quite subtle. Suppose you have a binary classifier. Is it more important to never mislabel false samples as true or to never mislabel true samples as false? How can you choose for model hyperparameters that satisfy the needs of your applications?

The answer turns out to be to choose the correct metric. In this section, we will discuss many different metrics for classification and regression problems. We will comment on the qualities each metric emphasizes. There is no best metric, but there are more suitable and less suitable metrics for different applications.

Metrics Aren't a Replacement for Common Sense!

Metrics are terribly blind. They only optimize for a single quantity. Consequently, blind optimization of metrics can lead to entirely unsuitable outcomes. On the web, media sites often choose to optimize the metric of "user clicks." Some enterprising young journalist or advertiser then realized that titles like "You'll never believe what happened when X" induced users to click at higher fractions. Lo and behold, clickbait was born. While clickbait headlines do indeed induce readers to click, they also turn off readers and lead them to avoid spending time on clickbait-filled sites. Optimizing for user clicks resulted in drops in user engagement and trust.

The lesson here is general. Optimizing for one metric often comes at the cost of a separate quantity. Make sure that the quantity you wish to optimize for is indeed the "right" quantity. Isn't it interesting how machine learning still seems to require human judgment at its core?

Binary Classification Metrics

Before introducing metrics for binary classification models, we think you will find it useful to learn about some auxiliary quantities. When a binary classifier makes predictions on a set of datapoints, you can split all these predictions into one of four categories (Table 5-1).

Table 5-1. Prediction categories

Category	Meaning
True Positive (TP)	Predicted true, Label true
False Positive (FP)	Predicted true, Label false
True Negative (TN)	Predicted false, Label false
False Negative (FN)	Predicted false, Label true

We will also find it useful to introduce the notation shown in Table 5-2.

Table 5-2. Positives and negatives

Category	Meaning
P	Number of positive labels
N	Number of negative labels

In general, minimizing the number of false positives and false negatives is highly desirable. However, for any given dataset, it is often not possible to minimize both false positives and false negatives due to limitations in the signal present. Consequently, there are a variety of metrics that provide various trade-offs between false positives and false negatives. These trade-offs can be quite important for applications. Suppose you are designing a medical diagnostic for breast cancer. Then a false positive would be to mark a healthy patient as having breast cancer. A false negative would be to mark a breast cancer sufferer as not having the disease. Neither of these outcomes is desirable, and designing the correct balance is a tricky question in bioethics.

We will now show you a number of different metrics that balance false positives and false negatives in different ratios (Table 5-3). Each of these ratios optimizes for a different balance, and we will dig into some of these in more detail.

Table 5-3. Binary metrics table

Metric	Definition
Accuracy	(TP + TN)/(P + N)
Precision	TP/(TP + FP)
Recall	TP/(TP + FN) = TP/P
Specificity	TN/(FP + TN) = TN/N
False Positive Rate (FPR)	FP/(FP + TN) = FP/N
False Negative Rate (FNR)	FN/(TP + FN) = FN/P

Accuracy is the simplest metric. It simply counts the fraction of predictions that were made correctly by the classifier. In straightforward applications, accuracy should be the first go-to metric for a practitioner. After accuracy, *precision* and *recall* are the most commonly measured metrics. Precision simply measures what fraction of the datapoints predicted positive were actually positive. Recall in its turn measures the fraction of positive labeled datapoints that the classifier labeled positive. *Specificity* measures the fraction of datapoints labeled negative that were correctly classified. The false positive rate measures the fraction of datapoints labeled negative that were misclassified as positive. False negative rate is the fraction of datapoints labeled positive that were falsely labeled as negatives.

These metrics all emphasize different aspects of a classifier's performance. They can also be useful in constructing some more sophisticated measurements of a binary

classifier's performance. For example, suppose that your binary classifier outputs class probabilities, and not just raw predictions. Then, there rises the question of choosing a *cutoff*. That is, at what probability of positive do you label the output as actually positive? The most common answer is 0.5, but by choosing higher or lower cutoffs, it is often possible to manually vary the balance between precision, recall, FPR, and TPR. These trade-offs are often represented graphically.

The receiver operator curve (ROC) plots the trade-off between the true positive rate and the false positive rate as the cutoff probability is varied (see Figure 5-1).

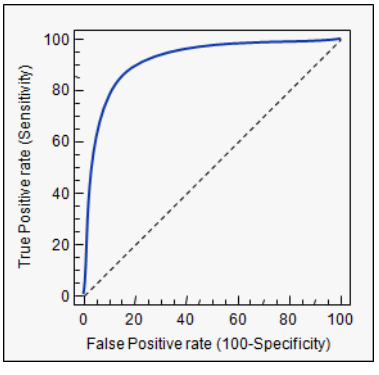

Figure 5-1. The receiver operator curve (ROC).

The area under curve (AUC) for the receiver operator curve (ROC-AUC) is a commonly measured metric. The ROC-AUC metric is useful since it provides a global picture of the binary classifier for all choices of cutoff. A perfect metric would have ROC-AUC 1.0 since the TPR would always be maximized. For comparison, a random classifier would have ROC-AUC 0.5. The ROC-AUC is often useful for imbalanced datasets, since the global view partially accounts for the imbalance in the dataset.

Multiclass Classification Metrics

Many common machine learning tasks require models to output classification labels that aren't just binary. The ImageNet challenge (ILSVRC) required entrants to build models that would recognize which of a thousand potential object classes were in provided images, for example. Or in a simpler example, perhaps you want to predict tomorrow's weather, where provided classes are "sunny," "rainy," and "cloudy." How do you measure the performance of such a model?

The simplest method is to use a straightforward generalization of accuracy that measures the fraction of datapoints correctly labeled (Table 5-4).