Performance Measures for Binary Classification[1]

Daniel Berrar

Machine Learning Research Group
School of Mathematics and Statistics
The Open University, Milton Keynes, United Kingdom
Email: daniel.berrar@open.ac.uk
and
Department of Information and Communications Engineering, School of Engineering,
Tokyo Institute of Technology, Tokyo, Japan

Abstract

This article is an introduction to some of the most fundamental performance measures for the evaluation of binary classifiers. These measures are categorized into three broad families: measures based on a single classification threshold, measures based on a probabilistic interpretation of error, and ranking measures. Graphical methods, such as ROC curves, precision-recall curves, TPR-FPR plots, gain charts, and lift charts, are also discussed. Using a simple example, we illustrate how to calculate the various performance measures and show how they are related. The article also explains how to assess the statistical significance of an obtained performance value, how to calculate approximate and exact parametric confidence intervals, and how to derive percentile bootstrap confidence intervals for a performance measure.

Keywords: 0-1 loss; accuracy; AUC; AUCH; AUCPR; average precision; balanced accuracy; bootstrap percentile confidence interval; Brier score; Clopper-Pearson confidence interval; Cohen's kappa; confusion matrix; cost function; cross-entropy; error rate; F_β-measure; F-measure; false discovery rate; G-measure; gain chart; Gini index; hinge loss; information score; Kolmogorv-Smirnov statistic; lift chart; lift; logloss; loss; loss function; MAP; Matthew's correlation coefficient; mean absolute error; mean squared error; negative predictive value; negative likelihood ratio; positive predictive value; positive likelihood ratio; precision-recall plot; precision; recall; ROC; root mean square error; scoring function; sensitivity; specificity; taKS; TPR-FPR plot; trapezoidal estimators; triplet loss; true positive rate; true negative rate; Type I error; Type II error; Wald confidence interval; Youden index

Key points

- This article provides an overview of the fundamental performance measures for binary classification.
- Graphical methods for evaluating classification performance, such as ROC and

[1]This article is the revised version of [2] for the 2nd edition of the *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier.

precision-recall curves, are presented as well.

- This article explains how to assess the statistical significance of a performance measure and how to calculate confidence intervals.

1. Introduction

Classification problems can be categorized into (1) binary, (2) multiclass, and (3) multilabel tasks. In binary classification tasks, only two classes are considered, which are commonly referred to as the *positive* and *negative* class; for example, healthy vs. diseased, underexpressed vs. overexpressed, smoker vs. non-smoker, etc. By contrast, multiclass tasks include more than just two classes. Some of the measures for binary classification tasks can be easily extended to multiclass problems [1, 3]. "Single-label" means that an instance (or case) belongs to only one class, whereas "multi-label" means that an instance can simultaneously belong to more than just one class. This article focuses on performance measures for single-label, binary classification tasks, with the goal to give an easily accessible introduction to the most commonly used quantitative measures and how they are related. Using a simplified example, we illustrate how to calculate these measures and give some general recommendations regarding their use.

In this article, the term "predictive model" should be understood to refer to not only fully specified models from machine learning; instead, the term also encompasses medical diagnostic tests, for example, a blood sugar test for diabetes.

We will begin with some basic notations. Let a data set D contain n instances (or cases) $\mathbf{x}_i$, $i = 1..n$, and let each instance be described by k attributes (or features or covariates). We assume that each instance belongs to exactly one class y_i, with $y \in \{0, 1\}$, where 1 denotes the positive class and 0 denotes the negative class. Some performance measures, such as the hinge loss (Definition 35), require that the negative class is encoded as -1. A *scoring classifier* is a mapping $C : X \to \mathbb{R}$ that produces a class membership score for each instance, for example, a signed distance to a decision boundary or a conditional probability $P(y = 1|X = \mathbf{x}_i)$. This class membership score expresses the degree of class membership of that instance in the positive class. Often, we will assume that the scores are scaled from 0 to 1 and that they can be interpreted as estimated class posterior probabilities, $C(\mathbf{x}_i) = p_i = P(y = 1|X = \mathbf{x}_i)$.

As D contains only positive and negative examples, the scoring classifier can either be used as a *ranker* or as a *crisp classifier*. A ranker uses the ordinal scores to order the instances from the most to the least likely to be positive. The ranker can be turned into a crisp classifier by setting a classification threshold t on the score: if $p_i > t$, then the predicted class label is $\hat{y} = 1$; otherwise, $\hat{y} = 0$.

The underlying concept of performance metrics are *scoring rules*, which assess the quality of probabilistic predictions [4, 5, 6]. Let a model be presented with an instance $\mathbf{x}_i$, which belongs to either the positive or negative class, $y = \{0, 1\}$. Let the model's *probabilistic belief* be the same as the true probability $p \in [0, 1]$ that the class of $\mathbf{x}_i$ is $y = 1$. The model outputs the *belief report* $q \in [0, 1]$. A scoring rule $R(y, q) \in \mathbb{R}$

assigns a reward based on the reported q and the real class y. A scoring rule is called *proper* if the model maximizes the expected reward by truthfully reporting its belief p. A scoring rule is called *strictly proper* if the reported belief is the only report that maximizes the expected reward. For example, consider the quadratic scoring rule $R(y,q) = 1-(y-q)^2$. The (true) probability that the instance $\mathbf{x}_i$ belongs to class 1 is p, and the (true) probability that it belongs to class 0 is $(1-p)$. The expected reward is then $E(R) = [1-(1-q)^2]p - [1-(0-q)^2](1-p) = 1-q^2+2pq-p$. Setting the first derivative with respect to q to zero gives $\frac{\partial R}{\partial q} = 2p-2q = 0$ or $q=p$. As the second derivative, $\frac{\partial R}{\partial q} = -2$, is negative, the reward is indeed (uniquely) maximized if $q=p$. Therefore, the model is incentivized to report the true probability p. The quadratic scoring rule is a strictly proper scoring rule and underlies various performance measures, for example, the Brier score (Definition 18).

As described in [3], the different performance measures can be categorized into three broad families:

1. Performance measures based on a single classification threshold;
    (a) Elementary performance measures;
    (b) Composite performance measures;
2. Performance measures based on a probabilistic interpretation of error;
3. Ranking measures.

We will illustrate the performance measures using a contrived example (Figure 1). Here, ten cases are described by two features, i.e., their x- and y-coordinates; five cases (#3, #6, #7, #9, and #10) belong to the positive class (represented by circles), while the five remaining cases (#1, #2, #4, #5, and #8) belong to the negative class (represented by squares). The classification task is to find a decision boundary, so that cases falling on one side are classified as members of the positive class, while cases falling on the opposite side are classified as members of the negative class.

In Figure 1, the decision boundary is represented by the solid vertical line. Note that this line is certainly not the optimal decision boundary for this classification problem; nonetheless, it can be used to discern the two classes: the more a case is located to the right of the boundary, the more likely it is a member of the positive class, and vice versa. Case #6 lies exactly on the boundary, so it is reasonable to assign a class membership score of 0.5, with the probabilistic interpretation that the case is equally likely to belong to the positive or negative class. To quantify the degree of class membership of the other cases, we calculate the distance between them and the boundary. For example, the distance between case #8 and the boundary is 0.25, which leads to a score of $0.5+0.25 = 0.75$. Case #3 lies on the opposite side of the boundary but has the same distance, so we use $0.5-0.25 = 0.25$ as its membership score for the positive class. Analogously, we can derive the scores for all ten cases and rank them as shown in Figure 2.

This contrived example is deliberately simplified, and real classification algorithms usually calculate the class membership scores in a more sophisticated way. But the example illustrates the key idea: a model separates positive and negative cases and quantifies their class membership by a score, which can be used to rank the cases from most likely to be positive to the least likely to be positive. Although the scores on

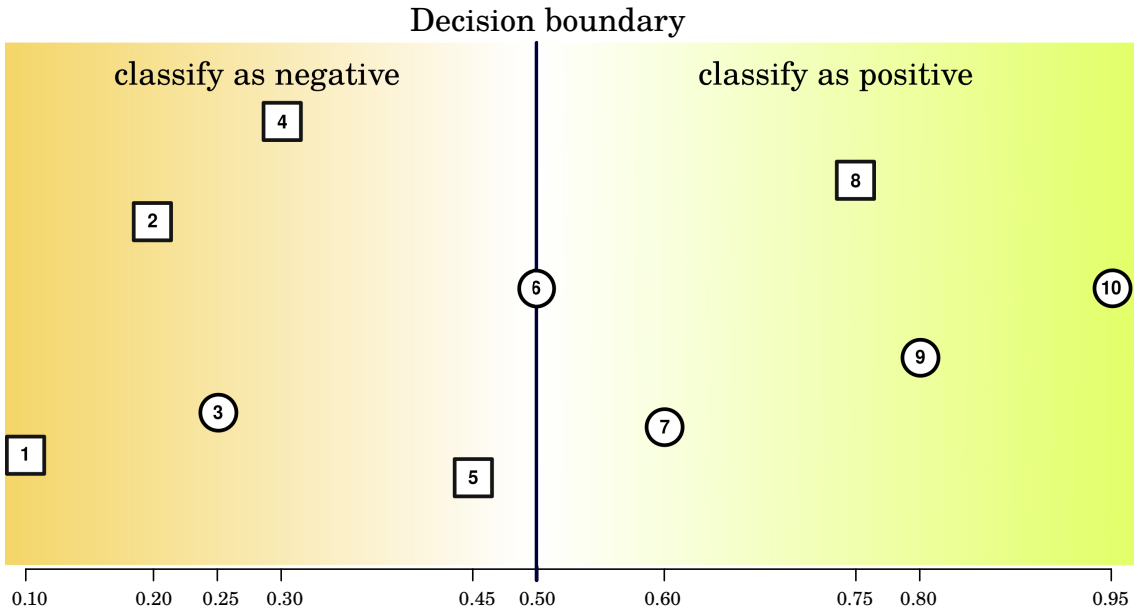


Figure 1: Simplified example. Ten instances of two classes (circles: positive class; squares: negative class) are classified based on a decision boundary (solid black line). Instances to the right of the boundary are predicted as positives, while instances to the left are predicted as negatives.

the horizontal axis in Figure 1 range from 0.10 to 0.95, they are strictly speaking not probabilities because they are not bound to the interval $[0, 1]$. However, to illustrate some of the performance measures, it can be convenient to interpret the scores in Figure 2 probabilistically.

