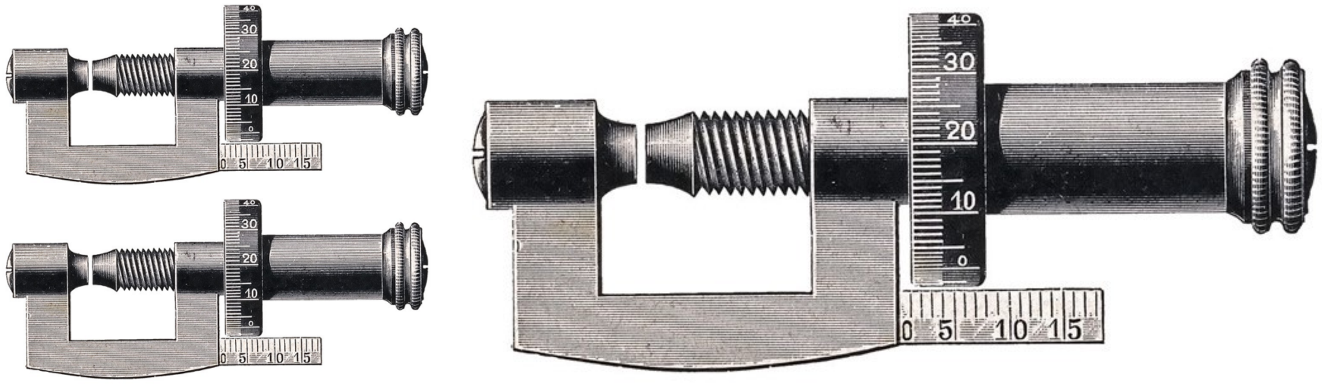
A Tale of Two Macro-F1's

Boaz Shmueli Follow
Aug 19, 2019 · 5 min read ★



After writing my 2-part series **Multi-Class Metrics Made Simple (Part I, Part II)** I received encouraging and useful feedback from readers, including claps, typo corrections, etc. So first, many thanks to all that responded! One email in particular came from a curious reader (who wished to remain anonymous, so I'll refer to this reader as "Enigma") and triggered an investigation into the way the **macro-averaged F1** score is calculated. This led me down a rather surprising rabbit hole, which I describe in this post. The bottom line is: there's more than one macro-F1 score; and data scientists mostly use

As a quick reminder, Part II explains how to calculate the macro-F1 score: it is the average of the per-class F1 scores. In other words, you first compute the per-class precision and recall for all classes, then combine these pairs to compute the per-class F1 scores, and finally use the arithmetic mean of these per-class F1-scores as the macro-F1 score. In the example in Part II, the F1 scores for classes Cat, Fish and Hen are 42.1%, 30.8%, and 66.7% respectively, and thus the macro F1-score is:

Macro-F1 = (42.1% + 30.8% + 66.7%) / 3 = 46.5%

But apparently, things are not so simple. In the email, "Enigma" included a reference to a highly-cited paper which defined the macro F1-score in a very different way: first, the macro-averaged precision and macro-averaged recall are calculated. Then, the harmonic mean of these two metrics is calculated as the final macro F1-score. In our example, the macro-precision and macro-recall are:

