uc3m | Universidad Carlos III de Madrid

HULAT Human Language & Accessibility Technologies

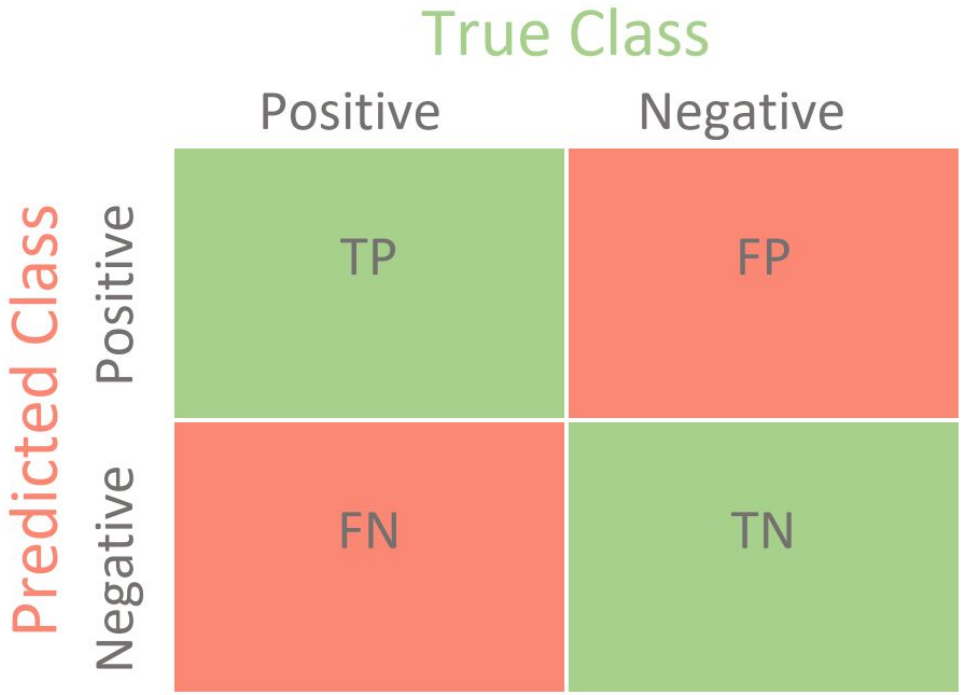
Evaluation metrics for NLP

Isabel Segura-Bedmar

Procesamiento de Lenguaje Natural con Aprendizaje Profundo

Máster en Ciencia y Tecnología Informática

Binary Classification Confusion Matrix



Evaluation 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)

Evaluation metrics for binary classification

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

Suppose that you have to develop a system to predict which COVID patients will die. Suppose that you have a dataset of Covid patients, with a 1:100 class imbalance. This means that for each patient who has died (positive example), there are 100 patients have survived (negative examples). In particular, suppose that you have 20 positive instances and 2000 negative instances

We can propose a very simple classifier that proposes nobody dies. Its confusion matrix is:
TP = 0, TN = 2000, FP=0, FN=20
Accuracy = 2000/2020 = 0,99 -> 99%

This is a very high accuracy, however, **the system is not able to correctly identify any of the patients who have died.**

Evaluation 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.

In unbalanced problems, 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, ignoring the distribution of major and minor examples expected in the test set on average.

Evaluation metrics for binary classification

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

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

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

Read about precision-recall trade-off

Evaluation 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.
- If the dataset is balanced, then macro-average and micro-average will be about the same.

Macro averages

To know the **overall performance** of the system. Give equal weight to each class, including rare ones.

F-measure is the harmonic average of $\text{Recall}_{\text{macro}}$ and $\text{Precision}_{\text{macro}}$

$$Recall_{macro} = \frac{\sum_{i=1}^{M} Recall_i}{M}, \quad where\ M\ is\ number\ of\ classes$$

$$Precision_{macro} = \frac{\sum_{i=1}^{M} Precision_i}{M}, \quad where\ M\ is\ number\ of\ classes$$

Micro averages

- Micro-averaged metrics **should be used when the size of datasets are variable**.
- Give equal weight to each sample (regardless of its class)
- Micro averages are dominated by those classes with more instances.

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

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

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

- Use micro-averaging when there is a need to weight each instance equally.
- Use macro-averaging score when all classes need to be treated equally to evaluate the overall performance of the classifier.
- If the dataset is balanced, then macro-average and micro-average will be about the same.