The algorithm versus the chimps:  
On the minima of classifier performance metrics

Joep Burger\textsuperscript{1} and Quinten Meertens\textsuperscript{2,3,4}

\textsuperscript{1} Statistics Netherlands, Research and Development, CBS-weg 11, PO Box 4481, 6401 CZ Heerlen, the Netherlands  
\texttt{j.burger@cbs.nl}

\textsuperscript{2} Center for Nonlinear Dynamics in Economics and Finance, University of Amsterdam, Roetersstraat 11, PO Box 15867, 1001 NJ Amsterdam, the Netherlands

\textsuperscript{3} Leiden Centre of Data Science, Leiden University, the Netherlands

\textsuperscript{4} Statistics Netherlands, Business Statistics (The Hague), the Netherlands  
\texttt{q.a.meertens@uva.nl}

Abstract. In this paper we seek the minima of performance metrics for binary classification to facilitate comparison between metrics and applications, and to assess the quality of inferential statistics made from non-probability samples. We use these minima to min-max normalize the performance metrics so that they can be interpreted as a percentage of the perfect classifier relative to the proverbial chimps at the zoo\textsuperscript{†} guessing at random. We compare our results with the balanced metrics that have been introduced recently, which are corrected for bias due to class imbalance.

Keywords: Inferential statistics · Non-probability samples · Statistical learning · Supervised machine learning · Binary classification.

1 Introduction

Imagine you have to do a school exam. The test consists of one hundred multiple choice questions. At each question you can choose from four possible answers (A, B, C or D). In the Netherlands, students pass when they score 6 points or more on a scale from 0 to 10. Is it sufficient to answer sixty questions correctly to pass the test? Maybe not, because the proverbial chimps at the zoo that choose answers randomly will on average answer twenty five questions correctly. The teacher could correct for that minimum by grading twenty five correct answers a 0 and then scaling linearly to maintain a perfect score when no mistakes are made by the student. Then, a student answering sixty questions correctly fails the test, having a score of 4.7 on the scale from 0 to 10. The student would now have to answer at least seventy questions correctly to pass the test. From a statistical point of view, an advantage of this so-called min-max normalization,


is that the students’ grades are now comparable with the grades at a second
school where the students can choose between two possible answers (A or B).
The proverbial chimps at the zoo that answer randomly will then answer fifty
question correctly, on average. After min-max normalization, a student at that
second school will only pass the test when answering eighty or more questions
correctly.

In supervised machine learning, algorithms instead of students are trained
to find the right answer to a multiple choice question. The algorithm learns, for
example, that in the case of a drawing the subject is a ‘moon’, ‘rose’ or ‘fish’.
(These are the first words Dutch children learn to read. Many algorithms are
still in elementary school and it is nice to use something else than pictures of
cats and cars as an example.) What final grade does an algorithm get for its
answers to new drawings? For this purpose a considerable list of performance
metrics has been developed (see, e.g., [7]). For some of them it is unclear what
the score of the control group would be: how well would the chimps at the zoo
perform?

In this paper, we will provide an answer to that question. The answer is
important from a statistical point of view: as students’ grades should preferably
be comparable between subjects and schools, we would like the performance of
algorithms to be comparable between metrics and applications. Moreover, we
would like to have a statistical interpretation of the actual value of a perform-
ance metric, such that the interpretation is independent of the metric and
application, similar to the min-max normalized grade of a multiple-choice test:
a grade equal to 6 can always be interpreted as 60 percent between guessing and
perfection. That statistical interpretation is essential when employing supervised
machine learning algorithms at national statistical institutes, such as Statistics
Netherlands, to produce official statistics.

Official statistics provide quantitative information about the status and de-
velopment of well-defined populations such as businesses or households. One of
the challenges is to produce such information at reasonable accuracy, cost and
time. A burning question in official statistics is how to make inference from non-
probability (NP) samples [8]. NP samples like social media messages or sensor
data can offset some disadvantages of questionnaires sent to units in a proba-
bility sample, such as response burden, high costs and a considerable time lag
between data collection and dissemination [1]. However, not all units in the pop-
ulation of interest have a positive and known probability of being included in an
NP sample. This rules out design-based estimators from sampling theory.

Alternatively, the data-generating mechanism of NP samples can be deduced
by modeling the relationship between features and the target variable in the NP
sample and use it to predict the missing data, assuming they are missing at ran-
dom. Statistical models could then be deployed, but more often machine learning
algorithms are used, because they are designed for prediction or extrapolation
and scale better with the number of features.

To assess the quality of the extrapolations, the quality of the predictions are
assessed on test sets for which the actual value is known. A range of performance
metrics exists to this end [7]. However, as noted before, it remains unclear what the proverbial chimps at the zoo would achieve by randomly guessing the value of the target variable. A typical example is high accuracy in imbalanced datasets: if a class has a relative frequency of 95%, it is easy to obtain a seemingly impressive accuracy of 95% by always ‘predicting’ the most common class.

In this paper we seek the minima of performance metrics for binary classification to facilitate comparison between metrics and to assess the quality of inferential statistics made from NP samples. We use these minima to min-max normalize the performance metrics so that they can be interpreted as percentage of perfection relative to the performance of the proverbial chimps at the zoo. We compare our results with balanced metrics [6], which have been corrected for bias due to class imbalance. In our view, the paper is a methodological contribution with preliminary simulation results that encourage a more thorough experimental study in the future.

