE*, also known as Root Mean Square Deviation, *RMSD*) [3–5], summarize the error values calculated for.

- Regression (e.g., support vector regression [6]),
- Time series forecasting (e.g., stock market forecasting [7], movie box office prediction [8], prediction of future [9] or ongoing epidemic events including COVID-19 [10], temperature distribution prediction in snap-curing ovens [11], and multivariate macroeconomic time series prediction [12]),
- Partly binary-class or multi-class classification applications (e.g., multiple-category water quality classification [13], multi-class intrusion detection [14]), and
- Extreme machine learning applications recently [15].

✉ Gürol Canbek
gurol@canbek.com

¹ Pointr, Ankara, Turkey

Probabilistic error/loss instruments, either alone or combined with confusion-matrix-derived instruments, are used to evaluate, publish, and compare classification performances in a wide variety of applications in existing and emerging fields, such as backpropagation neural networks [16, 17], monotonic classifications where predictive models should be monotonic [18], machine fault classification from vibration data [19], meteor detection in camera data [20], carp species identification from fish images [21] with convolutional neural networks, and groundwater level prediction from wells data with ensemble deep learning [22]. Researchers compare their predictions with other algorithms using multiple error/loss instruments (*e.g.*, comparing recurrent neural networks' performance via *MAE*, Mean Absolute Percentage Error (*MAPE*), and *RMSE* [23]). Other alternative examples that were published in the literature are provided in Sect. 4.

It is important to note that performance evaluation instruments are essential for ranking and comparing alternative classification methods such as feature selection, machine learning algorithm selection, and model tuning, in order to provide guidance on which method is the best. In this article, the term 'performance instrument' or 'instrument' is used as a generic term that is interchangeable with other terms used in the literature, such as 'performance measure' or 'performance metric'. This distinction is described in more detail in previous work [24]. Previous experiments with probabilistic error instruments have shown that there is a low correlation between the ranks generated by different instruments, indicating that different instruments may indicate different classifications as the best [25]. In many application domains, the choice of performance instrument becomes a convention that is followed in consecutive studies and may even be carried over to new domains without any additional evaluation. As a result, selecting instruments for emerging fields or interdisciplinary problem domains may be challenging. For example, in the rapidly growing field of deep learning, researchers lack clear guidelines for selecting a performance instrument, leading to the publication and evaluation of several alternatives, including the recent or rarely used ones [20].

Systematical assessment of the robustness of probabilistic error/loss instruments in the classification context is lacking in the literature. It is not appropriate to use probabilistic instruments developed for regression problems as they are in classification evaluations. Researchers often do not recognize the limitations of probabilistic instruments, especially in binary classification evaluations. This study proposes a benchmarking method, called BenchMetrics Prob (Benchmarking of Metrics for Probabilistic instruments) to evaluate the robustness of probabilistic error instruments. The study first reviews and describes

categories of classification performance evaluation instruments and assesses the robustness of probabilistic error instruments using hypothetical classifiers on synthetic datasets across seven simulation cases and fourteen sub-cases (Stage 1). In addition to this quantitative evaluation, a qualitative approach (Stage 2) with five criteria is proposed. As discussed in Sect. 4, there are few studies that provide high-level qualitative requirements for regression problems, not for classification problems. BenchMetrics Prob is an initial effort to create a theoretical model through the proposed quantitative criteria and a set of robustness principles through the qualitative criteria to compare the probabilistic performance instruments. The proposed benchmarking method can also be extended or compared through expert evaluations of the instruments in real-world problems in various or specific classification domains.

The evaluation of 31 instruments and instrument variants via BenchMetrics Prob has shown that some instruments should not be preferred in the performance evaluation of binary classification applications, or they should be used with knowledge of their limitations reported in this study. Therefore, this study recommends instruments from a large number of alternatives originally proposed for regression or time series forecasting to ensure that performance evaluation is in the context of binary classification. As an additional contribution, the study also suggests categorizing performance instruments as main and variants, making it easier to analyze and evaluate numerous instruments.

BenchMetrics Prob is a new study that complements and extends BenchMetrics, a benchmarking method proposed in the author's previous work [26]. BenchMetrics evaluated and compared thirteen confusion-matrix-derived metrics, along with two recently proposed metrics (optimized precision and index of balanced accuracy for *G*): True Positive Rate (*TPR*), True Negative Rate (*TNR*), Positive Predictive Value (*PPV*), Negative Predictive Value (*NPV*), Accuracy (*ACC*), Informedness (*INFORM*), Markedness (*MARK*), Balanced Accuracy (*BACC*), *G*, Normalized Mutual Information (*nMI*), *F1*, Cohen's Kappa (*CK*) [27], and Mathews Correlation Coefficient (*MCC*) [28]. BenchMetrics identified the weak and robust issues of individual confusion-matrix-derived metrics, ultimately finding that *MCC* is the most robust. BenchMetrics Prob, on the other hand, focuses on evaluating the robustness of probabilistic error/loss instruments specifically in a classification context.

The rest of the paper is structured as follows. Section 2 presents a different approach to describing three categories of binary classification performance evaluation instruments and their properties. Section 3 provides an overview of probabilistic error/loss instruments being evaluation in

Table 1 Performance evaluation instrument categories and attributes

Instrument type	Derived from	Classifier threshold (θ)	Performance insights
Confusion-matrix-derived	$M(TP, FP, FN, TN)$	Single/final	External, kinetic, production
Graphical-based	$\sum_{\theta=0}^1 M_{1\theta} \Delta M_{2\theta}$	Multiple	Internal, potential, development
Probabilistic error/loss	Loss: LogLoss (See Equation (B37)) Error: $\sum_{i=1,Sn}^g e_i(c_i, p_i)$	Single/final	Internal, kinetic, development/production
M	A 4-ary summary function (performance metric) <i>e.g.</i> , $ACC = (TP + TN)/(TP + FP + FN + TN)$ where ACC : Accuracy metric, TP : True Positive, FP : False Positive, FN : False Negative, and TN : True Negative)		
M_1, M_2	Two trade-off confusion-matrix-derived performance metrics <i>e.g.</i> , TPR vs. $1 - TNR$ where TPR : True Positive Rate and TNR : True Negative Rate		
$\sum_{\theta=0}^1 y \Delta x$	Area-under-curve (θ : design threshold value in $[0, 1]$ interval) x : $1 - TNR$ (or FPR) and y : TPR with $AUCROC$ (Area-under-ROC-curve, ROC: receiver-operating-characteristic) x : TPR and y : PPV with $AUCPR$ (area-under-precision-recall-curve where PPV : Positive Predictive Value or precision and TPR : True Positive Rate or recall)		
M_θ	A metric value for a classifier model having a specific θ (design) threshold value		
i	An instance to be classified		
Sn	Sample size		
c_i	The class value (ground truth) of an instance (0: negative, 1: positive in binary classification)		
p_i	The label score (prediction) in $[0, 1]$ for a classified instance with a θ (decision) threshold value in $(0, 1)$ interval, <i>e.g.</i> , 0.5		
e_i	Error function for the instance (see Table 10 for examples)		
g	Aggregation function for all error values (<i>e.g.</i> , arithmetic mean, see Table 10 for examples)		

this study. Section 4 examines the literature on evaluating probabilistic instruments from regression or classification contexts. Section 5 describes the proposed benchmarking method (BenchMetrics Prob) comprising two stages. This section explains the proposed qualitative criteria in Stage 1 and quantitative cases generated by a tool developed to analyze instruments quantitatively in Stage 2. The overall benchmarking results with instrument robustness rankings are provided in Sect. 6. Section 7 discusses the study in general, its limitations, and future work. The conclusions are outlined in Sect. 8. Appendix A gives preliminaries for classification and performance evaluation. Appendix B contains the equations of probabilistic error/loss instruments in distinct notation, organized by common error functions. Appendix C provides the distribution of aggregation and error functions of probabilistic error instruments. The BenchMetrics Prob instrument calculator and simulation tool are introduced in Appendix D, along with its graphical user interface, and basic usages.

Benchmarking tool and online material

This study provides a calculator/simulation tool for probabilistic error/loss instruments at <https://github.com/gurol/BenchMetricsProb>. Users can use the tool to reproduce the

controlled cases for hypothetical classifiers on synthetic datasets, which is used in Stage 2 quantitative benchmarking. It is possible to generate new simulation cases. The tool and results are provided as Microsoft® Excel® spreadsheet files (BenchMetricsProb_SimulationTool.xlsx and BenchMetricsProb_SimulationCasesResults.xlsx).

2 Binary classification performance evaluation instrument categories

Table 1 describes and compares three categories of performance evaluation instruments: confusion-matrix-derived, graphical-based, and probabilistic error/loss instruments. The categories are distinguished by three pillars of insights suggested in this study: “whether an instrument represents the internal vs. external performance of a classifier, representing potential vs. kinetic aspects of its classification model,” and “whether it is used during classifier model development or to declare the final performance of a classifier in production.” These are summarized in the fourth column of Table 1’s. Confusion-matrix-derived and graphical-based instruments are described in Appendix A.1 and Appendix A.2, respectively.

2.1 Probabilistic error/loss instruments in binary classification

Graphical-based instruments and probabilistic error/loss instruments, such as *MSE* or *LogLoss*, do not depend on a confusion matrix. However, unlike graphical ones, they do not depict the performance of multiple design thresholds. They summarize deviations from the true probability or prediction uncertainty of a classifier that is modeled with a single/final design threshold [29]. Although probabilistic error/loss instruments are preferred for regression rather than classification problems, or for multi-class rather than binary classification, they can be reported in binary classification (e.g., neural network classification models), usually with one or more confusion-matrix-derived metrics. A probabilistic instrument is a “reliability metric” [1, 30] instead of a “performance metric”. Probabilistic instruments provide insight for a *single* classification application, like confusion-matrix-derived instruments. However, they can be used to assess a classifier’s production performance and optimize model development. A 3D surface plot, for example, can display *MSE* with two dependent variables to visualize internal design variations [31] (see Fig. 10 in the referenced study). Likewise, *RMSE* is used to compare the kernels that will result in smooth functions for the classifier boundary and regression functions [32]. A probabilistic instrument (e.g., *RMSE*) can be used for a classifier producing a continuous outcome, whereas a confusion-matrix-derived instrument (e.g., *ACC*) can also be reported for the classifier evaluated according to binary outcome [33].

Contrary to zero–one loss metrics (e.g., *MCR*: Misclassification Rate, *FPR*: False Positive Rate, *FNR*: False Negative Rate, *FDR*: False Discovery Rate, and *FOR*: False Omission Rate), probabilistic error/loss instruments assess the performance error of scoring or soft (non-crisp) classifiers that label instances with a reported or attached belief value (score, probability or likelihood) based on a decision boundary. For example, instead of hard labeling with zero (0) or one (1), a binary classifier model with a $\Theta = 0.50$ internal decision boundary value (the right side is for positive labels, the left side is for negative ones) in $[0, 1]$ interval can correctly label a positive i^{th} instance ($c_i = 1$) as positive with a $p_i = 0.85$ score (also known as “soft label”) instead of the highest belief value where $p_i = 1$. In soft classification, a sample can be labeled with partial membership (positive with $p_i = 0.85$ and negative with $p_i = 1 - 0.85 = 0.15$). For another (j^{th}) instance with a true negative label ($c_j = 0$), the classifier assigns a score of $p_j = 0.40$ instead of $p_j = 0$. As a result, the probabilistic classification error (e_j) can be interpreted as a distance function for those instances where the classifier’s soft labeling is more uncertain ($|0.85 - 0.50| = 0.35$ for the positive i^{th} instance, compared to

$0.10 = |0.40 - 0.50|$ for the negative j^{th} instance) relative to the decision boundary ($\Theta = 0.5$).

Probabilistic metrics such as *MAE*, *MSE*, and *RMSE* are often expressed in a right-open interval $[0, \infty)$ in regression, but they become bound by the range of the dependent variable (e.g., $[0, 1]$ in a binary classification). For example, in a comparison between a binary classification and a regression model predicting weather temperatures on Earth (ranging from $\pm 40^\circ\text{C}$ or 100°F to -40°F), binary scoring classifiers produce absolute errors ranging from 0 and 1 for the “cold” (0) and “hot” (1) labels. For each i^{th} example: class c_i is either 0 “cold” or 1 “hot” ($c_i \in \{0, 1\}$) and the corresponding prediction value p_i is between 0 and 1 ($p_i \in [0, 1]$). Regression classifiers, on the other hand, produce absolute errors ranging from $0^\circ\text{C}/^\circ\text{F}$ to 80°C (140°F). A threshold like $20^\circ\text{C}/68^\circ\text{F}$ can be used to categorize the scalar outcomes into “cold” and “hot” labels. While Mean Relative Absolute Error (*MRAE*) and *LogLoss* also measure classification error, their summary functions yield a right-open interval in both binary classification and regression. However, interpreting probabilistic performance instruments can be challenging, especially when compared to well-known confusion-matrix-derived metrics according to some practitioners [34].

2.2 A general overview of instrument categories

As a toy example, let us consider a classifier that is modeled with a specific design threshold, which yields the following confusion matrix elements for a test dataset of size $Sn = 20$: $TP = 10$, $FP = 0$, $FN = 0$, and $TN = 10$ (number of True/False Positives/Negatives). In this best-performance case scenario where the classifier has correctly classified all instances in the dataset:

- Confusion-matrix-derived metrics produce the maximum value (e.g., $TPR = TNR = PPV = NPV = ACC = FI = CK = MCC = 1$), whereas loss instruments produce the minimum value (e.g., $FNR = FPR = FDR = FOR = MCR = 0$).
- A probabilistic error/loss instrument, such as *MAE*, could most likely be any value in $[0, 0.5]$ ¹ (ideally close to zero, but the possible range is in $[0, 1]$).
- A graphical-based metric such as *AUCROC* (Area-Under-Receiver-Operating-Characteristic-Curve) yields any value in $[0, 1]$ interval (ideally close to 1), indicating that the classifier has a perfect trade-off between *TPR* and *FPR*, independent of the current classification result

¹ For ten negative samples (e.g., $i = 1, \dots, 10$): $c_i = 0$ and example $p_i = 0.49$ then $|c_i - p_i| = 0.49$. For remaining ten positive samples (e.g., $i = 11, \dots, 20$): $c_i = 1$ and example $p_i = 0.51$ then $|c_i - p_i| = 0.49$. Hence, $MAE = 0.49$.

represented by confusion-matrix-derived and/or probabilistic instruments.

AUCROC measures the ability of a binary classifier to discriminate between positive and negative classes across all possible thresholds. A value of 0.90 indicates that the classifier is performing well in this regard, with a high true positive rate and a low false positive rate. On the other hand, the *ACC* and *MSE* are instruments that measure the overall performance of the classifier at a specific decision threshold. An *ACC* of 0.95 indicates that the classifier is correctly classifying 95% of the instances, while an *MSE* of 0.20 indicates that, on average, the difference between the predicted probabilities and actual labels is 0.20. It is important to note that the ideal threshold for a classifier depends on the specific problem and domain. Therefore, a high *AUCROC* score does not necessarily guarantee an ideal threshold, as different thresholds can produce different results for other metrics such as *ACC* and *MSE*.

Although confusion-matrix-derived and probabilistic error/loss instruments are not directly associated, the former can be expressed in terms of expected values formulated by the expected values of confusion-matrix elements ($\mathbb{E}(TP)$, $\mathbb{E}(FP)$, $\mathbb{E}(FN)$, and $\mathbb{E}(TN)$) [35]. For example, $\mathbb{E}(ACC)$, $\mathbb{E}(TPR)$, or $\mathbb{E}(F1)$, where $\mathbb{E}(I)$ denotes the expected value of an instrument I , can be expressed with these expected values. Although this study does not cover the performance evaluation of multi-class/multi-labeled classification [36–38], binary classification evaluation metrics can be adapted for such cases. This can be achieved by micro- or macro-averaging binary metrics over time [39, 40] or by making specific adaptations such as the one-versus-all approach [37, 38, 41].

3 Probabilistic error/loss performance evaluation instruments

Figure 1 illustrates a probabilistic-based classification scenario for a sample size of $Sn=4$, where the true scores are $c_i=[1, 0, 1, 0]$ and the prediction scores are $p_i=[0.8, 0.6, 0.4, 0.2]$ for the i^{th} example. The labels for each example are determined based on a decision boundary, which is the line or surface that separates different classes in a classification problem. The decision boundary in probabilistic-based classifiers is determined by the probability threshold used to classify a given input and can be linear or non-linear depending on the classifier used. The classifier assigns labels to the four examples based on their p_i scores using a 0.5 linear decision boundary, resulting in the following classification:

- $c_1=1$ (“positive”) is classified as “positive” with $p_1=0.8 \geq 0.5$ (true positive),

- $c_2=0$ (“negative”) is classified as “positive” with $p_2=0.6 \geq 0.5$ (false positive),
- $c_3=1$ (“positive”) is classified as “negative” with $p_3=0.4 < 0.5$ (false negative), and
- $c_4=0$ (“negative”) is classified as “negative” with $p_4=0.2 < 0.5$ (true negative).

