Performance Metrics for Classification*

March 3, 2022

1 Concepts

Before delving into the performance metrics themselves, it is important to make sure some concepts are clearly understood as they are consistently used across most performance metrics.

1.1 True Value vs Predicted Value

When evaluating the performance of a classification model, two concepts are key, the *real outcome* (usually called y) and the *predicted outcome* (usually called $\hat{y}$). For instance, a model can be trained to predict whether a person will develop a particular disease. In this case, it is trained with samples, e.g. a person's data, containing predictive information, such as age, gender, etc., and each person is labelled with a flag stating whether the disease will develop or not. In this case, the label can be whether the disease will happen ($y = 1$) or will not happen ($y = 0$).

A machine learning model aims at making sure that every time a sample is presented to it, the predicted outcome corresponds to the true outcome. The more the model's predictions are the same as the true values the higher is the performance of the model. There are many different ways of evaluating a model's performance, but in general, models make mistakes, lowering performance.

1.2 True Positive, True Negative, False Positive, False Negative

Each prediction from the model can be one of four types with regards to performance:

- **True Positive** (TP): A sample is **predicted to be positive** ($\hat{y} = 1$, e.g. the person is predicted to develop the disease) and its label is **actually positive** ($y = 1$, e.g., the person will actually develop the disease).
- **True Negative** (TN): A sample is **predicted to be negative** ($\hat{y} = 0$, e.g. the person is predicted to not develop the disease) and its label is **actually negative** ($y = 0$, e.g., the person will actually not develop the disease).

*References

- Eugenio Zuccarelli. Performance Metrics in Machine Learning. 2020

- **False Positive** (FP): A sample is **predicted to be positive** ($\hat{y} = 1$, e.g. the person is predicted to develop the disease) and its label is **actually negative** ($y = 0$, e.g., the person will actually not develop the disease). In this case, the sample is "falsely" predicted as positive.
- **False Negative** (FN): A sample is **predicted to be negative** ($\hat{y} = 0$, e.g. the person is predicted to not develop the disease) and its label is **actually positive** ($y = 1$, e.g., the person will actually develop the disease). In this case, the sample is "falsely" predicted as negative.

Even though the classes are usually labelled 1 and 0, these values are arbitrary and they can often be found labelled as 1 and -1, which is the reason "Positive" and "Negative" are used. Remembering False Positive and False Negative meanings is usually relatively tricky and it is common for data scientists to have to stop for a second to think about the meaning of each to remember which one represents what. An easy trick to remember the difference is focus first on the second part of the name ("Positive" or "Negative"). This relates to the prediction, basically saying "The sample is predicted to be Positive/Negative (belong to class 1/0)...". Then, we can look at the first part of the name to understand whether the prediction was correct or not ("True" or "False"). In this case, we are adding whether the prediction was correct or incorrect, and therefore if the sample was actually belonging to that class. For example, False Positive means that the sample is predicted to be Positive, but this is False/incorrect... as the sample is actually Negative.

1.3 Confusion Matrix

True Positive, True Negative, False Positive and False Negative are usually presented in a tabular format in the so-called **confusion matrix**, which is simply a table organizing the four values. Figure 1 shows such how the values are organized in a confusion matrix.

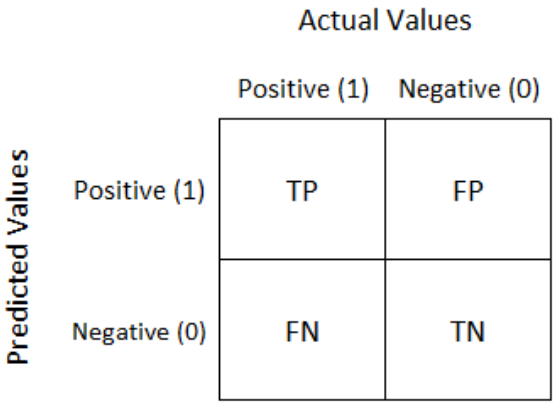

Figure 1: Confusion matrix

2 Performance Metrics

2.1 Accuracy

Accuracy is the fraction of predictions our model got right out of all the predictions. This means that we sum the number of predictions correctly predicted as Positive (TP) or correctly predicted as Negative (TN) and divide it by all types of predictions, both correct (TP, TN) and incorrect (FP, FN).

$$\text{Accuracy } = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

Accuracy ranges between 0 and 1. These extreme cases correspond to completely missing the predictions or having always correct predictions. For instance, if our model is able to perfectly predict, the model will have no False Positives or False Negatives, making the numerator be equal to the denominator, bringing the accuracy to 1. Conversely, if our system is always off, incorrectly predicting each time, the number of True Positives and True Negatives will be zero, making the equation be zero divided by something positive, leading to an accuracy equal to 0.