Note that many performance measures are known under different names. The reason is that the same measures were developed in different fields of science; for example, in epidemiology and medicine, the term "positive predictive value" is widely used, whereas in machine learning and information retrieval, the term "precision" is more common. Similarly, "sensitivity" is commonly used in the context of biomedical tests, whereas "recall" is more common in information retrieval. Mathematically, there is of course no difference between these synonyms.

A closely related concept to *performance metric* is the *loss function*, which is also known as *cost function*. In short, a loss function measures by how much a model's predictions diverge from the real target values. A classification can entail a penalty, and the objective during training is to find a model whose expected loss on the independent test set is as small as possible. The objective function is the function that we seek to optimize; in the case of the loss function, we wish to minimize it. But minimizing the loss during the training phase is usually not a wise strategy, as it generally leads to an overfitted model that performs well on the training set but not so well on new, unseen data. The perhaps most intuitive loss function in supervised learning is the 0-1 loss, which simply counts the number of misclassifications. Conceptually, the 0-1 loss is equivalent to the error rate (cf. Definition 2). In applied machine learning, however, this loss is rarely used because it is a non-convex function, and as a function that is not differentiable everywhere, it is more

	Case	Class	Score	TPR	FPR	Prec	Cum.class	Lift
t_1	10	1	0.95	1/5	0	1	1	2
t_2	9	1	0.8	2/5	0	1	2	2
t_3	8	0	0.75	2/5	1/5	2/3	2	4/3
t_4	7	1	0.6	3/5	1/5	3/4	3	3/2
t_5	6	1	0.5	4/5	1/5	4/5	4	8/5
	5	0	0.45	4/5	2/5	4/6	4	4/3
	4	0	0.3	4/5	3/5	4/7	4	8/7
	3	1	0.25	5/5	3/5	5/8	5	5/4
	2	0	0.2	5/5	4/5	5/9	5	10/9
t_{11}	1	0	0.1	5/5	5/5	5/10	5	10/10

Figure 2: Ranking table for the introductory example (Figure 1), with performance measures resulting from 11 different classification thresholds $t_1..t_{11}$. Each case *above* the threshold is classified as a positive. It is assumed that classification thresholds always fall between actual scores. TPR, true positive rate = recall = sensitivity; FPR, false positive rate = 1− specificity; Prec, precision = positive predictive value; Cum.class, cumulative class count.

difficult to minimize. The logloss (or cross-entropy) is therefore far more frequently used as the objective function.

2. Performance measured based on a single classification threshold

Consider Figure 2. Here, all cases *above* threshold t_5 (dotted line) are classified as positive cases, while all cases *below* the line are classified as negative cases. Several elementary performance measures can now be derived from such a single classification threshold. The classification results are often represented in a 2×2 table or *confusion matrix*, as shown in Figure 3a, with the elementary concepts of true positives (TP, a case is really a positive case and predicted as positive); false positive (FP, a case is really a negative case but predicted as a positive); false negative (FN, a case is really a positive case but predicted as negative); and true negative (TN, a case is really a negative case and predicted as negative). The number of false positives is also known as Type I error, and the number of false negatives is known as Type II error. The corresponding counts for the introductory example are shown in Figure 3b.

2.1. Elementary performance measures

From the confusion matrix in Figure 3a, several elementary performance measures can be derived.

Definition 1. Accuracy

The *accuracy* is the proportion of correct classifications,

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

For the introductory example (Figure 2), the accuracy is $\frac{3+4}{3+1+2+4} = 0.70$ for the classification threshold t_5.

(a)

Predicted \ Real	Positive class	Negative class	
Positive class	TP	FP (Type I error)	TP + FP
Negative class	FN (Type II error)	TN	FN + TN
	TP + FN	FP + TN	TP + FP + FN + TN

(b)

Predicted \ Real	Positive class	Negative class	
Positive class	3	1	4
Negative class	2	4	6
	5	5	10

Figure 3: (a) Confusion matrix for a binary classification task. (b) Confusion matrix for Figure 2.

Definition 2. Error rate

The *error rate* is the proportion of incorrect classifications,

$$\text{error rate} = \frac{FP + FN}{TP + FP + FN + TN} \tag{2}$$

For the introductory example (Figure 2), the error rate is $= 1 - \text{accuracy} = 0.30$ for the classification threshold t_5.

Definition 3. Sensitivity or recall or true positive rate

The *sensitivity* (or *recall* or *true positive rate*, TPR) is the number of correctly predicted positive cases divided by the number of all positive cases,

$$\text{sensitivity} = \frac{TP}{TP + FN} \tag{3}$$

For the introductory example (Figure 2), the sensitivity is $\frac{3}{5} = 0.60$. The sensitivity can also be stated as a conditional probability, $P(\hat{y} = 1 | y = 1)$.

Definition 4. Specificity or true negative rate

The *specificity* (or *true negative rate*, TNR) is the number of correctly predicted negative

cases divided by the number of all negative cases,

$$\text{specificity} = \frac{TN}{FP + TN} \tag{4}$$

For the introductory example (Figure 2), the specificity is $\frac{4}{5} = 0.80$. The specificity can also be stated as a conditional probability, $P(\hat{y} = 0|y = 0)$.

Definition 5. Precision or positive predictive value

The *precision* (or *positive predictive value*) is the number of correctly predicted positive cases divided by the number of all cases that are predicted as positive,

$$\text{precision} = \frac{TP}{TP + FP} \tag{5}$$

For the introductory example (Figure 2), the precision is $\frac{3}{4} = 0.75$. The precision can also be stated as a conditional probability, $P(y = 1|\hat{y} = 1)$.

Definition 6. Negative predictive value

The *negative predictive value* is the number of correctly predicted negative cases divided by the number of all cases that are predicted as negative,

$$\text{negative predictive value} = \frac{TN}{TN + FN} \tag{6}$$

For the introductory example (Figure 2), the negative predictive value is $\frac{4}{6} = 0.67$. The negative predictive value can also be stated as a conditional probability, $P(y = 0|\hat{y} = 0)$.

Definition 7. False discovery rate

The *false discovery rate* (FDR) is the number of false positives divided by the number of cases that are predicted as positive,

$$\text{FDR} = \frac{FP}{FP + TP} \tag{7}$$

For the introductory example (Figure 2), the false discovery rate is $\frac{1}{4} = 0.25$. The false discovery rate can also be stated as a conditional probability, $P(y = 0|\hat{y} = 1)$.

2.2. Composite performance measures derived from elementary measures

From the elementary performance measures, several composite measures can be constructed, such as the Youden index, which is defined as follows [7].

Definition 8. Youden index

The *Youden index* (or Youden's J statistic) is defined as

$$J = \text{sensitivity} + \text{specificity} - 1 \tag{8}$$

Often, the maximum Youden index is reported, i.e., $J_{\max} = \max_t\{\text{sensitivity}(t)+\text{specificity}(t)-1\}$, where t denotes the classification threshold for which J is maximal [8]. For the introductory example (Figure 2), the sum of sensitivity and specificity is maximized for t_6, for which both sensitivity and specificity are 0.8 (see Figure 5a), and $J_{\max} = 0.6$.

Definition 9. Positive likelihood ratio

The *positive likelihood ratio* (LR_+) is defined as

$$LR_+ = \frac{\text{sensitivity}}{1 - \text{specificity}} = \frac{P(\hat{y} = 1|y = 1)}{P(\hat{y} = 1|y = 0)} \tag{9}$$

For the introductory example (Figure 2), the positive likelihood ratio is $\frac{3/5}{1-4/5} = 3.0$.

Definition 10. Negative likelihood ratio

The *negative likelihood ratio* (LR_-) is defined as

$$LR_- = \frac{1 - \text{sensitivity}}{\text{specificity}} = \frac{P(\hat{y} = 0|y = 1)}{P(\hat{y} = 0|y = 0)} \tag{10}$$

For the introductory example (Figure 2), the negative likelihood ratio is $\frac{1-3/5}{4/5} = 0.5$.

Definition 11. Balanced accuracy

The *balanced accuracy* (BACC) is the average of sensitivity and specificity,

$$\text{BACC} = \frac{\text{sensitivity} + \text{specificity}}{2} \tag{11}$$

For the introductory example (Figure 2), the balanced accuracy is $\frac{1}{2}(\frac{3}{5} + \frac{4}{5}) = 0.70$.

Definition 12. F-measure

The *F-measure* (also known as F_1-score or simply F-score) is the harmonic mean of precision and recall,

$$F\text{-measure} = 2 \times \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{12}$$

The multiplication by the constant 2 scales the measure to 1 when both precision and recall are 1. For the introductory example (Figure 2), the F-measure is $2 \times \frac{3/4 \times 3/5}{3/4+3/5} = 0.67$.

Definition 13. F_β**-measure**

The F_β-measure is the general form of the F-measure,

$$F_\beta = (1 + \beta^2) \times \frac{\text{precision} \times \text{recall}}{\beta^2 \text{precision} + \text{recall}} \tag{13}$$

where the positive real constant β allows for an unequal weighting of precision and recall.

Definition 14. G**-measure**

The G-measure is the geometric mean of precision and recall,

$$G\text{-measure} = \sqrt{\text{precision} \times \text{recall}} \tag{14}$$

For the introductory example (Figure 2), the G-measure is $\sqrt{3/4 \times 3/5} = 0.671$ for the classification threshold t_5.

Matthews correlation coefficient [9] is a discretization of the Pearson correlation coefficient [10].

Definition 15. Matthews correlation coefficient

Matthews correlation coefficient (MCC) is defined as

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \tag{15}$$

with MCC $\in [-1, 1]$, where -1 indicates perfect negative correlation (i.e., the model predicts all negatives as positives, and vice versa), 0 indicates no correlation (i.e., the model predicts randomly), and $+1$ indicates perfect positive correlation (i.e., the model predicts all real positives as positives and all real negatives as negatives).