2 Imbalanced performance metrics

We assume that we have a test set of \( n \) data points, \( n_1 \) of which are labeled positive: \( y_i = 1 \), where \( y_i \) is the observed class of instance \( i \). The fraction \( \alpha = n_1/n \) is referred to as the base rate. The algorithm trained on a training set of \( N - n \) data points predicts for all \( n \) instances the probability that instance \( i \) belongs to the positive class: \( \hat{p}_i = P(y_i = 1) \). By choosing a cutoff \( 0 \leq c \leq 1 \) above which the probability \( \hat{p}_i \) is assigned to the positive class, a \( 2 \times 2 \) contingency table or confusion matrix can be constructed (Table 1). Optimizing cutoff \( c \) is discussed in Section 4.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Positive</th>
<th>Negative</th>
<th>( \sum )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>( X_c )</td>
<td>( n_1 - X_c )</td>
<td>( n_1 )</td>
</tr>
<tr>
<td>Positive</td>
<td>( Y_c )</td>
<td>( n_2 - Y_c )</td>
<td>( n_2 )</td>
</tr>
<tr>
<td>Negative</td>
<td>( X_c + Y_c )</td>
<td>( n - X_c - Y_c )</td>
<td>( n )</td>
</tr>
</tbody>
</table>

\[
TPR_c = \frac{X_c}{n_1} \quad TNR_c = \frac{n_2 - Y_c}{n_2} = 1 - \frac{Y_c}{n_2} \\
PPV_c = \frac{X_c}{X_c + Y_c} \quad NPV_c = \frac{n_2 - Y_c}{n - X_c - Y_c} \quad \alpha = \frac{n_1}{n}
\]

From this confusion matrix the following well-known performance metrics are derived (left two columns of Table 2). Accuracy (\( ACC_c \)) is the fraction of all cases that is predicted correctly. The true positive rate (\( TPR_c \)), also known as sensitivity or recall, is the fraction of positively labeled cases that is predicted correctly and the true negative rate (\( TNR_c \)), also known as specificity, is the fraction of negatively labeled cases that is predicted correctly. The positive predictive value (\( PPV_c \)), also known as precision, is the fraction of predicted
positive cases that is actually labeled ‘positive’ and the negative predictive value
\( (NPV_c) \) is the fraction of predicted negative cases that is actually labeled ‘negative’. The receiver operating characteristic curve, or ROC curve, plots \( TPR_c \)
against the complement of \( TNR_c \) for \( 0 \leq c \leq 1 \). The area under the ROC curve (\( AUC \)) is used as a performance metric. Note that \( TPR_c \) will decrease with \( c \) whereas \( TNR_c \) will increase with \( c \). This trade-off is captured by Youden’s J index \( (J_c) \) or Peirce Skill Score, which is also the vertical distance between the ROC curve and the diagonal. Note also that \( PPV_c \) will become unstable at higher values of \( c \), whereas \( NPV_c \) will become unstable at lower values of \( c \), because the respective denominators decrease there. This trade-off is captured by markedness \( (MRK_c) \). The Matthews correlation coefficient \( (MCC_c) \) is the correlation between the actual and predicted binary classifications. The positive \( F_1 \) score \( (PF_{1c}) \) is the harmonic mean of \( TPR_c \) and \( PPV_c \). Analogously, the negative \( F_1 \) score \( (NF_{1c}) \) is the harmonic mean of \( TNR_c \) and \( NPV_c \). The harmonic mean is more sensitive to one of the values being low than the arithmetic mean.

Table 2. Imbalanced performance metrics and their expected value when randomly guessing the positive class with probability \( g \).

<table>
<thead>
<tr>
<th>Metric</th>
<th>( Q )</th>
<th>( \mathbb{E}[Q(g)] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ACC_c )</td>
<td>( \frac{n_2 + X_c - Y_c}{n} )</td>
<td>( \alpha g + (1 - \alpha)(1 - g) )</td>
</tr>
<tr>
<td>( TPR_c )</td>
<td>( \frac{X_c}{n} )</td>
<td>( g )</td>
</tr>
<tr>
<td>( TNR_c )</td>
<td>( 1 - \frac{Y_c}{n_2} )</td>
<td>( 1 - g )</td>
</tr>
<tr>
<td>( PPV_c )</td>
<td>( \frac{X_c}{n_2 + \frac{X_c - Y_c}{n_2}} )</td>
<td>( \alpha + O\left(\frac{1}{n^2}\right) )</td>
</tr>
<tr>
<td>( NPV_c )</td>
<td>( \frac{Y_c}{n_2 + \frac{X_c - Y_c}{n_2}} )</td>
<td>( 1 - \alpha + O\left(\frac{1}{n^2}\right) )</td>
</tr>
<tr>
<td>( AUC )</td>
<td>( \int_0^1 TPR_c dTNR_c )</td>
<td>( \frac{1}{2} )</td>
</tr>
<tr>
<td>( J_c )</td>
<td>( TPR_c + TNR_c - 1 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( MRK_c )</td>
<td>( PPV_c + NPV_c - 1 )</td>
<td>( \alpha + O\left(\frac{1}{n^2}\right) )</td>
</tr>
<tr>
<td>( MCC_c )</td>
<td>( \frac{n_2 X_c + n_1 Y_c}{\sqrt{n_1 n_2 (X_c + Y_c)(n - X_c - Y_c)}} )</td>
<td>( 0 + O\left(\frac{1}{n^2}\right) )</td>
</tr>
<tr>
<td>( PF_{1c} )</td>
<td>( \frac{2n_2 X_c + n_1 Y_c}{n_2 + \frac{n_1 + X_c - Y_c}{n}} )</td>
<td>( 2\alpha g \left( \frac{1}{\alpha + g} - \frac{\alpha(1-g)}{\alpha + g} \right) + O\left(\frac{1}{n^2}\right) )</td>
</tr>
<tr>
<td>( NF_{1c} )</td>
<td>( \frac{2(n_2 + Y_c)}{n_2 + \frac{n_1 + X_c - Y_c}{n}} )</td>
<td>( 2(1 - \alpha)(1 - g) \left( \frac{1}{2 - \alpha - g} - \frac{(1-\alpha)g}{n(2-\alpha - g)^2} \right) + O\left(\frac{1}{n^2}\right) )</td>
</tr>
</tbody>
</table>

