Basic metrics for binary classification

Measuring what the model is supposed to do

Ernesto L. Garcia C., PhD
Appaloosa Engineering LLC
November 2021

Abstract

Basic performance measures for classification models are presented along with general selection guidelines and a discussion of the costs associated to a bad prediction. Reading time: 15 minutes.

Keywords: Machine learning, binary classifier, confusion matrix, precision, accuracy, recall, F1, baseline, sensitivity, specificity, prevalence.

Introduction

Machine learning success does not come from developing the most sophisticated models, it is the result of solving a business problem with the most efficient and effective model. Thus the importance of *performance metrics* which are some of the tools used to determine if we have a model that solves such business problem.

Performance metrics, or just *metrics*, are quantifiable measures used to assess the success or failure of a model's intended use. From a decision maker's perspective, when dealing with metrics and interacting with the machine learning team, we have to keep three things in mind:

1. There is no right metric that applies to *all* business cases.
2. Metrics are appropriate only when they measure the concept of what the model is supposed to do.
3. The decision maker should communicate what success or failure will be when testing a model.

This time we'll talk about the most common and simple metrics used for binary classification[1]: **Accuracy**, **precision**, **recall** or **sensitivity**, **specificity**, and **F1**. These classification metrics are best explained with the introduction of a **confusion table or matrix**, no pun intended ☺.

Confusion Table

The prediction of a binary classification model can be represented as one of two possible values per event: pass/fail, 1/0, positive/negative, etc. As convention we will use the value 1 or positive as the relevant result that the decision maker is looking for, and the one that usually occurs with less frequency.

These predictions are then compared against the actual value that events had in real life. Therefore, each event will have four possible pairs of predicted and actual values:

- Actual event is positive and model prediction is positive. Also called **True Positive or TP**,
- Actual event is positive and model prediction is negative. Also called **False Negative or FN**,
- Actual event is negative and model prediction is positive. Also called **False Positive or FP**,
- Actual event is negative and model prediction is negative. Also called **True Negative or TN**.

These combinations form the basis of a *Confusion Matrix or Table*, see figure 1. We use the word *confusion* because the table makes it easy to see whether the classifier is mislabeling or *confusing* the outcome that is supposed to predict.

Life is key lime pie for dessert when the model predicts reality, i.e., an event is either a True Positive (TP) or True Negative (TN). The problem is when we have False Positives (FP) or False Negatives (FN) because the consequence, or cost, of having these erroneous predictions can be severe depending on the context of the problem we are trying to solve. Consider for example the case of a vessel equipped with an algorithm that uses different sonar signals to classify objects undersea as proximity mines or rocks. In this case we will consider mines as the relevant event or *positive*. A False Positive will occur when the classifier declares signals as coming from a mine when in reality they come from a rock. A False Negative will happen when the classifier declares the object as a rock when in reality is a proximity mine. Both

[1] A binary classification model labels the outcomes, or events, of a process into two mutually exclusive groups: e.g., determining if a patient has cancer or not, classifying e-mails as *spam* or *no spam*, or deciding if an applicant is hired or not.

cases are model errors[2]. However, a False Negative has much more costly consequences for the vessels and their crew.

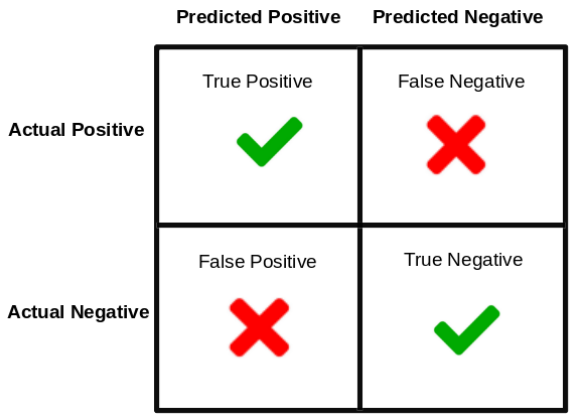

Figure 1: Confusion table.

False Positives can be costly too, as we have seen during the Covid-19 pandemic and reported in The New York Times: "False positives are generally very rare among tests that have been vetted by the Food and Drug Administration. [However], some rapid tests, which forgo sophisticated laboratory equipment and can deliver results in under an hour, have been criticized for returning high numbers of false positives, especially when used to screen people without symptoms...In places where the virus is relatively scarce, false positives may even outnumber actual positives, eroding trust in tests and, under some circumstances, prompting outbreaks of their own [1]."

Thus, rather than only consider the value of a metric by itself, the decision maker must also think about the economic consequences of an erroneous model decision (FP and FN) because all these metrics are calculated from the number of events that fall into the categories described by the confusion matrix.

Accuracy

Accuracy measures how often the classifier makes the correct prediction. It is the fraction of correctly classified events among the total number considered by the model as expressed by equation (1).

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{1}$$

Accuracy answers the question: *What is the fraction of correct predictions among the total number of predictions made by the model?*

[2]Statisticians refer to a False Positive as **Type I error**, and False Negative as **Type II error**.