For the introductory example (Figure 2), MCC $= \frac{3 \times 4 - 1 \times 2}{\sqrt{(3+2)(3+1)(4+1)(4+2)}} = 0.408$. Note that Eq. 15 may lead to the indeterminate form $\frac{0}{0}$, for example, if $TP + FN = 0$, which means that the classifier predicts all cases as instances of the negative class. As the positive class is never predicted, this is most likely an indication that something is wrong with the model. The MCC is suitable for imbalanced data sets [11]; however, this measure is not easily generalizable to more than two classes [1].

Definition 16. Lift

The *lift* measures how much better the predictions by the model, C, are compared to a *baseline* or *null model*. The lift for the positive class is defined as

$$\text{lift}(y = 1) = \frac{P(y = 1 | \hat{y}_c = 1)}{P(\hat{y}_{\text{null}} = 1)} \tag{16}$$

where $P(y = 1|\hat{y}_c = 1)$ denotes the probability that the case is really a positive, given that the model predicted that it is positive, and $P(\hat{y}_{\text{null}} = 1)$ is the probability that the null model predicts it as a positive.

Commonly, the null model is random guessing, so the probability of predicting a case as a positive is estimated as the proportion of positive cases in the training set, i.e., the prior probability of positive cases (which is also referred to as *prevalence*) [12]. Put simply, the lift tells us how much better our predictions are when we use our real model, compared to using just random guessing. To illustrate the lift, let us consider again the introductory example (Figure 3). For the classification threshold t_5, the model predicts 4 test cases as positives, and 3 of these predictions are correct, hence, $P(y = 1|\hat{y}_c = 1) = \frac{3}{4}$. Let us assume that the class ratio of positives and negatives is the same in the training set, i.e., half of the cases are positives and the other half are negatives; hence, $P(\hat{y}_{\text{null}} = 1) = \frac{1}{2}$. Therefore, the lift for the positive class is $\frac{3/4}{1/2} = 1.5$. So loosely speaking, we are doing 1.5 times better with the model than with random guessing. This can also be expressed in terms of *gain*: using the model, we expect to predict 3 out of 4 positives correctly, whereas with random guessing, we expect to predict only 2 correctly—hence, we "gain" 1 correct prediction. The lift and gain are usually calculated for all possible classification thresholds and visualized in a *lift chart* and *gain chart*, respectively. These charts are discussed in Section 4.

3. Performance measures based on a probabilistic understanding of error

Suppose that a model C_1 produces the score $P(y = 1|\mathbf{x}_-) = 0.9$ for a real negative test case $\mathbf{x}_-$, whereas another model C_2 produces the score $P(y = 1|\mathbf{x}_-) = 0.8$. Both models misclassify $\mathbf{x}_-$ as a positive case, but which model is making the more serious error? Here, it is useful to consider the deviation of the predicted class posterior probability from the real class label, which is coded as 1 for the positive and 0 for the negative class. Performance measures that take this deviation into account are based on a probabilistic understanding of error.

Definition 17. Mean absolute error

The *mean absolute error* (MAE) is defined as

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - p_i| \tag{17}$$

where $y_i \in \{0, 1\}$ and $p_i = P(y_i = 1|\mathbf{x}_i) = C(\mathbf{x}_i)$ is the predicted class membership score and n is the number of test cases.

For the introductory example (Figure 2), the mean absolute error is calculated as $\text{MAE} = \frac{1}{10}(|1-0.95|+|1-0.80|+|0-0.75|+|1-0.6|+|1-0.5|+|0-0.45|+|0-0.3|+|1-0.25|+|0-0.2|+|0-0.1|) = 0.370$.

Definition 18. Mean squared error or Brier score

The *mean squared error* (MSE) (or *Brier score* [13]) is defined as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2 \tag{18}$$

where $y_i \in \{0, 1\}$ and $p_i = P(y_i = 1|\mathbf{x}_i) = C(\mathbf{x}_i)$ is the predicted class membership score

and n is the number of test cases.

For the introductory example (Figure 2), MSE $= \frac{1}{10}[(1-0.95)^2 + (1-0.80)^2 + (0-0.75)^2 + (1-0.6)^2 + (1-0.5)^2 + (0-0.45)^2 + (0-0.3)^2 + (1-0.25)^2 + (0-0.2)^2 + (0-0.1)^2] = 0.192$. Note that the mean squared error cannot be extended to classification problems that involve more than two classes for which an ordinal relationship exists, for example, "win", "draw", and "lose" in sports match outcome prediction. Clearly, some classifications can be worse than others. For example, if the real match outcome is "win", and one classifier predicts "draw" while another classifier predicts "lose", then the error of the former classifier is less severe, as "win" is closer to "draw" than it is to "lose". To account for more than two classes with an ordinal relationship, the *ranked probability score* can be used [44].

Definition 19. Root mean square error

The *root mean square error* (RMSE) is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2} \tag{19}$$

where $y_i \in \{0, 1\}$ and $p_i = P(y_i = 1|\mathbf{x}_i) = C(\mathbf{x}_i)$ is the predicted class membership score and n is the number of test cases.

For the introductory example (Figure 2), RMSE $= \sqrt{0.192} = 0.438$.

The *logarithmic loss (logloss)* or *cross-entropy* is an information-theoretic measure. Note that the Kullback-Leibler divergence is the cross-entropy minus the entropy.

Definition 20. Logloss or cross-entropy

Let a model C produce scores $p_i \in]0, 1[$ for n instances $\mathbf{x}_i$, e.g., class posterior probabilities $C(\mathbf{x}_i) = P(y_i = 1|\mathbf{x}_i) = p_i$. The *logarithmic loss* (*logloss*) or *cross-entropy* is defined as

$$\text{logloss} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log_2(p_i) + (1 - y_i)\log_2(1 - p_i) \tag{20}$$

where $y_i \in \{0, 1\}$, and $p_i \neq 1$ and $p_i \neq 0$.

The smaller the logloss, the better the predictions. If $p_i = 0$ or $p_i = 1$, then the logloss is not defined because of $\log_2 0$; therefore, $\log_2 p_i$ is then calculated as $\log_2(\max\{p_i, \epsilon\})$, where ϵ is a small positive constant. Ferri *et al.* suggest $\epsilon = 10^{-5}$ [3]. For the introductory example (Figure 2), the logloss is calculated as follows.

$$
\begin{aligned}
\text{logloss} = -\frac{1}{10}[&1 \times \log_2(0.95) + (1-1) \cdot \log_2(1-0.95) \\
&+ 1 \times \log_2(0.8) + (1-1) \times \log_2(1-0.8) \\
&+ 0 \times \log_2(0.75) + (1-0) \times \log_2(1-0.75) \\
&+ 1 \times \log_2(0.6) + (1-1) \times \log_2(1-0.6) \\
&+ 1 \times \log_2(0.5) + (1-1) \times \log_2(1-0.5) \\
&+ 0 \times \log_2(0.45) + (1-0) \times \log_2(1-0.45) \\
&+ 0 \times \log_2(0.3) + (1-0) \times \log_2(1-0.3) \\
&+ 1 \times \log_2(0.25) + (1-1) \times \log_2(1-0.25) \\
&+ 0 \times \log_2(0.2) + (1-0) \times \log_2(1-0.2) \\
&+ 0 \times \log_2(0.1) + (1-0) \times \log_2(1-0.1)] \\
&= 0.798.
\end{aligned}
$$

One problem with the loss based on cross-entropy is that it does not take into account class imbalance. For example, consider a classification problem where the minority class (positive class) consists of 10 cases and the majority class (negative class) consists of 990 cases. The loss then mostly depends on the classification of the majority cases, which are perhaps easier to classify than the minority cases. The *balanced cross-entropy* addresses this problem by introducing a weighting factor $\alpha \in [0, 1]$ for the positive class and $1 - \alpha$ for the negative class, which can be taken as the inverse class frequencies [55]. In this example, $\alpha = \frac{990}{1000}$ and $1 - \alpha = \frac{10}{1000}$; alternatively, the weighting factor can be determined through cross-validation.

Definition 21. Balanced cross-entropy

Let a model C produce scores $p_i \in]0, 1[$ for n instances $\mathbf{x}_i$, e.g., class posterior probabilities $C(\mathbf{x}_i) = P(y_i = 1|\mathbf{x}_i) = p_i$. The *balanced cross-entropy*, bCE, is defined as

$$
\text{bCE} = -\frac{1}{n} \sum_{i=1}^{n} y_i \ \alpha \ \log_2(p_i) + (1 - y_i) \ (1 - \alpha) \ \log_2(1 - p_i) \tag{21}
$$

where $y_i \in \{0, 1\}$, and $\alpha \in [0, 1]$.

While the balanced cross-entropy addresses the problem of class imbalance, it does not take into account that some cases can be more easy to classify than others. Tsung-Yi et al. developed the *focal loss* in order to downweight the loss caused by the easy-to-classify cases [55].

Definition 22. Focal loss and average focal loss

Let a model C produce scores $p_i \in]0, 1[$ for n instances $\mathbf{x}_i$, e.g., class posterior probabilities

$C(\mathbf{x}_i) = P(y_i = 1|\mathbf{x}_i) = p_i$. Let

$$p_t = \begin{cases} p_i & \text{if } y_i = 1 \\ 1 - p_i & \text{if } y_i = 0 \end{cases}$$

The *focal loss* for an individual case is defined as

$$\text{FL}(p_t) = -(1 - p_t)^\gamma \ \log_2(p_t) \tag{22}$$

The *average focal loss*, FL_{avg}, is defined as

$$\text{FL}_{\text{avg}} = -\frac{1}{n} \sum_{i=1}^{n} y_i \ (1 - p_t)^\gamma \ \log_2(p_i) + (1 - y_i) \ (1 - p_t)^\gamma \ \log_2(1 - p_i) \tag{23}$$

where $y_i \in \{0, 1\}$. Here, $(1-p_t)^\gamma$ is the *modulating factor* with tunable *focusing parameter* $\gamma \geq 0$.

For a case that is relatively easy to classify, the model produces a relatively large p_t, i.e., close to 1. For such cases, the modulating factor is close to 0; hence, their contribution to the overall loss is very small, and therefore easy-to-classify cases are downweighted. For cases that are misclassified with relatively small p_t, the loss is less affected. The downweigthing of easy-to-classify cases depends on the focusing parameter γ. For $\gamma = 0$, the focal loss is equivalent to the cross-entropy; as γ increases, the downweighting effect increases as well. The value of $\gamma = 2$ was empirically found to work well in practice [55]. When the focal loss is used as objective function in the training phase, the model focuses more on correcting misclassified cases.