We will now formally introduce how to model the outcome of the predictions made by the proverbial chimps at the zoo. To that end, let \( g \) be the probability that a chimp at the zoo predicts the positive class. We assume that the chimps will all guess according to one and the same strategy out of the following three: they may toss a fair coin, throw a dice with \( n \) sizes, \( n_1 \) of which are labeled ‘positive’, or always guess the most common class (the mode), i.e.:
Then, let $X$ and $Y$ be independently distributed random variables with binomial distributions $X \sim \text{Bin}(n_1, g)$ and $Y \sim \text{Bin}(n_2, g)$. Each evaluation metric is a random variable as well. Table 3 gives the expected confusion matrix. The third column of Table 2 shows how to compute the expected value of each performance metric. The proofs are provided in Appendix A.1. Five of the metrics are linear functions in the random variables $X$ and $Y$, hence, it is trivial to compute their expected value. The expected value of the other six metrics take the form $E[f(X,Y)]$ for a nonlinear, real-valued function $f$. If $n$ is very small, these expectations could in theory be computed by the closed form expression

$$E[f(X,Y)] = \sum_{l=0}^{n_1} \sum_{m=0}^{n_2} f(l,m) \mathbb{P}(X = l) \mathbb{P}(Y = m).$$

In practice, however, it might take a relatively long time to evaluate this expression if $n$ gets large. So, unless $n$ is small, the approximations given in Table 2 should be used. In addition, note that both $X$ and $Y$ have a (very small, but strictly) positive probability of being 0, in which case $f$ might not be defined (e.g., for $PPV$ we find $0/0$). Mathematically, the correct way to deal with this is to exclude the event by conditioning the expectations on its complement. In practice, in particular for larger values of $n$, the obtained value will be very close to simply skipping the terms in the summation where $f$ is not defined.

**Table 3.** Expected confusion matrix when randomly guessing the positive class with probability $g$.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>$n_1g$</td>
<td>$n_2(1-g)$</td>
</tr>
<tr>
<td>Negative</td>
<td>$n_1(1-g)$</td>
<td>$n_2g$</td>
</tr>
<tr>
<td>$\sum$</td>
<td>$ng$</td>
<td>$n(1-g)$</td>
</tr>
</tbody>
</table>

### 3 Balanced performance metrics

Some performance metrics are biased due to class imbalance. Balanced performance metrics are obtained by rewriting the imbalanced performance metrics.
as a function of the imbalance coefficient $\delta = 2\alpha - 1$ and setting $\delta$ to 0, i.e. $\alpha$ to $\frac{1}{2}$ [6]. Table 4 shows the balanced metrics and an approximation of their expected value when randomly guessing the positive class with probability $g$. The derivations of the formulas in the third column can be found in Appendix A.2.

Table 4. Balanced performance metrics and their expected value when randomly guessing the positive class with probability $g$.

<table>
<thead>
<tr>
<th>Metric $Q^b$</th>
<th>Definition</th>
<th>$E[Q^b(g)]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC$^b_c$</td>
<td>$\frac{TPR_c}{TPR_c + TNR_c}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>TPR$^b_c$</td>
<td>$TPR_c$</td>
<td>$g$</td>
</tr>
<tr>
<td>TNR$^b_c$</td>
<td>$TNR_c$</td>
<td>$1 - g$</td>
</tr>
<tr>
<td>PPV$^b_c$</td>
<td>$\frac{TPR_c}{TPR_c + TNR_c + 1}$</td>
<td>$\frac{1}{2} + \frac{\delta(1-g)}{2n(1+\delta)(1-\delta)g} + O\left(\frac{1}{n^2}\right)$</td>
</tr>
<tr>
<td>NPV$^b_c$</td>
<td>$\frac{TNR_c}{TNR_c + TPR_c + 1}$</td>
<td>$\frac{1}{2} - \frac{\delta(1-g)}{2n(1+\delta)(1-\delta)(1-g)} + O\left(\frac{1}{n^2}\right)$</td>
</tr>
<tr>
<td>AUC$^b$</td>
<td>$2AUC - 1$</td>
<td>0</td>
</tr>
<tr>
<td>$J^b_c$</td>
<td>$J_c$</td>
<td>0</td>
</tr>
<tr>
<td>MRK$^b_c$</td>
<td>$\frac{PPV_c + NPV_c - 1}{TPR_c + TNR_c - 1}$</td>
<td>$\frac{\delta(1-2g)}{2n(1+\delta)(1-\delta)g(1-g)} + O\left(\frac{1}{n^2}\right)$</td>
</tr>
<tr>
<td>MCC$^b_c$</td>
<td>$\frac{TPR_c TNR_c - TPR_c TNR_c}{\sqrt{(TPR_c - TNR_c - 1)(TPR_c - TPR_c + 1)}}$</td>
<td>$\frac{\delta(1-2g)}{2n(1+\delta)(1-\delta)\sqrt{g(1-g)}} + O\left(\frac{1}{n^2}\right)$</td>
</tr>
<tr>
<td>PF$^b_{1c}$</td>
<td>$\frac{2TPR_c}{TPR_c + TNR_c + 2}$</td>
<td>$2g\left(\frac{\frac{1}{1+2g}}{\frac{1}{1+2g}} - \frac{2\left(1-\delta(1+2g)\right)(1-g)}{2n(1+\delta)(1-\delta)(1+2g)}\right) + O\left(\frac{1}{n^2}\right)$</td>
</tr>
<tr>
<td>NF$^b_{1c}$</td>
<td>$\frac{2TNR_c}{TNR_c + TPR_c + 2}$</td>
<td>$2(1-g)\left(\frac{1}{3-2g} - \frac{2\left(1+\delta(3-2g)\right)g}{2n(1+\delta)(1-\delta)(3-2g)}\right) + O\left(\frac{1}{n^2}\right)$</td>
</tr>
</tbody>
</table>