Accuracy and imbalanced data sets

Accuracy is a misleading metric when we are dealing with imbalanced data sets[3]. Consider a sample where 99% are labeled as negative values and 1% are positive. If we use a classifier model that predicts a negative value for all events then we will have an accuracy of 0.99 or 99%. In this case a high accuracy is achievable by a simplistic model with no skill that only predicts the most frequent class. The problem is that the rare and more relevant class for the customer is misclassified.

The take-away is: accuracy should not be used as a metric to *build* a model with samples that have a high imbalance of relevant events. Furthermore, accuracy can have catastrophic effects when used to train models aimed to the detection of very rare events if the sample data to create the model is not appropriately balanced. Precision, recall, specificity, F1 and other metrics[4] can be used when our data is imbalanced [2].

Precision

Precision, also called *positive predictive value*, evaluates the column that contains the True Positive **(TP)** cell as expressed by equation (2). It measures the fraction of true positive predictions relative to all positive predictions made by the model.

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

Precision answers the question: *From all the events the model predicted as positive, what was the fraction of actual positive?*

Recall

Recall, also called *sensitivity*, evaluates the row that contains the True Positive **(TP)** cell as expressed by equation (3). Therefore, it measures the fraction of true positive predictions relative to all actual positives from the sample.

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

Recall answers the question: *From all the actual positive events, what was the fraction "caught" by the model?*.

[3] A set is imbalanced when it is biased towards one class. This is, data sets that severely deviate from a distribution of 50% events with positive label and 50% events with negative label.

[4] Coming in a second article.

Precision and Recall as a pair

A well performing model should produce both high values of precision and recall. That is not always the case. If both precision and recall are lower than a previously established baseline, the model should be improved or replaced with a better one. When one of the metrics is high and the other is low the decision maker must consider the context and problem the model is solving. Usually, the machine learning team will be able to reduce one type of error (FN or FP) but they need guidance about which of these types of errors are costlier to the customer and the business. As a rule of thumb:

- Use recall when the cost of a False Negative is high
- Use precision when the cost of a False Positive is high and the cost of a False Negative is low

If you are considering both metrics at the same time:

- **If precision is low and recall is high** the model predicts most of the actual positive events (FN are low) but also predicts as positive many of the actual negative events (FP are high). **The model is over-predicting**.
- **If precision is high and recall is low** the model is good at predicting actual positive events (FP are low), but only catches a small proportion of actual positives (FN are high). **The model is under-predicting**.

F1: Aggregate measure of Precision and Recall

F1 is the harmonic mean of precision and recall[5] as expressed in equation (4):

$$F1 = \frac{2 * (Recall) * (Precision)}{Recall + Precision} \tag{4}$$

F1 takes values between 0 and 1. F1=1 means the model perfectly classifies each event or observation. An F1=0 comes from a model that is unable to classify any event or observation correctly.

F1 is low if either precision or recall are low. F1 has a high value if both precision and recall are high. If precision or recall are zero, then F1 will be zero irrespectively of the value of the other.

F1 is used in machine learning as an aggregated metric because only the positive class is of relevance while the number of negatives, in general, is large or unknown.

[5]The harmonic mean is often used to calculate the average of rates or ratios like precision and recall. It is the most appropriate average because it equalizes the weights of each data point.

Some suggestions on metrics practice

Baseline

How do you know if your metric is a good one? You need to calculate a baseline. A baseline in classification problems is obtained by selecting the class that has the most observations as the result for all predictions [3].

Prevalence metrics

In statistics, and in medicine, two metrics used are *sensitivity* and *specificity* as presented in equations (5) and (6).

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

Sensitivity is another name for recall. Specificity is the fraction of negative predictions among actual negative events. Specificity increases if false positives (FP) decrease. Thus, as precision, it is a useful metric when the penalty for false positives is high.

Sensitivity and specificity are special because their values do not depend on *prevalence* or the proportion of actual positives in the sample. Prevalence is defined in equation (7).

$$Prevalence = \frac{TP + FN}{TP + FN + FP + TN} \tag{7}$$

The implication for the decision maker is: once estimated in a sample obtained from a population these metrics may be applied to other populations where prevalence is different.

Conclusion

Because there is no right metric that spans all business cases, the decision maker should clearly define what success, and failure, will look in a model's test. This implies that we should select metrics that measure the concept of what a model is supposed to do and the associated costs of a bad outcome.

If you want to know more about metrics for classification, wikipedia provides a detailed summary in [4] and [5].

References

1. Why false positives merit concern, too. https://www.nytimes.com/2020/10/25/health/coronavirus-testing-false-positive.html. Accessed: 2021-10-29.

2. Paula Branco, Luis Torgo, and Rita Ribeiro. A survey of predictive modelling under imbalanced distributions. https://arxiv.org/abs/1505.01658, 2015. Accessed: 2021-11-08.

3. How to get baseline results and why they matter. https://machinelearningmastery.com/how-to-get-baseline-results-and-why-they-matter/. Accessed: 2021-12-07.

4. Evaluation of binary classifiers. https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers. Accessed: 2021-12-07.

5. Precision and recall. https://en.wikipedia.org/wiki/Precision_and_recall. Accessed: 2021-12-07.