As a minimal complete example, the entire classification results can be summarized as $TP=1$, $FP=1$, $FN=1$, and $TN=1$, representing all four possible outcomes of a binary classifier. There are two categories of probabilistic performance instruments: probabilistic loss and probabilistic error instruments. *LogLoss* (also known as “binary cross-entropy” in binary-class classification and “categorical cross-entropy” (*CCE*) in multi-class classification) is a well-known and widely used instrument in probabilistic loss instrument, while there are numerous probabilistic error instruments. For the example in Fig. 1, *LogLoss* can be calculated as follows:

$$\begin{aligned} 1 \text{ (“P”)} \log_2 0.8 &= \log_2 0.8 = -0.32 \text{ for } i=1 \\ (1-0 \text{ (“N”)}) \log_2 (1-0.6) &= -1.32 \text{ for } i=2 \\ 1 \text{ (“P”)} \log_2 0.4 &= \log_2 0.4 = -1.32 \text{ for } i=3 \\ (1-0 \text{ (“N”)}) \log_2 (1-0.2) &= -0.32 \text{ for } i=4 \end{aligned}$$

$$\begin{aligned} \text{LogLoss} &= -\frac{1}{Sn} \sum_i c_i \log_2 p_i + (1-c_i) \log_2 (1-p_i) \\ &= -\frac{1}{4}(-0.32 - 1.32 - 1.32 - 0.32) = 0.82. \end{aligned}$$

The error function, in its purest form, for the i^{th} instance or example is given by $e_i=c_i-p_i$. Mean Error (*ME*)² is the arithmetic mean of errors for all sample instances. For the sake of simplicity, *MAE* is used in the following example calculation for probabilistic error instruments.

$$\begin{aligned} |1 \text{ (“P”)} - 0.8| &= 0.2 \text{ for } i=1 \\ |0 \text{ (“N”)} - 0.6| &= 0.6 \text{ for } i=2 \\ |1 \text{ (“P”)} - 0.4| &= 0.4 \text{ for } i=3 \\ |0 \text{ (“N”)} - 0.2| &= 0.2 \text{ for } i=4 \end{aligned}$$

$$\text{MAE} = \frac{1}{Sn} \sum_i |c_i - p_i| = \frac{1}{2}(0.2 + 0.6 + 0.4 + 0.2) = 0.35.$$

Probabilistic error/loss instruments provide sufficient information for evaluating a classifier’s internal performance during model development. However, during production or validation/testing, a probabilistic instrument value should be evaluated and published with at least one confusion-matrix-derived instrument (e.g., *MSE* with *ACC* [42], *MAE* with *ACC* [43], and *MAE* with *F1* [44], *LogLoss* with *ACC* and *F1* [20]). This allows for a more comprehensive

² Also known as Measurement Error, Observational Error, or Mean Bias Error (*MBE*).

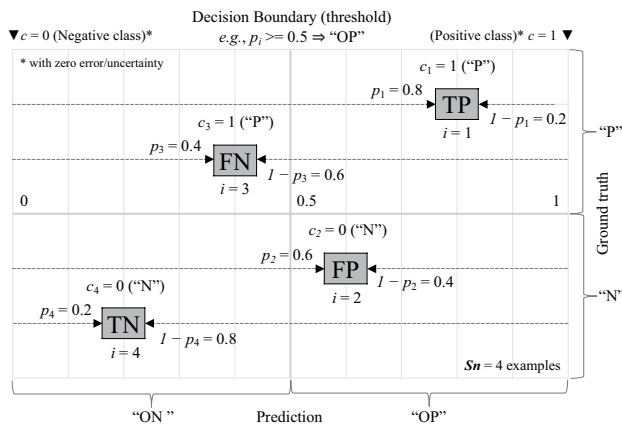


Fig. 1 Examples predicted by a probabilistic-based classifier with a decision boundary (e.g., 0.5) (“P”: Positive, “N”: Negative, “OP”: Outcome Positive, “ON”: Outcome Negative)

interpretation of the classification performance. Publishing both *ACC* and *MSE* values (e.g., $ACC = 0.99$ and $MSE = 0.40$) is more informative than publishing only the *ACC* value (e.g., $ACC = 0.99$). When two classifiers have the same accuracy, the one with the lower *MSE* is considered to be the better performer. Conversely, when two classifiers have the same *MSE*, the one with the higher *ACC* is considered to perform better. In some applications, such as recommender systems, the combination of confusion-matrix-derived and probabilistic instruments is used for specific purposes. For example, in, the quality of usage predictions is evaluated by confusion-matrix-derived instruments, whereas rating prediction calculation is evaluated by probabilistic instruments [45]. Finally, publishing all three instrument categories (e.g., *ACC*, *AUCROC*, and *RMSE* [46]) ensures completeness. It is important to note that there is no correlation between classification results obtained using confusion-matrix-derived and probabilistic instruments, as reported in the simulations in Subsection 5.2.2. Probabilistic instruments can be higher or lower than maximum/minimum *FPR*, *FNR*, *FDR*, *FOR*, and *MCR*.

4 Related works

In addition to *LogLoss*, which is a standard probabilistic loss instrument, various probabilistic error instruments have been defined and utilized for regression, time series forecasting, and partly classification applications [47]. Armstrong conducted a comprehensive evaluation of these instruments from a regression standpoint, which is summarized in the following paragraph [25]. Recently, Botchkarev emphasized the critical role of instruments in the decision-making process for selecting optimal machine learning algorithms.

He conducted a literature review of over 80 papers, categorized instruments as “primary”, “extended”, “composite”, and “hybrid”, reviewed 40 “primary” probabilistic error instruments, and compiled their generic components [47]. According to this recent review, there are few qualitative assessments with varying instruments that are either recommended or not recommended in the literature.

Armstrong discussed the following rationales for examining the instruments (“error measures”) from a regression standpoint [25]:

- “Ensure that error measures are valid”: The instruments should show agreement across applications, which can be measured using correlation. This rationale is addressed in the Discussion section below.
- “Avoid error measures with high sensitivity to the degree of difficulty”: This rationale is specific to forecasting applications and is related to the difficulty (*i.e.* volatility) of baseline models other than the model under evaluation.
- “Avoid high sensitivity to outliers”: When outliers are not of primary interest, they should be eliminated in error values (*e*) either by pre-processing or by using other proper aggregation functions (*g*) such as using median instead of mean.
- “Avoid biased error measures”: This rationale is addressed in Case 1 (over/under-predictions) in this study.
- “Ensure that scale does not affect error measures”: This rationale is addressed separately in the “scale invariance” subheading below.

Aside from those rationales, Armstrong provided some recommendations that should also be considered when selecting a probabilistic instrument for binary classification performance evaluation:

- “Do not use *RMSE* for comparisons across series”: Squaring errors were used in critical fields such as economics, where large errors could have serious consequences. Nevertheless, this is normally not required in the regression (or classification) context. Moreover, the reliability of *RMSE* as performance instrument decreases when data contain erroneous values or outliers.
- “Use multiple error measures”: Researchers should make decisions based on the results of multiple relevant instruments.

Hyndman and Koehler highlighted the following additional conditions also specific to regression or time series forecasting [48]:

- Avoid using instruments that produce zero on data where zero is meaningless (such as temperature forecasting on the Fahrenheit or Celsius scales)
- Avoid using instruments that measure errors on data having many zero values, such as intermittent demand data in economics.

The literature has also focused on investigating the robustness of specific instruments, especially those that have recently been proposed. For instance, Tofallis examined *MAPE*, which is a widely used instrument for evaluating prediction models in various fields, including software project estimation [49]. He showed that *MAPE* is a biased instrument due to its asymmetric characteristics (*i.e.* $MAPE(p, c)$ and $MAPE(c, p)$ produce different results). The use of *MAPE* favors a regression model with under-predictions. To overcome such biases in regression or time series forecasting, *SMAPE*, which is a symmetric version of *MAPE*, has been proposed [50].

However, there is a lack of research on the suitability of these instruments specifically for binary classification performance evaluation. Flach recommended *MSE* over *MAE* due to the calibration/expected loss properties and the higher value of *MAE* [51]. Kline and Berardi conducted a small experimental comparison that depends on sample size, neural network classifier models, and model complexity parameters such as the number of layers, and hidden nodes. Based on example datasets having 540 and 180 objects, they discovered that *LogLoss* outperformed *SSE* (also known as Residual Sum of Squares, *RSS*) for training [52]. Other studies have investigated the effect of noise in datasets on neural network loss functions, which are typically measured by *MSE* or *LogLoss*. An empirical study of deep neural network multi-class classifiers trained on six benchmark datasets with varying noise levels revealed that *MAE* is more robust than Categorical Cross-Entropy (*CCE*, *LogLoss* in multi-class classification) and *MSE* [53]. Kumar and Sastry proposed a parameterized version of *LogLoss* as a robust alternative and published empirical results that contradicted the previous empirical study's findings [54]. These studies mostly focus on differentiating between probabilistic loss instruments, specifically *LogLoss*, and a few probabilistic error instruments such as *MSE*, *MAE*, or *RMSE*.

The literature primarily focuses on probabilistic error instruments in the context of regression or time series forecasting, but using such instruments in classification applications involves different rationales. While the expected properties of these instruments have been extensively discussed, there is a lack of methodology for analyzing them from both qualitative and especially quantitative perspectives, as noted by Botchkarev [47]. To address this gap, BenchMetrics Prob offers a systematic approach that can be applied not only to classification but also to regression and time

series forecasting. This study provides a tool to facilitate the benchmarking process.

4.1 Previous works

It is clear that the proposed method, BenchMetrics Prob, is distinct from the author's previous works on performance evaluation. In [55], only confusion-matrix-derived binary classification instruments and their properties were reviewed, while in [24], the authors reviewed 69 instruments, including confusion-matrix-derived, graphical-based, and some probabilistic error instruments. The authors also proposed new formal concepts to identify characteristics and similarities and also developed a new exploratory table called PToPI (Periodic Table of Performance Instruments) to visualize the instruments similar to the periodic table of elements. In [26], the authors proposed a benchmarking method, BenchMetrics, to assess the robustness of 15 confusion-matrix-derived instruments, which is unrelated to probabilistic instruments. Thus, this article's proposed method is novel and distinct, benchmarking 31 probabilistic instruments.

5 BenchMetrics Prob: a proposed method for probabilistic error/loss performance-metrics benchmarking

This section describes the proposed benchmarking method comprising two stages: qualitative (Stage 1) and quantitative (Stage 2) evaluation, each of which is described in the following subsections. Stage 1 consists of five qualitative criteria and Stage 2 includes seven quantitative simulation cases. Besides describing the criteria and cases, the results of BenchMetrics Prob conducted on reviewed instruments are also provided. Section 6 provides the overall result of the tested method in terms of a final robustness ranking among all instruments.

5.1 Stage 1: proposed qualitative evaluation

This subsection introduces the five qualitative evaluation criteria proposed for assessing instruments in the classification context. Qualitative evaluation reveals the criteria to devalue the instruments that are appropriate only for regression and time-series performance evaluation. This is crucial because there are many existing instruments with different theoretical foundations. The proposed criteria can assist in selecting the most appropriate instrument for binary classification tasks. Furthermore, qualitative criteria can identify inconsistencies in instrument ranges, mathematical losses, and undefined outputs resulting from the instrument equation. These

issues severely limit the instruments' quality, robustness, and practicality.

5.1.1 Criterion 1: no interdependency in error functions

Scaled instruments, such as Mean Absolute Scaled Error (*MASE*), Median Absolute Scaled Error (*MdASE*), and Root Mean Squared Scaled Error (*RMSSE*), require two subsequent instances in their summary function (e.g., $|c_i - c_{i-1}|$ where $i = 2, \dots, Sn$). They are commonly used for evaluating time series regressions and forecasting problems with a sequential characteristic. However, these instruments are not applicable in binary classification problems where the sample ordering is arbitrary and irrelevant. Because this study specifically focuses on binary-class performance evaluation, the inapplicability of scaled instruments is a dominant criterion, and they are excluded from further evaluations.

5.1.2 Criterion 2: scale invariance

In regression, instrument values are based on the scale of the dependent variable (y), resulting in the same range of values (i.e. according to the range of y) for different models applied to the same dataset. However, instruments may exhibit variance when applied to datasets with different scales of the dependent variable (e.g., y_1 in $[0, 1]$ versus y_2 in $[0, 100]$). Most commonly used instruments, such as *MSE*, *RMSE*, *MAE*, and *MdAE* (Median Absolute Error), are not scale-invariant, as they are based on the scale of the dependent variable. Subtypes of squared and absolute instruments are proposed to avoid this variance [48]. However, in binary classification problems with two discrete values ($c \in \{0, 1\}$), all instruments are scale-invariant by definition. Because of the fixed dependent variable interval. Therefore, this criterion is not relevant for binary classification instruments.

5.1.3 Criterion 3: error/loss information preservation

Performance instruments, whether used in regression or classification, should not cause any information loss when aggregating the error values into a single value. Loss of information can occur when positive and negative errors cancel each other out. Certain error functions (e) when used with incapable aggregation functions (g) can cause this loss. For instance, Mean Error (*ME*) and Mean Percentage Error (*MPE*) are the instruments that combine the plain error ($e_i = c_i - p_i$) with positive and negative signs. Summing those signed errors can diminish the errors, leading to an instrument that yields 0 when the totals of negative and positive errors are the same. Due to this limitation, these two instruments, which also have a very low level of robustness ranking in BenchMetrics Prob, become practically useless. To avoid this issue, the errors are post-processed by taking

absolute values or squaring the values to avoid negative values before applying an aggregation function (g).

5.1.4 Criterion 4: valid outputs

Probabilistic performance instruments can produce invalid or undefined results due to division-by-zero errors. In regression, this may not be a common occurrence, but it should be considered in classification cases where both the class (c) and prediction (p) values are zero. There are two conditions in which instruments produce division-by-zero errors:

The first is in implausible cases, where certain instruments produce division-by-zero errors on datasets with specific characteristics:

- *nMSE* v1 and *nMSE* v4 on datasets with no positive-class examples ($c_i = 0$ only):
- *nMSE* v2 on datasets with only single-class examples ($c_i = 0$ or 1 only, hence, variance = 0)
- *nMSE* v3 on datasets with only single-class examples ($c_i = 0$ or 1 only, hence, the difference from the mean class value is zero)
- *nMSE* v5 on datasets with at least one negative-class example ($c_i \in \{0\} \vee c_i \in \{0, 1\}$)

The second is in plausible cases, where certain instruments produce division-by-zero errors (please refer to Table 11 for instrument name abbreviations):

- The instruments with percentage subtypes (*MPE*, *MAPE*, *MdAPE*, *RMSPE*, and *RMdSPE*) on datasets with at least one instance of the negative class ($c_i = 0$)
- The instruments with symmetric-percentage subtypes (*sMAPE*, *nsMAPE*, and *nsMdAPE*) on datasets with at least one instance of the negative class with 0 output ($c_i = 0$ and $p_i = 0$)
- The instruments with scaled subtyped (*MASE*, *MdASE*, and *RMSSE*) on datasets with a single-class only ($c_i = 0$ or 1 only, hence, difference = 0).

5.1.5 Criterion 5: parametric symmetry ($c \leftrightarrow p$)

This is informative in the classification context and indicates whether switching the c and p parameters in an instrument's equation makes it equal to itself. Note that Case 1, which describes balanced over/under-prediction in Stage 2, is sometimes called "unsymmetrical" in the literature, hence the inclusion of the term 'parametric' to distinguish the rationale. Upon examining the equations, squared error (except *nMSE* v2 – v4), absolute, and symmetric percentage subtyped instruments are parametrically symmetric instruments.

