TRAIL

TRUSTED AI LABS

Summer Workshop 25' London

Let's Tile together!

Implementation of a Python library for performance analysis

Project n°3 (part of GD 6)







KING'S College LONDON

KING'S INSTITUTE FOR ARTIFICIAL INTELLIGENCE

Wallonie - Bruxelles International.be

ncp WALLONIE

Wallonie service public SPW

S3

TRAIL

TRUSTED AI LABS

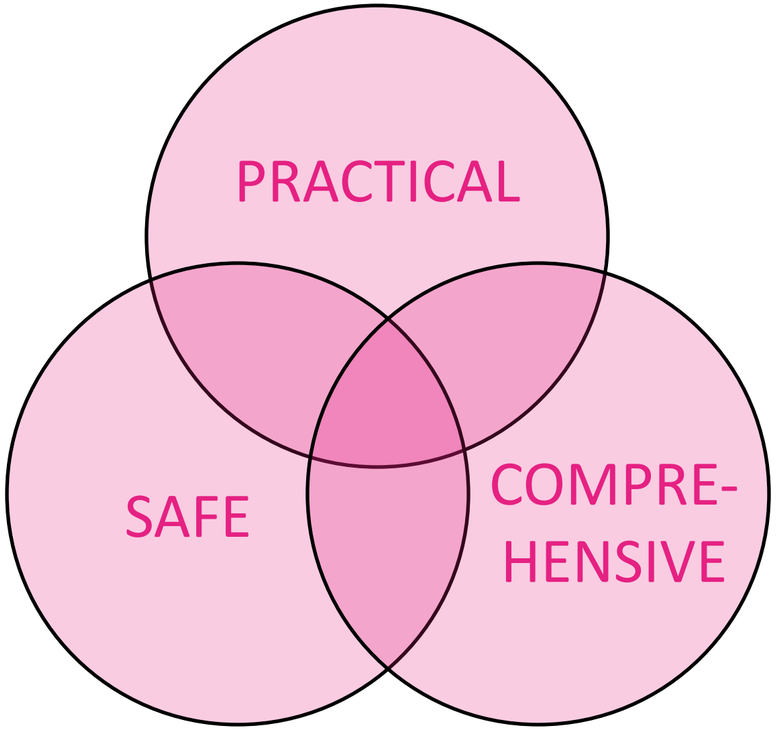
Which scores can we use to rank two-class classifiers?

bias index
specificity
error rate
negative prediction rate
inverse recall
F-Beta score
Peirce skill score
informedness
balanced accuracy
detection rate
true negative rate
Tanimoto coefficient
false omission rate
false negative rate
standardized positive predictive value
accuracy
critical success index
average conditional probability
Jaccard's coefficient
misclassification rate
markedness
negative predictive value
Bennet's S
positive predictive value
sensitivity
weighted accuracy
F-one score
positive likelihood ratio
determinant of the confusion matrix
Heidke skill score
markedness
rejection rate
precision
inverse precision
positive prediction rate
matching coefficient
false positive rate
prevalence
intersection over union
selectivity
Dice-Sørensen coefficient
true positive rate
standardized negative predictive value
Clayton skill score
negative likelihood ratio
similarity
G-measure
false discovery rate
recall
Youden's index
Cohen's κ statistic