Tsung-Yi et al. recommend an α-balanced variant of the focal loss. Let α_t denote α for $y_i = 1$, and let α_t denote $(1 - \alpha)$ for $y_i = 0$ according to Eq. 21. The α-balanced modulating factor is then $\alpha_t(1 - p_t)^\gamma$, which was found to improve Eq. 23 slightly [55].

A further information-theoretic measure is the *information score* [14], which is defined as follows.

Definition 23. Information score and relative information score

Let the real class of instance $\mathbf{x}_i$ be y_i. Let the prior probability of that class be $p(y_i)$. Let the predicted score for that class be p_i. The *information score* (IS) for the case $\mathbf{x}_i$ is defined as

$$I(\mathbf{x}_i) = \begin{cases} -\log_2(p(y_i)) + \log_2(p_i) & \text{if } p_i \geq p(y_i) \\ \log_2(1 - p(y_i)) - \log_2(1 - p_i) & \text{if } p_i < p(y_i) \end{cases} \tag{24}$$

The *relative information score* (I_r) is the ratio of the average information score over all n

test cases $\mathbf{x}_i$ and the entropy of the prior class distribution,

$$I_r = \frac{\frac{1}{n}\sum_{i=1}^{n} I(\mathbf{x}_i)}{-\sum_{k=1}^{K} p(y_k)\log_2(p(y_k))} \tag{25}$$

where n denotes the number of test cases, and K denotes the number of classes.

$I(\mathbf{x}_i)$ is positive if $p_i > p(y_i)$, negative if $p_i < p(y_i)$, and zero if $p_i = p(y_i)$. The information score takes into account the class prior probabilities and thereby accounts for the fact that a high accuracy can be easily achieved in tasks with a very likely majority class [15]. For the introductory example, the information score is calculated as follows.

$$\begin{aligned}
I(\mathbf{x}_1) &= -\log_2 0.5 + \log_2 0.95 = 0.926 \\
I(\mathbf{x}_2) &= -\log_2 0.5 + \log_2 0.80 = 0.678 \\
I(\mathbf{x}_3) &= \log_2(1-0.5) - \log_2(1-0.25) = -0.585 \\
I(\mathbf{x}_4) &= -\log_2 0.5 + \log_2 0.60 = 0.263 \\
I(\mathbf{x}_5) &= -\log_2 0.5 + \log_2 0.50 = 0 \\
I(\mathbf{x}_6) &= -\log_2 0.5 + \log_2 0.55 = 0.138 \\
I(\mathbf{x}_7) &= -\log_2 0.5 + \log_2 0.30 = 0.485 \\
I(\mathbf{x}_8) &= \log_2(1-0.5) - \log_2(1-0.25) = -0.585 \\
I(\mathbf{x}_9) &= -\log_2 0.5 + \log_2 0.80 = 0.678 \\
I(\mathbf{x}_10) &= -\log_2 0.5 + \log_2 0.90 = 0.848 \\
I_r &= \frac{\frac{1}{10}(0.926 + 0.678 - 0.585 + 0.263 + 0 + 0.138 + 0.485 - 0.585 + 0.678 + 0.848)}{-0.5\log_2 0.5 - 0.5\log_2 0.5} = 0.2846.
\end{aligned}$$

Cohen's kappa measures the agreement between two raters on a categorical variable [16]. Thus, if we assume that the real class labels are due to some unknown generating process, we can use this measure to compare how well the predictions of a model agree with that process (i.e., reality). This agreement is of course already quantified in the accuracy (Eq. 1); however, accuracy does not take into account that an agreement could be coincidental. Cohen's kappa adjusts the accuracy by calculating the probability of random agreements.

Definition 24. Cohen's kappa

Cohen's kappa (κ) is defined as

$$\kappa = \frac{P_0 - P_e^c}{1 - P_e^c} \tag{26}$$

where P_0 is the probability that the model's predicted class labels agree with the real class labels, and P_e^c is the probability of random agreement.

P_0 is estimated as the proportion of correctly predicted class labels, i.e., the accuracy. The probability of random agreement is estimated as follows: the probability that the classifier predicts a case as a positive is $P_+^c = \frac{TP+FP}{n}$. The probability that a case is really a positive is $P_+^r = \frac{TP+FN}{n}$. Hence, the probability that both agree just by chance on the positive cases is the product $P_+^c \times P_+^r$. It is of course assumed that the process which generates the real class labels is independent of the model. Analogously, we calculate the probability of random agreement on the negative cases, and we obtain the probability of total random agreement as

$$P_e^c = \frac{TP+FP}{n} \times \frac{TP+FN}{n} + \frac{FN+TN}{n} \times \frac{FP+TN}{n} \tag{27}$$

where n is the total number of test cases. For the introductory example (Figure 2), $\kappa = \frac{3+1}{10} \times \frac{3+2}{10} + \frac{2+4}{10} \times \frac{1+4}{10} = 0.50$.

4. Ranking measures

Ranking performance refers to how well the model orders or ranks the positive cases relatively to the negative cases based on the class membership score. There exist various graphical methods for representing ranking performance. Figure 4a shows the gain chart of the introductory example (Figure 2), which plots the cumulative class on the y-axis and the ranking index of the corresponding instance on the x-axis. For example, if we classify the 4 top-ranked instances as positives, then we classify 3 instances correctly, in contrast to only 2 instances that we expect to predict correctly by random guessing; hence, the gain is 1 correct prediction.

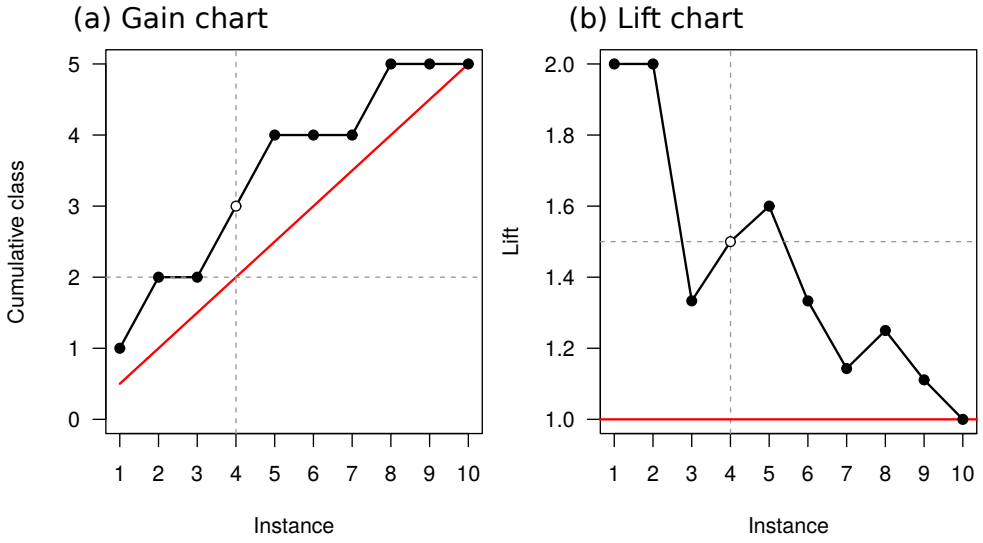


Figure 4: (a) Gain chart and (b) lift chart for the introductory example (Figure 2). The red line indicates the expected gain or lift for a random guesser.

In Figure 4a, the red line indicates the expected number of correct positive predictions (for a given number of predicted instances) by using random guessing. The *average gain* is a summary statistic of the gain chart.

Definition 25. Average gain

Let the test cases be ranked based on decreasing values of their class membership scores $p_i = P(y_i = 1|\mathbf{x}_i) = C(\mathbf{x}_i)$. Let Δg_j denote the difference between the gain of the trained model C, $g_c(j)$, and the gain of the null model, $g_{\text{null}}(j)$, when the j top-ranked instances are classified as positives,

$$\Delta g_j = g_c(j) - g_{\text{null}}(j) \tag{28}$$

The *average gain* is the average of all differences in gain,

$$\bar{g} = \frac{1}{n}\sum_{j=1}^{n} \Delta g_j \tag{29}$$

where n is the total number of test cases.

For example, assuming that the null model is randomly guessing, we calculate the average gain for the introductory example as follows: $\bar{g} = \frac{1}{10}[(1-0.5) + (2-1) + (2-1.5) + (3-2) + (4-2.5) + (4-3) + (4-3.5) + (5-4) + (5-4.5) + (5-5)] = 0.75$. Geometrically, the average gain is the average distance between the gain curve and the red line in Figure 4a.

Figure 4b shows the corresponding lift chart. Note that the expected lift for random guessing is 1.

Definition 26. Average lift

Let the test cases be ranked based on decreasing values of the class membership scores $p_i = P(y_i = 1|\mathbf{x}_i) = C(\mathbf{x}_i)$. Let $\text{lift}(j)$ denote the lift when the j top-ranked cases are classified as positive. The *average lift* is defined as

$$\overline{\text{lift}} = \frac{1}{n}\sum_{j=1}^{n} \text{lift}(j) \tag{30}$$

where n is the total number of test cases.

For the introductory example, we calculate the average lift as follows (see Figure 2): $\overline{\text{lift}} = \frac{1}{10}(2 + 2 + 4/3 + 3/2 + 8/5 + 4/3 + 8/7 + 5/4 + 10/9 + 1) = 1.427$.

The *receiver operating characteristic* (ROC) curve is one of the most widely used graphical tools to plot the performance of a binary classifier [17, 18, 19, 20, 21]. The ROC curve depicts the trade-offs between the false positive rate (or 1 minus specificity, shown on the x-axis) and the true positive rate (or sensitivity or recall, shown on the y-axis). These trade-offs correspond to all possible binary classifications that result from any dichotomization of the ranking scores. The points connecting the segments of the ROC curve are called *operating points*, and they correspond to the possible classification thresholds.