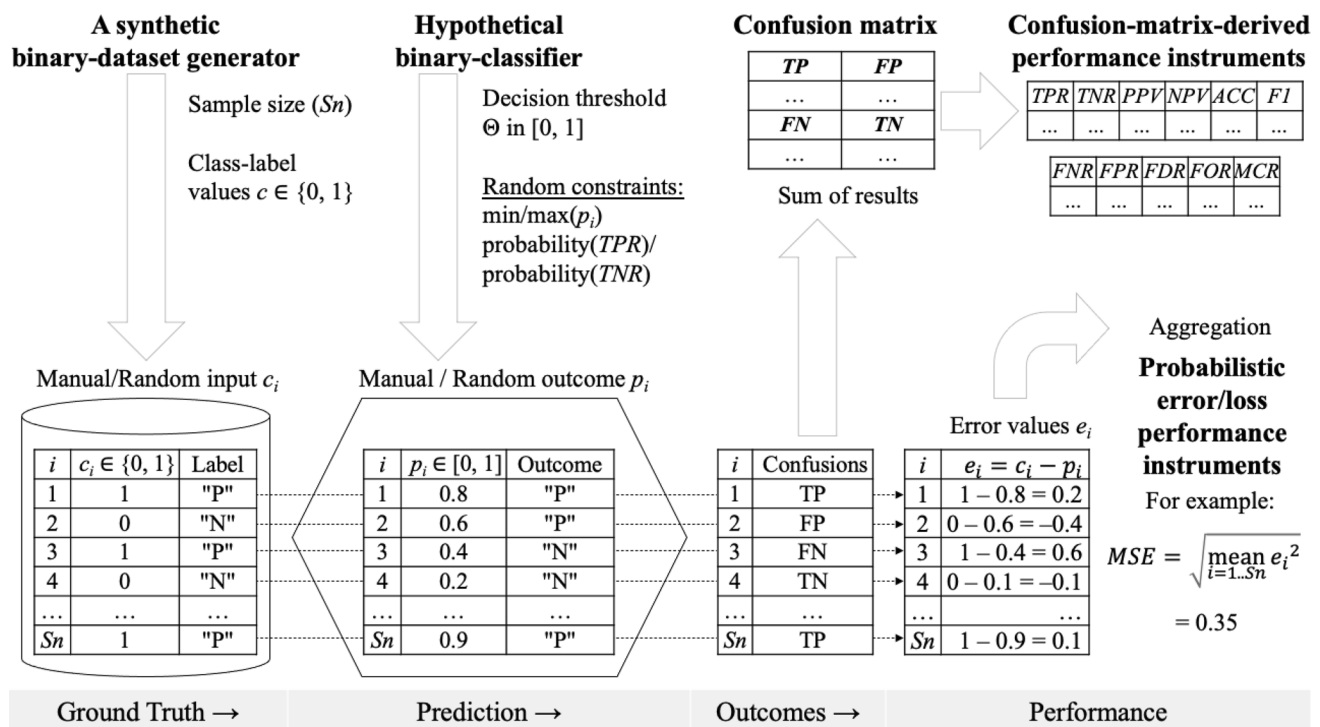


Fig. 2 The flow used in the simulation tool implemented to assess the performance of different hypothetical binary classifiers trained, validated, or tested on different synthetic datasets in terms of confusion-matrix-derived and probabilistic error/loss performance instruments

5.2 Stage 2: proposed quantitative evaluation via simulation cases

This subsection describes Stage 2 benchmarking, which involves proposed quantitative evaluation through simulation cases to rank the robustness of probabilistic error instruments in meeting the binary classification performance evaluation requirements at a fundamental level. The method was tested on 31 different probabilistic error instruments and their variants, which include varying error calculations and summary functions. Appendix B and Appendix C contain all the relevant equations and instrument abbreviations used in this study.

It is worth noting that this study covers the benchmarking of almost all probabilistic instruments, making it clear and precise to use the most robust instruments among the available alternatives. However, recently proposed probabilistic error instruments with unconventional summary functions, such as Mean Arc tangent Absolute Percentage Error (*MAAPE*) by Kim and Kim [56], LogCosh [57], and “extended” and “composite” instruments compiled in [47], are excluded. The instrument designed specifically for regression or time series forecasting, such as Mean Squared Trend Scaled Error (*MSTrdScE*), Mean Squared Volatility Scaled Error (*MSVolScE*), or Mean Squared Adjusted Percentage Error (*MSAdjPE*) [58] are also excluded.

5.2.1 Developed simulation tool and probabilistic error/loss instruments calculator

To evaluate probabilistic instruments under different conditions, a simulation tool was implemented. The tool, depicted in Fig. 2, consists of two modules:

- A synthetic binary dataset generator: This module generates random or manually assigned class label values (ground truth) $c_i \in \{0, 1\}$ for a given number of Sn samples.
- A hypothetical binary classifier: This module generates corresponding random or manually assigned outcome (prediction) variable values $p_i \in [0, 1]$ for the same samples.

The synthetic dataset generator allows for the creation of balanced or unbalanced datasets with varying levels of overlap between the two classes. The hypothetical binary classifier can produce any kind of probabilistic prediction. The combination of the synthetic dataset generator and hypothetical binary classifier enables the simulation of different classification scenarios that capture diverse levels of performances and issues such as imbalance.

In addition to the simulation tool, a probabilistic error/loss instruments calculator was developed. This calculator computes the values of 31 different probabilistic error/loss

instruments/instrument variants using the ground truth and predicted values from the simulated binary classification datasets. The equations and abbreviations for these instruments are provided in Appendices B and C, respectively.

The flow given in Fig. 2 mimics the natural way of binary classification performance evaluation by.

- Generating the ground truth examples (known labels/inputs),
- Setting the corresponding binary classifier predictions (outcome values),
- Measuring the low-level performance of the classifier by calculating
 - confusion matrix based on overall confusion and
 - error values per example, and
- Evaluating classification performances using both
 - Confusion-matrix-derived and
 - Probabilistic error/loss instruments.

In the first example (the row $i = 1$ in four tables) in Fig. 2, the simulation tool correctly identifies a given positive ground-truth example (“P”) as positive (“P”), which is a (True Positive) classification added to the *TP* element of the confusion matrix. The tool also calculates various error values for this input and outcome.

One type of error function is shown in the figure: the difference between the ground-truth class value (1) and the prediction value (0.8), yielding $e_i = c_i - p_i = 0.2$.

To aid researchers in their analysis, the simulation tool shown in Fig. 3 was prepared as a spreadsheet, providing ease and familiarity in computation. It can generate cases for any number of Sn examples, allowing the analysis of performance instruments under different conditions. The calculates:

- The confusion-matrix elements (base measures) based on a given threshold (Θ),
- Confusion-matrix-derived performance metrics (e.g., *TPR* and *ACC*),
- Error values using different error functions (e.g., $e_i = c_i - p_i$), and
- 31 probabilistic error/loss instruments/instrument variants, including *LogLoss*,

Based on predictions made by a hypothetical binary classifier for each instance of a synthetic binary-class dataset. Researchers can access the simulation tool online at <https://github.com/gurol/BenchMetricsProb>, and Appendix D provides a description of the tool and instructions on how to use it.

5.2.2 Simulation cases

Seven simulation cases, each with two subcases, are proposed to analyze the specific behaviors of probabilistic error/loss instruments. Each case, as described in the subheadings below, allows us to examine the behavior of probabilistic instruments under controlled conditions or trends. The cases are named based on a “robustness requirement” that an instrument should have in the classification scope.

In binary classification, it is important that robust instruments do not favor or show bias towards one class over the other class (e.g., positive over negative). To evaluate this balance requirement between two opposite classes, each case is analyzed by comparing its subcases for case $n = \{1, 2, 3, 6, \text{ and } 7\}$ (i.e. subcase $n.1$ and $n.2$ evaluate negative and positive classes, respectively). Table 2 summarizes the seven cases and subcases.

For example, Case 1 evaluates whether performance instruments yield balanced values for type I (false positive, i.e. ground-truth negative) and type II (false negative, i.e. ground-truth positive) errors. If the error values are not similar, the instrument is deemed not robust for Case 1 because it generates *unbalanced* error values between the two classes. Cases 2 and 3 assess the consistency (balance) in minimizing and maximizing prediction error trends. Cases 6 and 7 assess the decreasing trend balance of the instruments where the predictions are crisp (as if the classifier is binary, where c_i and $p_i \in \{0, 1\}$) and almost crisp (as if the classifier is almost binary where $c_i \in \{0, 1\}$ and $p_i \in \{0.01, 0.99\}$), respectively.

Case 4 assesses the effect of randomness in instruments measuring the performance of a random classifier making pure and stratified random predictions for the randomized examples under two subcases. In the stratified random subcase, the classifiers make random predictions exhibiting high performance (i.e. *TPR* $\gtrsim 0.8$ and *TNR* $\gtrsim 0.8$). It is expected that error values should be less for the latter (stratified) than the values for the former (pure random). Case 5 assesses the discrimination capability of the instruments measuring the increasing performance of crisp and almost crisp classifiers.

These proposed cases test and compare the instruments against defined robustness requirements that are true or natural for binary classifications. The detailed results provided in the following subsections show that some instruments do not satisfy certain cases, which are also highlighted in Sect. 6. For the experimentation, the simulation tool was configured according to the “Ground truth (c_i) and Prediction (p_i) conditions” in Table 2 to generate a simulated binary classification application comprising synthetic dataset instances and corresponding hypothetical classifiers’ predictions.

BenchMetrics Prob									
Probabilistic Error/Loss Instrument Calculator and Simulation Tool									
for hypothetical classifiers on synthetic datasets									
PFRR by Gated Cusback is licensed under CC BY-NC-ND 4.0 See License for details									
class label / prediction score values									
Use 1 and 2 (instead of 0 and 1) for c negative and positive to test MAPE and other metrics that require different uniformities, averaging, difference.									
c negative		0		min (p ₁)		0.5		1	
c positive		1		max (p ₁)		0.5		1	
Canonical and Partial 2nd Level Measures									
TC	P	N	FC	MCC	CK	F1	INFORM	Performance Metrics	
13	7	13	7	0.206	0.205	0.462	0.198	range/mean (FPR, FNR, FDR, MCR)	
OP	TP	FP	BIAS	ACC	MCR	BACC	G	Classifier Model Complexity Measure	
6	3	3	0.300	0.650	0.350	0.599	0.574	The penalty term for the number of model parameters (i.e. model complexity) in BIC is larger than in AIC (for BIC it should be S _n >> k)	
ON	FN	TN		PPV	FDR	TPR	FPR	Number of Model Parameters	
14	4	10		0.500	0.500	0.429	0.231	AIC	
Sn	PREV	NER	DET	FOR	NPV	FNR	TNR	BIC	
20	0.350	0.650	18	0.286	0.714	0.571	0.769	HOC	
Probabilistic Error / Loss Measures and Metrics									
Ground Truth									
Prediction									
Row (i)	Manual			Random			Manual		
start	Random	← select	Random	← select	Random	← select	Random	← select	Random
21	Manual	select	Random	select	Manual	select	Manual	select	Manual
Select ground-truth/prediction input methods. Enter your c ₁ and/or P ₁ values below for "Manual" inputs. (Press SHIFT + F9 to refresh (random) input data below to generate new classification examples and predictions.)									
22	Manual	select	Random	select	Manual	select	Manual	select	Manual
23	Manual	select	Random	select	Manual	select	Manual	select	Manual
24	Manual	select	Random	select	Manual	select	Manual	select	Manual
25	Manual	select	Random	select	Manual	select	Manual	select	Manual
26	Manual	select	Random	select	Manual	select	Manual	select	Manual
27	Manual	select	Random	select	Manual	select	Manual	select	Manual
28	Manual	select	Random	select	Manual	select	Manual	select	Manual
29	Manual	select	Random	select	Manual	select	Manual	select	Manual
30	Manual	select	Random	select	Manual	select	Manual	select	Manual
31	Manual	select	Random	select	Manual	select	Manual	select	Manual
32	Manual	select	Random	select	Manual	select	Manual	select	Manual
33	Manual	select	Random	select	Manual	select	Manual	select	Manual
34	Manual	select	Random	select	Manual	select	Manual	select	Manual
35	Manual	select	Random	select	Manual	select	Manual	select	Manual
36	Manual	select	Random	select	Manual	select	Manual	select	Manual
37	Manual	select	Random	select	Manual	select	Manual	select	Manual
38	Manual	select	Random	select	Manual	select	Manual	select	Manual
39	Manual	select	Random	select	Manual	select	Manual	select	Manual
40	Manual	select	Random	select	Manual	select	Manual	select	Manual
Increase/decrease hypothetical classifier performance per class for "Random" prediction (otherwise set as default 0.5): probability (TPR) >= 0.5 probability (FNR) >= 0.50									
end	Manual	select	Random	select	Manual	select	Manual	select	Manual
41	Manual	select	Random	select	Manual	select	Manual	select	Manual
42	Manual	select	Random	select	Manual	select	Manual	select	Manual
43	Manual	select	Random	select	Manual	select	Manual	select	Manual
44	Manual	select	Random	select	Manual	select	Manual	select	Manual
45	Manual	select	Random	select	Manual	select	Manual	select	Manual
46	Manual	select	Random	select	Manual	select	Manual	select	Manual
47	Manual	select	Random	select	Manual	select	Manual	select	Manual
48	Manual	select	Random	select	Manual	select	Manual	select	Manual
49	Manual	select	Random	select	Manual	select	Manual	select	Manual
50	Manual	select	Random	select	Manual	select	Manual	select	Manual
51	Manual	select	Random	select	Manual	select	Manual	select	Manual
52	Manual	select	Random	select	Manual	select	Manual	select	Manual
53	Manual	select	Random	select	Manual	select	Manual	select	Manual
54	Manual	select	Random	select	Manual	select	Manual	select	Manual
55	Manual	select	Random	select	Manual	select	Manual	select	Manual
56	Manual	select	Random	select	Manual	select	Manual	select	Manual
57	Manual	select	Random	select	Manual	select	Manual	select	Manual
58	Manual	select	Random	select	Manual	select	Manual	select	Manual
59	Manual	select	Random	select	Manual	select	Manual	select	Manual
60	Manual	select	Random	select	Manual	select	Manual	select	Manual
61	Manual	select	Random	select	Manual	select	Manual	select	Manual
62	Manual	select	Random	select	Manual	select	Manual	select	Manual
63	Manual	select	Random	select	Manual	select	Manual	select	Manual
64	Manual	select	Random	select	Manual	select	Manual	select	Manual
65	Manual	select	Random	select	Manual	select	Manual	select	Manual
66	Manual	select	Random	select	Manual	select	Manual	select	Manual
67	Manual	select	Random	select	Manual	select	Manual	select	Manual
68	Manual	select	Random	select	Manual	select	Manual	select	Manual
69	Manual	select	Random	select	Manual	select	Manual	select	Manual
70	Manual	select	Random	select	Manual	select	Manual	select	Manual
71	Manual	select	Random	select	Manual	select	Manual	select	Manual
72	Manual	select	Random	select	Manual	select	Manual	select	Manual
73	Manual	select	Random	select	Manual	select	Manual	select	Manual
74	Manual	select	Random	select	Manual	select	Manual	select	Manual
75	Manual	select	Random	select	Manual	select	Manual	select	Manual
76	Manual	select	Random	select	Manual	select	Manual	select	Manual
77	Manual	select	Random	select	Manual	select	Manual	select	Manual
78	Manual	select	Random	select	Manual	select	Manual	select	Manual
79	Manual	select	Random	select	Manual	select	Manual	select	Manual
80	Manual	select	Random	select	Manual	select	Manual	select	Manual
81	Manual	select	Random	select	Manual	select	Manual	select	Manual
82	Manual	select	Random	select	Manual	select	Manual	select	Manual
83	Manual	select	Random	select	Manual	select	Manual	select	Manual
84	Manual	select	Random	select	Manual	select	Manual	select	Manual
85	Manual	select	Random	select	Manual	select	Manual	select	Manual
86	Manual	select	Random	select	Manual	select	Manual	select	Manual
87	Manual	select	Random	select	Manual	select	Manual	select	Manual
88	Manual	select	Random	select	Manual	select	Manual	select	Manual
89	Manual	select	Random	select	Manual	select	Manual	select	Manual
90	Manual	select	Random	select	Manual	select	Manual	select	Manual
91	Manual	select	Random	select	Manual	select	Manual	select	Manual
92	Manual	select	Random	select	Manual	select	Manual	select	Manual
93	Manual	select	Random	select	Manual	select	Manual	select	Manual
94	Manual	select	Random	select	Manual	select	Manual	select	Manual
95	Manual	select	Random	select	Manual	select	Manual	select	Manual
96	Manual	select	Random	select	Manual	select	Manual	select	Manual
97	Manual	select	Random	select	Manual	select	Manual	select	Manual
98	Manual	select	Random	select	Manual	select	Manual	select	Manual
99	Manual	select	Random	select	Manual	select	Manual	select	Manual
100	Manual	select	Random	select	Manual	select	Manual	select	Manual

Entropic Instruments									
range/mean (MAE, MSE, RMSE, m-MAPE)									
p3.1x									
nsMAPE									
p3.1									
p3.2x									
MAPE(c, p)									
p1									
LogLoss									
2.31									
c ₁ - p ₁ / c ₁ + p ₁									
c ₁ log ₂ p ₁ + (1 - c ₁) log ₂ (1 - p ₁)									
0.00									
-0.46									
0.96									
54.54									
1.00									
1.00									
0.00									
-0.75									
0.00									
-4.41									

Fig. 3 Screenshot of the simulation tool (BenchMetrics Prob) for analyzing probabilistic error/loss instruments for hypothetical classifiers on synthetic datasets. For more information, see Appendix D

Table 2 Simulation cases for evaluation of the robustness of probabilistic error instruments (complete simulation data is provided in online material)