And an infinity of others …

Goal

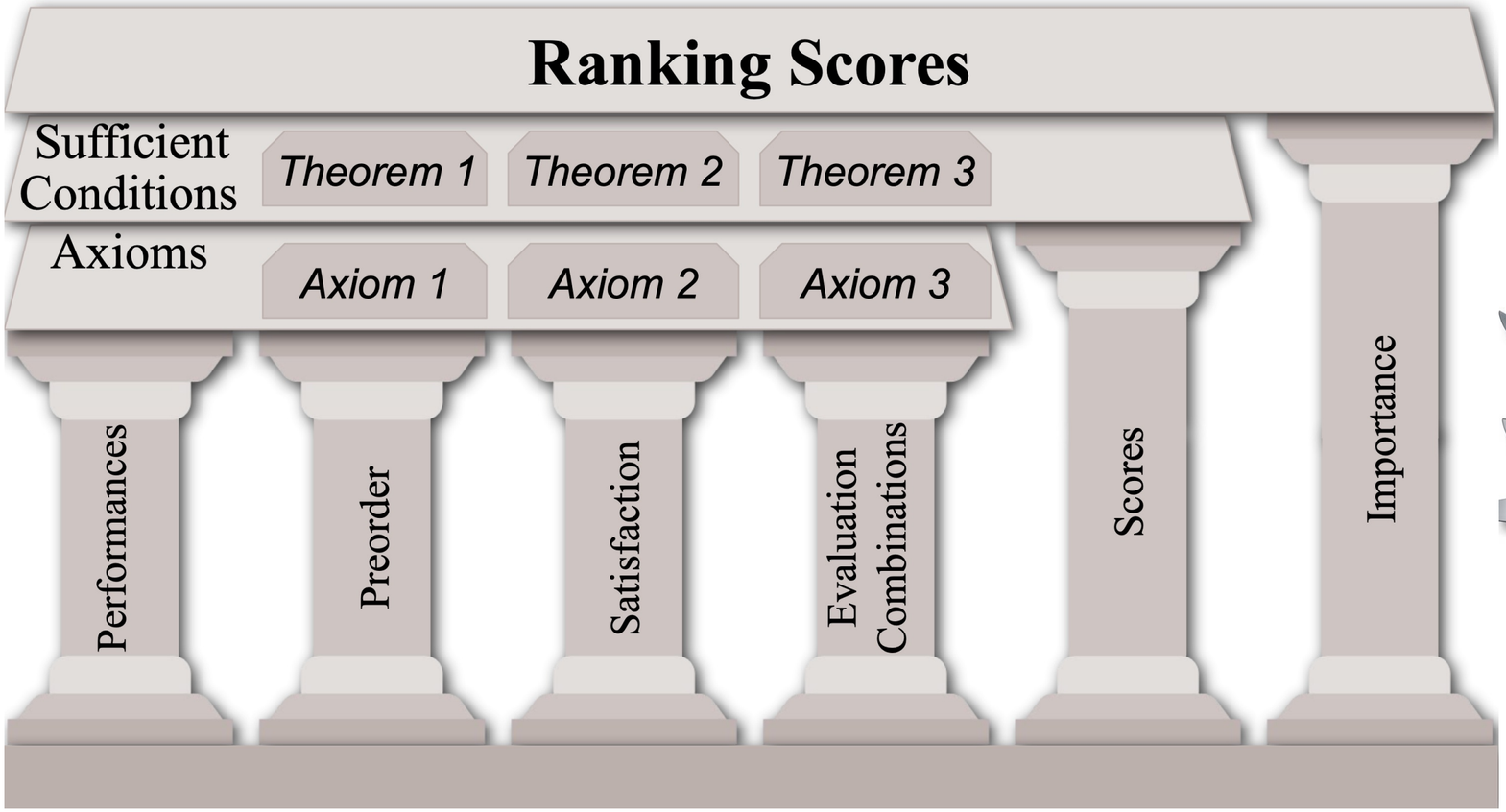
We want to provide visualization tools
for various user profiles:
theoretical analysts,
method designers,
benchmarkers,
app developers,
...



We want to have
safe tools, showing
only performance orderings
(scores) that make sense for ranking.

We want to
avoid biased analyses,
by considering a wide diversity of
performance orderings (scores)
that make sense for ranking.

We have strong theoretical foundations



S. Piérard, A. Halin, A. Cioppa, A. Deliège, and M. Van Droogenbroeck. Foundations of the theory of performance-based ranking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, Tennessee, USA, June 2025.

