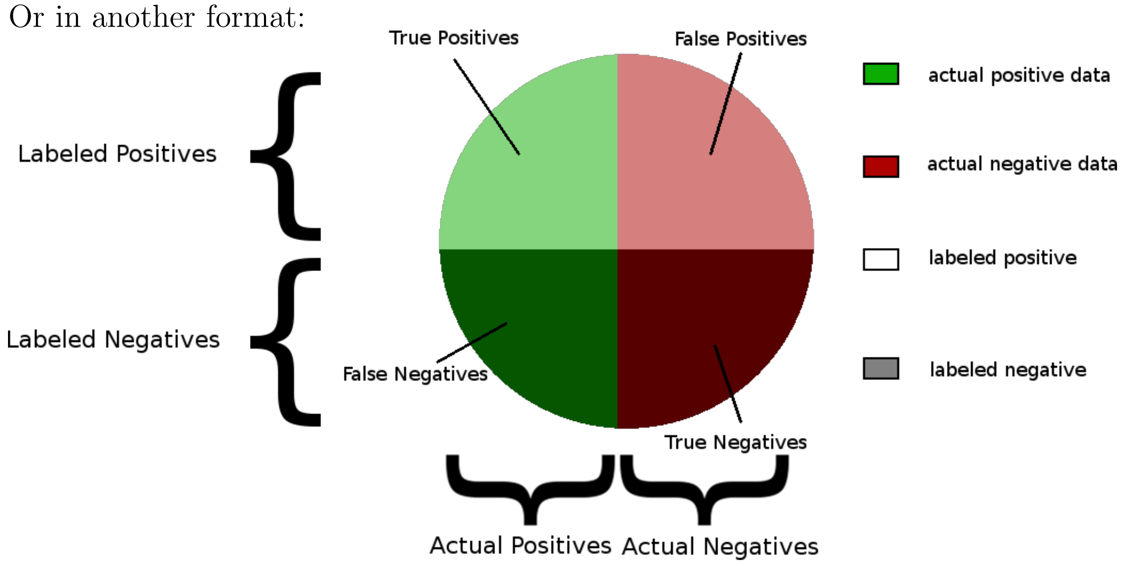
Crucial Metrics

As described in this article on Towards Data Science (Link), *the right metrics for classification tasks heavily depend on the context. Let's take a look at possible parameters/metrics to use in ORES API queries.*
Keep in mind the confusion matrix:

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Or in another format:

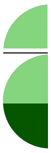


Note that the colors of all parts of the circle are always a mix of two colors from the legend on the right.

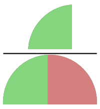
The following subsections describe interesting metrics, notably in the wikidatawiki context and, more precisely, itemquality model, and will be defined and described with the help of the already mentioned sources, not only from this file, but from other files within the repository as well.

0.1 Recall

- Recall (≡ True Positive Rate) is defined as the ability of a model to find all relevant cases within the dataset.

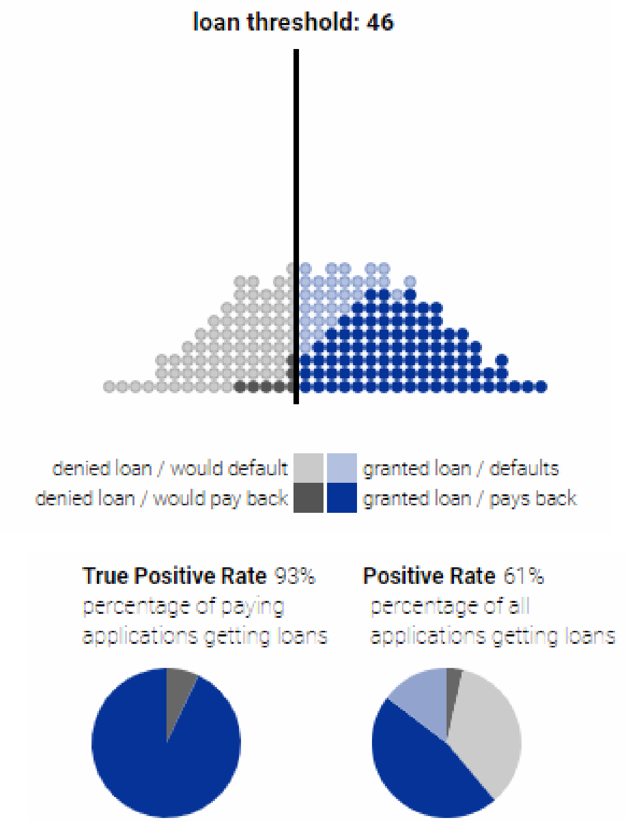
- Or: the portion, of all actual positive data, that was labeled as positive
- In other words Recall $= \frac{\texttt{TP}}{\texttt{TP+FN}}$
- Again in other words $= \frac{\texttt{correctly\_labeled\_as\_positives}}{\texttt{actual\_positives}}$
- $=$ 

0.2 Precision

- Ability of the model to find only relevant cases within the dataset
- The portion, of all as "positive" labeled data, that is actually positive
- $= \frac{\texttt{TP}}{\texttt{TP+FP}}$
- $= \frac{\texttt{correctly\_labeled\_as\_positives}}{\texttt{all\_labeled\_as\_positive}}$
- $=$ 

Recall vs Precision

When increasing one of these two, the other one naturally decreases. For an intuitive example, let's take a look at Google's Loan Threshold Simulation:



The dark grey / dark blue dots, representing clients that would actually pay back their loan, are more and more included (→ given loans) if we move the threshold further to the left.