Case Subcase	Description	Ground truth (c_i) and Prediction (p_i) conditions	Notes (input generation or results)	Number of example classifications
Case 1	Balanced over/under-prediction errors or type I <i>versus</i> type II errors			
1.1	Over-prediction ($p_i > c_i$) average values	$c_i = 1$ (“N”) $\in \{1, 2\}$, p_i in $[1.5, 2]$ (“OP”) and $\Theta = 1.5^{(1)}$	All outcomes are “FP” (Type I error)	20
1.2	Under-prediction ($p_i < c_i$) average values	$c_i = 2$ (“P”) $\in \{1, 2\}$, p_i in $[1, 1.5]$ (“ON”) and $\Theta = 1.5^{(1)}$	All outcomes are “FN” (type II error)	
Case 2	Showing a consistent trend with minimizing prediction errors			
2.1	All “TN” on negative-only samples with minimizing error	$c_i = 0$ (“N”) $\{0, 1\}$ and p_i in $[0, 0.4] \rightarrow [0, 0.00001]$ (“ON”)	In 7 steps	8
2.2	All “TP” on positive-only samples with minimizing error	$c_i = 1$ (“P”) $\{0, 1\}$ and p_i in $[0.5, 1] \rightarrow [0.9999, 1]$ (“OP”)		
Case 3	Showing a consistent trend with maximizing prediction errors			
3.1	All “FP” on negative-only samples with maximizing error	$c_i = 0$ (“N”) $\{0, 1\}$ and random p_i in $[0.5, 0.6] \rightarrow [0.99, 1]$ (“OP”)	In 9 steps	10
3.2	All “FN” on positive-only samples with maximizing error	$c_i = 1$ (“P”) $\{0, 1\}$ and random p_i in $[0.4, 0.5] \rightarrow [0, 0.1]$ (“ON”)		
Case 4	Random predictions for random class instances			
4.1	Random	Random p_i in $[0, 1]$ on random $c_i \in \{0, 1\}$ samples		20
4.2	Stratified Random with high performance	Random p_i in $[0, 1]$ on random $c_i \in \{0, 1\}$ datasets	$TPR \gtrsim 0.8$ and $TNR \gtrsim 0.8$	
Case 5	Showing a consistent trend with increasing performance of crisp/almost-crisp (binary) classifiers			
5.1	Increasing performance	c_i and $p_i \in \{0, 1\}$	$TP = 10 - i$, $FP = i$, $FN = i$, $TN = 10 - i$ where $i = 10.0.0$	11
5.2		$c_i \in \{0, 1\}$ and $p_i \in \{0.01, 0.99\}$		
Case 6	Showing a consistent trend with decreasing performance per class for crisp binary-classifiers on increasing dataset size			
6.1	Increasing FN and Sn	c_i and $p_i \in \{0, 1\}$	$FN = 4, 9, 14, 19$, and $24^{(2)}$	5
6.2	Increasing FP and Sn		$FP = 4, 9, 14, 19$, and $24^{(2)}$	
Case 7	Showing a consistent trend with decreasing performance per class for almost-crisp binary-classifiers on increasing dataset size			
7.1	Increasing FN and Sn	$c_i \in \{0, 1\}$ and $p_i \in \{0.01, 0.99\}$	$FN = 4, 9, 14, 19$, and $24^{(2)}$	5
7.2	Increasing FP and Sn		$FP = 4, 9, 14, 19$, and $24^{(2)}$	

(1) c values shifted to 1 and 2 instead of 0 and 1 to avoid division-by-zero errors(2) $Sn = 5, 10, 15, 20$, and 25

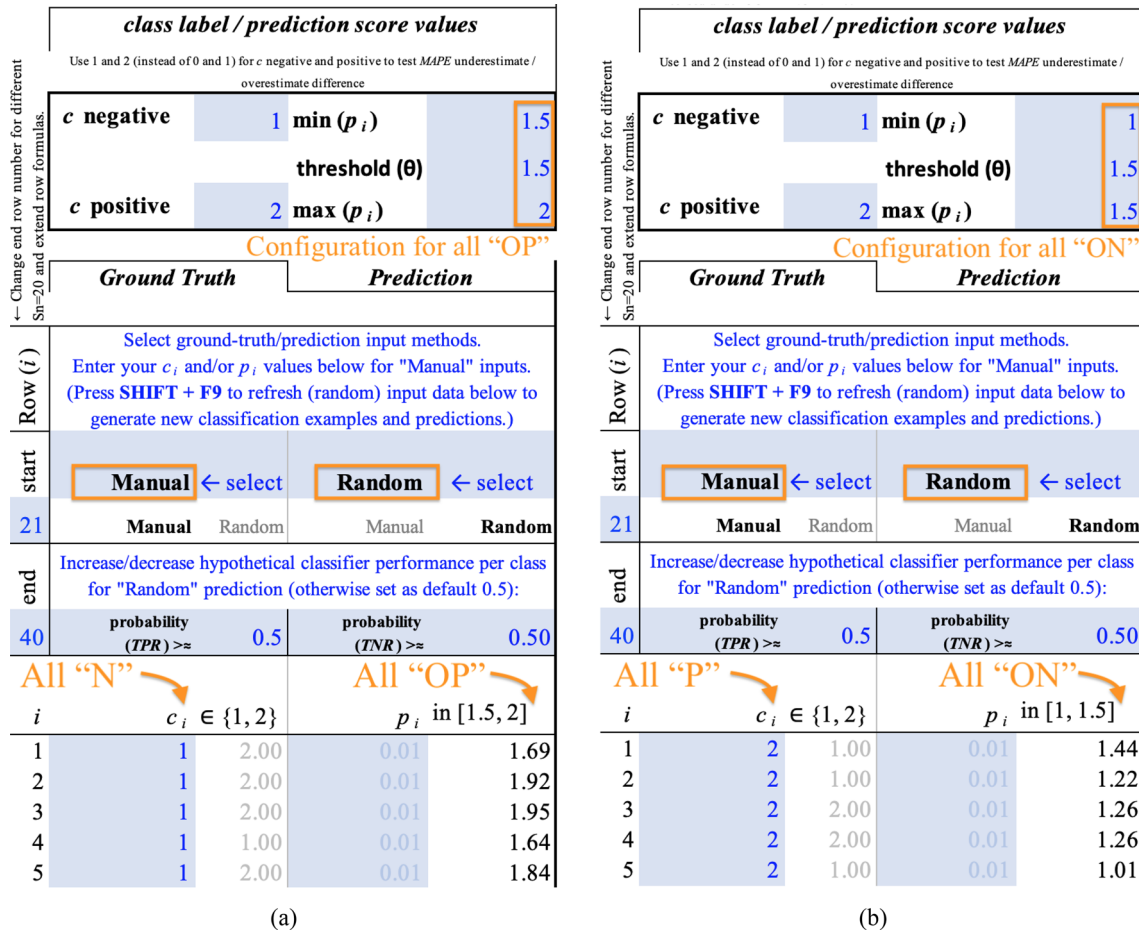
5.2.2.1 Case 1: balanced over/under-prediction errors or type I versus type II errors Case 1 is designed to assess the balance between over-prediction and under-prediction (*i.e.* are the predictions above or below the ground truth statistically unbiased?). This case is a concern in regression [49] and time series forecasting [59]. Comparing the error levels produced by a probabilistic instrument for each class reveals an imbalanced class evaluation, which is not preferred in a typical situation (*i.e.* giving more importance to one class). Case 1 is proposed to measure such a defect. It consists of two subcases to compare performance instrument results:

- **Case 1.1: Over-prediction (type I error):** False-positive prediction error occurs when a binary classifier predicts a negative class (“N”) labeled instance as positive (“OP”) with an error score where $p_i > c_i$.
- **Case 1.2 Under-prediction (type II error):** False-negative prediction error occurs when a binary classifier predicts a positive class (“P”) labeled instance as negative (“ON”) with an error score where $p_i < c_i$.

The simulation tool generates 20 simulated binary classification applications based on the configurations specified in the “Ground truth and Prediction Conditions” column in Table 2 and specifically shown in Table 3.

Table 3 Simulation cases for evaluation of the robustness of probabilistic error instruments (complete simulation data is provided online)

Input method settings	Synthetic dataset generator		Hypothetical binary classifier	
	Ground truth (known labels)		Predictions (decision threshold $\Theta = 1.5$)	
Case 1.1	Set manually (fixed)	$c_i = 1$ ("N")	Randomly generated	p_i in $[1.5, 2]$ ("OP")
Case 1.2		$c_i = 2$ ("P")		p_i in $[1, 1.5]$ ("ON")

**Fig. 4** Screenshots of the simulation tool (parts I–IV in Appendix D) for (a) Case 1.1 (all instances are negative and all predictions are outcome positive) and (b) Case 1.2 (all instances are positive and all

predictions are outcome negative). Please compare with Table 3 to understand how to use the simulation tool

Note that c and p values are shifted to 1 and 2 instead of 0 and 1, respectively, to avoid division-by-zero errors. Figure 4 shows the corresponding settings and generated instances (only the first five rows are shown here). Table 4 shows the results of Case 1, including the average instrument values (\bar{M}) in over-prediction (type I error) versus under-prediction (type II error), the delta percentage between, and the indication of unbalance determined by the 0.05 significance level.

The evaluation reveals that performance instruments, including the symmetric ones, tend to penalize over-predictions (type I) errors more than under-prediction (type II) errors, with an average of 1.00 and 0.83, respectively

(the difference is -17% , excluding N/As). Note the subcategorization of probabilistic instruments (subtypes, the first column in Table 4) used in this study is similar to common approaches in the literature [47].

5.2.2.2 Case 2: showing a consistent trend with minimizing prediction errors Case 2 is designed to test the hypothesis that probabilistic instruments will produce lower values as prediction errors are minimized. The simulation tool is configured as expressed in Table 2 to generate the following subcases:

Table 4 Case 1 (balanced over/under-prediction errors or type I versus type II errors) results per instrument

Subtype	Instruments	Average Instrument Values (\bar{M})			Balanced? ⁽²⁾
		Case 1.1: Over-prediction	Case 1.2 Under-prediction	Delta ⁽¹⁾	
(Raw)	<i>ME</i>	0.75	−0.74	−198%	No
Squared	<i>MSE</i>	0.59	0.57	−3%	
	<i>RMSE</i>	0.77	0.75	−2%	
	<i>MdSE</i>	0.57	0.55	−4%	
	<i>SSE</i>	11.73	11.37	−3%	
	<i>nMSE v1</i>	0.33	0.23	−32%	No
	<i>nMSE v2</i>	<u>division-by-0</u>	<u>division-by-0</u>	N/A	N/A
	<i>nMSE v3</i>	<u>division-by-0</u>	<u>division-by-0</u>	N/A	N/A
	<i>nMSE v4</i>	0.59	0.14	−76%	No
	<i>nMSE v5</i>	0.21	0.16	−24%	No
Absolute	<i>MAE</i>	0.75	0.74	−2%	
	<i>MdAE</i>	0.75	0.74	−2%	
	<i>MxAE</i>	0.98	0.97	−1%	
	<i>GMAE</i>	0.74	0.73	−2%	
	<i>MRAE</i>	<u>division-by-0</u>	<u>division-by-0</u>	N/A	N/A
Relative	<i>MdRAE</i>	<u>division-by-0</u>	<u>division-by-0</u>	N/A	N/A
	<i>GMRAE</i>	<u>division-by-0</u>	<u>division-by-0</u>	N/A	N/A
	<i>RAE</i>	<u>division-by-0</u>	<u>division-by-0</u>	N/A	N/A
	<i>RSE</i>	<u>division-by-0</u>	<u>division-by-0</u>	N/A	N/A
	<i>MPE</i>	0.75	−0.37	−149%	No
Percentage	<i>MAPE(p, c)</i>	0.75	0.37	−51%	No
	<i>MdAPE</i>	0.75	0.37	−51%	No
	<i>RMSPE</i>	0.77	0.38	−51%	No
	<i>RMdSPE</i>	0.75	0.37	−51%	No
	<i>nsMAPE</i>	0.27	0.23	−15%	No
Percentage (Symmetric)	<i>nsMdAPE</i>	0.27	0.23	−17%	No
	<i>LogLoss</i>	−2.41	−0.97	−60%	No

$$(1) \quad 100 \frac{\bar{M}_{\text{over-prediction}} - \bar{M}_{\text{under-prediction}}}{\bar{M}_{\text{over-prediction}}}$$

(2) Delta > 0.05

Bold values denote instances that are inappropriate or negative

Underlined table elements denote abnormal or non-robust cases

- *Case 2.1: Minimizing random prediction errors in negative-only datasets:* Prediction scores (p_i) are randomly generated from [0, 0.4] interval (high error margin) to [0, 0.00001] interval (low error margin) in seven steps: [0, 0.4], [0, 0.3], [0, 0.2], [0, 0.1], [0, 0.01], [0, 0.001], [0, 0.0001], [0, 0.00001]. In this minimizing error trend, negative random prediction scores are within the interval limits that are selected from a high error margin ($0 \leq p_i \leq 0.4 < \theta = 0.5$ decision threshold) to a low error margin ($0 \leq p_i \leq 0.00001 \ll \theta = 0.5$).
- *Case 2.2: Minimizing random prediction error in positive-only datasets:* Prediction scores (p_i) are randomly generated from [0.5, 1] interval (high error margin) to [0.9999, 1] interval (low error margin) in seven steps: [0.5, 1], [0.6, 1], [0.7, 1], [0.8, 1], [0.9, 1], [0.99, 1], [0.999, 1], [0.9999, 1]. In this minimizing trend, positive

random prediction scores are within the interval limits that are selected from a high error margin ($\theta = 0.5 \leq p_i \leq 1$) to a low error margin ($\theta = 0.5 \ll 0.9999 \leq p_i \leq 1$).

Case 2 demonstrates that there are some probabilistic instruments that do not align with the trend of minimizing prediction errors:

- The symmetric percentage errors (*sMAPE*, *nsMAPE*, and *nsMdAPE*), which are always equal to 1 according to Eq. (1), making a fixed aggregation function used in those instruments.
- *LogLoss* goes infinite in Case 2.1.

$$\text{sym}_{ei} = \frac{|e_i|}{|c_i + p_i|} = \frac{|c_i - p_i|}{|c_i + p_i|} = \frac{|0 - p_i|}{|0 + p_i|} = \frac{|-p_i|}{|p_i|} = 1, \quad (1)$$

Table 5 Case 4 (random predictions for random class instances) analysis (instruments are sorted by increasing order within subtype)

Subtype	Range	Instrument	Case 4.2 ⁽¹⁾	Position ⁽²⁾	Case 4.1 ⁽¹⁾
Squared	[0, 1]	<i>MdSE</i>	0.11	0 < 0.11 < 0.26 < 0.5 < 1.0	0.26
		<i>MSE</i>	0.19	0 < 0.19 < 0.34 < 0.5 < 1.0	0.34
		<i>RMSE</i>	0.43	0 < 0.43 < 0.5 < 0.58 < 1.0	0.58
		Intra position: ⁽³⁾ <i>MdSE</i> ≤ <i>MSE</i> < <i>RMSE</i>			
Squared	[0, ∞)	<i>nMSE</i> v4	0.40	0 < 0.4 < 0.65	0.65
		<i>nMSE</i> v1	0.79	0 < 0.79 < 1.37	1.37
		<i>nMSE</i> v2	0.73	0 < 0.73 < 1.38	1.38
		<i>nMSE</i> v3	0.77	0 < 0.77 < 1.46	1.46
		<i>SSE</i>	3.74	0 < 3.74 < 6.8	6.80
		<i>nMSE</i> v5	div-by-0	N/A	div-by-0
		Intra position: ⁽³⁾ <i>nMSE</i> (v4 ≤ v1 ≤ v2 ≤ v3) ≤ <i>SSE</i> (<i>nMSE</i> v5: div-by-0)			
Relative	[0, ∞)	<i>GMRAE</i>	0.56	0 < 0.56 < 0.82	0.82
		<i>MdRAE</i>	0.68	0 < 0.68 < 1.08	1.08
		<i>MRAE</i>	0.77	0 < 0.77 < 1.13	1.13
		<i>RAE</i>	15.38	0 < 15.38 < 22.58	22.58
		<i>RSE</i>	17.67	0 < 17.67 < 37.41	37.41
		Intra position: ⁽³⁾ <i>GMRAE</i> ≤ <i>MdRAE</i> ≤ <i>MRAE</i> ≤ <i>RAE</i> ≤ <i>RSE</i>			
Absolute	[0, 1]	<i>GMAE</i>	0.27	0 < 0.27 < 0.37 < 0.5 < 1.0	0.37
		<i>MdAE</i>	0.33	0 < 0.33 < 0.5 = 0.5 < 1.0	0.50
		<i>MAE</i>	0.36	0 < 0.36 < 0.5 = 0.5 < 1.0	0.50
		<i>MxAE</i>	0.88	0 < 0.5 < 0.88 < 0.96 < 1.0	0.96
		Intra position: ⁽³⁾ <i>GMAE</i> ≤ <i>MdAE</i> ≤ <i>MAE</i> ≤ <i>MxAE</i>			
Percentage (Symmetric)	[0, 1]	<i>nsMAPE</i>	0.63	0 < 0.5 < 0.63 < 0.68 < 1.0	0.68
		<i>nsMdAPE</i>	0.77	0 < 0.5 < 0.77 < 0.81 < 1.0	0.81
		Intra position: ⁽³⁾ <i>nsMAPE</i> ≤ <i>nsMdAPE</i> ⁽⁴⁾			
	[0, ∞)	<i>LogLoss</i>	1.31	0 < 1.31 < 1.60	1.60

Bold values denote instances that are inappropriate or negative

Underlined table elements denote abnormal or non-robust cases

(1) The instrument average values (\bar{M}) for Case 4.1 and Case 4.2 where $\bar{M}_{\text{Case4.2}} < \bar{M}_{\text{Case4.1}}$

(2) \bar{M} s are sorted according to instrument range [0, 1] and mid-range (0.5)

(3) \bar{M} s are sorted per range per instrument subtype

(4) *sMAPE* is [0, 2] and equal to $2 \times \text{nsMAPE}$

(5) *nMSE* variant 5 and (non-symmetric) percentage error instruments (*MPE*, *MAPE*(*p*, *c*), *MdAPE*, *RMSPE*, and *RmdSPE*) exhibit division-by-zero error

(6) Case 4.2 and Case 4.1 values for *ME* violate the condition specified in table footnote (1) above ($0.02 \not\leq -0.05$)

5.2.2.3 Case 3: showing a consistent trend with maximizing prediction errors

Case 3 serves as a maximizing error counterpart for Case 2:

- *Case 3.1: Maximizing random prediction errors on negative-only datasets:* Prediction scores (p_i) are randomly generated from the [0.5, 0.6] interval (low error margin) to the [0.99, 1] interval (high error margin) in nine steps: [0.5, 0.6], [0.5, 0.7], [0.5, 0.8], [0.5, 0.9], [0.5, 1], [0.6, 1], [0.7, 1], [0.8, 1], [0.9, 1], [0.99, 1]. In this maximizing error trend, negative random prediction scores are within the interval limits that are selected from a low

error margin ($\theta = 0.5 \leq p_i \leq 0.6$) to a high error margin ($\theta = 0.5 \ll 0.99 \leq p_i \leq 1$).

