have a budgetary constraint[3] that may affect your strategy. Say you have 100,000 total customers and a budget of $40,000 for the marketing campaign. You want to use the modeling results (the profit curves in Figure 8-2) to figure out how best to spend your budget. What do you do in this case? Well, first you figure out how many offers you can afford to make. Each offer costs $5 so you can target at most $40,000/$5 = 8,000 customers. As before, you want to identify the customers most likely to respond, but each model ranks customers differently. Which model should you use for this campaign? 8,000 customers is 8% of your total customer base, so check the performance curves at $x$=8%. The best-performing model at this performance point is Classifier 1. You should use it to score the entire population, then send offers to the highest-ranked 8,000 customers.

In summary, from this scenario we see that adding a budgetary constraint causes not only a change in the operating point (targeting 8% of the population instead of 50%) but also a change in the choice of classifier to do the ranking.

## ROC Graphs and Curves

Profit curves are appropriate when you know fairly certainly the conditions under which a classifier will be used. Specifically, there are two critical conditions underlying the profit calculation:

1. The *class priors*; that is, the proportion of positive and negative instances in the target population, also known as the *base rate* (usually referring to the proportion of positives). Recall that Equation 7-2 is sensitive to $p(\mathbf{p})$ and $p(\mathbf{n})$.
2. The *costs and benefits*. The expected profit is specifically sensitive to the relative levels of costs and benefits for the different cells of the cost-benefit matrix.

If both class priors and cost-benefit estimates are known and are expected to be stable, profit curves may be a good choice for visualizing model performance.

However, in many domains these conditions are uncertain or unstable. In fraud detection domains, for example, the amount of fraud changes from place to place, and from one month to the next (Leigh, 1995; Fawcett & Provost, 1997). The amount of fraud influences the priors. In the case of mobile phone churn management, marketing campaigns can have different budgets and offers may have different costs, which will change the expected costs.

[3] Another common situation is to have a *workforce constraint*. It's the same idea: you have a fixed allocation of resources (money or personnel) available to address a problem and you want the most "bang for the buck." An example might be that you have a fixed workforce of fraud analysts, and you want to give them the top-ranked cases of potential fraud to process.

One approach to handling uncertain conditions is to generate many different expected profit calculations for each model. This may not be very satisfactory: the sets of models, sets of class priors, and sets of decision costs multiply in complexity. This often leaves the analyst with a large stack of profit graphs that are difficult to manage, difficult to understand the implications of, and difficult to explain to a stakeholder.

Another approach is to use a method that can accomodate uncertainty by showing the entire space of performance possibilities. One such method is the Receiver Operating Characteristics (ROC) graph (Swets, 1988; Swets, Dawes, & Monahan, 2000; Fawcett, 2006). A ROC graph is a two-dimensional plot of a classifier with false positive rate on the *x* axis against true positive rate on the *y* axis. As such, a ROC graph depicts relative trade-offs that a classifier makes between benefits (true positives) and costs (false positives). Figure 8-3 shows a ROC graph with five classifiers labeled **A** through **E**.

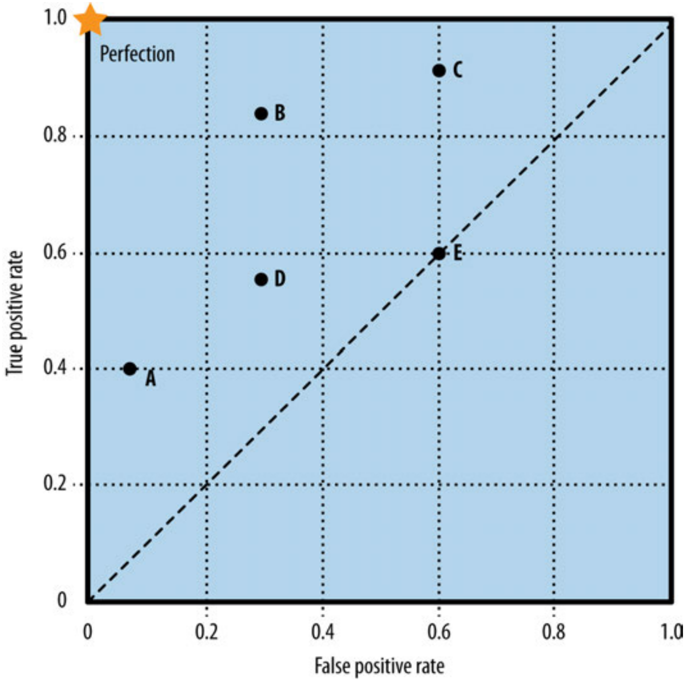


*Figure 8-3. ROC space and five different classifiers (A-E) with their performance shown.*