4 Min-max normalization and optimization

After establishing $E[Q(g)]$, min-max normalization can be applied to rescale each metric so that the proverbial chimps at the zoo score 0, on average:

$$Q^m_{mmn}(g) = \frac{Q_c - E[Q(g)]}{1 - E[Q(g)]}. \quad (1)$$

Note that $ACC^m_{mmn}(g)$ equals the Heidke Skill Score [4] or Cohen’s $\kappa$ [3] if $g$ is set to $\frac{X + Y}{n}$. This $g$ is, however, not a random guessing probability. The Heidke Skill Score min-max normalizes accuracy with the expected cell frequencies, which depend on the model.

Through K-fold cross validation or bootstrapping, we obtain the quality of $K$ classifiers trained on different partitions or bootstrap samples of the data. We propose to first average $Q^m_{mmn}$ per cutoff to determine the overall optimal cutoff $c^*$, that is:

$$c^* = \arg\max_c Q^m_{mmn}.$$ \quad (2)
Algorithm vs. Chimps

in which

\[ Q_{cmn} = \frac{1}{K} \sum_{k=1}^{K} Q_{kmn}^c. \]  

Then, we propose to use the distribution of \( Q_{kmn}^c(e^*) \) as a proxy for the quality of the predictions when applied to unlabeled data for making statistical inference.

5 Example: normalized \( F_1 \) scores

Figure 1 shows the \( F_1 \) performance of a fictitious binary classifier that predicts 60% of the actual positive instances correctly \((TPR = 0.6)\) and 80% of the actual negative cases \((TNR = 0.8)\), using \( g^{\text{min}} \) for normalization. Similar figures for accuracy and \( F_1 \) with \( g^{\text{prop}} \) can be found in Appendix B. When the test set is balanced \((\delta = 0, \text{i.e. } \alpha = 0.5)\), this classifier scores \( PF_1 = \frac{2}{3} \) and \( NF_1 = \frac{2}{3} \). The more abundant the positive class relative to the negative class, the higher the classifier scores on \( PF_1 \) and the lower on \( NF_1 \) (thin red line in left panels). One solution to this sensitivity to class imbalance is to balance the metric (thin red lines in right panels) by correcting for the bias. The alternative we propose is to min-max normalize the metric (thick red line in left panels) by relating it to the expected value when randomly guessing the positive class with probability \( g \) (thin blue lines).

Two interesting observations can be made. First, data sets with a different imbalance coefficient can be compared. The classifier performs best at \( \delta \approx -0.18 \) where \( PF_{1}^{\text{min}} \approx 0.34 \), i.e. 34% from perfection relative to tossing a fair coin. A classifier with the same \( TPR \) and \( TNR \) in an application with a higher \( \delta \) scores better on \( PF_1 \) (up to \( \frac{2}{3} \)), the same on \( PF_1^b \) but worse on \( PF_{1}^{\text{min}} \). Second, metrics can be compared. Before min-max normalization, the classifier scores equally well on \( PF_1 \) and \( NF_1 \) at \( \delta = \frac{1}{2} \) (small white points). After min-max normalization, however, the classifier scores equally well on \( PF_{1}^{\text{min}} \) and \( NF_{1}^{\text{min}} \) at \( \delta \approx 0.5 \) (large white points). Between \( \frac{1}{2} < \delta < 0.5 \), \( PF_1 > NF_1 \) but \( PF_{1}^{\text{min}} < NF_{1}^{\text{min}} \).

Note that \( \mathbb{E}[F_1] \) is sensitive to sample size \( n \) (see Tables 2 and 4), which becomes apparent when the metric is balanced and the sample is highly imbalanced (Fig. 1, right panel, blue line).

6 Conclusion

In this paper, we propose to rescale performance metrics through min-max normalization, where the minimum is set to the expected value when randomly guessing the positive class with probability \( g \). It should be explicitly specified which expected value the algorithm is trying to defeat. Our proposed normalization yields different results than correcting for bias due to class imbalance [6] or balancing the sample [e.g. 2; 5]. The min-max normalized metrics allow for a better comparison between applications and between metrics. Moreover, we propose to use the distribution across test sets of a normalized metric at the overall optimal cutoff as performance metric for inferential statistics. Future research could
Fig. 1. Performance of a fictitious binary classifier in relation to imbalance coefficient \( \delta \). \( g = 0.5, TPR = 0.6, TNR = 0.8, n = 1000 \). White points show where positive and negative \( F_1 \) intersect.
focus on generalizing the results from binary classification to multi-class classification and regression, and on metrics that compare the predicted probability directly with the actual label, without a cutoff for constructing the confusion matrix.
Bibliography


A Appendix - Proof of expected values

This appendix belongs to the paper by Burger & Meertens titled "The algorithm versus the chimps: On the minima of classifier performance metrics". It contains the proofs of the formulas provided in the third column of Table 2 and Table 4.

