G-AUC: An improved metric for classification model selection

Shashank Sadafule
Department of Physics
IIT Kharagpur,
Kharagpur, WB-721302, India.
Email: shashanksadafule25@gmail.com

Sobhan Sarkar
Information Systems & Business Analytics
IIM Ranchi,
Ranchi, Jharkhand-834008, India.
Email: sobhan.sarkar@iimranchi.ac.in

Shaomin Wu
Business/Applied Statistics,
University of Kent,
Canterbury-CT2 7FS, UK.
Email: s.m.wu@kent.ac.uk

***Abstract*—The performance of classification models is often measured using the metric, area under the curve (AUC). The non-parametric estimate of this metric only considers the ranks of the test instances and fails to consider the predicted scores of the model. Consequently, not all the valuable information about the model's output is utilized. To address this issue, the present paper introduces a new metric, called Gamma AUC (G-AUC) that can take into account both ranks as well as scores. The parameter G tackles the problem of overfitting scores into the metric. To validate the proposed metric, we tested it on 20 UCI datasets with 10 state-of-the-art models. Out of all the values of the parameter G that we tested, four of them got p-value less than 0.05 for the alternative hypothesis that, on the training sets, G-AUC has a greater correlation than AUC itself, with AUC on test sets. Furthermore, for all values of G considered, G-AUC always won majority of the times than AUC for selecting better models.**

***Keywords*—Classification, Performance metric, AUC, G-AUG.**

I. Introduction

A machine learning task in which the given input is classified into discrete classes is called classification. The number of classes could be two or more. For this paper, we will focus only on binary classification. An example of a binary classification task would be classifying images into 'cat' and 'dog'. In this case, the image is an input, and 'cat', and 'dog' are two classes into which it is distinguished. Many machine learning models could be used for the task of binary classification. Examples of these models include Support Vector Machines (SVM), Logistic Regression, Naive Bayes Classifier, and Neural Networks. The prediction that the above-mentioned models could give on a particular input could either be a positive or negative class, or a score corresponding to the probability that an instance could belong to either one of the classes. In this paper, having chosen the positive/negative class, we assume that a score corresponds to the probability that a given instance is predicted to have a positive class.

Different trained models do not always perform the same even after keeping the relevant parameters constant. Hence, it becomes necessary to build a metric that could estimate the performance of these trained models. Many metrics exist in the literature such as accuracy, recall, precision, and log-loss for estimating the performance of the binary classification model. But one of the shortcomings of these metrics is that they are quite vulnerable to the changes in class distribution. They do not perform the same since data in the training dataset and that in the test dataset are different. However, the Receiver Operating characteristic (ROC) curve has a unique property. It does not change if the class distribution in a test set changes, however, the underlying class-conditional distributions from which the data is obtained should remain intact.

The Receiver Operating Characteristic (ROC) is the curve showing the relationship between the model's true positive rate and its false positive rate. AUC, which is the area under the ROC curve, is a comprehensive measure of the model's performance taking into account all the possible thresholds used for classification. In short, it gives a summary of the ROC curve. If $AUC = 1$, it implies that the classifier is perfectly able to distinguish between negative and positive instances for all possible thresholds and if $AUC = 0$, it implies that the classifier predicts all positive instances as negative and vice versa for all possible thresholds. One of the ways of interpreting this metric is that it is the likelihood that a random positive instance will be ranked higher in score than a random negative instance.

The non-parametric version of AUC is broadly used within data mining as well as the machine learning community It is calculated by taking into account only the ranks that the model predicts based on its predicted scores. The non-dependency on any distribution assumption is one of its advantages. But the disadvantage is that the scores are ignored and only considered when ranking instances. Hence, when the scores changes and the ranking order remain the same, the AUC does not change. In this way, AUC fails to use this additional information on scores that the model predicts. In this paper, we investigate the parametric version of AUC, the Gamma AUC (G-AUC), which takes into account both ranks as well as scores of the instances, unlike AUC which only considers ranks. We have introduced a new parameter G which divides the unit interval into G equal parts. We propose that this metric could be used for model selection for the task of binary classification.

The rest of the paper is structured as follows: Relevant literature in the domain are reviewed in Section II. Section III introduces the proposed metric G-AUC along with the algorithm to calculate it. The methodology is also described to test the validity of the metric. In Section IV, the data sets and experimental setup used for analysis are discussed. Section V presents the experimental results for using G-AUC as metric

978-1-6654-9198-3/22/$31.00 ©2022 IEEE

for model selection. The main conclusions of the study and further scope for improvement are discussed in Section VI.