The ranking scores: a continuum of suitable scores

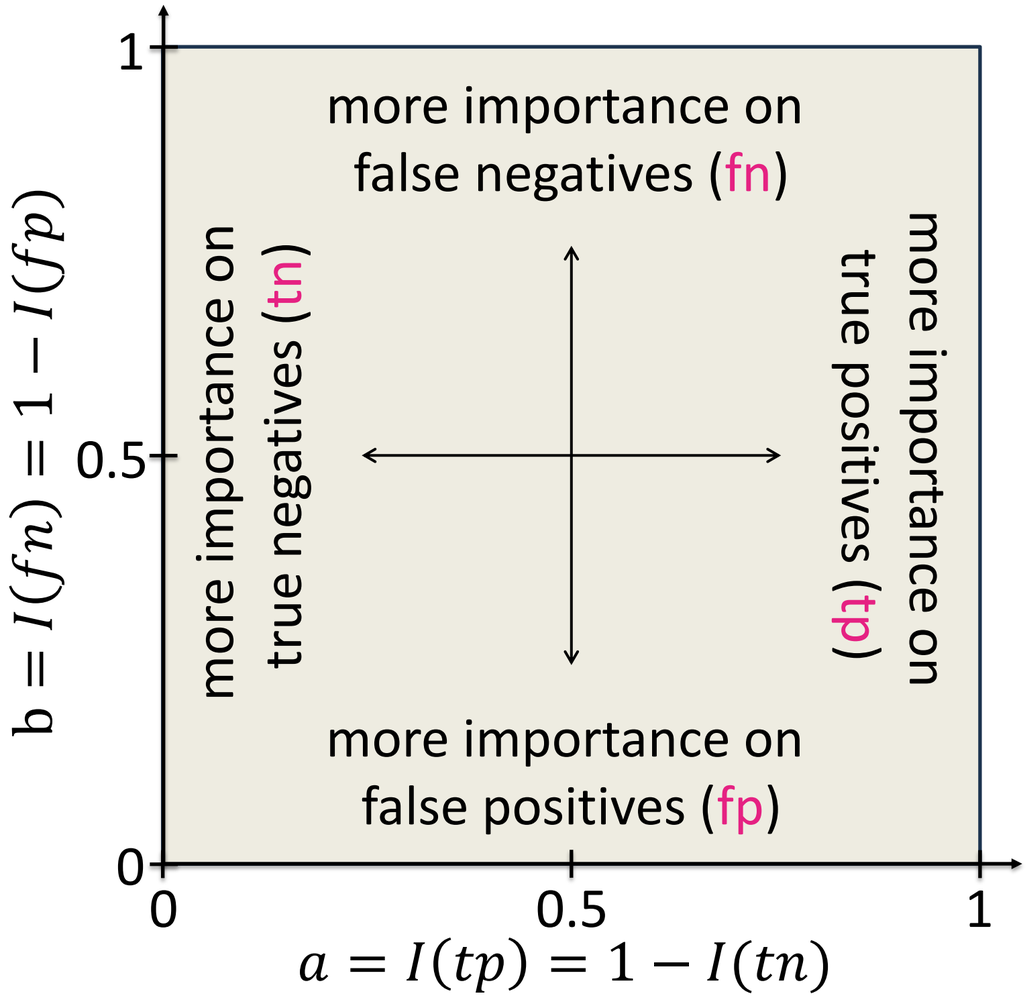
$$R_I : P \mapsto R_I(P) = \frac{\mathbf{E}_P[SI]}{\mathbf{E}_P[I]} = \frac{\sum_{\omega\in\Omega} S(\omega)\, I(\omega)\, P(\{\omega\})}{\sum_{\omega\in\Omega} I(\omega)\, P(\{\omega\})}$$

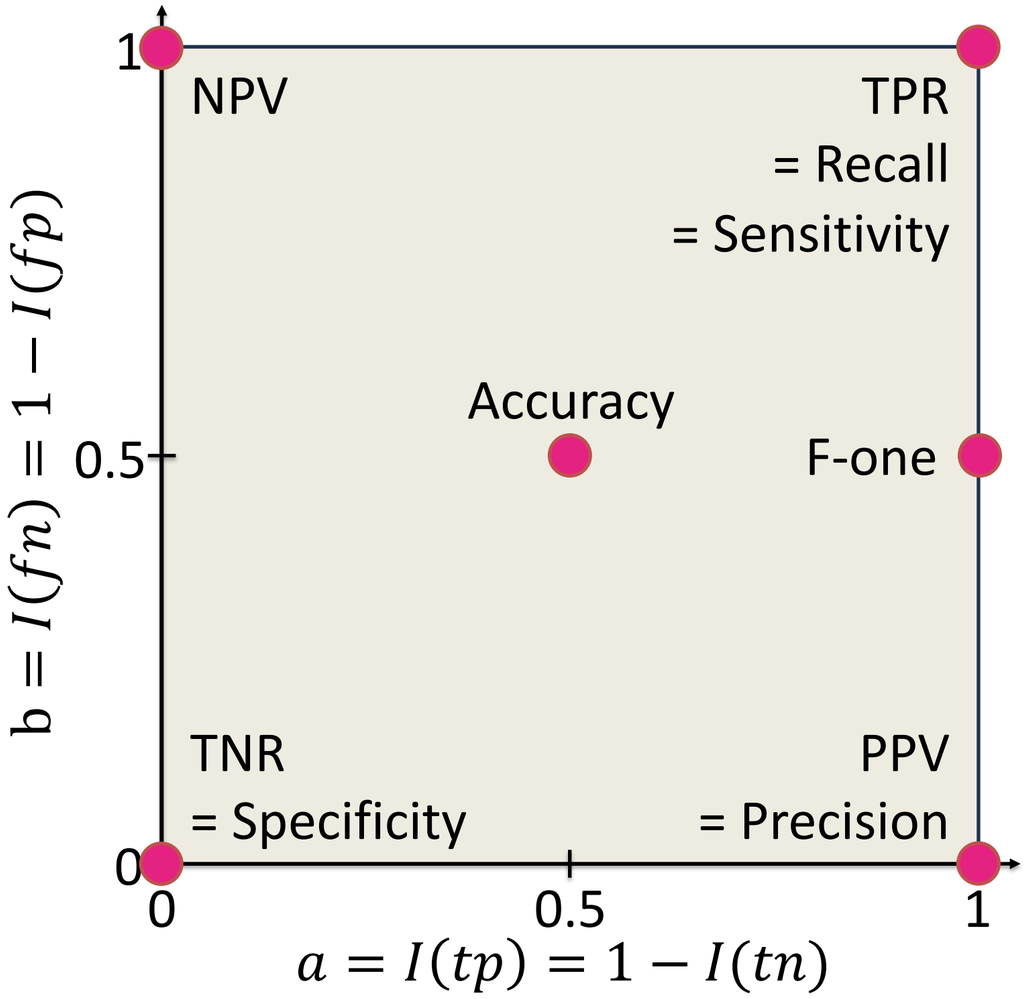
Let's particularize this family of scores to the case of two-class classification!