The key idea is to approximate an expectation of the form \( E[f(X,Y)] \), in which \( X \) and \( Y \) are random variables and in which \( f \) is an infinitely differentiable real-valued function, by inserting the Taylor series of \( f \) at \( (E[X],E[Y]) \). More specifically, let \( x_0 = E[X] \) and \( y_0 = E[Y] \) and consider the second-order Taylor series of \( f \) at \( (x_0,y_0) \):

\[
\begin{align*}
  f(x,y) &
  \approx f(x_0,y_0) + f_x(x_0,y_0)(x-x_0) + f_y(x_0,y_0)(y-y_0) \\
  &+ \frac{1}{2} f_{xx}(x_0,y_0)(x-x_0)^2 + \frac{1}{2} f_{yy}(x_0,y_0)(y-y_0)^2 \\
  &+ f_{xy}(x_0,y_0)(x-x_0)(y-y_0).
\end{align*}
\]

We then approximate the expectation \( E[f(X,Y)] \) by taking the expectation of the right-hand side of the above equation. Then, assuming that \( X \) and \( Y \) are uncorrelated, we find

\[
E[f(X,Y)] \approx f(x_0,y_0) + \frac{1}{2} f_{xx}(x_0,y_0) Var(X) + \frac{1}{2} f_{yy}(x_0,y_0) Var(Y).
\]

In the proofs below we will specify the order of the approximation in terms of the size \( n \) of the test dataset.

In the remainder of the appendix, \( n_1 \) and \( n_2 \) are positive integers that sum up to \( n \) and \( g \in (0,1) \) represents the probability that class 1 is predicted by the proverbial chimps at the zoo. Moreover, \( X \) will be a random variable distributed as \( \text{Bin}(n_1,g) \) and \( Y \) a random variable distributed as \( \text{Bin}(n_2,g) \). The random variables \( X \) and \( Y \) are assumed to be independent. The expectation and variance are given by

\[
\begin{align*}
  x_0 &= E[X] = n_1g, \quad \text{Var}(X) = n_1g(1-g),
\end{align*}
\]

and

\[
\begin{align*}
  y_0 &= E[Y] = n_2g, \quad \text{Var}(Y) = n_2g(1-g).
\end{align*}
\]

Finally, we will use the notation \( \alpha = n_1/n \) (and hence \( 1-\alpha = n_2/n \)) and \( \delta = 2\alpha - 1 = (n_1 - n_2)/n \).

A.1 Expected value of imbalanced metrics

This appendix contains the derivations of the approximations of the expected values of the imbalanced performance metrics, as presented in Table 2 of the main text.
Expected Positive Predictive Value (E[PPV])

The positive predictive value PPV can be written as \( f(X,Y) \) with \( f(x,y) = \frac{x}{x+y} \). Check that

\[
f_{xx}(x,y) = \frac{-2y}{(x+y)^3}, \quad f_{yy}(x_0,y_0) = \frac{2x}{(x+y)^3}.
\]

It follows that

\[
\mathbb{E}[PPV] \approx \frac{n_1 g}{ng} - \frac{n_2 g}{(ng)^3} \cdot n_1 g(1-g) + \frac{n_1 g}{(ng)^3} \cdot n_2 g(1-g) = \frac{n_1}{n} = \alpha. \tag{9}
\]

The higher order terms in the Taylor series of PPV are in \( O(1/n^2) \). It can be shown by looking at the terms or order 3 in the Taylor series. Only \( f_{xxx} \) and \( f_{yyy} \) remain, which are both \( O(1/n^3) \) when evaluated at \((x_0,y_0)\), and the third central moment of both \( X \) and \( Y \) are \( O(n) \).

Expected Negative Predictive Value (E[NPV])

The negative predictive value NPV can be viewed as the positive predictive value for the negative class, i.e., to compute NPV we first swap the roles of \( n_1 \) and \( n_2 \) and replace \( g \) by \( 1-g \) and then compute PPV. It follows that

\[
\mathbb{E}[NPV] = 1 - \alpha + O\left(\frac{1}{n^2}\right). \tag{10}
\]

Expected Area under the ROC curve (E[AUC])

If the threshold value \( c \) is equal to 1, then any coin toss prediction by the chimps is considered as tails, corresponding to the point \((0,0)\) on the ROC curve. Similarly, \( c = 0 \) corresponds to the point \((1,1)\) on the ROC curve. For any other value of the threshold value \( c \), the predictions by the chimps do not depend on \( c \), and thus we have \( TPR_c = X/n_1 \) and \( 1 - TNR_c = Y/n_2 \), for any \( 0 < c < 1 \). The ROC curve can be obtained by connected these three points, resulting in (the random variable!)

\[
AUC = \frac{1}{2} \frac{Y}{n_2} \frac{X}{n_1} + \left(1 - \frac{Y}{n_2}\right) \frac{X}{n_1} + \frac{1}{2} \left(1 - \frac{Y}{n_2}\right) \left(1 - \frac{X}{n_1}\right)
\]

\[
= \frac{1}{2} \left(\frac{X}{n_1} + 1 - \frac{Y}{n_2}\right). \tag{11}
\]

It then follows that \( \mathbb{E}[AUC] = \frac{1}{2} \).

