Precision and recall

Introduction to precision and recall

Precision and recall are important in machine learning, as it is used to define the desired accuracy of a machine learning algorithm.

Precision is the measure of how many bad cases are let through a machine learning algorithm.

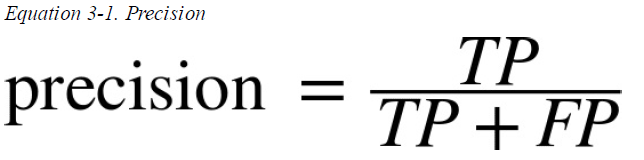
Recall is the measure of how many total cases are let through to the end user.

Discussion

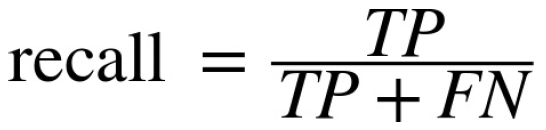
Generally, accuracy is not a preferred performance measure of a classifier. The example given in “Hands-On Machine Learning” page 116 is a good one at explaining the faultiness of accuracy. Let’s say we have a dataset consisting of numbers from 0 – 9. We want to guess how many of these numbers in our dataset is the number 5. Statistically, 10% of the dataset should consist of the number 5. So, by saying that the dataset didn’t contain a single 5, we would already have a 90 % accuracy measure. This of course works best on skewed datasets such as this. If the dataset was binary, the accuracy would be 50% if we were to guess that there were no 0’s in the set. Therefore, accuracy doesn’t really tell much about the dataset, even though the percentage of accuracy we get is good.

A better alternative to accuracy is using precision and recall. Precision is the measure of how many true positives out of the total positives are found by the algorithm. Let’s say we have three true positives and one false positive, we’d have a precision of 75%. On the other hand, if we had the same three true positives and classified two positives as false negatives, we’d have a recall rate of 60%.

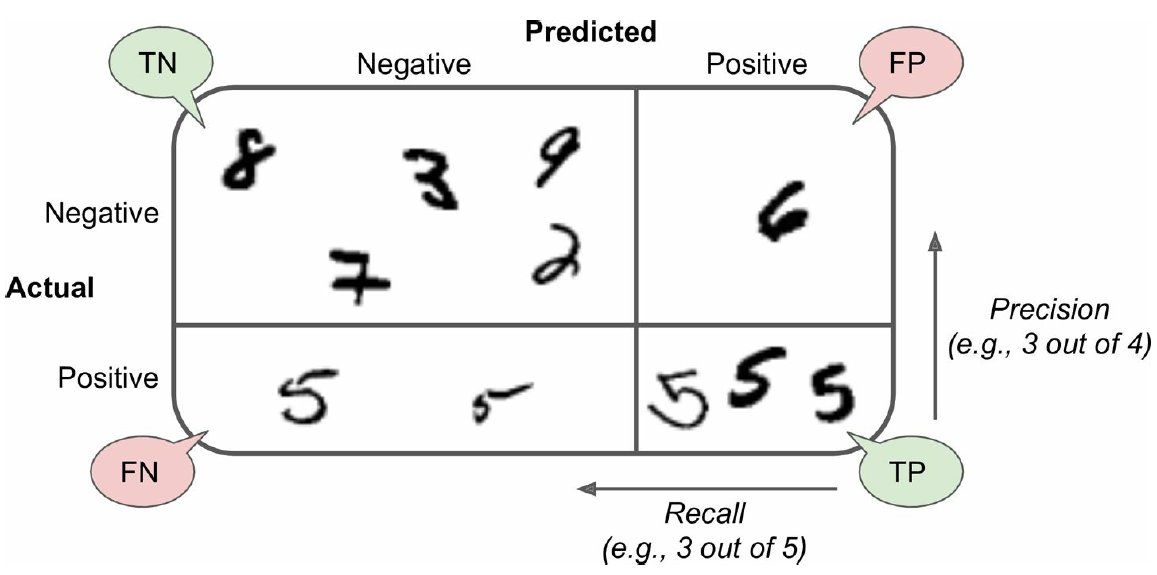
Precision is calculated by using the formula below:



Recall is calculated by using the formula below:



The connection between precision and recall is visualized by the figure below, also known as a confusion matrix:



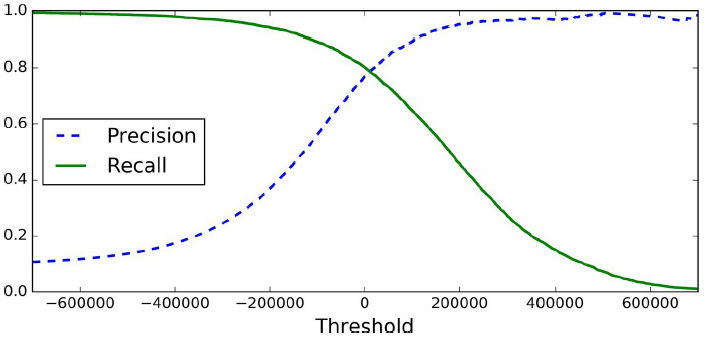
The confusion matrix shows a dataset like the one mentioned before. The dataset consists of the numbers 0 – 9, and we wish to find all the 5’s in the dataset. The matrix groups up the data in the set as False Positives, False Negatives, True Positives and True Negatives.

Whenever a number is classified as true it’s because it was deemed to be the desired data, in this case the number 5. Whenever a number is classified as Positive it’s because the algorithm picked up the number as being the desired number.

Instead of observing two numbers, the harmonic mean (also called F1) is used to describe precision and recall. The F1 score will show a much clearer picture of both the precision and recall values. If the F1 score is high, then both the precision and recall scores will be high. The F1 score is more likely to be high if the classifier as similar precision and recall.

When talking about precision and recall, there’s also the concept of precision/recall tradeoff. Whenever precision gets higher, recall will decrease and vice versa. One could argue that getting as high an F1 score as possible is optimal, but some applications favor higher recall or higher precision over the other. An example of such an application could be a shoplifter detection system. Here the system will favor recall, as it is not of top priority for the system to only return true positives, as long as it catches most if not all of the actual true positives. In this case it would result in the security guards getting a few false positives of shoplifters, but on the other hand the system would strive to detect as many shoplifters as possible.

A good visualization of precision/recall tradeoff can be seen below.



The best F1 score lies at the intersection between the graphs, but of course the chose precision/recall is very dependent on the project.

References

Hands-On Machine Learning with Scikit-Learn and TensorFlow