Evaluation metrics for NLP

Isabel Segura-Bedmar

1 y 2 Julio, Universidad Politécnica de Madrid (UPM)

Binary Classification Confusion Matrix

Prediction outcome

		p	n	total
actual value	p′	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Metrics for binary classification

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy fails for imbalanced classification (distribution of examples in the training dataset across the classes is not equal)

Metrics for binary classification

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Achieving 90 percent classification accuracy, or even 99 percent classification accuracy, may be trivial on an imbalanced classification problem.

Suppose that you are working on an imbalanced dataset with a 1:100 class imbalance, that is, each example of the minority class (class 1) will have a corresponding 100 examples for the majority class (class 0).

In problems of this type, the majority class represents "*normal*" and the minority class represents "*abnormal*," such as a fault, a diagnosis, or a fraud. **Good performance on the minority class will be preferred over good performance on both classes.**

On this problem, a model that predicts the majority class (class 0) for all examples in the test set will have a classification accuracy of 99 percent, mirroring the distribution of major and minor examples expected in the test set on average.

Metrics for binary classification

$$Presicion = \frac{TP}{TP + FP}$$

$$F1_{score} = 2 * \frac{precision * recall}{precision + recall}$$

$$Recall = \frac{TP}{TP + FN}$$

Metrics for multiclassification

Metrics for multi-classification

- Accuracy, precision, recall and F1 can be easily expanded to the multi classification problem.
- We have to calculate these metrics for each class.
- Macro and micro averages allow to combine these metrics for all classes, providing a single score (https://slideplayer.com/slide/6194398/)
- If the dataset is balanced, then macro-average and micro-average will be about the same.

Macro averages

To know how the system performs overall across the sets of data. You should not come up with any specific decision with this average.

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

**where l is the number of classes*

Source: https://towardsdatascience.com/a-tale-of-two-macro-f1s-8811ddcf8f04

Micro averages

$$Recall_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} TP_i + \sum_{i=1}^{M} FN_i}$$

$$Precision_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} TP_i + \sum_{i=1}^{M} FP_i}$$

Source: https://slideplayer.com/slide/6194398/

When to use micro-averaging and macro-averaging scores?

- "Use micro-averaging score when there is a need to weight each instance or prediction equally."
- "Use macro-averaging score when all classes need to be treated equally to evaluate the overall performance of the classifier with regard to the most frequent class labels."
- "Use weighted macro-averaging score in case of class imbalances (different number of instances related to different class labels). The weighted macro-average is calculated by weighting the score of each class label by the number of true instances when calculating the average."