$$\Omega = \{tn, fp, fn, tp\}$$
$$S(tn) = S(tp) = 1$$
$$S(fp) = S(fn) = 0$$

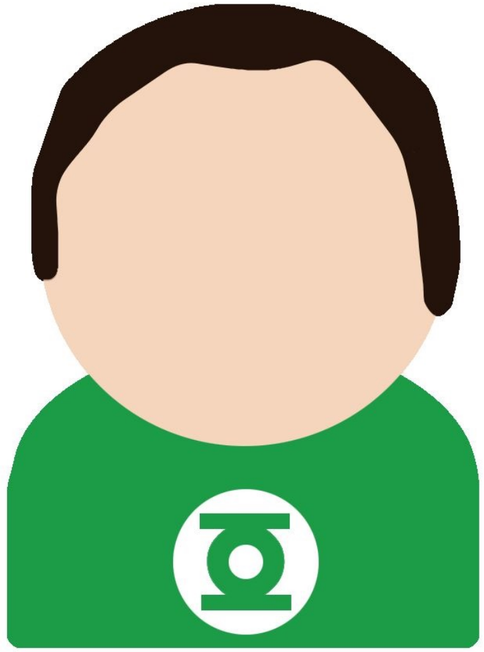
$$R_I(P) = \frac{I(tn)\, P(\{tn\}) + I(tp)\, P(\{tp\})}{I(tn)\, P(\{tn\}) + I(fp)\, P(\{fp\}) + I(fn)\, P(\{fn\}) + I(tp)\, P(\{tp\})}$$