Accuracy, however, is not a great metric, especially when the data is imbalanced. When there is a significant disparity between the number of positive and negative labels, Accuracy does not tell the full story. For instance, let's consider an example where we have 100 samples, 95 of which labelled as belonging to class 0, and 5 labelled as class 1. In this case, a poorly built "dummy" model which always predicts class 0, achieves a 95% accuracy, which indicates a very strong model. However, this model is not really predictive and accuracy is not the right performance metric to evaluate the power of this model. If we used only accuracy to evaluate this model, we would end up providing stakeholders, and clients eventually, with a model that is not performant or predictive.

2.2 Precision

To overcome the limitations of Accuracy, we usually use Precision, Recall and Specificity. Precision tells what **proportion of positive predictions** was actually correct. It achieves this by counting the samples correctly predicted as positive (TP) and dividing it by the total positive predictions, correct or incorrect (TP, FP).

$$\text{Precision } = \frac{TP}{(TP + FP)} \tag{2}$$

2.3 Recall (Sensitivity, True Positive Rate, Hit Rate)

Similarly to Precision, Recall aims at measuring what **proportion of actual positives** was identified correctly. It does so by dividing the correctly predicted positive samples (TP) by the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (TP, FN).

$$\text{Recall } = \frac{TP}{(TP + FN)} \tag{3}$$

2.4 Specificity (True Negative Rate, Selectivity)

Symmetrically to Recall, Specificity aims at measuring what **proportion of actual negatives** was identified correctly. It does so by dividing the correctly predicted negative samples by the total number of negatives, either correctly predicted as negative or incorrectly predicted as positive (TN, FP).

$$\text{Specificity } = \frac{TN}{(FP + TN)} \tag{4}$$

Considering the example to show the shortcomings of Accuracy, if we use Precision, Recall and Specificity, we get: Accuracy = 0.95 and Recall = 0. By using additional performance metrics instead of Accuracy, we can better understand that a model predicting the majority class all the time is actually a low-performance model (Recall = 0) even though Accuracy is high (Accuracy = 0.95).

2.5 Area Under the ROC Curve (AUC)

As we've seen, one of the issues of Accuracy is that it can lead to overly inflated performance if the distribution of the classes is not very well balanced. AUC, which stands for "Area Under the ROC Curve" (see Section 3.1), measures the entire two-dimensional area underneath the entire ROC curve. It is an aggregate measure of performance across all possible classification thresholds. Another way of interpreting AUC is as the probability that the model ranks a random positive sample higher than a random negative sample. AUC is a great metric, especially when dealing with imbalanced classes, and is one of the most frequently used performance measures in classification, even though it can be used only in binary classification settings (i.e. not with more than 2 classes as target). Some of the properties that make it a preferred metric are:

- **Scale-Invariance**. AUC measures how well predictions are ranked, instead of their absolute values.
- **Classification-Threshold-Invariance**. AUC measures the quality of the model's predictions regardless of what classification threshold is chosen.

2.6 F1 Score

The F1 score is a less known performance metric, indicating the harmonic mean of Precision and Recall. The highest value of an F1 Score is 1, indicating perfect Precision and Recall, and the lowest possible value is 0 if either the Precision or the Recall is zero.

$$F1 \text{ Score } = \frac{2TP}{(2TP + FP + FN)} \tag{5}$$

3 Performance Charts

Additionally, performance measures can be not only communicated as single numbers but also as charts. Some common charts showing a machine learning model's performance are the ROC Curve and the Precision/Recall Curve.