- *Case 3.2: Maximizing random prediction error on positive-only datasets:* Prediction scores (p_i) are randomly generated from the [0.4, 0.5] interval (low error margin) to the [0, 0.1] interval (high error margin) in nine steps: [0.4, 0.5], [0.3, 0.5], [0.2, 0.5], [0.1, 0.5], [0, 0.5], [0, 0.4], [0, 0.3], [0, 0.2], [0, 0.1], [0, 0.01]. In this maximizing error trend, positive random prediction scores are within the interval limits that are selected from a low error margin ($0.4 \leq p_i \leq \theta = 0.5$) to a high error margin ($0 \leq p_i \leq 0.01 \ll \theta = 0.5$).

Case 3 has shown that only these instruments conflict with the minimizing trend:

- The symmetric percentage errors ($nsMAPE$ and $nsMdAPE$) are always 1 in Case 3.1, as explained in Case 2.
- $LogLoss$ goes zero in Case 3.1.
- ME and MPE go to -1 in Case 3.2.

5.2.2.4 Case 4: random predictions for random class instances Case 4 is an evaluation of instruments that measure random prediction p_i in $[0, 1]$ on random class instances $c_i \in \{0, 1\}$. This case provides a pure perspective for evaluating and comparing the instruments without any side effects. There are two subcases in Case 4:

- *Case 4.1: Random predictions and datasets:* Class instances and corresponding predictions are generated randomly.
- *Case 4.2: Stratified random predictions and datasets:* Class instances are generated randomly, but the corresponding predictions are generated proportionally based on a true positive/negative rate equal to or greater than 0.8, which is the expected minimal accuracy for a classifier.

Table 5 lists average instrument values (\bar{M}) for Case 4.2 stratified random first then Case 4.1 random because the former presents less error ($TNR \geq 0.8$ when $c_i = 0$ or $TPR \geq 0.8$ when $c_i = 1$). The instruments are grouped by instrument subtype and range.

The following requirements are expected for evaluating probabilistic error/loss instruments:

- No division-by-zero error in \bar{M}
- $\bar{M}_{Case4.2} < \bar{M}_{Case4.1}$

$nMSE$ variant 5 and (non-symmetric) percentage error instruments, namely MPE , $MAPE$, $MdAPE$, $RMSPE$, and $RMdSPE$, exhibit division-by-zero error for randomly generated inputs.

Table 5 also provides more insights into the instruments:

- **Intra positions:** As explained in Table 1, individual error instruments are identified by aggregation function (g). Table 5 lists the relative positions of the instruments with the same range within the same subtype (e.g., squared error for $[0, 1]$ and $[0, \infty)$ ranges). The positions identified by geometric/arithmetic means and root mean square (also known as quadratic) are consistent with known mathematical inequalities [60]: Harmonic mean \leq Geometric mean \leq Arithmetic mean \leq Quadratic mean (e.g.,

$MSE < RMSE$ or $GMRAE \leq MRAE \leq RSE$). Because mean and median aggregation functions depend on the data distribution there is no mathematically identifiable ordering. However, median instruments yield lower values than mean instruments ($MdSE \leq MSE$, $MdRAE \leq MRAE$, $MdRAE \leq MRAE$, and $MdAE \leq MAE$), except symmetric percentage instruments ($nsMAPE \leq nsMdAPE$).

- **Range and mid-range:** Instrument range, specifically mid-range (i.e. the middle value of the range), is also a property to evaluate the instruments. The closed intervals, namely $[0, 1]$ and $[-1, 1]$ ranges, are meaningful to examining mid-ranges. The results in Table 5's "Position" column, show that mid-ranges differ for the instruments even within the same subtype and range. Therefore, it might seem obvious, but the values of different instruments cannot be compared (e.g., comparing MSE in one classification application with $RMSE$ in the other one). The instruments yielding 0.5 average values for random simulation data (Case 4.1) are $MdAE$ and MAE .

Note that there are two instruments with $[-1, 1]$ ranges: ME and MPE . ME violates the condition $\bar{ME}_{Case4.2} < \bar{ME}_{Case4.1}$ ($0.02 \not\leq -0.05$) and MPE yields a division-by-zero error.

5.2.2.5 Case 5: showing a consistent trend with increasing performance of crisp/almost-crisp (binary) classifiers Case 5 examines extreme prediction scores to emulate crisp or almost-crisp binary classifiers where.

- *Case 5.1 Crisp classifiers:* p_i is either 0 (for "Outcome Negative") or 1 (for "Outcome Positive").
- *Case 5.2 Almost-crisp classifiers:* p_i is either 0.01 (for "Outcome Negative") or 0.99 (for "Outcome Positive").

The aim of Case 5 is to assess the trend consistency with a measurable performance difference among consecutive classification application simulations with a test sample size (Sn) of 20. The confusion-matrix for each simulation starts.

- From the lowest performance $\{TP=0, FP=10, FN=10, TN=0\}$
- To the highest performance $\{TP=10, FP=0, FN=0, TN=10\}$

where the elements are changed by $\{TP=+1, FP=-1, FN=-1, TN=+1\}$.

Note that there is no need to evaluate the decreasing performance because it is the inverse of the increasing performance simulation. Consistency with increasing performance is the same as consistency with decreasing performance from the last to the first simulated binary classification application.

Table 6 Case 5 (showing a consistent trend with increasing performance of crisp/almost-crisp (binary) classifiers)

Subtype	Instrument	Case 5.1			Case 5.2			Case 5.1	Case 5.2	Average discrimination rate	
		Number of unique values	From	Middle	To	Number of unique values	From				Middle
(Raw) Squared	ME	1	0			1	0		9%	9%	9%
	MSE	11	1		0	11	0.98		100%	100%	100%
	RMSE	11	1		0	11	0.99		100%	100%	100%
	MdSE	3	five 1 s	0.5	five 0 s	3	five 0.98 s	0.49	five 0.001 s	27%	27%
	SSE	11	20		0	11	19.6		100%	100%	100%
	nMSE v1	11	4		0	11	3.92		100%	100%	100%
	nMSE v2	11	3.8		0	11	3.72		100%	100%	100%
	nMSE v3	11	4		0	11	3.92		100%	100%	100%
	nMSE v4	11	2		0	11	1.96		100%	100%	100%
	nMSE v5	0	division-by-0			0	division-by-0		0%	0%	0%
Absolute	MAE	11	1		0	11	0.99		100%	100%	100%
	MdAE	3	five 1 s	0.5	five 0 s	3	five 0.99 s	0.5	five 0 s	27%	27%
	MxAE	2	ten 1 s		0	2	ten 0.99		18%	18%	18%
	GMAE	0	division-by-0			11	0.99		0%	100%	50%
	MRAE	11	2		0	11	1.98		100%	100%	100%
Relative	MdRAE	3	five 2 s	1	five 0 s	3	five 1.98 s	1	five 0.02 s	27%	27%
	GMRAE	0	division-by-0			11	1.98		0%	100%	50%
	RAE	11	40		0	11	39.6		100%	100%	100%
	RSE	11	80		0	11	78.41		100%	100%	100%
	MPE	0	division-by-0			0	division-by-0		0%	0%	0%
Percentage (Symmetric) Loss	MAPE										
	MdAPE										
	RMSPE										
	RMdSPE										
	nsMAPE	0	division-by-0			11	0.99		0%	100%	50%
	nsMdAPE					2	ten 0.99 s		0%	18%	9%
	LogLoss	0	division-by-0			1	0		0%	9%	5%

Bold values denote instances that are inappropriate or negative
Underlined table elements denote abnormal or non-robust cases

Table 7 Case 6 and Case 7 analysis

Instrument	Case 6.1				Case 6.2				Case 7.1				Case 7.2				Case 6.1 Case 6.2	Case 7.1 Case 7.2	Average Discrimination Rate	
	Unique values	From	Mid- dle (or single)	To	Unique values	From	Mid- dle (or single)	To	Unique values	From	Mid- dle (or single)	To	Unique values	From	Mid- dle (or single)	To				Unique values / Simulation Steps (=5)
(Raw)																				
<i>ME</i>	-5	-0.6		-0.9	5	0.6		0.9	-5	-0.6		-0.9	5	0.6		0.9	-100%	100%	100%	0%
Squared																				
<i>MSE</i>	1	1	1		1	1	1		1	0.98	0.98		1	0.98	0.98		20%		20%	20%
<i>RMSE</i>	1	1	1		1	1	1		1	0.99	0.99		1	0.99	0.99		20%		20%	20%
<i>MdSE</i>	1	1	1		1	1	1		1	0.98	0.98		1	0.98	0.98		20%		20%	20%
<i>SSE</i>	5	5		25	5	5		25	5	4.9		24.5	5	4.9		24.5	100%		100%	100%
<i>nMSE v1</i>	5	6.3		26	5	6.3		26	5	6		20.8	5	6.2		25.8	100%		100%	100%
<i>nMSE v2</i>	5	5		25	5	5		25	5	4.9		24.4	5	4.9		24.5	100%		100%	100%
<i>nMSE v3</i>	5	6.3		26	5	6.3		26	5	6.1		25.5	5	6.1		25.5	100%		100%	100%
<i>nMSE v4</i>	-5	1.3		1	5	25		5	-5	1.2		1	5	5		25	-100%	100%	100%	0%
<i>nMSE v5</i>	0	division-by-0			0	division-by-0			0	division-by-0			0	division-by-0			0%		0%	0%
Absolute																				
<i>MAE</i>	1	1	1		1	1	1		1	0.99	0.99		1	0.99	0.99		20%		20%	20%
<i>MdAE</i>																				
<i>MxAE</i>																				
<i>GMAE</i>																				
Relative																				
<i>MRAE</i>	5	4.3	24	24	5	4.3	24	24	5	4.2	23.8	23.8	5	4.2	23.8	23.8	100%		100%	100%
<i>MdRAE</i>	5	5	25	25	5	5	25	25	5	5.0	24.8	24.8	5	5.0	24.8	24.8				
<i>GMRAE</i>		3.8	22	22		3.8	22	22		3.8	21.8	21.8		3.8	21.8	21.8				
<i>RAE</i>		21.3	601	601		21.3	601	601		21	595	595		21	595	595				
<i>RSE</i>		102	15001	15001		102	15001	15001		100	14703	14703		100	14703	14703				
Percentage																				
<i>MPE</i>	0	division-by-0			0	division-by-0			0	division-by-0			0	division-by-0			0%		0%	0%
<i>MAPE</i>																				
<i>MdAPE</i>																				
<i>RMdAPE</i>																				
<i>RMSPE</i>																				
<i>RMdSPE</i>																				
Percentage (Symmetric)																				
<i>nsMAPE</i>	1	1	1	1	1	1	1	1	1	0.98	0.98		1	~1	~1		20%		20%	20%
<i>nsMdAPE</i>																				
Loss																				
<i>LogLoss</i>	0	division-by-0			0	division-by-0			5	5.3	6.4	6.4	-5	1.3	1.3	0.3	0%	100%	-100%	0%

Bold values denote instances that are inappropriate or negative
Underlined table elements denote abnormal or non-robust cases

Table 8 Evaluation details of the robustness of probabilistic error instruments for binary classification performance evaluation. The ranks are the average of the two stages

				Ranks			Stage 1: Qualitative robustness evaluation criteria ⁽¹⁾					Stage 2: Quantitative robustness evaluation cases ⁽²⁾						
Subtype	Abbreviation	Interval	Instrument Name	Overall	Stage 1	Stage 2	1 ⁽³⁾	2	3	4	5	1	2	3	4	5	6 – 7	
Squared Error	<i>SSE</i>	[0, ∞)	Sum Squared Error	1	1	1												
	<i>MSE</i>	[0, 1]	Mean Squared Error	2	1	2											20%	
	<i>RMSE</i>	[0, 1]	Root Mean Square Error	2	1	2											20%	
	<i>MedSE</i>	[0, 1]	Median Squared Error	6	1	6										27%	20%	
	<i>nMSE</i>	[0, ∞)	Normalized Mean Squared Error (v1)	9	10	9				No ⁽⁴⁾		0%	50%	50%				
		[0, ∞)	Normalized Mean Squared Error (v2)	16	20	10				No ⁽⁴⁾	No	0%	0%	0%				
		[0, ∞)	Normalized Mean Squared Error (v3)	16	20	10				No ⁽⁴⁾	No	0%	0%	0%				
		[0, ∞)	Normalized Mean Squared Error (v4)	16	20	10				No ⁽⁴⁾	No	0%	50%	50%			0%	
	[0, ∞)	Normalized Mean Squared Error (v5)	22	10	24				No ⁽⁴⁾		0%	50%	50%	0%	0%	0%		
Absolute Error	<i>MAE</i>	[0, 1]	Mean Absolute Error	2	1	2											20%	
	<i>GMAE</i>		Geometric Mean Absolute Error	5	1	5										50%	20%	
	<i>MedAE</i>		Median Absolute Error	6	1	6										27%	20%	
	<i>MxAE</i>		Maximum Absolute Error	8	1	8										18%	20%	
Loss	<i>LogLoss</i>	[0, ∞)	LogLoss	10	1	21						0%	50%	50%		5%	0%	
Relative Error	<i>MRAE</i>	[0, ∞)	Mean Relative Absolute Error	10	12	10					No	0%	0%	0%				
	<i>RAE</i>		Relative Absolute Error	10	12	10					No	0%	0%	0%				
	<i>RSE</i>		Relative Squared Error	10	12	10					No	0%	0%	0%				
	<i>GMRAE</i>		Geometric Mean Relative Absolute Error	16	12	18					No	0%	0%	0%		50%		
	<i>MedRAE</i>		Median Relative Absolute Error	21	12	20					No	0%	0%	0%		27%		
Percentage (Symmetric)	<i>sMAPE</i>	[0, 2]	Symmetric Mean Absolute Percentage Er.	14	12	16				No ⁽⁵⁾		0%	50%	50%		50%	20%	
	<i>nsMAPE</i>	[0, 1]	Normalized Symmetric Mean Abs. Per. Er.	14	12	16				No ⁽⁵⁾		0%	50%	50%		50%	20%	
	<i>nsMdAPE</i>	[0, 1]	Normalized Symmetric Median Abs. Per. Er.	20	12	19				No ⁽⁵⁾		0%	50%	50%		9%	20%	
Percentage Error	<i>MAPE</i>	[0, 1]	Median Absolute Percentage Error	23	23	22				No ⁽⁵⁾	No	0%	50%	50%	0%	50%	0%	
	<i>MAPE</i>	[0, 1]	Mean Absolute Percentage Error	25	23	24				No ⁽⁵⁾	No	0%	50%	50%	0%	0%	0%	
	<i>RMSPE</i>	[0, 1]	Root Mean Square Percentage Error	25	23	24				No ⁽⁵⁾	No	0%	50%	50%	0%	0%	0%	
	<i>RmdSPE</i>	[0, 1]	Root Median Square Percentage Error	25	23	24				No ⁽⁵⁾	No	0%	50%	50%	0%	0%	0%	
	<i>MPE</i>	[-1, 1]	Mean Percentage Error	28	28	28				No	No ⁽⁵⁾	No	0%	50%	-50%	0%	0%	0%
	(Raw)	<i>ME</i>	[-1, 1]	Mean Error	24	23	23				No	No	0%	100%	0%	0%	9%	0%

Notes:

(1) Only unsatisfied criteria are shown for the readability (all the other criteria results are “Yes”), (2) Only unsatisfied cases (less than 100%) are shown for the readability (all the other case results are 100%), (3) Only the scaled subtyped instruments (*MASE*, *MedAE*, and *RMSSE*) do not satisfy Criterion 5. Hence, these instruments that are specifically used regression and time series forecasting are excluded and not shown above. See Subsection 5.1.1 for further information, (4) For unrealistic cases, (5) For possible cases

Stage 1 Criteria:

Criterion 1: No interdependency in error functions, **Criterion 2:** Scale invariance, **Criterion 3:** Error/loss information preservation, **Criterion 4:** Valid outputs, **Criterion 5:** Parametric symmetry ($c \leftrightarrow p$)

Stage 2 Cases:

Case 1: Balanced over/under-prediction errors or type I versus type II errors, **Case 2:** Showing a consistent trend with minimizing prediction errors,

Case 3: Showing a consistent trend with maximizing prediction errors, **Case 4:** Random predictions for random class instances,

Case 5: Showing a consistent trend with increasing performance of crisp/almost-crisp (binary) classifiers,

Case 6 – 7: Showing a consistent trend with decreasing performance per class for crisp/almost-crisp (binary) classifiers on increasing dataset size

Bold values denote instances that are inappropriate or negative

Underlined table elements denote abnormal or non-robust cases

The simulation instance inputs were manually provided in the simulation tool by satisfying the confusion matrix elements for each simulation. For example, for the first simulation with $\{TP = 10, FP = 0, FN = 0, TN = 10\}$ confusion-matrix:

- To achieve $TP = 10$, the first 10 instances are manually generated using $c_i = 1$ (“P”) and $p_i = 1$ (“OP”) and
- To achieve $TN = 10$, the second 10 instances are manually generated using $c_i = 0$ (“N”) and $p_i = 1$ (“ON”).