II. Related works

To evaluate how the classification model performs on the test dataset, various metrics like *accuracy* , *Brier score*, and *AUC* are used [1]. Accuracy evaluates classification performance of the classifier by measuring total percentage of test instances that are classified correctly. But it very sensitive to imbalances in classes in a dataset. On the other hand, brier score only measures the probability estimation performance of the model. Other metrics like sensitivity, specificity, precision and recall perform well on skewed datasets. But all of these metrics consider only three out of four values in the confusion matrix, completely ignoring one of the important pieces of information. This issue is tackled by the Mathew Correlation Coeffiecient (MCC) which measures correlation between true class and predicted class of test instances.

However, emperical comparison of AUC against MCC is done by [2] on both balanced and skewed datasets based on criteria of degree of consistensy and degree of discriminancy. They found that both the metrics are statistically consistent but AUC is more discriminant than MCC implying that AUC is a better metric among the two for evaluating binary classifiers. It is also shown by [3], [4] that non-parametric AUC is better than accuracy for measuring the performance of the classification model.

In the medical community, particularly in radiology [5], the ROC curve have been used for a long time. Further improvements for using it in machine learning models have been made by the introduction of the ROC Convex Hull (ROCCH) method proposed by [6], [7]. AUC, the area under ROC curve, can be calculated under non-parametric [8], semi-parametric [9], and parametric assumptions [10]. Within data mining as well as the machine learning community, the non-parametric one is highly used out of above-mentioned versions. It evaluates the likelihood that a random positive instance will be ranked higher in score than a random negative instance. Mann–Whitney–Wilcoxon statistic test of ranks is also similar to this metric [8].

A handful of AUC variations have been developed that incorporate the score differences between positive and negative instances. [11] developed pAUC that incorporated predicted probabilities. sAUC is another variant that considers scores as well as ranks [12] and softAUC is AUC's approximation which is differentiable. But [13] argues through critical analysis that none of above-mentioned variants are as effective as AUC with regards to model selection and evaluation. They also showed that these variants are less favorable toward models that generate small score differences between positive and negative instances. It is also possible to apply AUC to multi-class classification problems [14]. Furthermore, [15] extended the defination of sAUC and pAUC for application in multi-class classification problems.

A. Research issues

Although AUC takes into account the ranks of the test instances, one of its disadvantages is that it only considers their scores when ranking instances, losing some information that could be provided by them. On the contrary, the sAUC takes an entire score of the instance under consideration into account. This leads the scores to overfit the metric.

B. Research contribution

In order to address the above-mentioned issues, we have proposed a new classification performance metric, called Gamma-AUC (G-AUC). It can take into account both ranks and scores, however, it is developed to perform in a more sophisticated way such that it does not overfit by taking into account the entire score because of its parameter G.

III. Proposed Methodology

This section provides a step-by-step approach to the development of G-AUC and the methodology used to test the validity of the metric. Since G-AUC is a modified version of AUC with the introduction of a parameter G, it becomes necessary to gain a good understanding of how AUC is calculated.

A. AUC

To calculate the AUC of a trained model, divide the dataset's negative and positive instances. Denote the total count of negative and positive instances by n and m respectively. Let $\{x_1, x_2, ..., x_n\}$ and $\{y_1, y_2, ...y_m\}$ be the model's predicted scores for negative and positive instances. Then AUC could be calculated using (1) [16].

$$\hat{\theta} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \Psi_{ij} \tag{1}$$

where
Ψ_{ij} = 1 if $(y_i - x_j) > 0$. Else it is set as 0. In this way, AUC is the normalized count of the total number of negative and positive pairs in which the positive instance has a greater score than the negative one.

The algorithm for calculating AUC is provided in [16]. In this algorithm, we provide as input the merged sequence of negative and positive instances, sorted in the decreasing order of their predicted scores. The algorithm transverses through each instance and if the positive instance is encountered, the variable c is incremented by 1 or else the variable AUC is incremented by c. In the end, the variable AUC is divided by the product of m and n to get the final value.