A *discrete* classifier is one that outputs only a class label (as opposed to a ranking). As already discussed, each such classifier produces a confusion matrix, which can be summarized by certain statistics regarding the numbers and rates of true positives, false positives, true negatives, and false negatives. Note that although the confusion matrix contains four numbers, we really only need two of the rates: either the true positive rate or the false negative rate, and either the false positive rate or the true negative rate. Given one from either pair the other can be derived since they sum to one. It is conventional to use the true positive rate (*tp rate*) and the false positive rate (*fp rate*), and we will keep to that convention so the ROC graph will make sense. Each discrete classifier produces an (*fp rate*, *tp rate*) pair corresponding to a single point in ROC space. The classifiers in Figure 8-3 are all discrete classifiers. Importantly for what follows, the *tp rate* is computed using only the actual positive examples, and the *fp rate* is computed using only the actual negative examples.

> Remembering exactly what statistics the *tp rate* and *fp rate* refer to can be confusing for someone who does not deal with such things on a daily basis. It can be easier to remember by using less formal but more intuitive names for the statistics: the *tp rate* is sometimes referred to as the *hit rate*—what percent of the actual positives does the classifier get right. The *fp rate* is sometimes referred to as the *false alarm rate* —what percent of the actual negative examples does the classifier get wrong (i.e., predict to be positive).

Several points in ROC space are important to note. The lower left point (0, 0) represents the strategy of never issuing a positive classification; such a classifier commits no false positive errors but also gains no true positives. The opposite strategy, of unconditionally issuing positive classifications, is represented by the upper right point (1, 1). The point (0, 1) represents perfect classification, represented by a star. The diagonal line connecting (0, 0) to (1, 1) represents the policy of guessing a class. For example, if a classifier randomly guesses the positive class half the time, it can be expected to get half the positives and half the negatives correct; this yields the point (0.5, 0.5) in ROC space. If it guesses the positive class 90% of the time, it can be expected to get 90% of the positives correct but its false positive rate will increase to 90% as well, yielding (0.9, 0.9) in ROC space. Thus a random classifier will produce a ROC point that moves back and forth on the diagonal based on the frequency with which it guesses the positive class. In order to get away from this diagonal into the upper triangular region, the classifier must exploit some information in the data. In Figure 8-3, **E**'s performance at (0.6, 0.6) is virtually random. **E** may be said to be guessing the positive class 60% of the time. Note that no classifier should be in the lower right triangle of a ROC graph. This represents performance that is worse than random guessing.

One point in ROC space is superior to another if it is to the northwest of the first (*tp rate* is higher and *fp rate* is no worse; *fp rate* is lower and *tp rate* is no worse, or both are better). Classifiers appearing on the lefthand side of a ROC graph, near the *x* axis, may be thought of as "conservative": they raise alarms (make positive classifications) only with strong evidence so they make few false positive errors, but they often have low true positive rates as well. Classifiers on the upper righthand side of a ROC graph may be thought of as "permissive": they make positive classifications with weak evidence so they classify nearly all positives correctly, but they often have high false positive rates. In Figure 8-3, **A** is more conservative than **B**, which in turn is more conservative than **C**. Many real-world domains are dominated by large numbers of negative instances (see the discussion in "Sidebar: Bad Positives and Harmless Negatives" on page 188), so performance in the far left-hand side of the ROC graph is often more interesting than elsewhere. If there are very many negative examples, even a moderate false alarm *rate* can be unmanageable. A ranking model produces a set of points (a curve) in ROC space. As discussed previously, a ranking model can be used with a threshold to produce a discrete (binary) classifier: if the classifier output is above the threshold, the classifier produces a **Y**, else an **N**. Each threshold value produces a different point in ROC space, as shown in Figure 8-4.