Likewise, the second simulation with $\{TP = 9, FP = 1, FN = 1, TN = 9\}$ confusion-matrix:

- To achieve $TP = 9$, the first 9 instances are manually generated using $c_i = 1$ (“P”) and $p_i = 1$ (“OP”),
- To achieve $TN = 9$, the second 9 instances are manually generated using $c_i = 0$ (“N”) and $p_i = 1$ (“ON”),
- To achieve $FP = 1$, the next instance ($i = 19$) is manually generated using $c_i = 0$ (“N”) and $p_i = 1$ (“OP”), and
- To achieve $FN = 1$, the last instance ($i = 20$) is manually generated by $c_i = 0$ (“P”) and $p_i = 1$ (“ON”).

Case 5 demonstrates that all instruments are in line with the trend, but the simulations for the following instruments show unexpected behavior.

- The median aggregation instruments (*MedSE*, *MedAE*, and *MedRAE*) were unable to distinguish the performance increase and produced three unique values rather than eleven.³
- MxAE* producing three distinct values ranging from 2 to 0 (five 2 s, one 1, and five 0 s).
- For all percentage subtype instruments (including symmetric ones), instruments with geometric mean aggregation functions (*GMAE* and *GMRAE*), and *nMSE* v5 producing a division-by-zero error in Case 5.1.

Table 6 summarizes the results of Case 4 in terms of the average discrimination rate of the two subcases. Robust

³ *MedSE*: From 1 to 0 with three unique values (five 1 s, one 0.5, and five 0 s), *MedAE*: From 1 to 0 with three unique values (five 1 s, one 0.5, and five 0 s) and *MedRAE*: From 2 to 0 with three unique values (five 2 s, one 1, and five 0 s).

instruments should distinguish those classification applications. The case demonstrated that percentage subtyped instruments cannot distinguish different applications. *LogLoss* and *ME* also had a low-level discrimination rate. Another finding is that the discrimination capability of median and partially geometric mean aggregation functions degrades within each subtype.

5.2.2.6 Case 6/7: showing a consistent trend with decreasing performance per class for crisp/almost-crisp (binary) classifiers on increasing dataset size Case 6 and Case 7 investigate the ability of probabilistic error/loss instruments to distinguish between classification applications with decreasing performance per class. Case 7 evaluates almost-crisp binary classifiers similar to Case 5.2. The results of Cases 6 and 7 are presented in Table 7, in terms of the average discrimination rate of four subcases. Robust instruments should be able to distinguish between these classification applications.

Similar to Case 5, the percentage subtyped instruments were unable to distinguish between different applications in Cases 6 and 7. Furthermore, *LogLoss* and *ME* had the lowest level of discrimination rate compared to Case 5. Absolute subtyped instruments produced only one value for all combinations. However, unlike Case 5, the median and geometric mean aggregation functions did not show a degradation in their discrimination capability within each subtype.

6 Results

In total, 31 probabilistic instrument variants were evaluated via BenchMetrics Prob, qualitatively in Stage 1 and quantitatively in Stage 2. The overall benchmarking results are summarized in Table 8, which also provides specific deficiencies found in some instruments. The instruments are ranked based on their average rates for the seven simulation cases and positioned per subtype to see which alternatives are better in one subtype (*e.g.*, *MAE* in absolute subtyped instruments). Here are the key findings:

- Criterion 3: positive and negative errors cancel each other out in *ME* and *MPE*. This completely misleads the interpretation of results. In the worst-case scenario, they can yield even zero (no error) even when the individual errors are high (*e.g.*, half of all the errors are 0.9 and the others are -0.9).
- Criterion 4: division-by-zero occurs in all percentage error instruments (*e.g.*, *MPE* and *MAPE*) when the samples have at least one negative class ($N > 0$).
- Criterion 1: Scaled error instruments (*MASE*, *MdASE*, and *RMSSE*) for regressions and time series forecasting [61] are not applicable in binary classification

Table 9 Overall robustness ranking results presented in two instrument categories (measures and metrics)

Robustness	← (increasing)	
↑ (increasing)	Metrics: In a closed interval	
	Main Instrument	Variant 1
	2 nd	2 nd
	MSE	RMSE
	2 nd	6 th
	MAE	MdSE
	14 th	8 th
	nsMAPE (sMAPE)	MdAE
	23 rd	25 th
	MdAPE	RMSPE
↑ (increasing)	Measures: In an opened interval [0, ∞)	
	Main Instrument	Variant 2
	1 st	25 th
	SSE	MPE
	9 th	28 th
	nMSE v1	RMdSPE
	10 th	21 st
	LogLoss	GMRAE
	10 th	RdRAE
	MRAE	

Bold values denote instances that are inappropriate or negative

Three scaled subtyped instruments (*MASE* main and *MdASE*, and *RMSSE* variants) were excluded from binary classification performance evaluation

because there is no innate sequence/order in dataset samples for classification.

- Case 1: Instruments yield unbalanced over-prediction/type I errors ($p_i > c_i$, false positive) versus under-prediction/type II ($p_i < c_i$, false negative) errors, penalizing over-prediction/type I errors more than under-prediction/type II errors. This finding corresponds to class imbalance bias in some confusion-matrix-derived instruments (e.g., *ACC* or *FI*) [62]. However, the rate of unbalance is lowest ($|\Delta| < 0.05\%$) for squared (except *nMSE* variants) and absolute subtyped instruments.
- Cases 2 and 3: Relative and partial percentage (including symmetric percentage), *ME*, and *LogLoss* instruments exhibit conflicting minimizing and maximizing trends.
- Case 4: Percentage and *ME* exhibited conflicting relativity in random and stratified random prediction errors.
- Case 5: Most of the instruments had conflicting behaviors in increasing performance of crisp and almost-crisp binary classifiers. Median and geometric mean aggregation functions exhibit confliction considering the base form for squared, absolute, and relative absolute subtypes (e.g., *MdSE* vs. *MSE*; *GMAE* or *MdAE* vs. *MAE*; and *GMRAE* or *MdRAE* vs. *MRAE*).
- Cases 6 and 7: The top-ranked instruments, such as *MSE*, *RMSE*, and *MAE*, significantly failed with respect to distinguishing ability and consistency with decreasing performance per class.

Out of 31 instruments/instrument variants, the analysis distinguished the robustness of nine main instruments with a total of 19 variants. These instruments are presented in two performance instrument categories (metrics and measures) in two-dimensional tabular form in Table 9. The final ranks were calculated by ranking the average ranks of Stage 1 and Stage 2:

- The most robust instrument is *SSE* which is a measure with $[0, \infty)$ interval.
- The second robust instruments are metrics with a closed interval $[0, 1]$: *MSE*, *RMSE* variant, and *MAE*.

Although *SSE* is more robust than *MSE*, *RMSE*, and *MAE*, it has one major disadvantage: low interpretability. The second robust instrument exhibits deficiency only in Cases 6 and 7. Metrics with $[0, 1]$ intervals are easier to interpret and compare than measures with open intervals. This could be why it is rarely used in the literature. Among the most reviewed instruments in the regression context,

namely *MSE*, *RMSE*, *MAE* (starting from the 1990s), and *MAPE* (from the 2000s) [47], *MAPE* exhibits a low level of robustness in the classification context. Note that the individual stage ranks are consistent per each instrument except *LogLoss*, which is the most robust (1st) one in Stage 1, but the 10th most robust one in Stage 2.

7 Discussions

One of the practical problems in analyzing probabilistic error instruments, which was also observed in this study, is that there are already numerous instruments. Furthermore, the variance of the error functions (*e*), aggregation functions (*g*), and their combinations are high. This study also compiled the frequency distribution of elements forming a probabilistic error instrument in Table 11, which helps to identify the most representative ones. The instruments' high-level derivation is managed by defining the main instrument as the most basic aggregation form (e.g., mean) as representing other forms (such as geometric mean, median, min/max) as variants, often achieved through normalization or exceptional aggregation functions. The identical instruments described in the literature with various equations were enumerated with numbers (e.g., *nMSE* v1 – v5). The proposed performance instrument subtype categorization seen in Tables 4, 5, 6, 7, 8 and main/variant distinction per subtype, which are not found in the related literature, considerably simplify the initial comprehension and consecutive analysis by allowing us to focus only on the most representative ones. As a result of such categorization, future instrument variant proposals with unusual aggregation functions should first behave more robustly in the same subtype. One consequence of this approach in current research practices is *RMSE*, which is simply the square root of *MSE*, is a variant of *MSE*. In other words, *MSE* is the main and *RMSE* is the variant in the “squared error” subtype. The distinction between measures with open intervals and metrics with closed intervals was also employed in this study. Hence, the evaluated instruments can be considered as nine main instruments (five metrics and four measures) instead of over 31 main instruments and instrument variants.

The proposed benchmark demonstrated that the probabilistic error/loss instruments behave differently in classification than in regression. Instrument robustness refers to the ability of an error instrument to provide consistent and reliable performance evaluation across different scenarios. The findings revealed specific robustness issues in instruments for binary classification performance evaluation. Some undesirable rationales observed in regression,

on the other hand, are inapplicable in the classification context. Scale invariance, for example, which is partially satisfied in regression, is valid by definition in classification. Hence, the initial expectation that there would be minor issues when the instruments originally designed for regression are used for binary classification was not met. Instrument robustness in classification differs from that in regression.

The most important result of BenchMetrics Prob, which is provided in Table 8 and merits its own discussion, is that the most robust metric is not the single but three instruments, namely *MSE*, *RMSE*, and *MAE*. This finding should be evaluated first from a regression standpoint, where the metrics measure the difference between the predicted values and true values in a continuous output:

- In general, *MSE* is a commonly used for regression problems and penalizes large differences between the predicted and actual values. *MSE* is sensitive to outliers and the scale of the target variable, meaning that large differences in the target variable will have a disproportionate effect on the result.
- *RMSE* is simply the square root of *MSE*, which rescales the results so that they are in the same units as the target variable. This makes it easier to interpret the results, and often results in more human-interpretable numbers, as the values are in the same units or range as the target variable. *MSE*, on the other hand, is more computationally efficient, as it does not require taking the square root.
- Empirical studies showed that *MAE* can be more robust to outliers [63], as it only considers the absolute value of the errors, whereas *RMSE* and *MSE* give more weight to larger errors. Hence, they can become much larger than *MAE*.

The choice of evaluation metric in regression depends on the specific goals and requirements of the problem and the desired properties of the evaluation metric. For instance, if large errors are crucial, then *RMSE* may be a better choice. On the other hand, if all errors are equally important, then *MAE* may be the preferred choice. The choice of evaluation metric also depends on the specifics of the data, such as the distribution of errors and the presence of outliers. The literature poses contradictory claims recommending among *MSE*, *RMSE*, and *MAE* in regression [4, 64].

In binary classification problems where target variables are bounded in a specific special range $[0, 1]$, three evaluation metrics that are also in the same range become suitable. The range enforces this order: $RMSE \geq MAE \geq MSE$ where $|e_i| \geq e_i^2$ and $RMSE = \sqrt{MSE} \geq MSE$. Note that the equality holds only when *MAE* is either 0 or 1. However,

this order is not valid in regression where the ranges are higher than 1, and the distribution of errors may vary. For example, *RMSE* can be less than *MSE*.

The final choice between *RMSE*, *MSE*, and *MAE* in binary classification depends on the specific goals and requirements of the problem and the desired properties of the evaluation metric. For instance, if large errors are more important than small errors, then *RMSE* can be a better choice. On the other hand, if all errors are equally important, then *MAE* can be the preferred choice. The order also suggests that among these three robust metrics, *MSE* implies the best performance since it is closer to zero error.

The study also reveals that some uncommon aggregation functions (*g*), such as median for mean or geometric mean for mean, which are considered or suggested as an improvement or novelty over existing ones from a regression perspective, present robustness issues in binary classification. The conventional forms are better. Future claims to improve the existing metrics by proposing variants replacing the aggregation functions with uncommon ones might not be validated by BenchMetrics Prob. This study also addressed the observed confusion in the literature regarding the definition of *nMSE*. There are five variants of *nMSE* compiled in the literature (see Eqs. (B.14 – B.18)). Taking BenchMetrics Prob results into account, the variants are numbered from the most robust one. Hence, *nMSE* v1 with Eq. (B.14) should be used in a binary classification context.

The theoretical implications of this study are.

- To define objective and measurable criteria and cases for evaluating the robustness of probabilistic error/loss instruments in binary classification, and
- To establish a well-founded testing methodology for them.

The practical implications are.

- To increase awareness regarding the problematic instruments and their specific weaknesses, and
- To assist researchers in terms of selecting a robust instrument for their application domain in addition to the conventional ones used in the domain when necessary.

A further implication of transitioning to robust instruments is the possibility that different machine learning approaches in the literature may come to prominence in specific domains that require using those metrics as a standard practice.

7.1 Limitations

This study proposes five qualitative criteria and 14 quantitative subcases to evaluate the robustness of probabilistic error/loss instruments in binary classification. These criteria and cases provide a comprehensive evaluation of existing instruments. The quantitative evaluation provides a more precise measurement of robustness, while the qualitative evaluation highlights the *essential* properties of instruments to be used in classification problems. However, there may be other factors in addition to the proposed criteria and especially cases. Interested researchers can extend the existing methodology by incorporating these factors.

One limitation of the existing instruments is the division-by-zero defect, which is a known issue in performance error/loss instruments in regression. Although some corrections such as adding a small value or practically the lowest non-zero value of observed data have been proposed in the literature [65], they may not be suitable for binary classification, as divide-by-zero inherently creates uncertainties in performance evaluations, especially when taking negative-class variable c as zero and generating zero p is less likely. To make some quantitative evaluations proposed (e.g., under/over-predictions), class variable c was taken as $\{1, 2\}$ instead of $\{0, 1\}$. However, this approach should be considered a general limitation for probabilistic instruments in classification. Therefore, instruments should be defined and used to eliminate division-by-zero errors.

The number of example classifications in the simulation cases may appear small. However, higher numbers were also verified by other experiments conducted separately, and this number does not affect the criteria evaluation's validity. The number of examples is sufficient to represent, tabulate, interpret, and understand the criteria's focus.

7.2 Future works

This study presents a starting point for benchmarking classification instruments from both qualitative and quantitative perspectives. As further research is conducted in this area, the following directions could be explored:

- **Number of classes:** This study focuses on binary classification. However, the proposed benchmarking method could be extended to cover n -ary class classification (initially trinary classification) to provide a more comprehensive evaluation of instruments.
- **Simulation tool:** The current simulation tool is a spreadsheet, which enhances the usability and comprehension of the method. However, additional analysis functionalities could be provided by developing

a benchmarking API. This would make it possible to increase the number of simulation example instances, allowing for a more detailed evaluation of classification instruments.

- **Extending quantitative cases:** Incorporating agreements (i.e. correlations) among instruments in different cases, including the simulation cases proposed in this study, could provide additional insights. The initial attempt to address this was described with “ensure error measures are valid” criteria [25]. Note that this will not affect the robustness rank found by BenchMetrics Prob but provide similarities in the instruments.

As more research is conducted in the field, further directions for improvement and extension of the benchmarking method can be identified and pursued.

8 Conclusion

This study evaluated probabilistic-based error/loss instruments in classification performance evaluation. These instruments, originally developed for regression, time series forecasting, and non-crisp classifiers evaluation, were adapted to measure classification performance. While the instruments are believed to be reliable in evaluating binary-class classification performance, not all instruments are equally suitable or robust in this context. *MSE* and *RMSE* are commonly used to evaluate and report the performance of binary classifiers like artificial neural networks, but there are many other instruments available, making it unclear which ones are best for binary classification performance evaluation.

The study comprehensively evaluates probabilistic error/loss instruments for binary classification performance evaluation. It proposes a two-stage systematic benchmarking method, BenchMetrics Prob, to assess the robustness of any existing and future probabilistic performance instruments:

- Stage 1 evaluates the instruments based on five qualitative criteria to identify the theoretically expected properties for binary classification.
- Stage 2 evaluates the instruments based on seven quantitative cases that reflect different conditions of a binary classifier. The cases are generated using the simulation tool provided online.

BenchMetrics Prob tested 31 probabilistic instruments and ranked their robustness. The results showed that:

- *SSE*, a measure in an opened interval $[0, \infty)$, is the most robust instrument but has low interpretability.

- *MSE*, *RMSE* as the *MSE* variant, and *MAE* metrics with a closed interval in $[0, 1]$, are the second most robust instrument.