B. G-AUC: The proposed metric

The new metric G-AUC is a modified version of the original AUC with the introduction of a new multiplicative factor **g**. The new parameter G divides the unit interval into G equal parts. For a particular instance with score s, g takes the value of the interval's upper bound in which the score s lies after the unit interval is divided into G equal parts. For example, if we set $G = 2$, then it divides the interval [0,1] into 2 equal parts, [0,0.5) and [0.5,1]. Now, if an instance with score s is encountered and it lies between [0,0.5) i.e., within the first interval, then g takes the value of this interval's upper bound which is 0.5; otherwise, it takes the value of second interval's upper bound which is 1.

To calculate G-AUC, we denote the total count of negative and positive instances in the set by n and m respectively. Let $\{x_1, x_2, ..., x_n\}$ and $\{y_1, y_2, ...y_m\}$ be the model's predicted scores for negative and positive instances respectively. Then G-AUC is calculated using (2).

$$G - AUC = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} g_i \Psi_{ij} \tag{2}$$

where $g_i = \frac{n}{G}$ such that the score s of instance lies in the n^{th} interval produced by the parameter G. $\Psi_{ij} = 1$ if $(y_i - x_j) > 0$. Else it is set as 0.

The algorithm for calculating G-AUC is a modification of the algorithm for calculating AUC. For input, we provide the parameter G and the merged sequence of positive and negative instances along with their scores, sorted in decreasing order of the scores. The algorithm transverses through each instance and if the positive instance is encountered, instead of adding 1 in the variable c as it is done in the AUC algorithm, it adds the interval's upper bound where the score of that positive instance lies after the unit interval is divided into G equal parts. We find that interval using the while loop in the algorithm.

Algorithm 1 G-AUC

Input: parameter G and a sequence of actual class of test instances along with their predicted scores s, sorted in decreasing order of s

Output: G-AUC of the model

```
Initialisation: θ ← 0, c ← 0, i ← 0
for each following instance in the sequence do
    if instance is positive then
        while i < G do
            if s > i/N and s <= (i+1)/N then
                c = c + (i+1)/N
                break
            end if
            i ← i + 1
        end while
    else
        θ ← θ + c
    end if
end for
G − AUC = θ/(mn)
```

For testing the validity of G-AUC, we have divided our research analysis into two parts. In the first part, we explore the nature of G-AUC as the parameter G increases. We plot the values of G-AUCs with increasing G for different datasets and models. For this task, we have trained four state-of-the-art models namely Random Forest Classifier, Naive Bayes, Logistic Regression, and the Support Vector Machine (SVM) on six data sets used from UCI repository [17]. The data sets used are Breast Cancer, Tic Tac Toe, House Votes, WDBC, Mushrooms, and Kr vs Kp. We execute the first part of our analysis in the following manner:

- *Step 1*: Select a particular data set (say House Votes) and train a particular model (say SVM) on that data set.
- *Step 2*: Get two separate lists of the model's predicted score and actual class on that data set.
- *Step 3*: Sort both lists in descending order of the score.
- *Step 4*: Use algorithm 1 to get values of G-AUC's with G ranging from 2 to 100.
- *Step 5*: Repeat steps 1 to 4 for all the other data sets and plot on the same graph.
- *Step 6*: Repeat steps 1 to 5 for all the other models mentioned above.

For the second part, we test if G-AUC outperforms AUC for selection of better models having higher AUC value on the test sets. We also analyze the values of G which are optimal for model selection. Let S_1 be the Spearman's rank correlation between AUCs on training set and AUCs on test set and S_2 be that between G-AUCs on training set and AUCs on test set. If S_2 is greater than S_1, then that would count as a win for G-AUC because it implies that G-AUC has outperformed AUC for selecting models with higher AUC on the test set. We have used Spearman's rank correlation for our task because it works for a monotonic relationship rather than just a linear relationship, as is the case with Pearson correlation. We train 10 state-of-the-art models for this task, namely Random Forest Classifier, Logistic Regression, Decision Tree, Bernoulli Naive Bayes, Multi-Layer Perceptron, Support Vector Classifier, Nu-Support Vector Classifier, Ada Boost Classifier, Quadratic Determinant Analysis, and Gaussian Naive Bayes. The datasets used are listed in Table I. We execute this task in the following manner:

