the classifier. First employed at the dawn of radar, these curves show the trade-offs over the continuum of cutoff values (relative costs) of the two competing goals; that is, they reveal how changing cutoffs affect the classifier’s accuracy versus precision, or false alarm versus false dismissal rate, or Type I versus Type II error, or false positive versus false negative rate, etc. One receiver (classifier) dominates another in a cost region if it is better by both criteria. The cost region of interest is that closest to the estimated relative cost. Thus, it is possible for an ROC curve to reveal that one classifier is better than another without anyone being able (or willing) to exactly specify the relative costs.

Note that each point along the ROC curve corresponds to a separate confusion matrix. And choice of a particular confusion matrix as best implies selection of a relative cost trade-off (e.g., cost matrix). Sometimes an analyst can present alternative solutions to domain experts in this “backwards” way to discover what underlying cost trade-offs match their intuition. Intuition is exceedingly powerful—a psychologist friend Dan Elash calls it “unarticulated lessons from experience”—so uncovering it by eliciting preferences between alternative solutions can be a very useful technique to know.

If a tool does not allow for cost matrices and only maximizes PC, you can give important classes the proper weight by manipulating the data. Undersampling is when the majority class is sampled and its other excess cases are ignored. This is tough for many analysts to be willing to do, given the great value of data, unless the data are very plentiful. More common is oversampling, where the rare cases are simply duplicated. In the previous example, then, six more copies of the default cases would be made so they can have seven times their original influence on the overall model. Be very careful when sampling either way, however, as problems can ensue (as explained in Mistake 9 of Chapter 20). Especially remember to oversample only the cases in the training data, and make sure that none of their copies appear in the evaluation data.

For estimation problems, you can also duplicate the cases of important classes to increase their influence on the resulting model.

**Error Metric: Ranking**

The error metrics described in the preceding sections calculate a global score, over all the data, but some problems really require a local score. That is, they may need to pay attention only to cases near a boundary. For example, a model to estimate political leanings might need to work well only at the boundary between two parties, where one could go either way, if it is to be used to identify persuadable constituents where time or money spent reaching out to them might affect their affiliation. (Hence, the influence of “battleground” states when a winner-takes-all-delegates score function is used, and the ignoring of “safe/lost” states by both parties.) On the other hand, if you are trying to identify the peaks of the curve (e.g., who might donate or might volunteer for campaigning), then an extreme is the most interesting region. Still, in both cases, the exact accuracy of the rest of the population is not important.

To optimize an estimation problem of this form, you would have to define a custom cost (or score) function and use a global search algorithm (and a lot of time) to find the best model parameters (for a given model structure). But if your data mining tool allows