The study also found high variance among error functions and aggregation functions for probabilistic error instruments. It simplified our understanding by categorizing a large number of instruments into subtypes and classifying them as main or variant instruments. The study also found that uncommon aggregation functions had robustness issues in the classification context. The instruments proposed as an improvement for regression performance evaluation, such as “absolute percentage”, “symmetric absolute percentage”, “relative absolute” errors (e.g., *MAPE*, *sMAPE*, and *MRAE*, respectively), and *LogLoss* showed low robustness in classification and should be avoided for performance evaluation, comparison, and reporting. These results have important implications for researchers and practitioners seeking to select appropriate evaluation metrics for binary classification.

The choice of robust metrics (*MSE*, *RMSE*, and *MAE*) for binary classification evaluation depends on specific goals. As elaborated in the Discussion section, *MAE* is the most convenient and robust metric, while *RMSE* may be preferred if the emphasis on larger errors is required. This study's findings on instrument rankings and robustness should guide researchers to choose the right probabilistic evaluation metrics and contribute to establishing common best practices in performance evaluation and reporting. Future research could extend BenchMetrics Prob to evaluate the robustness of probabilistic performance instruments for multiclass classification.

Appendix A: Preliminaries

Classification and Binary Classification: Classification is a specific problem in machine learning at which a classifier (i.e. a computer program) improves its performance through learning from experience. In a supervised approach, the experience is gained by providing labeled examples (i.e. training dataset) of one or more classes with common properties or characteristics. In a binary classification or two-class classification, a classifier separates an example given into two classes. The classes are named positive (e.g., malicious software or spam) and negative (e.g., benign software or non-spam) in general.

Classification Performance and Confusion Matrix: The performance of the trained classifier (i.e. to what degree it predicts the labels of known examples) is then improved or evaluated on different labeled examples (i.e. validation or test datasets). At this stage, the classifier is supposed to be ready to predict the class of additional unknown or unlabeled

instances. The binary classification performances in training, validation, or test datasets are presented by a confusion matrix, also known as a “ 2×2 contingency table” or “four-fold table”, (i.e. the number of correct and incorrect classification per positive and negative classes).

Confusion-matrix-derived instruments

Confusion-matrix-derived instruments are a convenient, familiar, and frequently used instrument category. Along with well-known metrics such as accuracy (*ACC*), true positive rate (*TPR*), and *F1*, other specific metrics such as Cohen's Kappa (*CK*) [27] and Mathews Correlation Coefficient (*MCC*) [28] have been used in the evaluation of crisp classifiers that assign instances to either a positive (value: one) or negative (value: zero) class absolutely (also known as “hard label”) [24, 55]. The performance measured by these instruments can be interpreted as.

External: They present observed results without an explicit connection to internal design parameters. The classifier is modeled with a single/final optimum configuration (i.e. a model threshold).

Production-ready: They provide an estimate of the classifier's performance in a production environment for the intended problem domain when compared to other classifiers.

Kinetic: They represent a classifier's performance summarizing a specific application of the samples of a dataset (e.g., the *ACC* or *MSE* values that are measured for the first run or iteration in *k*-fold cross-validation).

Graphical-based instruments

Graphical-based performance instruments are not based on a single instance of confusion-matrix elements yielded from a specific application. Instead, they list a classifier's performance panorama by varying a decision threshold (i.e. full operating range of a classifier) in terms of metric pairs that involve trade-off (e.g., *x*: *FPR* and *y*: *TPR* for *ROC*, receiver-operating-characteristic) [1]. A graph is used to visualize the variance and the area under the curve provides a single value (e.g., *AUCROC*, Area-Under-ROC-Curve) to summarize the variance [66]. These instruments represent the classifiers' internal capability designed with different possible settings and provide insight into the classifiers' potential during model development. However, since a classifier is eventually deployed with a single decision threshold in a production environment, a confusion-matrix-derived (e.g., *ACC*) and/or a probabilistic error/loss instrument (e.g., *MSE*) should be used and reported to represent the final performance.

A graphical-based instrument (e.g., *AUCPR*) can also be included to show the *classifiers' potential* when used with different decision thresholds.

Appendix B

Probabilistic error/loss instruments' equations

See Table 10.

Appendix C

Probabilistic error instruments aggregation and error function frequency distribution

Table 11 lists the frequency distribution of aggregation (*g*) and error (*e_i*) functions described in Table 1. The most used aggregation functions are mean and square(d) mean and error functions (shown in underlined) are absolute and percentage.

Table 10 Instruments' equations categorized into performance measures and metrics and instrument subtypes

Probabilistic Error/Loss Base Equations			
$e_i = c_i - p_i$ (B.1)	$\%e_i = \frac{e_i}{c_i}$ (B.2)	$\Delta c_i = c_i - \bar{c}$ (B.3)	
$rel_e_i = \frac{e_i}{\Delta c_i}$ (B.4)	$sca_e_i = \frac{e_i}{\text{mean}_{i=2..Sn} c_i - c_{i-1} }$ (B.5)	$sym_e_i = \frac{e_i}{ c_i + p_i }$ (B.6)	
$c_i \in \{0, 1\}$: ground truth class for <i>i</i> th instance (0 for negative, 1 for positive class), In <i>LogLoss</i> : $p_i \in [0, 1]$ scores produced by a model <i>C</i> for each of <i>Sn</i> samples, In others: $p_i = P(p_i = 1 x_i) = C(x_i)$: predicted class membership score where $p_i \geq \theta$ outcome is positive; otherwise, the outcome is negative (decision threshold θ in $[0, 1]$)			
Probabilistic Error/Loss Measures			
$LogLoss = -\frac{1}{Sn} \sum_i c_i \log_2 p_i + (1 - c_i) \log_2 (1 - p_i)$ (B.7)			
(Relative absolute error measures)			
$MRAE = \text{mean}_{i=1..Sn} rel_e_i $ (B.8)	$MdRAE = \text{median}_{i=1..Sn} rel_e_i $ (B.9)	$GMRAE = \text{geomean}_{i=1..Sn} rel_e_i $ (B.10)	
$RAE = \text{sum}_{i=1..Sn} rel_e_i $ (B.11)	$RSE = \text{sum}_{i=1..Sn} rel_e_i^2$ (B.12)		
(Squared error measures, continued from MSE, RMSE, and MdSE squared error metrics below)			
$SSE = \text{sum}_{i=1..Sn} e_i^2$ (B.13)	$nMSEv1 = \text{mean}_{i=1..Sn} \frac{e_i^2}{c \cdot p}$ (B.14)	$nMSEv2 = \text{mean}_{i=1..Sn} \frac{e_i^2}{\text{var}(c)}$ (B.15)	
$nMSEv3 = \text{mean}_{i=1..Sn} \frac{e_i^2}{\text{mean}_{j=1..Sn} \Delta c_j^2}$ (B.16)	$nMSEv4 = \text{mean}_{i=1..Sn} \frac{e_i^2}{\text{mean}_{j=1..Sn} c_j^2}$ (B.17)	$nMSEv5 = \text{mean}_{i=1..Sn} \frac{e_i^2}{c_i \cdot p_i}$ (B.18)	
Probabilistic Error Metrics			
$ME = \text{mean}_{i=1..Sn} e_i$ (B.19)			
(Squared error metrics)			
$MSE = \text{mean}_{i=1..Sn} e_i^2$ (B.20)	$RMSE = \sqrt{\text{mean}_{i=1..Sn} e_i^2}$ (B.21)	$MdSE = \text{median}_{i=1..Sn} e_i^2$ (B.22)	
(Absolute error metrics)			
$MAE = \text{mean}_{i=1..Sn} e_i $ (B.23)	$MxAE = \max_{i=1..Sn} e_i $ (B.25)	$GMAE = \text{geomean}_{i=1..Sn} e_i $ (B.26)	
$MdAE = \text{median}_{i=1..Sn} e_i $ (B.24)			
(Percentage error metrics)			
$MPE = \text{mean}_{i=1..Sn} \%e_i$ (B.27)	$MAPE = \text{mean}_{i=1..Sn} \%e_i$ (B.28)	$MdAPE = \text{median}_{i=1..Sn} \%e_i$ (B.29)	
$RMSPE = \sqrt{\text{mean}_{i=1..Sn} \%e_i^2}$ (B.30)	$RmdSPE = \sqrt{\text{median}_{i=1..Sn} \%e_i^2}$ (B.31)		
(Symmetric percentage error metrics)			
$sMAPE = \text{mean}_{i=1..Sn} sym_ \%e_i $ (B.32)	$nsMAPE = \text{mean}_{i=1..Sn} \left \frac{sym_ \%e_i}{2} \right $ (B.33)		
$nsMdAPE = \text{median}_{i=1..Sn} \left \frac{sym_ \%e_i}{2} \right $ (B.34)			
(Absolute scaled errors for time series forecasting, not applicable for binary classification)			
$MASE = \text{meansca}_{i=1..Sn} e_i$ (B.35)	$MdASE = \text{mediansca}_{i=1..Sn} e_i$ (B.36)	$RMSSE = \sqrt{\text{meansca}_{i=1..Sn} e_i^2}$ (B.37)	

Table 11 Aggregation function and error function (underlined> frequencies

Subtype	Instrument ⁽¹⁾ Abbreviation	Frequencies																	
		Instrument Name	2	3	4	Root	Geomet- ric (mean)	15	Mean	7	Median	1	3	9	Square(d)	5	Relative	Absolute	Percentage
(Raw) Squared	<i>ME</i>	Mean Error						X											
	<i>MSE</i>	Mean Squared Error						X						X					
	<i>RMSE</i>	Root Mean Square Error			X			X						X					
	<i>MdSE</i>	Median Squared Error							X					X					
	<i>SSE</i>	Sum Squared Error											X	X					
Absolute	<i>nMSE</i>	Normalized Mean Squared Error (v1—5)	X					X						X					
	<i>MAE</i>	Mean Absolute Error						X									X		
	<i>MdAE</i>	Median Absolute Error							X								X		
	<i>MxAE</i>	Maximum Absolute Error									X						X		
	<i>GMAE</i>	Geometric Mean Absolute Error				X	X										X		
Relative	<i>MRAE</i>	Mean Relative Absolute Error						X								X		X	
	<i>MdRAE</i>	Median Relative Absolute Error								X						X		X	
	<i>GMRAE</i>	Geometric Mean Relative Absolute Error				X	X									X		X	
	<i>RAE</i>	Relative Absolute Error											X			X		X	
	<i>RSE</i>	Relative Squared Error											X	X		X			
Percentage	<i>MPE</i>	Mean Percentage Error						X											X
	<i>MAPE</i>	Mean Absolute Percentage Error						X									X		X
	<i>MdAPE</i>	Median Absolute Percentage Error								X							X		X
	<i>RMSPE</i>	Root Mean Square Percentage Error			X			X						X					X
	<i>RMdSPE</i>	Root Median Square Percentage Error			X					X				X					X
Percentage (Symmet- ric)	<i>sMAPE</i>	Symmetric Mean Absolute Percentage Error		X				X									X		X
	<i>nsMAPE</i>	Normalized Symmetric Mean Absolute Percentage Error	X	X				X									X		X
	<i>nsMdAPE</i>	Normalized Symmetric Median Absolute Percentage Error		X						X							X		X
	<i>MASE</i>	Mean Absolute Scaled Error						X									X		
	<i>MdASE</i>	Median Absolute Scaled Error								X							X		
Scaled	<i>RMSSE</i>	Root Mean Squared Scaled Error			X			X										X	

(1)The instruments with gray background are measures in an opened interval $[0, \infty)$ whereas the others are performance metrics in a closed interval

BenchMetrics Prob

Probabilistic Error/Loss Instrument Calculator and Simulation Tool

for hypothetical classifiers on synthetic datasets

PTToPI by Gürol Canbek is licensed
under CC BY-NC-ND 4.0 [See "License for use"](#)

class label / prediction score values

Use 1 and 2 (instead of 0 and 1) for c negative and positive to test $MAPE$ underestimate / overestimate difference

c negative	0 min (p_i)	0
	threshold (θ)	0.5
c positive	1 max (p_i)	1

(a)

Select ground-truth/prediction input methods.
Enter your c_i and/or p_i values below for "Manual" inputs.
(Press **SHIFT + F9** to refresh (random) input data below to
generate new classification examples and predictions.)

	Row (i)	Ground Truth	Prediction
	start	Random ← select	Random ← select
	21	Manual Random	Manual Random
	end	Increase/decrease hypothetical classifier performance per class for "Random" prediction (otherwise set as default 0.5):	
	40	probability (TPR) \geq 0.5	probability (TNR) \geq 0.50

(b)

Fig. 5 **a** Class label/prediction score values settings (screenshot of BenchMetrics Prob—Part I) **b** Ground truth/prediction input method settings (screenshot of BenchMetrics Prob—Part II)

Appendix D

Introduction to BenchMetrics Prob calculator and simulation tool

The BenchMetrics Prob, depicted in Fig. 2, is a spreadsheet-based tool that is designed to prepare cases for evaluating the robustness of probabilistic error/loss instruments. The tool can be accessed online at <https://github.com/gurol/BenchMetricsProb>. The user interface of the tool is divided into nine parts:

- I. Class label/prediction score values settings
- II. Ground truth/prediction input method settings
- III. Synthetic dataset instances
- IV. Hypothetical classifier predictions
- V. Classification examples/outputs/confusions
- VI. Confusion matrix and other measures
- VII. Performance metrics/measure results
- VIII. Different error function results
- IX. Probabilistic error/loss performance instrument results

Part I. Class label/prediction score values settings

The first part of the tool allows the users to define the class label and prediction score values. By default, the tool is set up for conventional binary classification, as shown in Fig. 5a below, where the minimum prediction score ($\min(p_i)$) for the negative class is set to 0 and the maximum prediction score ($\max(p_i)$) for the positive class is set to 1.

The decision threshold (class-decision boundary) is set to the middle of $[0, 1]$ prediction score interval. However, the

i	$c_i \in \{0, 1\}$	
1	0	1.00
2	0	0.00
3	0	1.00
4	0	0.00

Fig. 6 Synthetic dataset instances (screenshot of BenchMetrics Prob—Part III)

i	$c_i \in \{0, 1\}$		p_i in $[0, 1]$	
1	0	1.00	0.01	0.19
2	0	0.00	0.01	0.62
3	0	1.00	0.01	0.89
4	0	0.00	0.01	0.26

Fig. 7 Hypothetical classifier predictions (screenshot of BenchMetrics Prob – Part IV)

user can change these values by editing the cells with blue text/background color. For example, if the users want to avoid division-by-zero errors, they can set $\min(p_i)$ for the negative class to 1, $\max(p_i)$ for the positive class to 2, and $\theta = 1.5$.

Part II. Ground truth/prediction input method settings

In the second part of the tool, shown in Fig. 5b, the users can choose between “manual” or “random” input methods for the ground truth and prediction values. If the users select the random input method, the tool will generate

i	$c_i \in \{0, 1\}$		$p_i \in [0, 1]$		Classification Examples/Outputs		
					Ground-Truth	Prediction	Confusion
1	0	1.00	0.01	0.19	P	ON	FN
2	0	0.00	0.01	0.62	N	OP	FP
3	0	1.00	0.01	0.89	P	OP	TP
4	0	0.00	0.01	0.26	N	ON	TN

Fig. 8 Classification examples/outputs (screenshot of BenchMetrics Prob—Part VII)

values according to the class label and prediction score values defined in Part I above. The user can also set the classifier's prediction by adjusting TPR and/or TNR to be above a given value. If the users set these values to 0.5, the

Canonical and Partial 2nd Level Measures			
TC	P	N	FC
11	12	8	9
OP	TP	FP	$BIAS$
13	8	5	0.650
ON	FN	TN	
7	4	3	
Sn	$PREV$	NER	DET
20	0.600	0.400	4

Fig. 9 Confusion matrix and other measures (screenshot of BenchMetrics Prob—Part V)

tool will generate purely random values. Setting a higher value made the classifier predict stratified random values. Additionally, the users can specify the number of samples (Sn) by defining the starting (e.g., 21) and ending (e.g., 40) row numbers in the sheet. Note that refreshing the sheet using the SHIFT and F9 shortcut keys will change the random values.