- *Step 1*: Select a particular data set (say House Votes) and divide it into 80% training set and 20% testing set.
- *Step 2*: Select a particular model (say Logistic Regression) and train that model on the training set.
- *Step 3*: Get four separate lists of the model's predicted score and actual class on a training set and a test set.
- *Step 4*: Sort all the lists in descending order of their respective score on the training/test set.
- *Step 5*: For a particular value of G, calculate 3 metrics, i) G-AUC on a training set, ii) AUC on a training set, and iii) AUC on a test set, with the help of algorithm provided in [16] for calculating AUC and algorithm 1 for calculating G-AUC using required set's actual class and predicted scores as the input.
- *Step 6:* Repeat step 1 through step 5 for all 10 models mentioned above. Now we get 3 lists of the metrics mentioned in this step.
- *Step 7*: Get S_1, the spearman's rank correlation between the list of AUCs on training set and the list of AUCs on test set and S_2, the spearman's rank correlation between the list of G-AUCs on training set and list of AUCs on test set.
- *Step 8*: Repeat step 1 through 7 for all the datasets mentioned in Table I. For 20 datasets, we get 20 pairs of (S_1, S_2).
- *Step 9*: With a level of confidence of 0.05, use paired t-test to test the alternative hypothesis $S_2 > S_1$ and get the corresponding p-value.
- *Step 10*: Repeat step 1 through 9 for G ranging from 2 to 20.

We have analyzed G-AUC for the values of G ranging from 2 to 20 because it is within this range that G-AUC shows maximum variation in its value for majority of the models. This is discussed in Section V. If for any value of G, we get

the p-value to be less than the level of confidence we conclude that the particular G value is optimal for G-AUC for selecting models having higher AUC on the test set.

IV. Experiments and Data

The analysis has been done on a computer with 8 GB Ram and Intel Core i5-9300H CPU at 2.40GHz with 4 Core(s) and 8 Logical Processor(s). The programming language used is Python using different libraries namely sci-kit learn [18], Pandas, NumPy, and Matplotlib. In our experiment, twenty datasets from the UCI repository are selected for our analysis. Their names, number of features, size, and percentage of the majority class present are listed in Table I.

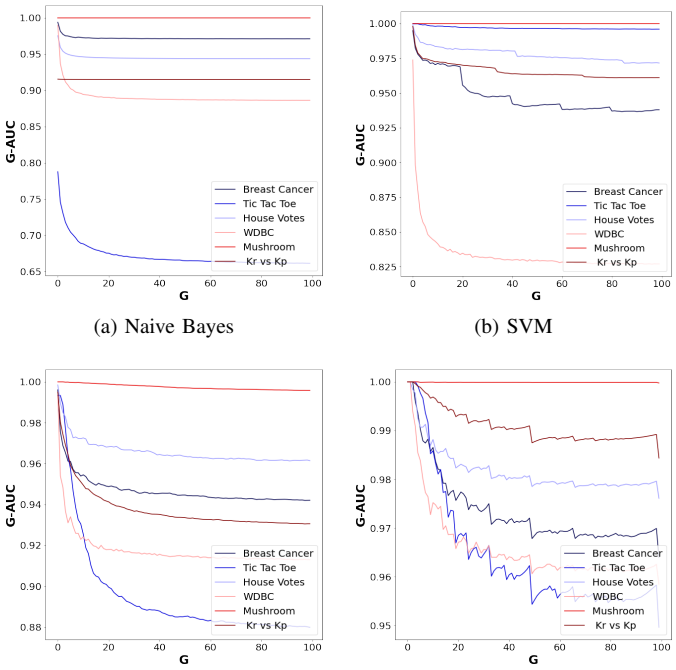
TABLE I: Datasets used in our analysis

	Data sets	Features	Size	Majority Class
1	Breast Cancer	9	683	65.01
2	Blood Transfusion	4	748	76.20
3	Mamographic Mass	5	830	51.45
4	Vertebral Column	6	310	67.74
5	Qualitative Bankruptcy	18	250	57.20
6	MONK-2	6	601	65.72
7	Somerville Happiness Survey	6	143	53.85
8	Raisin	7	900	50.00
9	Heart Failure Clinical Records	12	299	67.89
10	Statlog (Heart)	13	270	55.56
11	Statlog - Australian Credit Approval	14	690	55.51
12	Parkinsons	22	195	75.38
13	Ionosphere	34	351	64.10
14	Musk	166	476	56.51
15	SPECT	22	267	79.40
16	Tic Tac Toe	27	958	65.34
17	House Votes	48	435	61.38
18	Divorce Prediction	54	170	50.59
19	Connectionist Bench	60	208	53.37
20	Breast Cancer Coimbra	9	116	55.17

V. Results & Discussions

In the first part, we have explored the nature of G-AUC as G increases. We have plotted the values of G-AUCs with increasing G for different datasets. We selected a particular model and trained that model on six datasets. Then on the training set, we got the values of G-AUCs with G ranging from 2 to 100. We have shown the results for different models in Fig. 1.