Expected Matthews Correlation Coefficient (E[MCC])

The Matthews Correlation Coefficient (MCC) can be written as \( f(X,Y) \) in which

\[
f(x,y) = \frac{n_2 x - n_1 y}{\sqrt{n_1 n_2 (x+y)(n-x-y)}}. \tag{12}
\]
Introducing the function \( D(x, y) = n(x + y) - (x + y)^2 \), the above simplifies to
\[
 f = (n_1n_2)^{-\frac{1}{2}}(n_2x - n_1y)D^{-\frac{1}{2}}.
\] (13)

Both first order partial derivatives of \( D \) are equal to \( n - 2(x + y) \). The identity \( n_2x_0 - n_1y_0 = 0 \) then implies that only the following term remains in \( f_{xx}(x_0, y_0) \):
\[
f_{xx}(x_0, y_0) = 2 \cdot (n_1n_2)^{-\frac{1}{2}} \cdot n_2 \cdot (-\frac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0, y_0) \cdot (2 - n(x_0 + y_0))
\]
\[
= -n_2(1 - 2g)
\]
\[
= \frac{n_2(1 - 2g)}{n^2/1n_2g(1 - g)^3}.
\] (14)

Notice that \( f_{xx}(x_0, y_0) = O(1/n^2) \), and hence \( f_{xx}(x_0, y_0) \) \( \text{Var}(X) = O(1/n) \).

Similarly, we obtain
\[
f_{yy}(x_0, y_0) = 2 \cdot (n_1n_2)^{-\frac{1}{2}} \cdot (-n_1) \cdot (-\frac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0, y_0) \cdot (2 - n(x_0 + y_0))
\]
\[
= \frac{n_1(1 - 2g)}{n^2/1n_2g(1 - g)^3}.
\] (15)

Interestingly, we have derived that
\[
f_{yy}(x_0, y_0) \text{Var}(Y) = -f_{xx}(x_0, y_0) \text{Var}(X).
\] (16)

In particular, we have \( f_{yy}(x_0, y_0) \) \( \text{Var}(Y) = O(1/n) \). Finally, as \( f(x_0, y_0) = 0 \), we have shown that
\[
\mathbb{E}[MCC] = 0 + O \left( \frac{1}{n^2} \right).
\] (17)

**Expected Positive \( F_1 (\mathbb{E}[PF_1]) \)** The positive \( F_1 \) score \( (PF_1) \) can be written as \( f(X, Y) \) for \( f(x, y) = 2x/(n_1 + x + y) \). Check that
\[
f_{xx}(x, y) = \frac{-4(n_1 + y)}{(n_1 + x + y)^2}, \quad f_{yy}(x, y) = \frac{4x}{(n_1 + x + y)^3}.
\] (18)

We leave it to the reader to check that \( f_{xxx}(x_0, y_0) = O(1/n^3) \) and \( f_{yyy}(x_0, y_0) = O(1/n^3) \). It follows that
\[
\mathbb{E}[PF_1] = \frac{2n_1g}{n_1 + ng} - \frac{2(n_1 + n_2g)}{(n_1 + ng)^3} \cdot n_1g(1 - g) + \frac{2n_1g}{(n_1 + ng)^3} \cdot n_2g(1 - g) + O \left( \frac{1}{n^2} \right)
\]
\[
= \frac{2n_1g}{n_1 + ng} - \frac{2n_1^2g(1 - g)}{(n_1 + ng)^3} + O \left( \frac{1}{n^2} \right)
\]
\[
= 2n_1g \left( \frac{1}{n_1 + ng} - \frac{n_1(1 - g)}{(n_1 + ng)^3} \right) + O \left( \frac{1}{n^2} \right)
\]
\[
= 2n_1g \left( \frac{1}{\alpha + g} - \frac{\alpha(1 - g)}{n(\alpha + g)^3} \right) + O \left( \frac{1}{n^2} \right).
\] (19)

Notice that \( \mathbb{E}[PF_1(X, Y)] - PF_1(\mathbb{E}[X], \mathbb{E}[Y]) = O(1/n) \) and not \( O(1/n^2) \). Moreover, the difference is strictly negative.
Expected Negative \( F_1 \) (\( E[\mathcal{N}F_1] \)) The approximation of the expectation of the negative \( F_1 \) score (\( \mathcal{N}F_1 \)) can be obtained from that of \( PF_1 \) by first swapping \( n_1 \) and \( n_2 \) and replacing \( g \) by \( 1 - g \). In particular, we find

\[
E[\mathcal{N}F_1] = 2(1 - \alpha)(1 - g)\left(\frac{1}{2 - \alpha - g} - \frac{(1 - \alpha)g}{n(2 - \alpha - g)^2}\right) + O\left(\frac{1}{n^2}\right), \quad (20)
\]

Again, notice that \( E[\mathcal{N}F_1(X, Y)] - \mathcal{N}F_1(E[X], E[Y]) = O(1/n) \) and not \( O(1/n^2) \), and that the difference is strictly negative.

A.2 Expected value of balanced metrics

This appendix contains the derivations of the approximations of the expected values of the balanced performance metrics, as presented in Table 4 of the main text. The derivations are similar to those in Appendix A.1, although the outcomes are slightly different.

Expected balanced Positive Predictive Value (\( E[PPV^b] \)) The balanced positive predictive value \( PPV^b \) can be written as \( f(X, Y) \) with \( f(x, y) = (x/n_1)/(x/n_1 + y/n_2) \). Check that

\[
f_{xx}(x, y) = \frac{-2y/n_2}{n_1^2(x/n_1 + y/n_2)^3}, \quad f_{yy}(x_0, y_0) = \frac{2x/n_1}{n_2^2(x/n_1 + y/n_2)^3}. \quad (21)
\]

It follows that

\[
E[PPV^b] = \frac{1}{2} - \frac{n_1g^2(1-g)}{n_1^2(2g)^3} + \frac{n_2g^2(1-g)}{n_2^2(2g)^3} + O\left(\frac{1}{n^2}\right)
\]

\[
= \frac{1}{2} + \frac{(n_1 - n_2)(1-g)}{8n_1n_2g} + O\left(\frac{1}{n^2}\right)
\]

\[
= \frac{1}{2} + \frac{\delta(1-g)}{2n(1+\delta)(1-\delta)g} + O\left(\frac{1}{n^2}\right). \quad (22)
\]

Observe that \( E[PPV^b(X, Y)] - PPV^b(E[X], E[Y]) = O(1/n) \), in contrast to \( E[PPV(X, Y)] - PPV(E[X], E[Y]) = O(1/n^2) \). However, the absolute value of the term of order \( 1/n \) can be bounded from above by \( (1-g)/(8g) \).