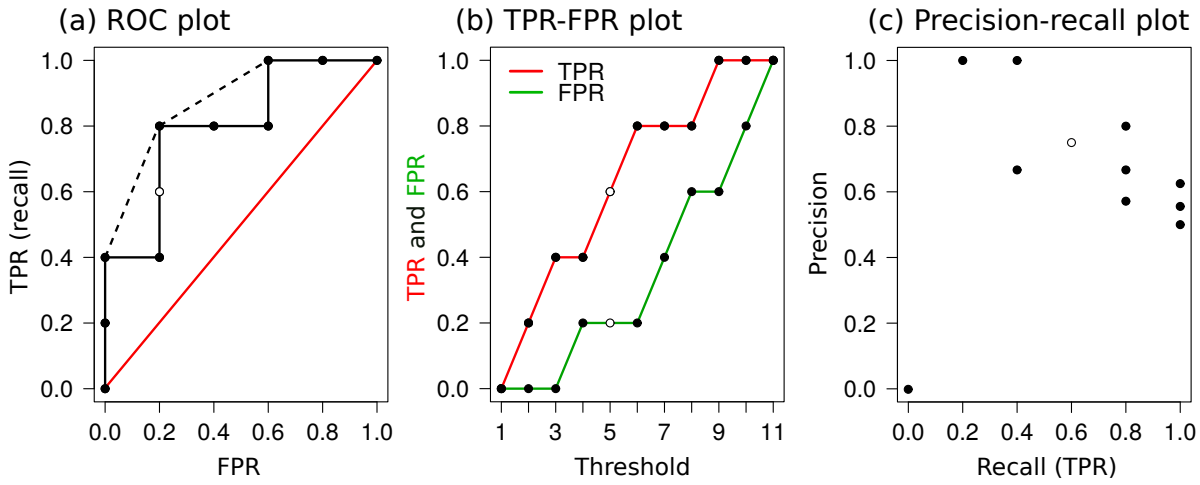


Figure 5: Three different methods for the graphical representation of ranking performance. (a) ROC plot, showing the ROC curve (solid line) and the ROC convex hull (with dotted lines spanning the concavities). The red line is the expected performance of a random guesser. (b) TPR-FPR plot. (c) Precision-recall plot. The point marked by a white circle results from thresholding the ranking scores at t_5.

Figure 5a shows the ROC plot for the introductory example. The diagonal red line is the expected ROC curve of a random guesser. The area under the diagonal is 0.5. Hence, any "good" model should produce a ROC curve above the red line, with an *area under the curve* (AUC) larger than 0.5. The AUC is a commonly used summary statistic of the ROC curve and can be interpreted as a conditional probability because it is equivalent to a Wilcoxon rank-sum statistic [22, 23, 24]: given any randomly selected positive and negative case, the AUC is the probability that the model assigns a higher score to the positive case (i.e., ranks it before the negative case). Following Hilden's notation [25], the AUC is defined as follows.

Definition 27. Area under the ROC curve

Let $P(p_+ > p_-|\mathbf{x}_+ \text{ and } \mathbf{x}_-)$ denote the probability that a randomly selected actual positive case, $\mathbf{x}_+$, has a higher ranking score, p_+, than a randomly selected negative case, $\mathbf{x}_-$, i.e., $p_+ > p_-$. Here, a *higher* ranking score means that $\mathbf{x}_+$ is ranked *before* $\mathbf{x}_-$. Let $n_+ > 0$ be the number of positive instances and $n_- > 0$ be the number of negative instances, and $n = n_+ + n_-$. Let $\text{TPR}(t_i)$ and $\text{FPR}(t_i)$ denote the true positive rate and false positive rate for a threshold t_i, respectively, where $k = n + 1$ is the number of possible thresholds. The *area under the ROC curve* (AUC) is defined as

$$\text{AUC} = P(p_+ > p_-|\mathbf{x}_+ \text{ and } \mathbf{x}_-) = \int_0^1 \text{TPR}(t_i) d\text{FPR}(t_i) \tag{31}$$

From an empirical ROC curve, the AUC can be calculated as

$$\text{AUC} = \frac{S_- - 0.5n_-(n_- + 1)}{n_+ n_-} \tag{32}$$

where S_- is the sum of the ranks of the negative instances.

For the introductory example, we obtain $S_- = 3 + 6 + 7 + 9 + 10 = 35$, $n_- = 5$, $n_+ = 5$, and AUC $= \frac{35 - 0.5 \times 5 \times 6}{5 \times 5} = 0.8$, which corresponds to the shaded area in Figure 6a. Note that the expected AUC of a random model is 0.5, whereas the AUC of a perfect model is 1.

A closely related measure is the *Gini index*, which is defined as follows.

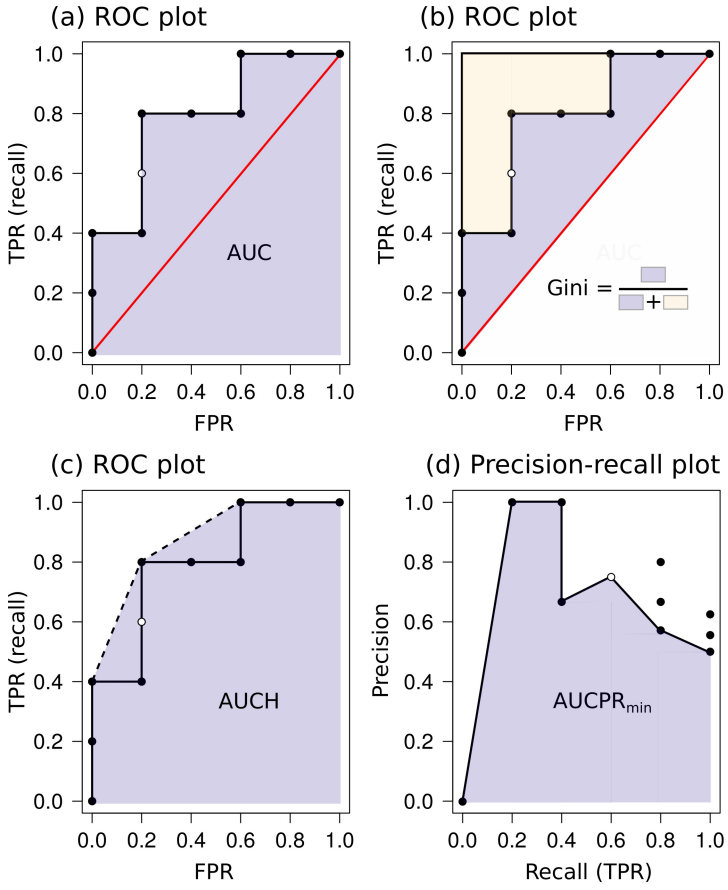


Figure 6: Different graphical representations of ranking performance for the model from the introductory example (cf. Figure 2). The point (0.2, 0.6) in ROC space and the point (0.6, 0.75) in precision-recall space result from thresholding the ranking scores at t_5. (a) ROC plot, with AUC = 0.8. The red line is the expected performance of a random guesser. (b) ROC plot, with Gini index $= \frac{0.3}{0.5} = 0.6 = 2 \times \text{AUC} - 1$. (c) ROC plot, with AUCH = 0.88. (d) Precision-recall plot, with a trapezoidal estimator for the area under the curve, $\text{AUCPR}_{\text{min}} = 0.6476$.

Definition 28. Gini index or Gini coefficient

The *Gini index* (or *Gini coefficient*) is defined as

$$\text{Gini} = \frac{A}{B} \tag{33}$$

where A is the area between the ROC curve and the diagonal line from (0,0) to (1,1), and B is the area above that line, with $B = 0.5$.

The Gini index can be easily derived from the AUC. Consider Figure 6b. A is the purple area. As $\text{AUC} = A + 0.5$, we have $A = \text{AUC} - 0.5$. Then, since $\text{Gini} = \frac{A}{0.5} = 2A$, we obtain $\text{Gini} = 2 \times \text{AUC} - 1$. For the introductory example, we obtain $\text{Gini} = 2 \times 0.8 - 1 = 0.6$.

A further measure that can be derived from the ROC curve is the *area under the ROC convex hull* (AUCH). The ROC convex hull is the hull that encloses the operating points of the ROC curve [26, 20]. The line segment $(0, 0)$ to $(0, 0.4)$, the dotted lines, and the segment $(0.6, 1.0)$ to $(1.0, 1.0)$ in Figure 5a represent the convex hull for the introductory example; the corresponding AUCH is shown in Figure 6c. Note that in machine learning, a curve is usually called "convex" if any straight line interpolating between two points on the curve is never above the curve. This interpretation is different from the definition of a convex curve in mathematics.

Definition 29. Area under the ROC convex hull

The *area under the ROC convex hull* (AUCH) is the area under the curve that results from the interpolation between the following k points in ROC space, which are ordered based on increasing values of their abscissa: the origin $(x_i, y_i) = (0, 0)$, the minimum set of points spanning the concavities, and the point (1,1). The AUCH can be calculated for an empirical ROC curve by using the trapezoid rule,

$$\text{AUCH} = \sum_{i=1}^{k-1} y_i(x_{i+1} - x_i) + 0.5(y_{i+1} - y_i)(x_{i+1} - x_i) \tag{34}$$

For the introductory example, we obtain $\text{AUCH} = 0.88$.

In Figure 5b, the true positive rate (TPR) and false positive rate (FPR) are plotted as a function of the classification threshold in a *TPR-FPR plot*, from which the *Kolmogorov-Smirnov statistic* can be derived.

Definition 30. Kolmogorov-Smirnov statistic

The *Kolmogorov-Smirnov statistic* (KS) is the maximum value of the absolute difference between two cumulative distributions. When we assess the ranking ability of a model, these distributions are given by the true positive rates, $\text{TPR}(t_i)$, and false positive rates, $\text{FPR}(t_i)$, for all classification thresholds t_i.

$$\text{KS} = \max\{|\text{TPR}(t_i) - \text{FPR}(t_i)|\} \tag{35}$$

The KS statistic has a simple geometrical interpretation as the maximum distance between the TPR and FPR curves. In Figure 5b, the distance between the TPR and FPR curve is maximal for threshold t_6, and $\text{KS} = 0.8 - 0.6 = 0.2$. The KS statistic was shown to be sensitive to various types and levels of noise [27, 28] and is therefore not recommended for model evaluation and comparison. A closely related but more robust measure is the *truncated average Kolmogorov-Smirnov statistic* (taKS) [27], which calculates the average distance between the TPR- and FPR-curves.