We get two observations from Fig 1. The first observation is that the value of G-AUC decreases as G increases. The decrement in greater for initial values of G, hence G-AUC shows maximum variation in this range. This is expected because, for greater G, the unit interval [0,1] is divided into finer parts which lead to the addition of lower values of g in G-AUC. Consider the following example, for $G = 2$, the unit interval is divided into 2 parts i.e. [0,0.5], (0.5,1] and for $G = 4$, the unit interval is divided into 4 parts i.e. [0,0.25], (0.25,0.5], (0.5,0.75] and (0.75,1]. If we encounter the positive score $s = 0.7$, g will take up value 1 for $G = 2$ and 0.75 for $G = 4$ which is lower. Similarly for other scores, the value of g would be lower as G increases, finally leading to a lower value of G-AUC. It could also be seen that the metric varies in a different manner for different models for increasing values of G. Secondly, we could see that the value of G-AUC reaches a constant value as G increases. This value is different for different models. This happens because as G increases,



(a) Naive Bayes (b) SVM

(c) Logistic Regression (d) Random Forest

Fig. 1: Nature of G-AUC on different models for increasing values of the parameter G

g begins to take similar values as the unit interval is finely divided. In this way, it does not make much of a difference.

Table II shows that for G-AUC with $G = 10$, we get 15 significant wins out of 20 over the original AUC for the selection of better models. This is because for those 15 datasets, the Spearman's rank correlation between G-AUCs on training set and AUCs on test set (S_2) is greater than that between AUCs on training set and AUCs on test set (S_1). The negative results of the rest of the 5 datasets might be

TABLE II: List of Spearman's rank coefficients for different datasets for $G = 10$

	Name	S_1	S_2	Win
1	Breast Cancer	0.311	0.614	1
2	Blood Transfusion	0.321	0.624	1
3	Mamographic Mass	0.042	0.576	1
4	Vertebral Column	0.073	0.576	1
5	Qualitative Bankruptcy	0.745	0.467	0
6	MONK-2	0.900	0.467	0
7	Somerville Happiness Survey	0.115	0.188	1
8	Raisin	0.508	0.613	1
9	Heart Failure Clinical Records	0.705	0.733	1
10	Statlog (Heart)	0.439	0.693	1
11	Statlog - Australian Credit Approval	0.748	0.600	0
12	Parkinsons	0.610	0.110	0
13	Ionosphere	0.418	0.529	1
14	Musk	0.089	0.389	1
15	SPECT	0.055	0.479	1
16	Tic Tac Toe	0.791	0.539	0
17	House Votes	0.308	0.675	1
18	Divorce Prediction	-0.059	0.455	1
19	Connectionist Bench	0.537	0.794	1
20	Breast Cancer Coimbra	0.546	0.636	1

TABLE III: Results for paired t-tests for different values of parameter G with the alternative hypothesis $S_2 > S_1$.

G	wins	p-value	G	wins	p-value
2	13	0.084	12	14	0.07
3	13	0.103	13	15	**0.039**
4	15	0.107	14	14	**0.039**
5	14	0.086	15	13	0.106
6	14	0.065	16	13	0.088
7	15	**0.049**	17	14	0.096
8	14	0.054	18	15	0.055
9	13	0.188	19	13	0.111
10	15	**0.041**	20	13	0.074
11	14	0.108	-	-	-

because of the variability lying within the attributes of those datasets. If we perform a paired t-test on (S_1, S_2) with the confidence interval of 0.05 and alternative hypothesis $S_2 > S_1$, we get the p-value of 0.041. This means we could reject the null hypothesis $S_1 = S_2$ and accept the alternative hypothesis $S_2 > S_1$. Similarly, paired t-tests were done on G-AUC with parameters G ranging from 2 to 20 and the results are displayed in Table III.

In Table III, it could be seen that for parameter G taking values 7, 10, 13, and 14, we get p-value less than the level of confidence of 0.05 implying that the alternative hypothesis $S_2 > S_1$ is statistically true for these values. Additionally, G-AUC wins a majority of the time for selection of better model than AUC itself for all values of G considered. We believe that these are good results and further investigation on the metric could be done in the future.