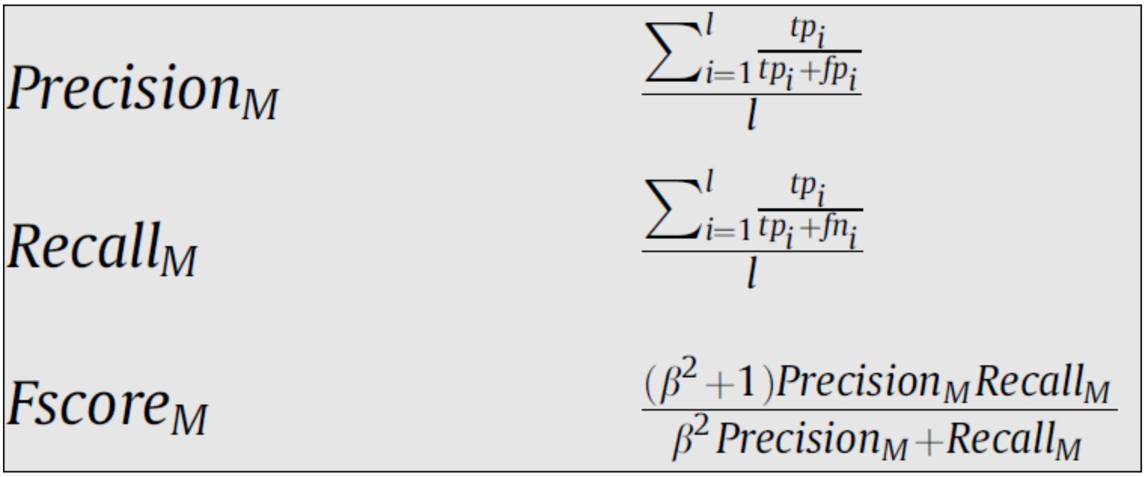
Macro-precision = (31% + 67% + 67%) / 3 = 54.7%

Macro-recall = (67% + 20% + 67%) / 3 = 51.1%

And thus using the second method, which I designate with an asterisk (*):

The reference that “Enigma” sent me is “A systematic analysis of performance measures for classification tasks” by Marina Sokolova and Guy Lapalme [Ref 1]. This paper has around 2000 citations according to Google Scholar.

Here is the definition of the Macro F1-score from the Sokolova paper:



$Precision_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fp_i}}{l}$
$Recall_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fn_i}}{l}$
$Fscore_M$	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M + Recall_M}$

(Note that *l* is the number of classes and *M* stands for Macro. In addition, β is a parameter that can be used to tune the relative importance of precision and recall. β=1 gives equal weights to precision and recall, and is what I used all along in my posts.)

Indeed, as can be seen from the above paper excerpt, the Sokolova paper opts for calculating Macro-F1* and not Macro-F1.

Back to our question: should we use Macro-F1 or Macro-F1*? Both of them attempt at summarizing the per-class precision and recall values into a single number, but in different ways. In the absence of a real domain-specific problem where there is a cost defined for each type of error, summarizing the performance of the model using either metrics provides an easy yet somehow vague way of comparing models. When deploying a real system, these numbers are likely to be unsuitable and you will need to provide your own metric based on the business problem at hand. In this sense, both metrics share the same flaw. The important thing to remember is to be consistent and use the same metric when comparing models.

As to which one is more popular: if history is written by the victors, then — like it or not — metrics are defined by software packages. Python's sklearn library is the most popular machine learning package, and it provides the `sklearn.metrics.f1_score` function, which computes Macro-F1 (and not Macro-F1*). Thus, most data scientists and researchers use the Python library function to compute Macro-F1, usually without giving it a second thought — exactly as I did before "Enigma"'s email landed in my inbox.

. . .

Update (18 December, 2019): I received an email from Juri Opitz, who recently co-authored a paper on this exact topic, with interesting results. Specifically, the authors recommend using Macro-F1 and *not Macro-F1**, as it is less sensitive to error type distribution. I recommend checking out the paper; see [Ref 4] at the end of this post.

References

[Ref 1] Sokolova, Marina, and Guy Lapalme. "A systematic analysis of performance measures for classification tasks." *Information processing & management* 45.4 (2009): 427–437.

[Ref 2] Yang, Yiming, and Xin Liu. "A re-examination of text categorization methods." *SIGIR*. Vol. 99. №8. 1999.

[Ref 3] Lewis, David D., et al. "Training algorithms for linear text classifiers." *SIGIR*. Vol. 96. 1996.

[Ref 3] Lewis, David D., et al. "Training algorithms for linear text classifiers." *SIGIR*. Vol. 96. 1996.

[Ref 4] Opitz, Juri, and Sebastian Burst. "Macro F1 and Macro F1." *arXiv preprint arXiv:1911.03347* (2019).

Machine Learning Measurement Classification Sklearn Data