# Classification Evaluation Metrics

2021 - 2022

# Accuracy

Perhaps the most obvious way of measuring classifier performance is accuracy. Accuracy tells us of all the observations in a dataset, how many are classified correctly? This is given as a ratio or a percentage and can be derived from the correct (true positive and true negative classifications) as shown below. Given a balanced dataset, a higher accuracy score means that a classifier is making more correct classifications, though accuracy runs into issues when used with unbalanced data (Akosa, 2017) as with high prevalence data, a high accuracy score does not necessarily mean high performing classifier.

$$Accuracy = \frac{(True\ positive + True\ negative)}{Total\ observations}$$

# True positive rate/Recall

Recall or the true positive rate, gives the ratio of the observations that have been correctly classified as positive over the total of observations that are labelled positive found by summing the true positive and false negative observations., given below. Or, when an observation should be classified as positive, how often is it classified positive? In Bayesian probability, true positive rate is the conditional probability that a labelled positive value is classified positive. True positive rate gives a measure of how well your classifier is replicating the labelled positive class in its classified positive judgements as a high true positive rate will mean maximising true positives and minimising false negatives. In an equivalent way, it is possible to compute the true negative rate, we just replace the true positive with true negative and false negative with false positive

$$True\ positive\ rate = \frac{True\ positive}{True\ Postive + False\ Negative}$$

# False positive rate

This gives the ratio of the observations that have been incorrectly classified positive over the total of observations that are labelled negative calculated by summing false positive and true negative observations. Or, when an observation should be classified negative, how often is it classified positive, given below. False positive rate measures how well your classifier avoids mislabelling false values as true. So, a lower false positive means the classifier minimises the number of false positives and maximise the true negatives.

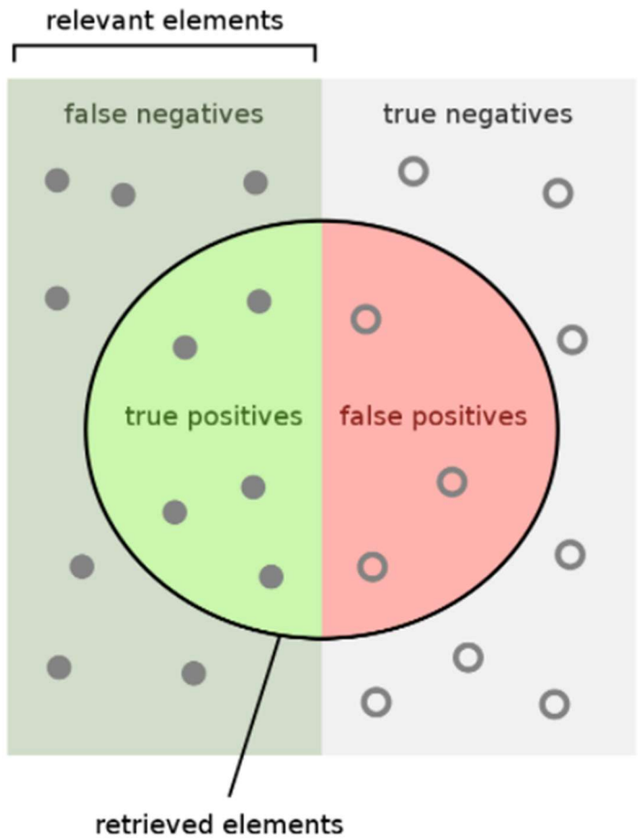
$$False\ positive\ rate = \frac{False\ positives}{False\ positives + True\ Negatives}$$

# Precision

Precision is like the true positive rate, except it uses the false positive instead of false negative to give the total classified positive in the denominator meaning it gives the true positives over total classified true. Or, for a classified positive observation, how likely is this observation labelled positive? Precision can also be considered the Bayesian posterior probability that an observation is labelled positive, given that it has been classified as positive. So, a classifier with a high precision will have a high number of true positives with a low number of false positives. The main difference between precision and recall/true positive rate is precision is more focused on the certainty if the predictions for the true class are correct, rather than how much of the labelled true class are predicted true by the classifier.

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

# Precision and recall




How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

# F-score

F-score is the harmonic mean of recall and precision. It is useful as often, with real-world problems, a classifier needs to make a compromise between reducing false positives at the expense of increasing false negatives, or vice versa. The F-score tells us how well the classifier makes this compromise. It is computed from the harmonic mean as if one of precision or recall scores is 0, F-score would also be zero indicating a poor classifier. F-score is calculated as below.

$$F = 2 * \left( \frac{precision * recall}{precision + recall} \right)$$

# Log-Loss (1/2)

Log-loss, or logarithmic loss, gets into the finer details of a classifier. In particular, if the raw output of the classifier is a numeric probability instead of a class label of 0 or 1, then log-loss can be used. The probability can be understood as a gauge of confidence. If the true label is 0 but the classifier thinks it belongs to class 1 with probability 0.51, then even though the classifier would be making a mistake, it's a near miss because the probability is very close to the decision boundary of 0.5. Log-loss is a "soft" measurement of accuracy that incorporates this idea of probabilistic confidence.

$$\text{log-loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log p_i + (1 - y_i) \log (1 - p_i)$$

# Log-Loss (2/2)

Formulas like this are incomprehensible without years of grueling, inhuman training. Let's unpack it. $p_i$ is the probability that the ith data point belongs to class 1, as judged by the classifier. $y_i$ is the true label and is either 0 or 1. Since $y_i$ is either 0 or 1, the formula essentially "selects" either the left or the right summand. The minimum is 0, which happens when the prediction and the true label match up. (We follow the convention that defines 0 log 0 = 0.)