VI. Conclusions

The performance of the classification model at every possible classification threshold is shown by the ROC curve. AUC is the area under this curve which is a metric used on a test set that measures the overall performance of the classification model. To select better models, that give a higher value of AUC on the test set, [12] introduced scored AUC or sAUC. It has an added advantage in considering ranks and predicted scores. However, the scores overfit this metric.

In this paper, we introduced G-AUC which has the potential to be used as a metric to select models with higher AUC on the test set. It considers ranks and scores predicted by the model, addressing the issue of overfitting of the scores. The new parameter G divides the unit interval into G equal parts to avoid overfitting of scores in the metric. For every pair of positive and negative instance in which positive instance has greater score than negative one, the interval's upper bound in which the score of positive instance lie is added in G-AUC. We observe that the value of G-AUC decreases and reaches a constant value as the parameter G increases. This happens because as G increases the unit interval is divided into finer parts, leading to the addition of lower valued upper bounds in G-AUC. The metric is also evaluated on 20 UCI datasets and it is found to win a majority of times over AUC itself for the selection of the models with higher values of AUC on the test set, thereby selecting better models.

However, G-AUC has a few limitations. It is proposed only for binary classification problems. The implications of G-AUC on skewed dataset have been not tested. The size of the dataset for which the metric works best could also be further investigated. In future studies, this analysis could be expanded to multi-class classification too.

References

[1] M. Hossin, M. N. Sulaiman, A review on evaluation metrics for data classification evaluations, International Journal of Data Mining & Knowledge Management Process 5 (2) (2015) 1.

[2] C. Halimu, A. Kasem, S. S. Newaz, Empirical comparison of area under roc curve (auc) and mathew correlation coefficient (mcc) for evaluating machine learning algorithms on imbalanced datasets for binary classification, in: Proceedings of the 3rd international conference on machine learning and soft computing, 2019, pp. 1–6.

[3] J. Huang, C. X. Ling, Using auc and accuracy in evaluating learning algorithms, IEEE Transactions on knowledge and Data Engineering 17 (3) (2005) 299–310.

[4] C. X. Ling, J. Huang, H. Zhang, et al., Auc: a statistically consistent and more discriminating measure than accuracy, in: IJCAI, Vol. 3, 2003, pp. 519–524.

[5] J. A. Swets, Roc analysis applied to the evaluation of medical imaging techniques., Investigative Radiology 14 (2) (1979) 109–121.

[6] F. Provost, T. Fawcett, Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions in: Proceedings of the third international conference on knowledge discovery and data mining (1997).

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

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

[9] F. Hsieh, B. W. Turnbull, Nonparametric and semiparametric estimation of the receiver operating characteristic curve, The Annals of Statistics 24 (1) (1996) 25–40.

[10] X.-H. Zhou, D. K. McClish, N. A. Obuchowski, Statistical Methods in Diagnostic Medicine, John Wiley & Sons, 2009.

[11] C. Ferri, P. Flach, J. Hernández-Orallo, A. Senad, Modifying roc curves to incorporate predicted probabilities, in: Proceedings of the second workshop on ROC analysis in machine learning, Vol. 4140, International Conference on Machine Learning, 2005, pp. 33–40.

[12] S. Wu, P. Flach, A scored auc metric for classifier evaluation and selection, in: Second Workshop on ROC Analysis in ML, Bonn, Germany, 2005.

[13] S. Vanderlooy, E. Hüllermeier, A critical analysis of variants of the auc, Machine Learning 72 (3) (2008) 247–262.

[14] D. J. Hand, R. J. Till, A simple generalisation of the area under the roc curve for multiple class classification problems, Machine learning 45 (2) (2001) 171–186.

[15] B. Van Calster, V. Van Belle, G. Condous, T. Bourne, D. Timmerman, S. Van Huffel, Multi-class auc metrics and weighted alternatives, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1390–1396.

[16] S. Wu, P. Flach, C. Ferri, An improved model selection heuristic for auc, in: European Conference on Machine Learning, Springer, 2007, pp. 478–489.

[17] D. Dua, C. Graff, UCI machine learning repository, [date accessed: 05-2022] (2017).
URL `http://archive.ics.uci.edu/ml`

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.