Expected balanced Negative Predictive Value (\( E[NPV^b] \)) The balanced negative predictive value \( NPV^b \) can be viewed as the balanced positive predictive value for the negative class, i.e., to compute \( NPV^b \) we first swap the roles of \( n_1 \) and \( n_2 \) and replace \( g \) by \( 1 - g \) and then compute \( PPV^b \). It follows that

\[
E[NPV^b] = \frac{1}{2} - \frac{\delta g}{2n(1+\delta)(1-\delta)(1-g)} + O\left(\frac{1}{n^2}\right). \quad (23)
\]

Again, observe that \( E[NPV^b(X, Y)] - NPV^b(E[X], E[Y]) = O(1/n) \), in contrast to \( E[NPV(X, Y)] - NPV(E[X], E[Y]) = O(1/n^2) \). However, the absolute value of the term of order \( 1/n \) can be bounded from above by \( g/(8(1-g)) \).
Expected balanced Markedness \((E[MRK^b])\)  The expectation of the balanced markedness \((MRK^b)\) can be approximated as follows:

\[
E[MRK^b] = E[PPV^b] + E[NPV^b] - 1 = 1 + \frac{\delta(1 - g)}{2n(1 + \delta)(1 - \delta)g} + 1 - \frac{\delta g}{2n(1 + \delta)(1 - \delta)(1 - g)} - 1 + O\left(\frac{1}{n^2}\right)
\]

It shows that \(E[MRK^b(X,Y)] - MRK^b(E[X],E[Y]) = O(1/n)\), in contrast to \(E[MRK(X,Y)] - MRK(E[X],E[Y]) = O(1/n^2)\). However, if \(g = \frac{1}{2}\), then the term of order \(1/n\) is zero. If \(g \neq \frac{1}{2}\), then the absolute value of the term of order \(1/n\) can be bounded from above by \((1 - 2g)/(8g(1 - g))\).

Expected balanced Matthews Correlation Coefficient \((E[MCC^b])\)  The balanced Matthews Correlation Coefficient \((MCC^b)\) can be written as \(f(X,Y)\) in which

\[
f(x,y) = \frac{x/n_1 - y/n_2}{\sqrt{(x/n_1 + y/n_2)(2 - x/n_1 - y/n_1)}}
\]

Introducing the function \(D(x,y) = 2(x/n_1 + y/n_2) - (x/n_1 + y/n_2)^2\), the above simplifies to

\[
f = (x/n_1 - y/n_2)D^{-\frac{1}{2}}.
\]

The first order partial derivatives of \(D\) are equal to \(2/n_1 \cdot (1 - x/n_1 - y/n_2)\). The identity \(x_0/n_1 - y_0/n_2 = 0\) then implies that only the following term remains in \(f_{xx}(x_0, y_0)\):

\[
f_{xx}(x_0, y_0) = 2 \cdot (1/n_1) \cdot (-\frac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0, y_0) \cdot 2/n_1 \cdot (1 - x_0/n_1 - y_0/n_2)
\]

Similarly, we obtain

\[
f_{yy}(x_0, y_0) = 2 \cdot (-1/n_2) \cdot (-\frac{1}{2}) \cdot D^{-\frac{3}{2}}(x_0, y_0) \cdot 2/n_2 \cdot (1 - x_0/n_1 - y_0/n_2)
\]

Finally, as \(f(x_0, y_0) = 0\), it follows that

\[
E[MCC^b] = \frac{-(1 - 2g)n_1 g(1 - g)}{8n_1^2 \sqrt{g^3(1 - g)^3}} + \frac{(1 - 2g)n_2 g(1 - g)}{8n_2^2 \sqrt{g^3(1 - g)^3}} + O\left(\frac{1}{n^2}\right)
\]

\[
= \frac{(n_1 - n_2)(1 - 2g)}{8n_1 n_2 \sqrt{g(1 - g)}} + O\left(\frac{1}{n^2}\right)
\]

\[
= \frac{\delta(1 - 2g)}{2n(1 + \delta)(1 - \delta)\sqrt{g(1 - g)}} + O\left(\frac{1}{n^2}\right).
\]
Once again, observe that $E[MCC^b(X, Y)] - MCC^b(E[X], E[Y]) = O(1/n)$, in contrast to $E[MCC(X, Y)] - MCC(E[X], E[Y]) = O(1/n^2)$. However, if $g = \frac{1}{2}$, then the term of order $1/n$ is zero. If $g \neq \frac{1}{2}$, then the absolute value of the term of order $1/n$ can be bounded from above by $(1 - 2g)/(8\sqrt{g(1-g)})$.