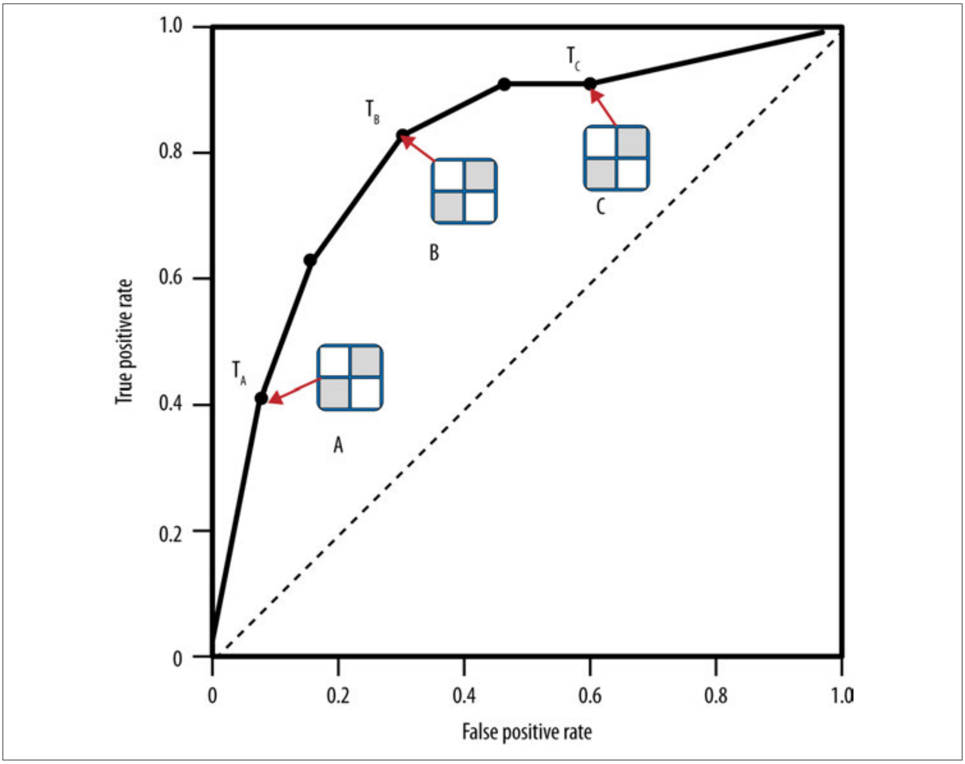
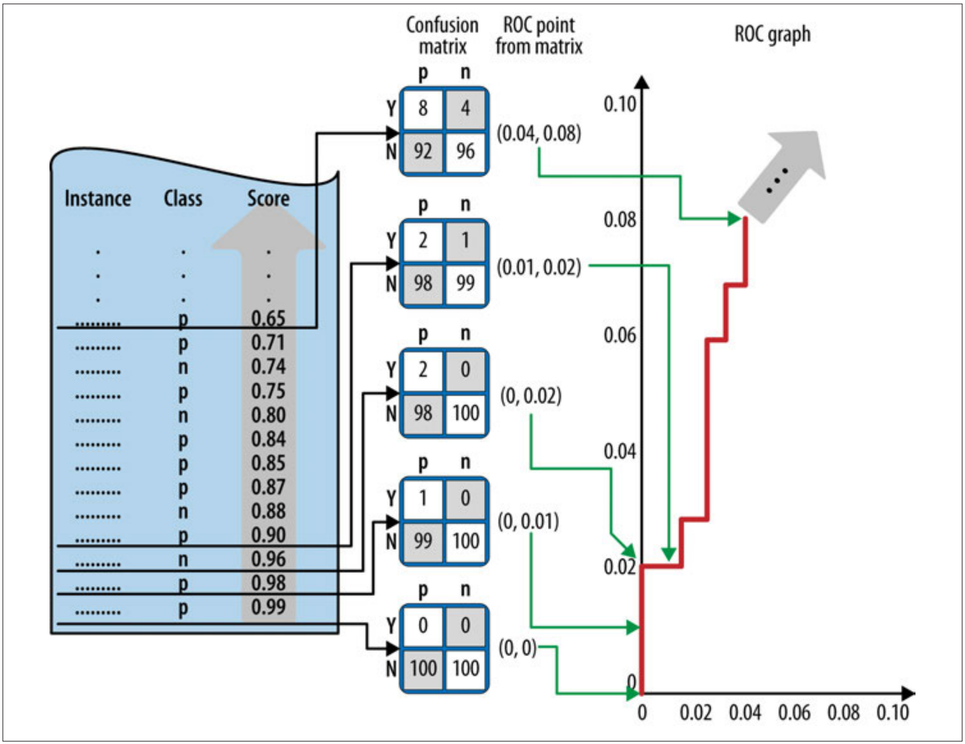


*Figure 8-4. Each different point in ROC space corresponds to a specific confusion matrix.*

Conceptually, we may imagine sorting the instances by score and varying a threshold from −∞ to +∞ while tracing a curve through ROC space, as shown in Figure 8-5.