3.1 ROC Curve (Receiver Operating Characteristic Curve)

A ROC curve is a graph showing the performance of a classification model at all classification thresholds (see Figure 2). The chart's y-axis is the True Positive Rate, while the x-axis is the False Positive Rate and the plot consists of the TPR and FPR values varying the threshold. The worst-case scenario (random chance) consists of a 45 degrees diagonal line. The best-case scenario consists of an angled line, going vertically first and horizontally after. Lowering the classification threshold, the model classifies more items as positive, increasing both False Positives and True Positives.

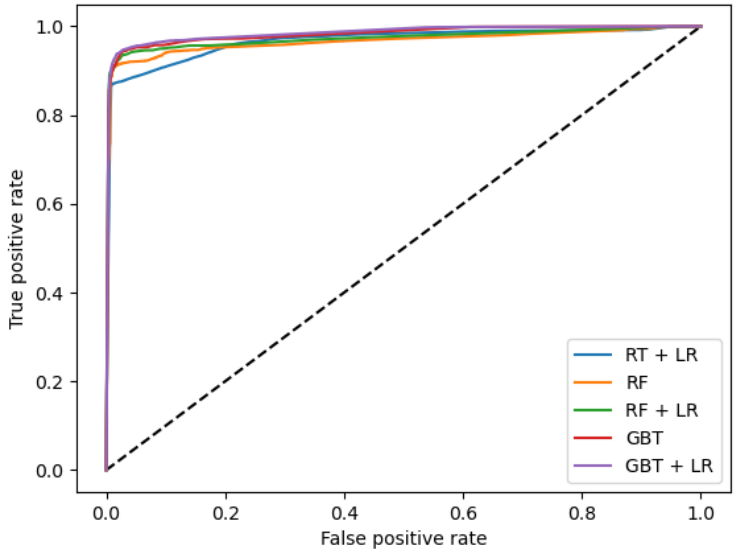

Figure 2: ROC curve

3.2 Precision/Recall Curve

Similarly to the ROC curve, a Precision/Recall curve plots performance over a y-axis showing Precision and an x-axis which is Recall (see Figure 3). Each point is evaluated at different threshold values. The best-case scenario is a flipped version of the ROC curve's best-case scenario, basically consisting of a horizontal line then becoming vertical. Differently, the worst-case scenario, random chance, is seen as a horizontal line at Precision = 0.5.

4 Impact of Choosing the Right Performance Metric

Choosing the right metric is key, especially in cases where False Positives and False Negatives do not have the same impact. Ideally, we would want to have a perfect prediction both in terms of False Positive and False Negative (both zero), but with machine learning models there is usually a tradeoff between detecting False Positives or False Negatives well. For instance, if our model predicts whether a person has got a deadly disease, like cancer, it could be said that False Positives are more important. We want to make sure that if that person has the disease, we correctly flag them. We are less concerned if we accidentally misclassify a person as having the disease even though they didn't have it. Conversely, if our model predicts whether a person is innocent or not, it might be argued

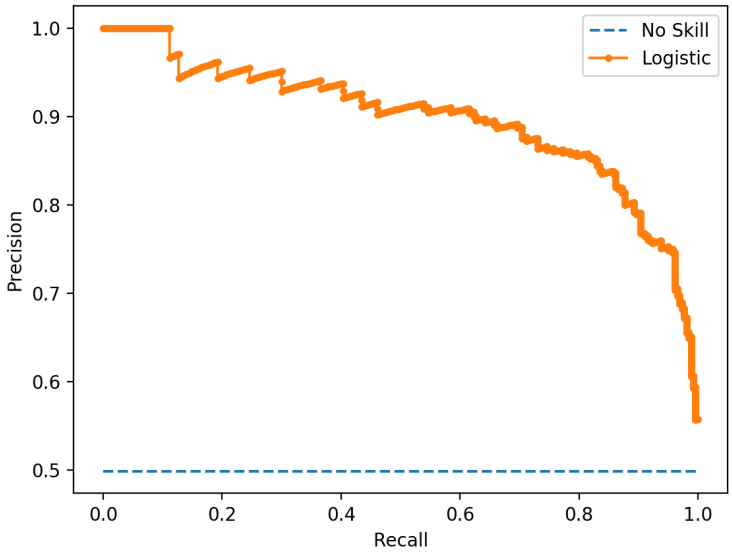

Figure 3: Precision/Recall curve

that False Negatives are more important. We want to make sure that no innocent person is incorrectly jailed.