The *Tile*: a 2D map of the performance orderings induced by the ranking scores

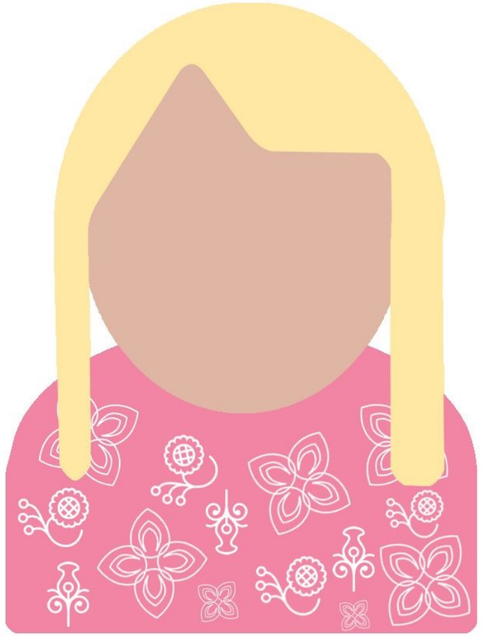




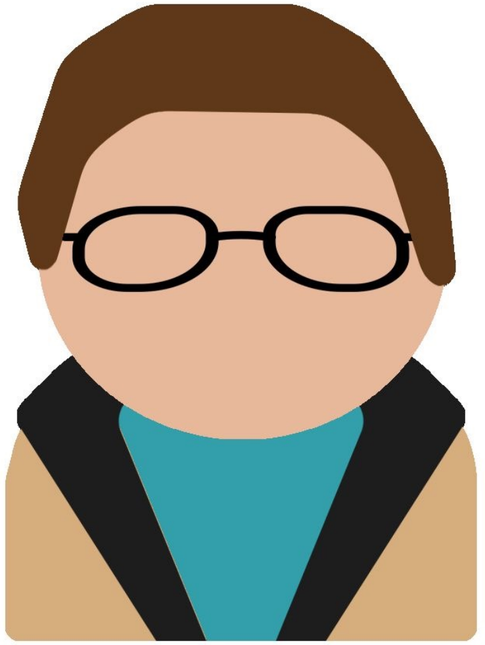
Let me introduce you to four friends …



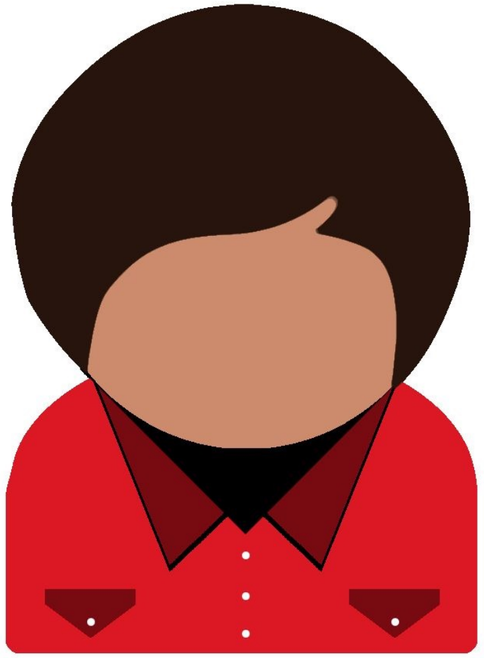
The theoretical analyst



The method designer