Definition 31. Truncated average Kolmogorov-Smirnov statistic

Let $\text{TPR}(t_i)$ and $\text{FPR}(t_i)$ denote the true positive and false positive rate for a classification threshold t_i, respectively, where $k = n + 1$ is the number of possible thresholds. The *truncated average Kolmogorov-Smirnov statistic* (taKS) is the average distance between the TPR- and FPR-curves, excluding the start- and endpoints $(0, 0)$ and $(1, 1)$,

$$\text{taKS} = \frac{1}{k-2} \sum_{i=2}^{k-1} [\text{TPR}(t_i) - \text{FPR}(t_i)] \tag{36}$$

For the introductory example, we obtain $\text{taKS} = \frac{1}{9} \times [(0.2 - 0) + (0.4 - 0) + (0.4 - 0.2) + (0.6 - 0.2) + (0.8 - 0.2) + (0.8 - 0.4) + (0.8 - 0.6) + (1 - 0.6) + (1 - 0.8)] = 0.33$.

In precision-recall plots (Figure 5c), the precision is plotted as a function of the recall (or true positive rate or sensitivity). A frequently used summary measure of a precision-recall plot is the *average precision*, which corresponds to the *area under the precision-recall curve* (AUCPR).

Definition 32. Average precision or area under the precision-recall curve

Let n_+ be the number of positive instances and n_- be the number of negative instances, with $n_+ > 0$ and $n_- > 0$ and $n_+ + n_- = n$. Let $h_+(t_i)$ denote the hits, i.e., the number of positive instances at or above the threshold t_i, $i = 1..k$, where $k = n + 1$ is the number of possible classification thresholds. Accordingly, let $h_-(t_i)$ denote the number of negative instances at or above the threshold. The recall at threshold t_i is $r(t_i) = \text{TPR}(t_i) = \frac{h_+(t_i)}{n_+}$. The precision is $p(t_i) = \frac{h_+(t_i)}{i-1}$ for $i > 1$ and 0 for $i = 1$. The *average precision* (AP) is the area under the precision-recall curve,

$$\text{AP} = \int_0^1 p(t_i) dr(t_i) \tag{37}$$

From an empirical precision-recall curve, AP can be calculated by using the trapezoidal rule,

$$\text{AP} = \sum_{i=1}^{k} p(t_i) \Delta r_i = \sum_{i=1}^{k} p(t_i)[r(t_i) - r(t_{i-1})] \tag{38}$$

with $r(t_0) = 0$.

For the introductory example, we calculate the average precision as follows (see Figure 2): AP $= 1 \times 1/5 + 1 \times (2/5 - 1/5) + 2/3 \times (2/5 - 2/5) + 3/4 \times (3/5 - 2/5) + 4/5 \times (4/5 - 3/5) + 4/6 \times (4/5 - 4/5) + 4/7 \times (4/5 - 4/5) + 5/8 \times (5/5 - 4/5) + 5/9 \times (5/5 - 5/5) + 5/5 \times (5/5 - 5/5) = 0.835$.

The term "average precision" could be misunderstood as the average over the precisions resulting from all possible thresholds; in the introductory example, this average is $\frac{1}{10} \times (1 + 1 + 2/3 + 3/4 + 4/5 + 4/6 + 4/7 + 5/8 + 5/9 + 5/10) = 0.714$. Note that the expected AP of a random model is $\frac{n_+}{n}$, whereas the AP of a perfect model is 1.

From an empirical precision-recall plot, it is not obvious how the area should be calculated because the precision normally does not change monotonically with increasing recall (cf. Figure 5c) [29, 30]. From a plot like the one shown in Figure 5c, we can construct different curves by interpolating through different points, and consequently, we obtain (slightly) different areas under the curve, which can be calculated by using *trapezoidal estimators*.

Definition 33. Trapezoidal estimators of the area under the empirical precision-recall curve

Let $p_{\min}(t_i)$ and $p_{\max}(t_i)$ denote the minimum and maximum precision, respectively, for a recall at threshold t_i, $i = 1..k$, where $k = n + 1$ is the number of possible thresholds. The *area under the empirical precision-recall curve* (AUCPR) can be estimated by $\text{AUCPR}_{\text{minmax}}$, $\text{AUCPR}_{\max}$ (upper trapezoid), or $\text{AUCPR}_{\min}$ (lower trapezoid), where

$$\text{AUCPR}_{\min} = \sum_{i=1}^{k-1} \frac{p_{\min}(t_i) + p_{\min}(t_{i+1})}{2} [r(t_{i+1}) - r(t_i)] \tag{39}$$

$$\text{AUCPR}_{\text{minmax}} = \sum_{i=1}^{k-1} \frac{p_{\min}(t_i) + p_{\max}(t_{i+1})}{2} [r(t_{i+1}) - r(t_i)] \tag{40}$$

$$\text{AUCPR}_{\max} = \sum_{i=1}^{k-1} \frac{p_{\max}(t_i) + p_{\max}(t_{i+1})}{2} [r(t_{i+1}) - r(t_i)] \tag{41}$$

The *mean average precision* (MAP) is the extension of AP to more than two classes [31]. MAP is simply the arithmetic mean of the average precisions, which we obtain by considering each class in turn as the positive class. MAP is a widely used performance measure in information retrieval [31].

The *triplet loss* was developed for image recognition [54]. The rationale for the triplet loss is that a given positive image x_i^a (called *anchor*) should be closer to all other positive images x_i^p than to any other negative image x_i^n. For example, x_i^a may be a picture of a person's face, x_i^p is another picture of the same person's face, and x_i^n is the picture of another person's face. This means that

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

for all possible triplets $f(x_i^a)$, $f(x_i^p)$, and $f(x_i^n)$ from the data set, and where α is a margin between

positive and negative pairs.

Definition 34. Triplet loss

Let $f(x) \in \mathbb{R}^d$ represent an embedding of an image x into a d-dimensional Euclidean space, with the Euclidean norm $\|f(x)\|_2 = 1$. The *triplet loss* is defined as

$$\mathcal{L}_t = \sum_{i=1}^{N} \max\{0, (\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha)\} \tag{42}$$

where N is the number of images, and $\|\times\|_2^2$ denotes the squared Euclidean norm.

The *hinge loss* is used for training maximal margin classifiers, i.e., algorithms that try to find a decision boundary or hyperplane, subject to some constraints, that maximizes the margin between the cases of the positive and negative class. An example of such a classifier is the support vector machine [56].

Definition 35. Hinge loss

Let a data set contain n cases $\mathbf{x}_i$ that belong to either the positive class, $y = +1$, or the negative class, $y = -1$. Let a model C produce scores $s_i \in \mathbb{R}$ for the cases, which are interpretable as signed distances from a decision boundary that separates the positive from the negative class. The *hinge loss* for the i^{th} case is defined as

$$\mathcal{L}_h = \max\{0,\ 1 - y_i \times s_i\} \tag{43}$$

The *average hinge loss* is defined as

$$\bar{\mathcal{L}}_h = \frac{1}{n} \sum_{i=1}^{n} \max\{0,\ 1 - y_i \times s_i\} \tag{44}$$

Note that the negative class must be encoded as -1, not as 0, like in the previous examples. Figure 7 shows the hinge loss and the 0-1 loss for $y = +1$.

If the predicted score, s_i, and the true class, y_i, have the same sign, then the corresponding case lies on the correct side of the boundary. If in addition the absolute value of s_i is larger than or equal to 1, then the hinge loss is 0. The case is considered sufficiently far from the boundary and therefore entails no penalty. If the score and the true class have opposite signs, then the case lies on the wrong side of the boundary. The hinge loss is now a linear function of the score. Also, all cases that are on the correct side of the boundary but not sufficiently far from the boundary, that is, $|s_i| < 1$, contribute linearly to the hinge loss.

Figure 8a shows the same data as Figure 1, except that the numbers on the horizontal axis are signed distances from the decision boundary. These signed distances are the scores, s_i. Figure 8b shows the ranking of the cases based on decreasing scores, together with their hinge losses. Note that cases #8 and #3 are misclassified, as they lie on the wrong side of the boundary. As they have the same distance from the

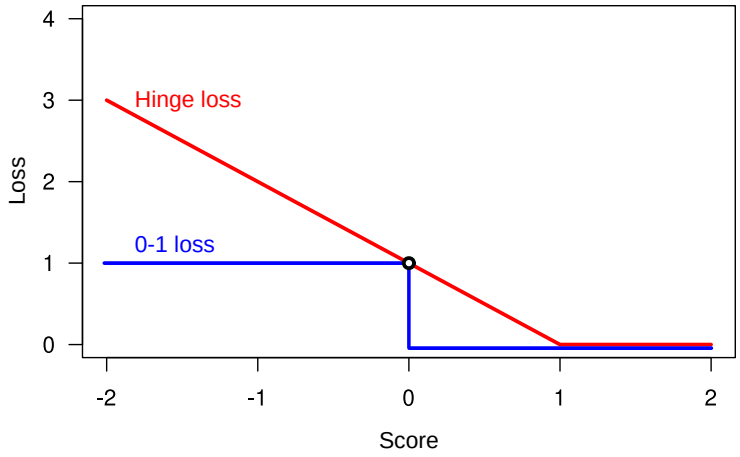
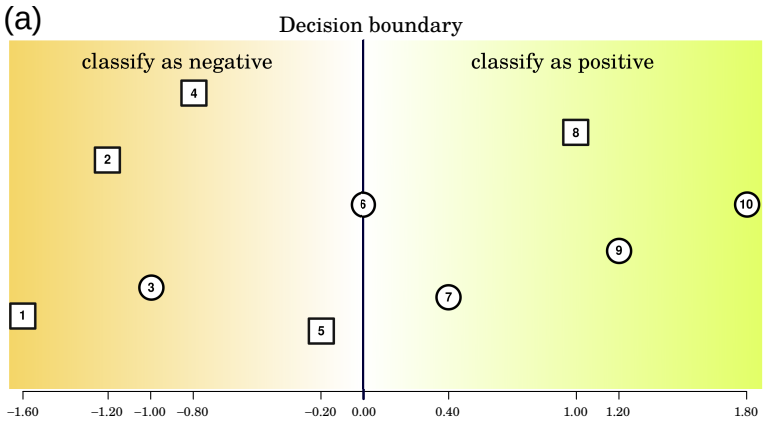


Figure 7: Hinge loss (red) and 0-1 loss (blue) for $y = +1$.

boundary, they lead to the same hinge loss of $\mathcal{L}_h = 2$.