But so are clients that would not. Thus moving the threshold to the left increases the **recall** (**tpr**) but decreases the **precision** and vice versa when moving to the right.

0.3 f1

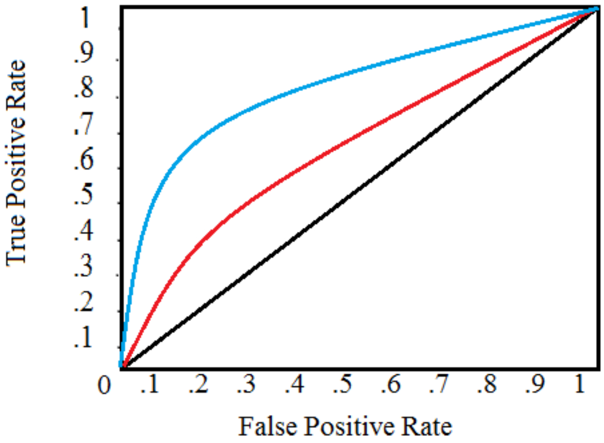
- F1-Score: The optimal combination of recall and precision: the harmonic mean of the two
- **f1** $= 2 * \frac{\texttt{precision}*\texttt{recall}}{\texttt{precision}+\texttt{recall}}$
- Note: harmonic mean is used instead of average to punish extreme values (e.g. precision 1.0 and recall 0.0 → average 0.5, but F1 $= 0$)

0.4 fpr

- We have already mentioned the **TPR**, now, the false positive rate (**FPR**) is the probability of a false alarm
- The portion, of all negatives, that were labeled as positives (→ false positives):
- **fpr** $= \frac{\texttt{FP}}{\texttt{FP+TN}}$

0.5 roc_auc

- Summarize the performance of a classifier over all possible thresholds
- The receiver operating characteristic (ROC) curve plots the TPR versus FPR as a function of the model's threshold for classifying a positive



Two classifiers are shown, one in blue, one in red. The black line is a random classifier.

Keep in mind that on the X- and Y-axis we have the portion, of all negative and of all positive data, respectively, that was predicted as positive

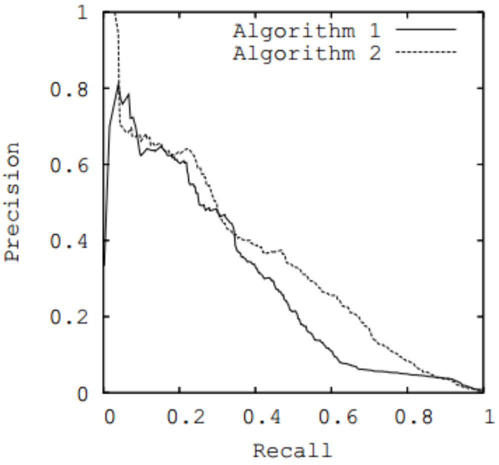
As we increase the threshold we include fewer and fewer data (also data labeled as positive), thus moving down to the bottom left (→ decision threshold = 1.0) on the corresponding curve; top right → threshold = 0.0

That means at the top right corner all data is predicted as positive and in the bottom left corner all data is predicted as negative

- Now, to evaluate / quantify a given ROC, we calculate the area under the curve (**AUC**)
- The AUC is a metric between 0 and 1, where higher values signify better capability of achieving a blend of precision and recall (since a higher peaking curve means higher **tpr**)
- Note: the random classifier will achieve an AUC of 0.5.

0.6 pr_auc

- This one stands for **Precision Recall AUC** (see: `http://www.chioka.in/differences-between-roc-auc-and-pr-auc/`)
- Sample PR-curve:



Instead of the top left corner for the ROC-curve, here, we want to be in the top right corner for our classifier to be perfect

- **pr_auc** is, as expected, the area under the PR-curve. The higher its value, the better the model

0.7 roc_auc vs pr_auc

see: `https://www.kaggle.com/general/7517`

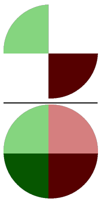
- In short, if the class imbalance problem exists, **pr_auc** is more appropriate than **roc_auc**

If TNs are not meaningful to the problem or there are a lot more negatives than positives, **pr_auc** is the way to go (it does not account for TNs).

- Intuitive explanation:
  - If the model needs to perform equally on the positive and negative class → **roc_auc**
  - If it's not interesting how the model performs on negative class → **pr_auc** (example: detecting cancer; find all positives and make sure they're correct!)

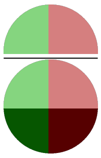
0.8 accuracy

- **accuracy**$= \frac{\texttt{TP+TN}}{\texttt{Total}} =$



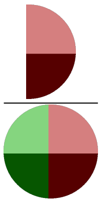
0.9 match_rate

- "The proportion of observations matched/not-matched"
- **match_rate** $= \frac{\texttt{TP+FP}}{\texttt{Total}} =$



0.10 filter_rate

- "The proportion of observations filtered/not-filtered" (?)
- **filter_rate** $= 1 - \texttt{match\_rate}$
- $= \frac{\texttt{TN+FN}}{\texttt{Total}} =$



0.11 !<metric>

- Any <metric> with an exclamation mark is the same metric for the negative class
- e.g. recall $= \frac{\texttt{TP}}{\texttt{TP+FN}} \Rightarrow$ **!**recall $= \frac{\texttt{TN}}{\texttt{TN+FP}}$
- Example usage: find all items that are not "E" class → look at **!**recall for "E" class.

0.11.1 Existing !<metric>s

- !f1
- !precision
- !recall