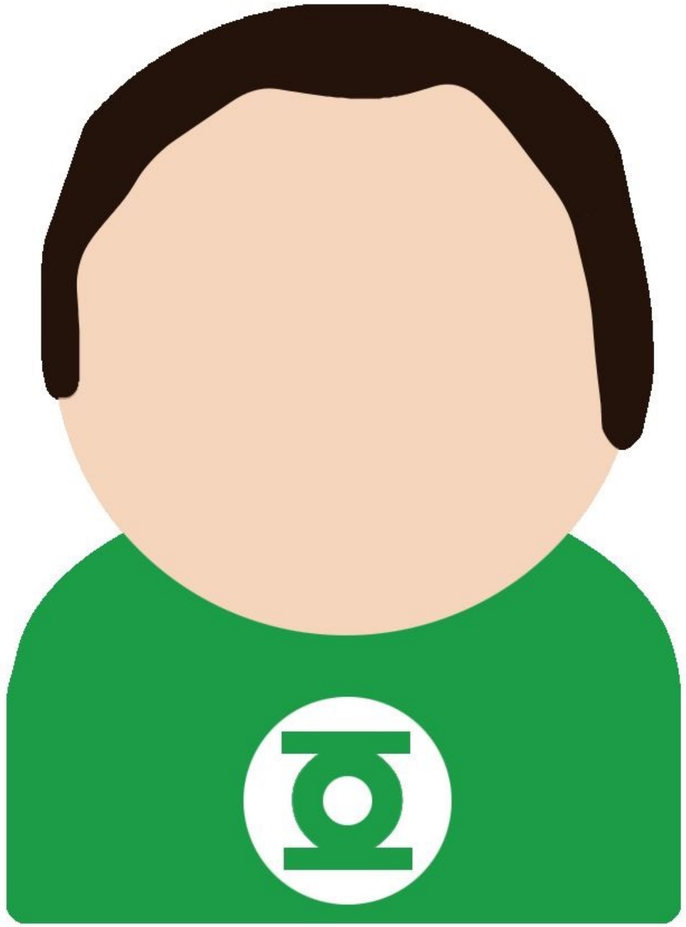


The benchmarker



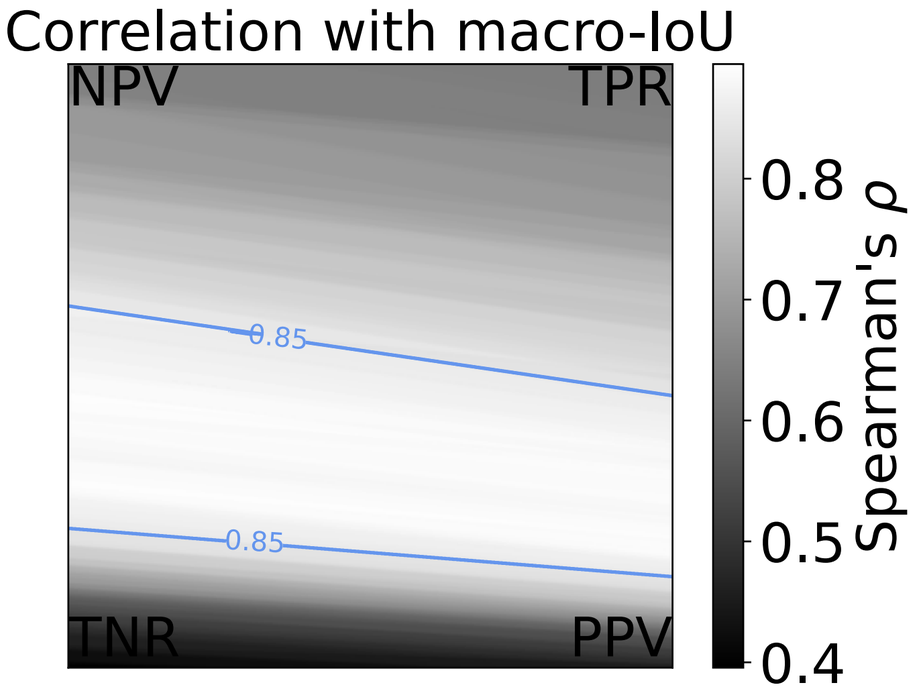
The app developer

Example of Flavor: *Correlation Tiles*



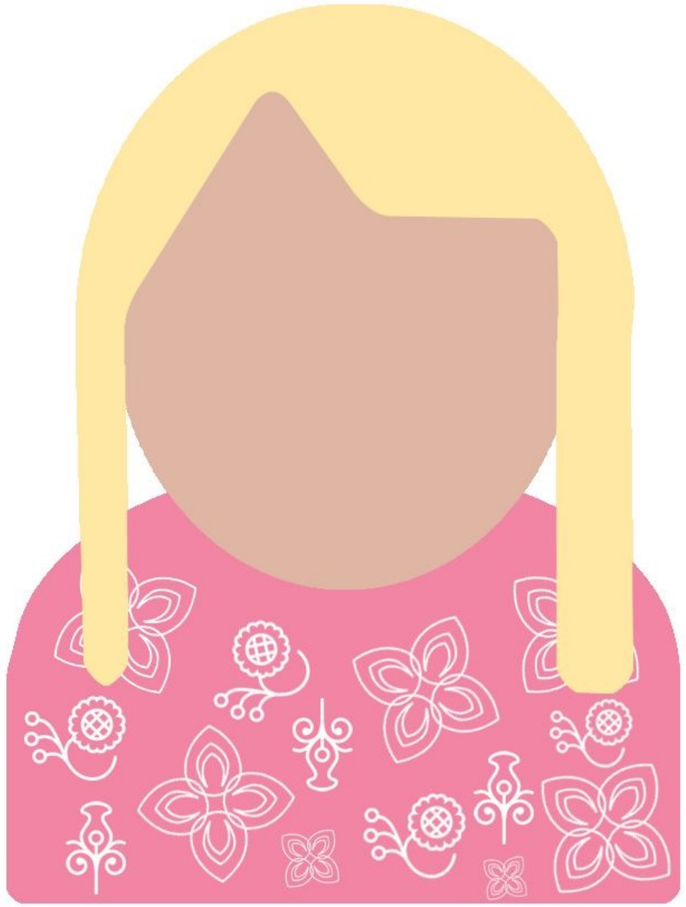
For a theoretical analyst

« What is the behavior of this particular score? »