(b)

Case	Class	Score	Hinge loss
10	+1	1.80	0.0
9	+1	1.20	0.0
8	−1	1.00	2.0
7	+1	0.40	0.6
6	+1	0.00	1.0
5	−1	−0.20	0.8
4	−1	−0.80	0.2
3	+1	−1.00	2.0
2	−1	−1.20	0.0
1	−1	−1.60	0.0

Figure 8: (a) The same ten cases from two classes as shown in Figure 1, except that the numbers on the horizontal axis reflect signed distances from the decision boundary. (b) Hinge loss of the ten cases.

5. Evaluating an observed value of a performance measure

Once we have the value of our performance measure, how should we interpret it? For example, assume that we obtain a sensitivity of 0.80—how do we know whether this is a good result or not? The performance may of course be evaluated based on domain expert knowledge. If such knowledge is not available, however, then we could consider the statistical significance of the observed performance, keeping of course in mind the caveats and pitfalls of the p-value and that confidence intervals are always more meaningful than p-values [32, 45]. The statistical literature contains various approaches for calculating a confidence interval for a proportion. Here, we will present only two of them: a *simple asymptotic confidence interval* based on the normal approximation and the *exact Clopper-Pearson confidence interval*. We will assume that the

prediction for an instance $\mathbf{x}_i$ is independent from the prediction for another instance $\mathbf{x}_j$, $i \neq j$. Also, we will assume that the test instances were randomly sampled from the target population. In practice, both assumptions may be violated, of course.

5.1. Significance testing with random permutation tests

Random permutation tests are non-parametric Monte Carlo procedures that make fewer distributional assumptions than parametric significance tests. Such tests can be used to assess the statistical significance of an observed result. There exist various random permutation tests; for an overview, see [33, 34]. In our context, the basic procedure of a random permutation test is as follows:

1. Train the classifier on the training set and apply it to the test set. Calculate the value v of the statistic of interest, i.e., the value of the performance measure.
2. Randomly permute the class labels of the training set. Thereby, any association between the data set attributes and the class labels is destroyed.
3. Retrain the classifier on the *permuted* training set and apply it to the test set. Calculate the statistic again (i.e., the value w_i of the performance measure of interest).
4. Repeat steps (2) and (3) many times (e.g., $k = 10\,000$ times) to generate the empirical distribution of the statistic under the null hypothesis of no association between the class labels and the data set attributes.

The p-value is defined as the probability of observing results as extreme as (or more extreme than) the observed results, given that the null hypothesis is true and given the stopping intentions [35, 45]. The p-value quantifies the compatibility between the null hypothesis and the observed result: the smaller the p-value, the less compatible is the null hypothesis with the observed result. We can calculate a permutation-based p-value for our observed performance v by counting how many values w_i are at least as extreme as v. For example, assume that $v = 0.80$ is the observed sensitivity. Let us further assume that we performed $10\,000$ random permutations and that among these, 300 values w_i are at least 0.80. Then the one-sided, empirical p-value is $\frac{300}{10\,000} = 0.03$. Thus, under the null hypothesis that there is no association between the attributes and the class labels, the probability of observing a sensitivity of at least 0.80 is estimated as 0.03.

5.2. Approximate confidence interval for a proportion

Let p denote the (unknown) proportion in a population. For a sufficiently large sample, the sample proportion, $\hat{p}$, is approximately normally distributed with a mean of p. The standard deviation of the sampling distribution of the sample proportion (i.e., the standard error of the sample proportion) is $\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$, which is estimated as $s_{\hat{p}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$. The *Wald interval* is an approximate $(1-\alpha)100\%$ confidence interval for a population proportion [36].

Definition 36. Wald confidence interval

Let $\hat{p}$ denote a proportion estimated from a sample of size n. If $n\hat{p} > 5$ and $n(1-\hat{p}) > 5$, then an approximate asymptotic $(1-\alpha)100\%$ confidence interval for the population proportion p is given by the *Wald interval*,

$$\hat{p} \pm z_{1-\alpha/2} \times \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{45}$$

where $z_{1-\alpha/2}$ is the quantile of the standard normal distribution for probability $1-\alpha/2$.

For example, a 95% Wald interval for the sensitivity in the introductory example is calculated as follows. With $TP = 3$ and $FP = 2$, we obtain $\hat{p} = \frac{3}{5}$ (= sensitivity). The interval is then

$$\hat{p} \pm z_{1-\alpha/2} \times \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = 0.6 \pm 1.96 \times \sqrt{\frac{0.6 \times 0.4}{5}} = 0.6 \pm 0.4294 = [0.1706, 1.0294]. \tag{46}$$

We see that the upper bound exceeds 1, which, obviously, is the maximum sensitivity; hence, the confidence interval includes values that are impossible. The approximation is not good because the conditions $n\hat{p} = 5 \times 0.6 > 5$ and $n(1-\hat{p}) = 5 \times (1-0.6) > 5$ do not hold; therefore, an exact confidence interval should be calculated.

5.3. Exact confidence interval for a proportion

An exact confidence interval based on the F-distribution is given by the Clopper-Pearson interval [37].

Definition 37. Exact Clopper-Pearson confidence interval

Let $\hat{p} = \frac{r}{n}$ denote a proportion estimated from a sample of size n. An exact $(1-\alpha)100\%$ *Clopper-Pearson confidence interval* for the population proportion p is given by

$$\left[\frac{r}{r + (n-r+1)F_{1-\alpha/2;2(n-r+1),2r}}, \frac{(r+1)F_{1-\alpha/2;2(r+1),2(n-r)}}{(n-r) + (r+1)F_{1-\alpha/2;2(r+1),2(n-r)}}\right] \tag{47}$$

where $F_{1-\alpha/2;df_1,df_2}$ is the quantile function of the F-distribution with df_1 and df_2 degrees of freedom.

For instance, a 95% confidence interval for the sensitivity in the introductory example is calculated as follows.

$$\left[\frac{3}{3 + (5-3+1)F_{1-0.05/2;2(5-3+1),2\times 3}}, \frac{(3+1)F_{1-0.05/2;2(3+1),2(5-3)}}{(5-3) + (3+1)F_{1-0.05/2;2(3+1),2(5-3)}}\right] = [0.1466, 0.9473].$$

5.4. Bootstrap percentile confidence interval

To derive a confidence interval analytically, it is necessary to calculate the standard error of the sample statistic, which is not trivial for the more intricate performance measures. When conventional parametric confidence intervals are difficult to calculate or when their assumptions (for example, an approximate normal

distribution) are violated, deriving a *bootstrap confidence interval* is an interesting alternative. The *bootstrap* is a data resampling method for assessing the accuracy of statistical estimates [38, 39, 42, 43]. There exist different types of bootstrap confidence intervals [40, 41]. One of the simplest ones is the *percentile bootstrap* [38]. The basic procedure for deriving a bootstrap percentile confidence interval is as follows [42]:

1. From the available learning set, L, which contains a total of n instances, generate a bootstrap set, B, by randomly and uniformly sampling n instances with replacement.
2. Repeat step (1) b times to generate B_i bootstrap sets, $i = 1..b$.
3. Build model C using the set B_i as training set, and apply C to the corresponding out-of-bag set T_i, which contains the elements from L that are not in B_i.
4. Calculate the value of the performance measure $\hat{\theta}_i$ from T_i.
5. Repeat steps (3) and (4) for all b bootstrap sets and calculate all $\hat{\theta}_i$, $i = 1..b$.

Definition 38. Bootstrap percentile confidence interval

Let L denote a learning set from a population. Let θ denote the unknown value of a performance measure for a predictive model C for that population. Let B_i denote the i^{th} bootstrap set (i.e., i^{th} training set) that was sampled uniformly with replacement from L, and let the number of instances in B_i and L be the same. Let T_i denote the i^{th} out-of-bag test set, with $T_i = L \setminus B_i$. The model C is trained on B_i and then applied to T_i. Let $\hat{\theta}_i$ denote the resulting value of the performance measure. A $(1-\alpha)100\%$ *bootstrap percentile confidence interval* ($\mathrm{CI}_{\mathrm{boot}}$) for θ is given by

$$\mathrm{CI}_{\mathrm{boot}} = [\hat{\theta}_{\alpha/2}, \hat{\theta}_{1-\alpha/2}] \tag{48}$$

where $\hat{\theta}_{\alpha/2}$ and $\hat{\theta}_{1-\alpha/2}$ are the $\alpha/2$ and $1-\alpha/2$ percentiles, respectively, of the empirical distribution of $\hat{\theta}$.

In R, bootstrap confidence intervals can be generated with the function `boot.ci` of the package `boot` [46, 47].

6. Discussion

Which performance measure should be used in practice? This question does not have a definitive answer, as the different measures quantify different aspects of the performance. If a predictive model is developed for a concrete application, then it is often clear which performance measure matters most. On the other hand, if the general usefulness of a new classifier is to be assessed, then the choice of the most appropriate measure is often not obvious. In that case, reporting several measures can be meaningful.

When the classes are highly imbalanced, accuracy and error rate are not really meaningful. For example, assume that the class ratio is 9:1 in both the training and test set. A model that always predicts the majority

class (i.e., a majority voter) is expected to achieve an accuracy of 90%, although it has not learned anything from the data beyond the class priors. For data sets with highly imbalanced classes, the average precision (Definition 32) is a recommended ranking measure. The average precision is also easily extended to more than two classes as mean average precision. Also, the balanced cross-entropy (Definition 21) and the focal loss (Definition 22) are good performance measures when the classes are imbalanced.

Performance measures that are based on a probabilistic interpretation of error are, in general, preferable to measures that rely on a single classification threshold. For example, let us assume that model A made the predictions shown in Table 2, and let us assume that model B made the same predictions, except that case #8 was misclassified with a score of 0.79. This is clearly a worse prediction than that of model A, which produced the score 0.75 for case #8. However, for t_5, the threshold-based measures would not discriminate between A and B, in contrast to the measures that are based on a probabilistic interpretation of error.