Whenever we pass a positive instance, we take a step upward (increasing true positives); whenever we pass a negative instance, we take a step rightward (increasing false positives). Thus the "curve" is actually a step function for a single test set, but with enough instances it appears smooth.[4]



| Instance | Class | Score |
|---|---|---|
| . | . | . |
| . | . | . |
| . | . | . |
| ......... | p | 0.65 |
| ......... | p | 0.71 |
| ......... | n | 0.74 |
| ......... | p | 0.75 |
| ......... | n | 0.80 |
| ......... | p | 0.84 |
| ......... | p | 0.85 |
| ......... | p | 0.87 |
| ......... | n | 0.88 |
| ......... | p | 0.90 |
| ......... | n | 0.96 |
| ......... | p | 0.98 |
| ......... | p | 0.99 |

*Figure 8-5. An illustration of how a ROC "curve" (really, a stepwise graph) is constructed from a test set. The example set, at left, consists of 100 positives and 100 negatives. The model assigns a score to each instance and the instances are ordered decreasing from bottom to top. To construct the curve, start at the bottom with an initial confusion matrix where everything is classified as* **N***. Moving upward, every instance moves a count of 1 from the* **N** *row to the* **Y** *row, resulting in a new confusion matrix. Each confusion matrix maps to a (fp rate, tp rate) pair in ROC space.*

An advantage of ROC graphs is that they decouple classifier performance from the conditions under which the classifiers will be used. Specifically, they are independent of the class proportions as well as the costs and benefits. A data scientist can plot the

[4] Technically, if there are runs of examples with the same score, we should count the positive and negatives across the entire run, and thus the ROC curve will have a sloping step rather than square step.

performance of classifiers on a ROC graph as they are generated, knowing that the positions and relative performance of the classifiers will not change. The region(s) on the ROC graph that are of interest may change as costs, benefits, and class proportions change, but the curves themselves should not.

Both Stein (2005) and Provost & Fawcett (1997, 2001) show how the operating conditions of the classifier (the class priors and error costs) can be combined to identify the region of interest on its ROC curve. Briefly, knowledge about the range of possible class priors can be combined with knowledge about the cost and benefits of decisions; together these describe a family of tangent lines that can identify which classifier(s) should be used under those conditions. Stein (2005) presents an example from finance (loan defaulting) and shows how this technique can be used to choose models.

## The Area Under the ROC Curve (AUC)

An important summary statistic is the *area under the ROC curve* (AUC). As the name implies, this is simply the area under a classifier's curve expressed as a fraction of the unit square. Its value ranges from zero to one. Though a ROC curve provides more information than its area, the AUC is useful when a single number is needed to summarize performance, or when nothing is known about the operating conditions. Later, in "Example: Performance Analytics for Churn Modeling" on page 223, we will show a use of the AUC statistic. For now it is enough to realize that it's a good general summary statistic of the predictiveness of a classifier.

> As a technical note, the AUC is equivalent to the Mann-Whitney-Wilcoxon measure, a well-known ordering measure in Statistics (Wilcoxon, 1945). It is also equivalent to the Gini Coefficient, with a minor algebraic transformation (Adams & Hand, 1999; Stein, 2005). Both are equivalent to the probability that a randomly chosen positive instance will be ranked ahead of a randomly chosen negative instance.

## Cumulative Response and Lift Curves

ROC curves are a common tool for visualizing model performance for classification, class probability estimation, and scoring. However, as you may have just experienced if you are new to all this, ROC curves are not the most intuitive visualization for many business stakeholders who really ought to understand the results. It is important for the data scientist to realize that clear communication with key stakeholders is not only a primary goal of her job, but also is essential for doing the right modeling (in addition to doing the modeling right). Therefore, it can be useful also to consider visualization frameworks that might not have all of the nice properties of ROC curves, but are more intuitive. (It is important for the business stakeholder to realize that the theoretical