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).

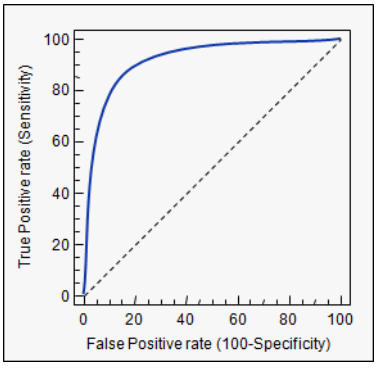

Figure 5-1. The receiver operator curve (ROC).

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).

Table 5-4. Multiclass classification metrics

Metric	Definition
Accuracy	Num Correctly Labeled/Num Datapoints

We note that there do exist multiclass generalizations of quantities like precision, recall, and ROC-AUC, and we encourage you to look into these definitions if interested. In practice, there's a simpler visualization, the *confusion matrix*, which works well. For a multiclass problem with k classes, the confusion matrix is a $k \times k$ matrix. The (i, j)-th cell represents the number of datapoints labeled as class i with true label class j. Figure 5-2 illustrates a confusion matrix.

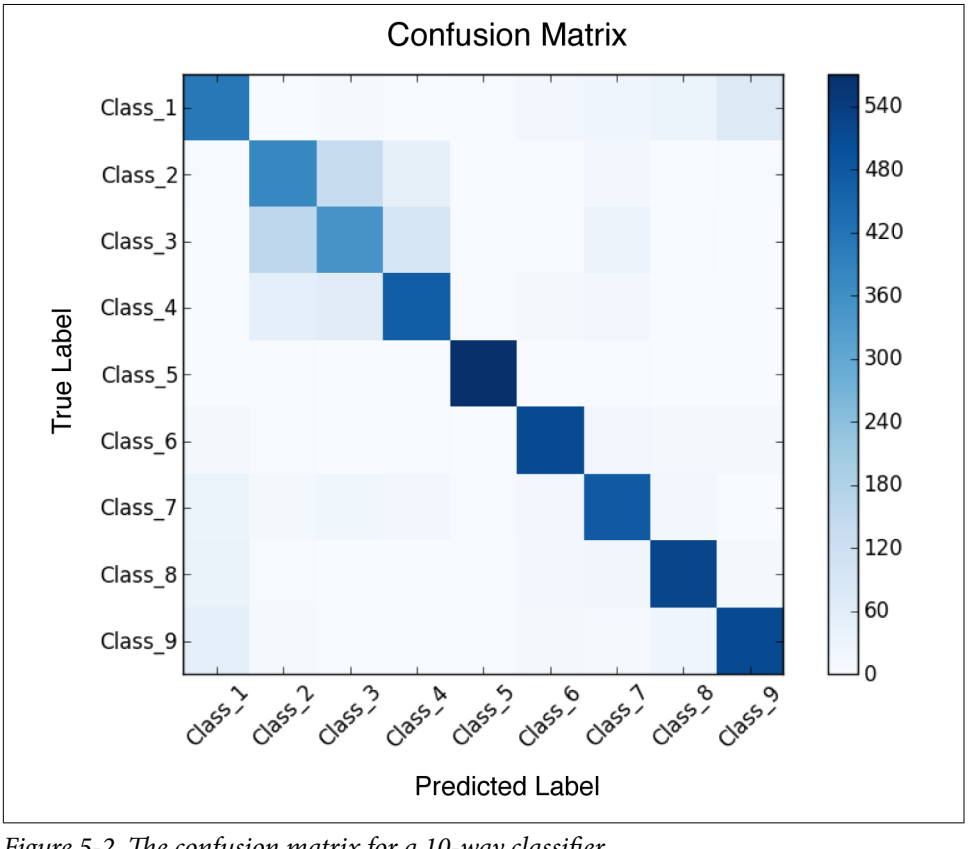

Figure 5-2. The confusion matrix for a 10-way classifier.

Don't underestimate the power of the human eye to catch systematic failure patterns from simple visualizations! Looking at the confusion matrix can provide quick understanding that dozens of more complex multiclass metrics might miss.

Regression Metrics

You learned about regression metrics a few chapters ago. As a quick recap, the Pearson R^2 and RMSE (root-mean-squared error) are good defaults.

We only briefly covered the mathematical definition of R^2 previously, but will delve into it more now. Let x_i represent predictions and y_i represent labels. Let $\bar{x}$ and $\bar{y}$ represent the mean of the predicted values and the labels, respectively. Then the Pearson R (note the lack of square) is

$$R = \frac{\Sigma_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\Sigma_{i=1}^{N}(y_i - \bar{y})^2}}$$

This equation can be rewritten as

$$R = \frac{\text{cov}(x, y)}{\sigma(x)\sigma(y)}$$

where cov represents the covariance and σ represents the standard deviation. Intuitively, the Pearson R measures the joint fluctuations of the predictions and labels from their means normalized by their respective ranges of fluctuations. If predictions and labels differ, these fluctuations will happen at different points and will tend to cancel, making R^2 smaller. If predictions and labels tend to agree, the fluctuations will happen together and make R^2 larger. We note that R^2 is limited to a range between 0 and 1.

The RMSE measures the absolute quantity of the error between the predictions and the true quantities. It stands for root-mean-squared error, which is roughly analogous to the absolute value of the error between the true quantity and the predicted quantity. Mathematically, the RMSE is defined as follows (using the same notation as before):

$$\text{RMSE} = \sqrt{\frac{\Sigma_{i=1}^{N}(x_i - y_i)^2}{N}}$$

Hyperparameter Optimization Algorithms

As we mentioned earlier in the chapter, hyperparameter optimization methods are learning algorithms for finding values of the hyperparameters that optimize the chosen metric on the validation set. In general, this objective function cannot be differentiated, so any optimization method must by necessity be a black box. In this section, we will show you some simple black-box learning algorithms for choosing