Graphical tools such as ROC curves, lift charts, TPR-FPR plots, and precision-recall curves generally paint a clearer picture of the predictive performance than summary statistics. Clearly, it is often desirable to use a single value, for example, in order to tabulate the results of various models so that they can be ranked from best to worst, which is commonly done in data mining competitions. However, we have to keep in mind that we lose important information when we condense a two-dimensional plot into a single scalar. Thus, whenever possible, performance plots are preferable to scalars.

Ranking measures that are derived from such plots are popular evaluation measures; particularly, the AUC is widely used in machine learning. In contrast to accuracy, for example, the AUC is relatively robust to class imbalances and other types of noise [27, 28]. Therefore, the AUC is not a bad choice as a performance measure, although several shortcomings have been pointed out[25, 48, 49, 50]. For example, consider a drug discovery study that aims at ranking thousands of chemical compounds based on their toxicological effect. Here, it is typically possible to follow up on only a small number of top-ranked compounds, which should be predicted very accurately. By contrast, it may be irrelevant how well the lower-ranked compounds were predicted. This is also known as the *early retrieval problem*: although we are interested in only the top-ranked instances, the AUC also reflects the quality of the predictions of the remaining instances, which may not be interesting.

There exists an equivalence between ROC curves and precision-recall curves, in the sense that they contain the same points for the same predictive model [30]. However, for data sets with highly imbalanced classes, precision-recall curves are more informative than ROC curves because precision-recall curves emphasize the performance with respect to the top-ranked cases [51]. In other words, if the correctness of the few top-ranked instances matters most (like in the mentioned drug discovery study), then precision-recall curves are preferable to ROC curves. This might explain why the average precision is more commonly used than the AUC in information retrieval [52].

Importantly, the performance measure in the training phase should be the same as the measure in the validation and test phase, since a model that was optimized to achieve, say, a high AUC on the training set

is not necessarily expected to achieve, say, a low logloss on the test set. There are further important aspects that need to be considered when we evaluate a predictive model or learning algorithm, such as the choice of benchmark data sets, data resampling strategies, statistical tools to compare different models or algorithms, etc. These aspects are beyond the scope of this article; for a more in-depth discussion, see for example [53].

References

[1] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, H. Nielsen, Assessing the accuracy of prediction algorithms for classification: an overview, Bioinformatics 16 (5) (2000) 412–424.

[2] D. Berrar, Performance measures for binary classification, in: S. Ranganathan, K. Nakai, C. Schönbach, M. Gribskov (Eds.), Encyclopedia of Bioinformatics and Computational Biology, 1st edition, Elsevier, 2018, pp. 546–560.

[3] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognition Letters 30 (2009) 27–38.

[4] A. Buja, W. Stuetzle, Y. Shen, Loss functions for binary class probability estimation and classification: Structure and applications, manuscript, available at `www-stat.wharton.upenn.edu/~buja`, accessed 24 May 2023, (2005).

[5] T. Gneiting, A. Raftery, Strictly proper scoring rules, prediction, and estimation, Journal of the American Statistical Association 102 (477) (2007) 359–378.

[6] J. Witkowski, P. Atanasov, L. Ungar, A. Krause, Proper proxy scoring rules, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI Press, 2017, pp. 743–749.

[7] W. Youden, Index for rating diagnostic tests, Cancer 3 (1) (1950) 32–35.

[8] M. Ruopp, N. Perkins, B. Whitcomb, E. Schisterman, Youden index and optimal cut-point estimated from observations affected by a lower limit of detection, Biometrical Journal 50 (3) (2008) 419–430.

[9] B. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochimica et Biophysica Acta—Protein Structure 405 (2) (1975) 442–451.

[10] D. Powers, Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation, Journal of Machine Learning Technologies 2 (2011) 37–63.

[11] S. Boughorbel, F. Jarray, M. El-Anbari, Optimal classifier for imbalanced data using Matthews correlation coefficient metric, PLoS One 12 (6) (2017) e0177678.

[12] W. Dubitzky, M. Granzow, D. Berrar, Comparing symbolic and subsymbolic machine learning approaches to classification of cancer and gene identification, in: S. Lin, K. Johnson (Eds.), Methods of Microarray Data Analysis, Kluwer Academic Publishers, 2001, pp. 151–166.

[13] G. Brier, Verfication of forecasts expressed in terms of probability, Monthly Weather Review 78 (1) (1950) 1–3.

[14] I. Kononenko, I. Bratko, Information-based evaluation criterion for classifier's performance, Machine Learning 6 (1) (1991) 67–80.

[15] N. Lavrač, D. Gamberger, S. Džeroski, Noise elimination applied to early diagnosis of rheumatic diseases, in: N. Lavrač, E. T. Keravnou, B. Zupan (Eds.), Intelligent Data Analysis in Medicine and Pharmacology, Springer US, Boston, MA, 1997, pp. 187–205.

[16] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1) (1960) 37–46.

[17] A. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (3) (1997) 1145–1159.

[18] T. Fawcett, ROC graphs: Notes and practical considerations for researchers, Technical Report HPL-2003-4, HP Laboratories (2004) 1–38.

[19] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (2006) 861–874.

[20] P. Flach, ROC analysis, in: C. Sammut, G. Webb (Eds.), Encyclopedia of Machine Learning, Springer, 2010, pp. 869–875.

[21] D. Berrar, P. Flach, Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them), Briefings in Bioinformatics 13 (1) (2012) 83–97.

[22] D. Bamber, The area under the ordinal dominance graph and the area below the receiver operating characteristic curve, Journal of Mathematical Psychology 12 (1975) 387–415.

[23] J. Hanley, B. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1) (1982) 29–36.

[24] J. Hanley, B. McNeil, A method of comparing the areas under receiver operating characteristic curves derived from the same cases, Radiology 148 (3) (1983) 839–843.

[25] J. Hilden, The area under the ROC curve and its competitors, Medical Decision Making 11 (2) (1991) 95–101.

[26] F. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning 42 (3) (2001) 203–231.

[27] D. Berrar, An empirical evaluation of ranking measures with respect to robustness to noise, Journal of Artificial Intelligence Research 49 (6) (2014) 241–267.

[28] D. Berrar, On the noise resilience of ranking measures, in: A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, D. Liu (Eds.), 23rd International Conference on Neural Information Processing (ICONIP), Kyoto, Japan, Proceedings, Part II, Springer, 2016, pp. 47–55.

[29] K. Boyd, K. H. Eng, C. D. Page, Area under the precision-recall curve: Point estimates and confidence intervals, in: H. Blockeel, K. Kersting, S. Nijssen, F. Železný (Eds.), Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III, Springer, Berlin, Heidelberg, 2013, pp. 451–466.

[30] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 233–240.

[31] S. Beitzel, E. Jensen, O. Frieder, MAP, in: L. Liu, M. T. Özsu (Eds.), Encyclopedia of Database Systems, Springer US, Boston, MA, 2009, pp. 1691–1692.

[32] D. Berrar, Confidence curves: an alternative to null hypothesis significance testing for the comparison of classifiers, Machine Learning 106 (6) (2017) 911–949.

[33] P. Good, Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses, Springer Series in Statistics, 2000.

[34] M. Ojala, G. Garriga, Permutation tests for studying classifier performance, Journal of Machine Learning Research 11 (2010) 1833–1863.

[35] T. Sellke, M. Bayarri, J. Berger, Calibration of p values for testing precise null hypotheses, The American Statistician 55 (1) (2001) 62–71.

[36] A. Wald, J. Wolfowitz, Confidence limits for continuous distribution functions, The Annals of Mathematical Statistics 10 (2) (1939) 105–118.

[37] C. Clopper, E. Pearson, The use of confidence or fiducial limits illustrated in the case of the binomial, Biometrika 26 (4) (1934) 404–413.

[38] B. Efron, Nonparametric standard errors and confidence intervals, Canadian Journal of Statistics 9 (2) (1981) 139–158.

[39] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, 1994.

[40] B. Efron, Better bootstrap confidence intervals, Journal of the American Statistical Association 82 (397) (1987) 171–185.

[41] T. DiCiccio, B. Efron, Bootstrap confidence intervals, Statistical Science 11 (3) (1996) 189–228.

[42] D. Berrar, W. Dubitzky, Bootstrapping, in: W. Dubitzky, O. Wolkenhauer, K.-H. Cho, H. Yokota (Eds.), Encyclopedia of Systems Biology, Springer, 2013, pp. 158–163.

[43] D. Berrar, Introduction to the non-parametric bootstrap, in: S. Ranganathan,M. Gribskov, K. Nakai, C. Schonbach, Encyclopedia of Bioinformatics and Computational Biology, 1st edition, 2018, pp. 766–773.

[44] D. Berrar, P. Lopes, W. Dubitzky, Incorporating domain knowledge in machine learning for soccer outcome prediction, Machine Learning 108 (1) (2019) 97–126.

[45] D. Berrar, Using p-values for the comparison of classifiers: pitfalls and alternatives, Data Mining and Knowledge Discovery 36 (3) (2022) 1102–1139.

[46] A. Davison, D. Hinkley, Bootstrap Methods and Their Applications, Cambridge University Press, 1997.

[47] A. Canty, B. Ripley, boot: Bootstrap R (S-Plus) Functions. R package version 1.3-20.

[48] N. Adams, D. Hand, Comparing classifiers when the misallocation costs are uncertain, Pattern Recognition 32 (7) (1999) 1139–1147.

[49] D. Hand, Measuring classifier performance: a coherent alternative to the area under the ROC curve, Machine Learning 77 (2009) 103–123.

[50] C. Parker, On measuring the performance of binary classifiers, Knowledge and Information Systems 35 (2013) 131–152.

[51] T. Saito, M. Rehmsmeier, The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, PLoS One 10 (3) (2015) e0118432.

[52] W. Su, Y. Yuan, M. Zhu, A relationship between the average precision and the area under the ROC curve, in: Proceedings of the 2015 International Conference on the Theory of Information Retrieval, ICTIR 2015, ACM, New York, NY, USA, 2015, pp. 349–352.

[53] N. Japkowicz, M. Shah, Evaluating Learning Algorithms—A Classification Perspective, Cambridge University Press, 2011.

[54] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: A unified embedding for face recognition and clustering, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 815–823.

[55] L. Tsung-Yi, P. Goyal., R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[56] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 2nd edition, 1999.