Part III. Synthetic dataset instances

Part II of the BenchMetrics Prob tool, shown in Fig. 6, generates the synthetic dataset instances for evaluation. The dataset instances (“ i ”) are numbered sequentially in the first column, starting from 1. In the cells with blue text/background colors in the second column, users can manually enter the class labels for each instance when the ground truth input method is selected as “Manual”. The values should be either 0 or 1 for default binary classification problems. When the ground truth input method is set to “Random”, automatically generated

Performance Metrics				Entropic Instruments		
MCC	CK	$F1$	$INFORM$	range/mean (FPR, FNR, FDR, FOR, MCR)		
0.043	0.043	0.640	0.042	[0.33, (0.47), 0.63]		
ACC	MCR	$BACC$	G	Classifier Model Complexity Measure		
0.550	0.450	0.521	0.500	The penalty term for the number of model parameters (i.e. model complexity) in BIC is larger than in AIC (for BIC it should be $Sn \gg k$)		
PPV	FDR	TPR	FPR	Number of Model Parameters (k)	MI	nMI
0.615	0.385	0.667	0.625	3	0.001	0.001
FOR	NPV	FNR	TNR	HOC	HC	
0.571	0.429	0.333	0.375	1.904	0.971	
				AIC	BIC	HO
				7.077	4.980	0.934

Fig. 10 Performance metrics/measure results (screenshot of BenchMetrics Prob – Part VI)

Confusion	$e_i = c_i - p_i$	$ e_i = c_i - p_i $	e_i^2	$ rel e_i = \frac{ e_i }{\Delta c_i}$	$ p_i - c_i = \frac{ c_i - p_i }{ c_i }$	$ p_i - c_i / p_i $	$ c_i - p_i / c_i + p_i $	$c_i \log_2 p_i + (1 - c_i) \log_2 (1 - p_i)$	
FN	-0.81	0.81	0.66	2.03	0.81	4.31	0.68	-2.41	0.00
FP	0.62	0.62	0.39	1.04	#SAYI/O!	1.00	1.00	0.00	-0.68
TP	-0.11	0.11	0.01	0.28	0.11	0.13	0.06	-0.17	0.00
TN	0.26	0.26	0.07	0.43	#SAYI/O!	1.00	1.00	0.00	-1.97

Fig. 11 Different error function results (screenshot of BenchMetrics Prob—Part VIII)

[illegible]

Fig. 12 Probabilistic error/loss performance instrument results (screenshot of BenchMetrics Prob—Part IX)

random class labels are displayed in the third column. These values should not be changed by the users. The total number of instances generated can be changed in Part II by adjusting the "Number of samples (Sn)" parameter. The formulas should be pasted to the rows for the new instances.

Part IV. Hypothetical classifier predictions

The same approach in Part III is applied to the predictions of the hypothetical classifier per corresponding synthetic dataset instances. Figure 7 shows the predictions along with the dataset instances where p_i values are either manually entered in the fourth column (shown in blue text/background) or automatically generated in the last column. Note that the tool takes the columns according to current “random”/“manual” settings shown in Fig. 5b.

Part V. Classification examples/outputs/confusions

Having generated synthetic dataset examples and corresponding hypothetical classifier’s prediction outputs, Part IV summarizes ground truth, predictions, and confusion status. Figure 8 shows all possible confusions (*e.g.*, the first instance is “positive” but predicted as “outcome negative” so that the instance is classified as “false negative”).

Part VI. Confusion matrix and other measures

Part VI shown in Fig. 9 provides the confusion matrix and other measures based on the matrix (total of 15 measures). The values summarize the current case’s classification performance as a crisp classifier.

Part VII. Performance metrics/measure results

In Part VII, shown in Fig. 10, the tool provides various performance metrics and measures derived from the confusion matrix, with a total of 21 instruments. The zero-one loss metrics described in Sect. 2.4 are shown in red text color. In addition, two measures of classifier model complexity are calculated based on the number of model parameters (k). These measures are:

Akaike Information Criterion (*AIC*): A measure of the quality of a model that penalizes models for the number of parameters used. Lower *AIC* values indicate a better model fit.

Bayesian Information Criterion (*BIC*): Similar to AIC, BIC also penalizes models for the number of parameters used. However, *BIC* has a stronger penalty for model complexity than *AIC*. Lower *BIC* values indicate a better model fit.

The *AIC* and *BIC* values can be used to compare different models and select the one with the best fit.

Part VIII. Error function results

Part VIII shows the results of error function based on class labels (c_i) and prediction scores (p_i), as shown in Fig. 11. Probabilistic error/loss instruments summarize those errors listed in rows into a single figure (in Part IX below) according to their aggregation functions.

Part IX. Probabilistic error/loss performance instrument results

Part IX, shown in Fig. 12, lists the results of probabilistic performance instruments for the predictions made on dataset instances. The instruments are grouped into subtypes (shown in black background text).

Note that the current instrument outputs, including confusion-matrix-based ones and the configuration setting, are listed in the sixth row in a separate worksheet ('simulation cases'). You can copy and paste the row into another row to create a simulation case for your analysis.

The benchmarking results were already provided in this sheet per seven cases.

Data availability The datasets generated during and/or analyzed during the current study are available in the GitHub repository, <https://github.com/gurol/BenchMetricsProb>.

Declarations

Conflict of interest The author declares that he has no conflict or competing interests.

References

- Japkowicz N, Shah M (2011) Evaluating learning algorithms: a classification perspective. Cambridge University Press, Cambridge
- Abdualgalil B, Abraham S (2020) Applications of machine learning algorithms and performance comparison: a review. In: International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020. pp 1–6
- Qi J, Du J, Siniscalchi SM et al (2020) On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Process Lett* 27:1485–1489. <https://doi.org/10.1109/LSP.2020.3016837>
- Karunasingha DSK (2022) Root mean square error or mean absolute error? Use their ratio as well. *Inf Sci (Ny)* 585:609–629. <https://doi.org/10.1016/j.ins.2021.11.036>
- Pham-Gia T, Hung TL (2001) The mean and median absolute deviations. *Math Comput Model* 34:921–936. [https://doi.org/10.1016/S0895-7177\(01\)00109-1](https://doi.org/10.1016/S0895-7177(01)00109-1)
- Zhang Z, Ding S, Sun Y (2020) A support vector regression model hybridized with chaotic krill herd algorithm and empirical mode decomposition for regression task. *Neurocomputing* 410:185–201. <https://doi.org/10.1016/j.neucom.2020.05.075>
- Atsalakis GS, Valavanis KP (2009) Surveying stock market forecasting techniques—part II: soft computing methods. *Expert Syst Appl* 36:5932–5941. <https://doi.org/10.1016/j.eswa.2008.07.006>
- Ru Y, Li B, Liu J, Chai J (2018) An effective daily box office prediction model based on deep neural networks. *Cogn Syst Res* 52:182–191. <https://doi.org/10.1016/j.cogsys.2018.06.018>
- Zhang X, Zhang T, Young AA, Li X (2014) Applications and comparisons of four time series models in epidemiological surveillance data. *PLoS ONE* 9:1–16. <https://doi.org/10.1371/journal.pone.0088075>
- Huang C-J, Chen Y-H, Ma Y, Kuo P-H (2020) Multiple-Input deep convolutional neural network model for COVID-19 Forecasting in China (preprint). *medRxiv*. <https://doi.org/10.1101/2020.03.23.20041608>
- Fan Y, Xu K, Wu H et al (2020) Spatiotemporal modeling for non-linear distributed thermal pProcesses based on KL decomposition, MLP and LSTM network. *IEEE Access* 8:25111–25121. <https://doi.org/10.1109/ACCESS.2020.2970836>
- Hmamouche Y, Lakhal L, Casali A (2021) A scalable framework for large time series prediction. *Knowl Inf Syst*. <https://doi.org/10.1007/s10115-021-01544-w>
- Shakhari S, Banerjee I (2019) A multi-class classification system for continuous water quality monitoring. *Heliyon* 5:e01822. <https://doi.org/10.1016/j.heliyon.2019.e01822>
- Sumaiya Thaseen I, Aswani Kumar C (2017) Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *J King Saud Univ - Comput Inf Sci* 29:462–472. <https://doi.org/10.1016/j.jksuci.2015.12.004>
- Ling QH, Song YQ, Han F et al (2019) An improved learning algorithm for random neural networks based on particle swarm optimization and input-to-output sensitivity. *Cogn Syst Res* 53:51–60. <https://doi.org/10.1016/j.cogsys.2018.01.001>
- Pwasong A, Sathasivam S (2016) A new hybrid quadratic regression and cascade forward backpropagation neural network. *Neurocomputing* 182:197–209. <https://doi.org/10.1016/j.neucom.2015.12.034>
- Chen T (2014) Combining statistical analysis and artificial neural network for classifying jobs and estimating the cycle times in wafer fabrication. *Neural Comput Appl* 26:223–236. <https://doi.org/10.1007/s00521-014-1739-1>
- Cano JR, Gutiérrez PA, Krawczyk B et al (2019) Monotonic classification: An overview on algorithms, performance measures and data sets. *Neurocomputing* 341:168–182. <https://doi.org/10.1016/j.neucom.2019.02.024>
- Jiao J, Zhao M, Lin J, Liang K (2020) A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing* 417:36–63. <https://doi.org/10.1016/j.neucom.2020.07.088>
- Cecil D, Campbell-Brown M (2020) The application of convolutional neural networks to the automation of a meteor detection pipeline. *Planet Space Sci* 186:104920. <https://doi.org/10.1016/j.pss.2020.104920>
- Banan A, Nasiri A, Taheri-Garavand A (2020) Deep learning-based appearance features extraction for automated carp species identification. *Aquac Eng* 89:102053. <https://doi.org/10.1016/j.aquaeng.2020.102053>
- Afan HA, Ibrahim Ahmed Osman A, Essam Y et al (2021) Modeling the fluctuations of groundwater level by employing ensemble deep learning techniques. *Eng Appl Comput Fluid Mech* 15:1420–1439. <https://doi.org/10.1080/19942060.2021.1974093>
- Lu Z, Lv W, Cao Y et al (2020) LSTM variants meet graph neural networks for road speed prediction. *Neurocomputing* 400:34–45. <https://doi.org/10.1016/j.neucom.2020.03.031>
- Canbek G, Taskaya Temizel T, Sagioglu S (2022) PToPI: a comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics. *SN Comput Sci* 4:1–30. <https://doi.org/10.1007/s42979-022-01409-1>
- Armstrong JS (2001) Principles of forecasting: a handbook for researchers and practitioners. Springer, Boston
- Canbek G, Taskaya Temizel T, Sagioglu S (2021) BenchMetrics: A systematic benchmarking method for binary-classification performance metrics. *Neural Comput Appl* 33:14623–14650. <https://doi.org/10.1007/s00521-021-06103-6>
- Cohen J (1960) A coefficient of agreement for nominal scales. *Educ Psychol Meas* 20:37–46. <https://doi.org/10.1177/001316446002000104>
- Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *BBA Protein Struct* 405:442–451. [https://doi.org/10.1016/0005-2795\(75\)90109-9](https://doi.org/10.1016/0005-2795(75)90109-9)
- Hodson TO, Over TM, Foks SS (2021) Mean squared error, deconstructed. *J Adv Model Earth Syst* 13:1–10. <https://doi.org/10.1029/2021MS002681>
- Ferri C, Hernández-Orallo J, Modroiu R (2009) An experimental comparison of performance measures for classification. *Pattern Recognit Lett* 30:27–38. <https://doi.org/10.1016/j.patrec.2008.08.010>
- Shen F, Zhao X, Li Z et al (2019) A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation. *Phys A Stat Mech Appl*. <https://doi.org/10.1016/j.physa.2019.121073>
- Reddy CK, Park JH (2011) Multi-resolution boosting for classification and regression problems. *Knowl Inf Syst* 29:435–456. <https://doi.org/10.1007/s10115-010-0358-0>

33. Smucny J, Davidson I, Carter CS (2021) Comparing machine and deep learning-based algorithms for prediction of clinical improvement in psychosis with functional magnetic resonance imaging. *Hum Brain Mapp* 42:1197–1205. <https://doi.org/10.1002/hbm.25286>
34. Zammito F (2019) What's considered a good Log Loss in Machine Learning? <https://medium.com/@fzammito/whats-considered-a-good-log-loss-in-machine-learning-a529d400632d>. Accessed 15 Jul 2020
35. Baldwin B (2010) Evaluating with Probabilistic Truth: Log Loss vs. O/I Loss. <http://lingpipe-blog.com/2010/11/02/evaluating-with-probabilistic-truth-log-loss-vs-o-i-loss/>. Accessed 20 May 2020
36. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inf Process Manag* 45:427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
37. Pereira RB, Plastino A, Zadrozny B, Mersmann LHC (2018) Correlation analysis of performance measures for multi-label classification. *Inf Process Manag* 54:359–369. <https://doi.org/10.1016/j.ipm.2018.01.002>
38. Kolo B (2011) Binary and multiclass classification. Weatherford Press
39. Carbonero-Ruz M, Martínez-Estudillo FJ, Fernández-Navarro F et al (2017) A two dimensional accuracy-based measure for classification performance. *Inf Sci (Ny)* 382–383:60–80. <https://doi.org/10.1016/j.ins.2016.12.005>
40. Madjarov G, Gjorgjevikj D, Dimitrovski I, Džeroski S (2016) The use of data-derived label hierarchies in multi-label classification. *J Intell Inf Syst* 47:57–90. <https://doi.org/10.1007/s10844-016-0405-8>
41. Hossin M, Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. *Int J Data Min Knowl Manag Process* 5:1–11. <https://doi.org/10.5121/ijdkp.2015.5201>
42. Tavanaei A, Maida A (2019) BP-STDP: approximating backpropagation using spike timing dependent plasticity. *Neurocomputing* 330:39–47. <https://doi.org/10.1016/j.neucom.2018.11.014>
43. Mostafa SA, Mustapha A, Mohammed MA et al (2019) Examining multiple feature evaluation and classification methods for improving the diagnosis of Parkinson's disease. *Cogn Syst Res* 54:90–99. <https://doi.org/10.1016/j.cogsys.2018.12.004>
44. Di Nardo F, Morbidoni C, Cucchiarelli A, Fioretti S (2021) Influence of EMG-signal processing and experimental set-up on prediction of gait events by neural network. *Biomed Signal Process Control* 63:102232. <https://doi.org/10.1016/j.bspc.2020.102232>
45. Alharthi H, Inkpen D, Szpakowicz S (2018) A survey of book recommender systems. *J Intell Inf Syst* 51:139–160. <https://doi.org/10.1007/s10844-017-0489-9>
46. Pakdaman Naeini M, Cooper GF (2018) Binary classifier calibration using an ensemble of piecewise linear regression models. *Knowl Inf Syst* 54:151–170. <https://doi.org/10.1007/s10115-017-1133-2>
47. Botchkarev A (2019) A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdiscip J Inform Knowledge Manag* 14:45–79. <https://doi.org/10.2894/4184>
48. Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. *Int J Forecast* 22:679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
49. Tofallis C (2015) A better measure of relative prediction accuracy for model selection and model estimation. *J Oper Res Soc* 66:1352–1362. <https://doi.org/10.1057/jors.2014.103>
50. Shin Y (2017) Time series analysis in the social sciences: the fundamentals. Time series analysis in the social sciences: the fundamentals. University of California Press, Oakland, pp 90–105
51. Flach P (2019) Performance evaluation in machine learning: The good, the bad, the ugly and the way forward. In: 33rd AAAI Conference on Artificial Intelligence. Honolulu, Hawaii
52. Kline DM, Berardi VL (2005) Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Comput Appl* 14:310–318. <https://doi.org/10.1007/s00521-005-0467-y>
53. Ghosh A, Himanshu Kumar B, Sastry PS (2017) Robust loss functions under label noise for deep neural networks. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17). Association for the Advancement of Artificial Intelligence, San Francisco, California USA, pp 1919–1925
54. Kumar H, Sastry PS (2019) Robust loss functions for learning multi-class classifiers. In: Proceedings - 2018 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2018. Institute of Electrical and Electronics Engineers Inc., pp 687–692
55. Canbek G, Sagioglu S, Temizel TT, 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). IEEE, Antalya, Turkey, pp 821–826
56. Kim S, Kim H (2016) A new metric of absolute percentage error for intermittent demand forecasts. *Int J Forecast* 32:669–679. <https://doi.org/10.1016/j.ijforecast.2015.12.003>
57. Ayzel G, Heistermann M, Sorokin A, et al (2019) All convolutional neural networks for radar-based precipitation nowcasting. In: Procedia Computer Science. Elsevier B.V., pp 186–192
58. Xu B, Ouenniche J (2012) Performance evaluation of competing forecasting models: a multidimensional framework based on MCDA. *Expert Syst Appl* 39:8312–8324. <https://doi.org/10.1016/j.eswa.2012.01.167>
59. Khan A, Yan X, Tao S, Anerousis N (2012) Workload characterization and prediction in the cloud: A multiple time series approach. In: Proceedings of the 2012 IEEE Network Operations and Management Symposium, NOMS 2012. pp 1287–1294
60. Gwanyama PW (2004) The HM-GM-AM-QM inequalities. *Coll Math J* 35:47–50
61. Prestwich S, Rossi R, Armagan Tarim S, Hnich B (2014) Mean-based error measures for intermittent demand forecasting. *Int J Prod Res* 52:6782–6791. <https://doi.org/10.1080/00207543.2014.917771>
62. 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 Recognit* 91:216–231. <https://doi.org/10.1016/j.patcog.2019.02.023>
63. Trevisan V (2022) Comparing robustness of MAE, MSE and RMSE. In: Towar. Data Sci. <https://towardsdatascience.com/comparing-robustness-of-mae-mse-and-rmse-6d69da870828>. Accessed 6 Feb 2023
64. Hodson TO (2022) Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geosci Model Dev* 15:5481–5487. <https://doi.org/10.5194/gmd-15-5481-2022>
65. Tabataba FS, Chakraborty P, Ramakrishnan N et al (2017) A framework for evaluating epidemic forecasts. *BMC Infect Dis*. <https://doi.org/10.1186/s12879-017-2365-1>
66. Gong M (2021) A novel performance measure for machine learning classification. *Int J Manag Inf Technol* 13:11–19. <https://doi.org/10.5121/ijmit.2021.13101>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.