The beautiful thing about this definition is that it is intimately tied to information theory: log-loss is the cross entropy between the distribution of the true labels and the predictions, and it is very closely related to what's known as the relative entropy, or Kullback–Leibler divergence. Entropy measures the unpredictability of something. Cross entropy incorporates the entropy of the true distribution, plus the extra unpredictability when one assumes a different distribution than the true distribution. So log-loss is an information-theoretic measure to gauge the "extra noise" that comes from using a predictor as opposed to the true labels. By minimizing the cross entropy, we maximize the accuracy of the classifier.

# AUC (1/2)

AUC stands for area under the curve. Here, the curve is the receiver operating characteristic curve, or ROC curve for short. This exotic sounding name originated in the 1950s from radio signal analysis, and was made popular by a 1978 paper by Charles Metz called "Basic Principles of ROC Analysis." The ROC curve shows the sensitivity of the classifier by plotting the rate of true positives to the rate of false positives (see Figure 2-2). In other words, it shows you how many correct positive classifications can be gained as you allow for more and more false positives. The perfect classifier that makes no mistakes would hit a true positive rate of 100% immediately, without incurring any false positives—this almost never happens in practice.

# AUC (2/2)

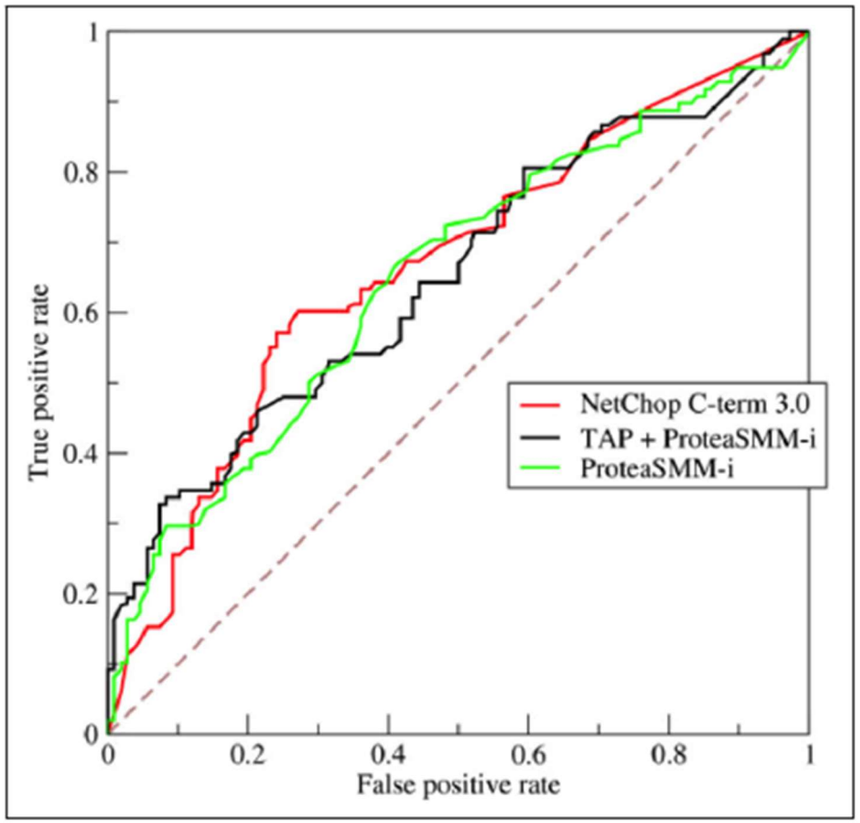



*Figure 2-2. Sample ROC curve (source: Wikipedia)*

The ROC curve is not just a single number; it is a whole curve. It provides nuanced details about the behavior of the classifier, but it's hard to quickly compare many ROC curves to each other. In particular, if one were to employ some kind of automatic hyperparameter tuning mechanism (a topic we will cover in Chapter 4), the machine would need a quantifiable score instead of a plot that requires visual inspection. The AUC is one way to summarize the ROC curve into a single number, so that it can be compared easily and automatically. A good ROC curve has a lot of space under it (because the true positive rate shoots up to 100% very quickly). A bad ROC curve covers very little area. So high AUC is good, and low AUC is not so good.