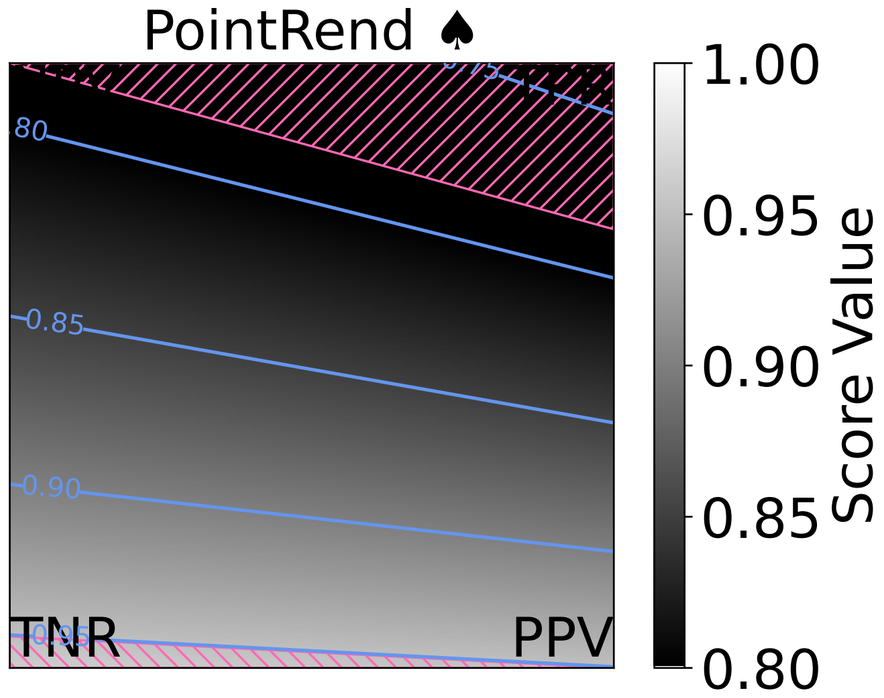
Suggested interpretation. The score is not perfectly rank-correlated with any of the ranking scores. This is usually the sign that it cannot be used to establish meaningful performance orderings. However, it has a correlation of about 0.85 with the accuracy (center of the Tile), which is quite high.

Example of Flavor: *Value Tiles*



For a method designer

« What are the strengths and weaknesses of the method I'm developing? »

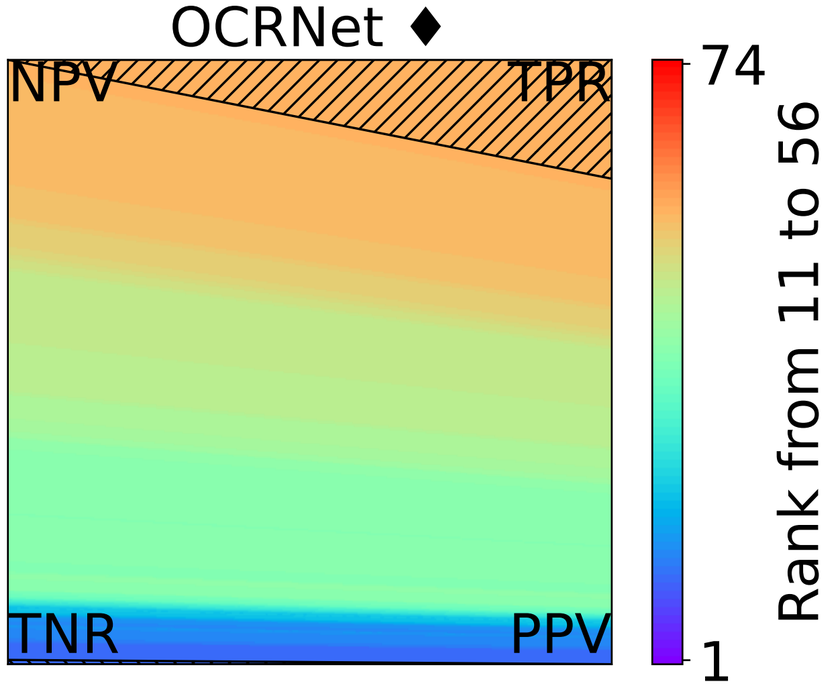


Suggested interpretation. The values taken by the canonical ranking scores show that the strength of the method is when importance is put on the TNs and FPs and its weakness is when the importance is put on the TPs and FNs. Moreover, the region of the Tile where no-skill classifiers perform better is huge.