**Expected balanced Positive $F_1$ ($E[PF_1^b]$)** The balanced positive $F_1$ score ($PF_1^b$) can be written as $f(X, Y)$ for $f(x, y) = (2x/n_1)/(x/n_1 + y/n_2 + 1)$. Check that

$$f_{xx}(x, y) = \frac{-4(y/n_2 + 1)}{n_1^2(x/n_1 + y/n_2 + 1)^3}, \quad f_{yy}(x, y) = \frac{4x/n_1}{n_2^2(x/n_1 + y/n_2 + 1)^3}.$$  

(30)

It follows that

$$E[PF_1^b] = \frac{2g}{1 + 2g} - \frac{2n_1(1 + g)g(1 - g)}{n_1^2(1 + 2g)^3} + \frac{2n_2g^2(1 - g)}{n_2^2(1 + 2g)^3} + O\left(\frac{1}{n^2}\right)$$

$$= 2g \left(\frac{1}{1 + 2g} - \frac{n_2 - (n_1 - n_2)g}{n_1n_2(1 + 2g)^3}\right) + O\left(\frac{1}{n^2}\right)$$

$$= 2g \left(\frac{1}{1 + 2g} - \frac{2(1 - \delta(1 + 2g))(1 - g)}{n(1 + \delta)(1 - \delta)(1 + 2g)^3}\right) + O\left(\frac{1}{n^2}\right)$$

(31)

The term of order $1/n$ is bounded from above by $2g^2(1 - g)/(2g + 1)^3$, which is at most $8/243 \approx 0.033$ at $g = 2/5$. Moreover, it is bounded from below by $-2g(1 - g^2)/(2g + 1)^3$, which is at least $-4/243 \cdot (7\sqrt{7} - 10) \approx -0.14$ at $g = (\sqrt{7} - 2)/3 \approx 0.22$.

**Expected balanced Negative $F_1$ ($E[NF_1^b]$)** The approximation of the expectation of the balanced negative $F_1$ score ($NF_1^b$) can be obtained from that of $PF_1^b$ by first swapping $n_1$ and $n_2$ and replacing $g$ by $1 - g$. In particular, we find

$$E[NF_1^b] = 2(1 - g) \left(\frac{1}{3 - 2g} - \frac{2(1 + \delta)(3 - 2g)g}{n(1 + \delta)(1 - \delta)(3 - 2g)^3}\right) + O\left(\frac{1}{n^2}\right),$$

(32)

The term of order $1/n$ is bounded from above by $2g(1 - g^2)/(2(1 - g) + 1)^3$, which is at most $8/243 \approx 0.044$ at $g = 3/5$, and bounded from below by $-2(1 - g)(1 - (1 - g)^2)/(2(1 - g) + 1)^3$, which is at least $-4/243 \cdot (7\sqrt{7} - 10) \approx -0.14$ at $g = (5 - \sqrt{7})/3 \approx 0.78$.

**B Appendix - Performance of fictitious binary classifier**

This appendix shows how a fictitious binary classifier with $TPR = 0.6$ and $TNR = 0.8$ performs on accuracy (Figs. 2 and 3) and $F_1$ (Figs. 1 and 4 when
it is min-max normalized with the expected value when randomly guessing the positive class with probability $g = 0.5$ (Figs. 2 and 1) or $g = \alpha$ (Figs. 3 and 4), as a function of imbalance coefficient $\delta$. Shown are imbalanced metrics (left panels) and balanced metrics (right panels), which have been corrected for bias due to class imbalance.

When $g^{\text{unif}} = \frac{1}{2}$ is chosen as control, $\mathbb{E}[ACC(g^{\text{prop}})] = \frac{1}{2}$ (Fig. 2, left panel, blue line). When $g^{\text{prop}} = \alpha$ is chosen as control, $\mathbb{E}[ACC(g^{\text{prop}})]$ is a quadratic function (Fig. 3, left panel, blue line; see Table 2). As a result, the classifier is outperformed ($ACC^{\text{mmn}}(g^{\text{prop}}) < 0$) by this strategy when imbalance is large (here when $\delta < -\frac{1 - \sqrt{41}}{10} \approx -0.74$ or $\delta > -\frac{1 + \sqrt{41}}{10} \approx 0.54$).

**Fig. 2.** Accuracy of a fictitious binary classifier in relation to imbalance coefficient $\delta$. $g = 0.5$, $TPR = 0.6$, $TNR = 0.8$, $n = 1000$.

Before min-max normalization, the classifier scores equally well on $PF_1$ and $NF_1$ at $\delta = \frac{1}{7}$ (Fig. 4, small white points). After min-max normalization using $g^{\text{prop}}$, however, the classifier scores equally well on $PF_{1}^{\text{mmn}}$ and $NF_{1}^{\text{mmn}}$ at $\delta = -\frac{1}{3}$ (large white points). For $\delta < -\frac{1}{3}$ and $\delta > \frac{1}{3}$, the regular $F_1$ and normalized $F_{1}^{\text{mmn}}(g^{\text{prop}})$ disagree on whether the model performs better on the positive or the negative class.

By definition, the balanced $F_1$ is insensitive to class imbalance. After min-max normalization with $g^{\text{prop}}$, however, it is sensitive again to class imbalance. The larger the imbalance coefficient, the lower the classifier scores on $PF_{1}^{\text{mmn,b}}(g^{\text{prop}})$ and the higher on $NF_{1}^{\text{mmn,b}}(g^{\text{prop}})$ (Fig. 4, right panels).
Fig. 3. Accuracy of a fictitious binary classifier in relation to imbalance coefficient $\delta$. $g = \alpha$, $TPR = 0.6$, $TNR = 0.8$, $n = 1000$.

Fig. 4. $F_1$ of a fictitious binary classifier in relation to imbalance coefficient $\delta$. $g = \alpha$, $TPR = 0.6$, $TNR = 0.8$, $n = 1000$. White points show where positive and negative $F_1$ intersect.