Example of Flavor: *Ranking Tiles*

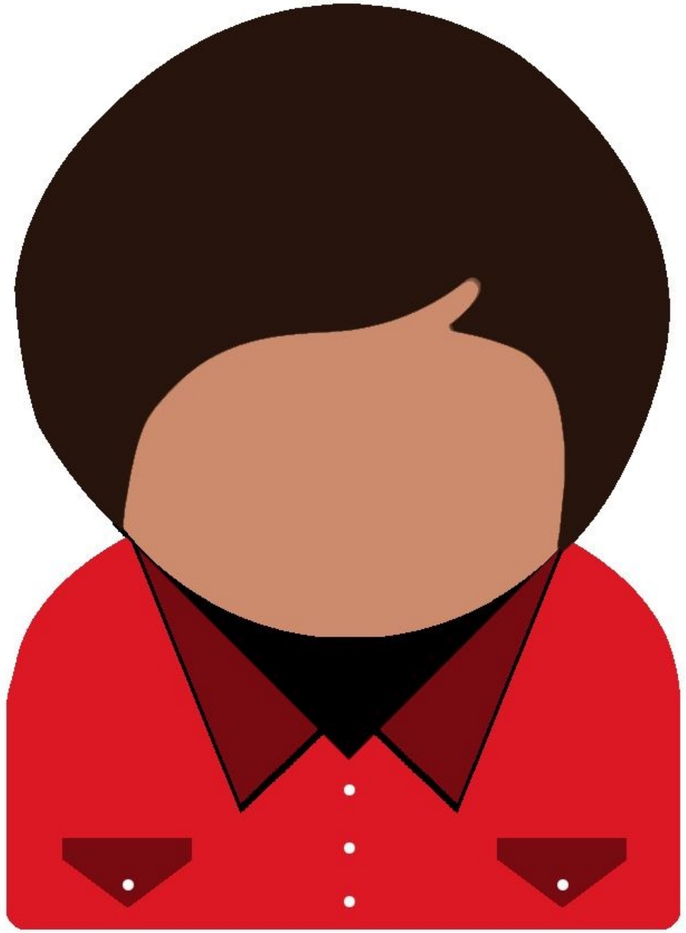
For a benchmarker

« How does this particular method rank w.r.t. the application specific preferences ? »



Suggested interpretation. The method is ranked 11th to 56th , among 74, depending on the application-specific preferences. It performs better than most methods in the bottom of the Tile, where more importance is put on false positives than on false negatives.

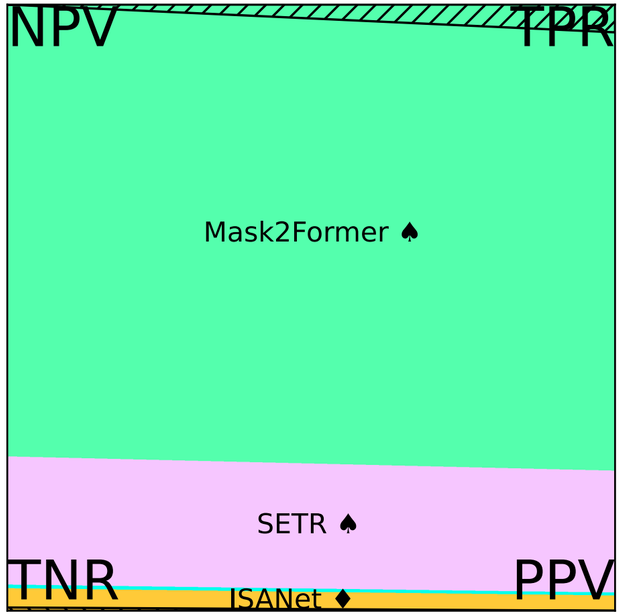
Example of Flavor: *Entity Tiles*



For an app developer

« Show me the most appropriate method w.r.t. the application specific preferences! »

Who's first?



Suggested interpretation. Unless you give much more importance to the false positives than to the false negatives (bottom of the tile), Mask2Former and SETR are the methods you should implement.

Objectives

- **A first version of an open-source library to produce Tiles** (functional and documented)
- **Easy to use by all researchers**
- **Flexible to handle a wide range of flavors**
- **That we plan to published on GitHub, TRAIL factory, PyPI**
- **And ready to be submitted to a journal**
  (e.g. JOSS, *The Journal of Open Source Software*)

Opportunities

To share your talents

- in software architecture design;
- in software development and documentation;
- and all others!

To become author of an open-source library

To discover the Tile in all its details

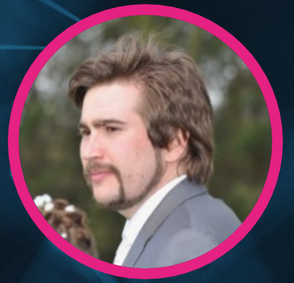
TRAIL
TRUSTED AI LABS

Summer Workshop 25’ London

Thank you for your attention !






KING'S College LONDON

KING'S INSTITUTE FOR ARTIFICIAL INTELLIGENCE

Wallonie - Bruxelles International.be

ncp WALLONIE

Wallonie service public SPW

S3

TRAIL
TRUSTED AI LABS