Data Mining Misconceptions #2: How Much Data …

By Tim Graettinger

"How much data do I need for data mining?" In my experience, this is the most-frequently-asked of all frequently-asked questions about data mining. It makes perfect sense that this is a concern – data is the raw material, the primary resource, for any data mining endeavor. Data can be difficult and expensive to collect, maintain, and distribute. And those activities are just the prerequisites for extracting any value from it via data mining.

The question of data quantity does not stand apart from data quality, sampling, and a host of other related issues. Entire books describe these other aspects in great detail. My goal is more modest: to discuss the detection, diagnosis, and treatment for the most common mistake related to data quantity - trying to do too much with the data at hand.

Working with Pat and Liam

Pat and Liam are long-time friends (and clients). Our first collaboration took place over ten years ago when they had a project that was in trouble. Pat and Liam had engaged a firm to build a direct marketing model to identify the best people to contact for an insurance product. The trouble was that the model performed well on the data used to build it, but its performance was terrible on other data held out for independent testing. A mutual friend introduced me to Pat and Liam, thinking that I might be able to help. Early in our initial conversation, Pat said, "We have tons of data, almost a million records from the first direct mail campaign." My spider-sense[1] was tingling. I had some suspicions and asked, "What was the overall response rate for that campaign?" Liam told me that it was about 0.1%, or 1 in 1000. My suspicion was that they were data-limited, in an important way. As we'll soon see, my suspicion was right on target.

The Quantity-Action Matrix

The Quantity-Action matrix in Table 1 outlines four situations that can occur, based on:

- the amount of data available (enough or not enough)
- the data mining action taken (do nothing or do something)

As evident from the table, two situations are not problems (doing nothing when sufficient data is not available and doing something when sufficient data is available). Two situations are problematic, however:

- doing nothing when sufficient data is available for data mining is a missed opportunity
- doing data mining (or more accurately, trying to do too much) with limited data is trouble

In the jargon of data mining, this latter problem is termed "overfitting", and it is my focus in this article.

		Data Mining Action	
		Do Nothing	**Do Something**
Data Quantity	**Enough**	Missed Opportunities	Non-Problem
	Not Enough	Non-Problem	Overfitting

Table 1 - The Quantity-Action Matrix

Detect, Diagnose, and then Treat…

Pat and Liam's problem had the classic symptom of overfitting – a model performs well on data used to build it, but poorly on data withheld for testing. As an analogy, think about custom-tailored clothes. They are cut very specifically for an individual, and they are unlikely to fit anyone else nearly as well, if at all. To understand how overfitting occurs in practice, we will use a greatly simplified example of data for a direct marketing model much like Pat and Liam's. We will use my favorite, three-step approach to problem solving: detect, diagnose, and treat. In doing so, I will illustrate overfitting and demonstrate how to fix it – so you can recognize and remedy it in your own work.

Step One: Detect

Good data mining practice requires splitting any data table into at least two segments[2]. One segment is used to build, or "train", the model. The other is a testing, or "hold out" segment that is used to validate the built model. Test data simulates how the model will likely perform in actual, production use. For simplicity in our example, we will randomly select 50% of the data for training and 50% for testing.

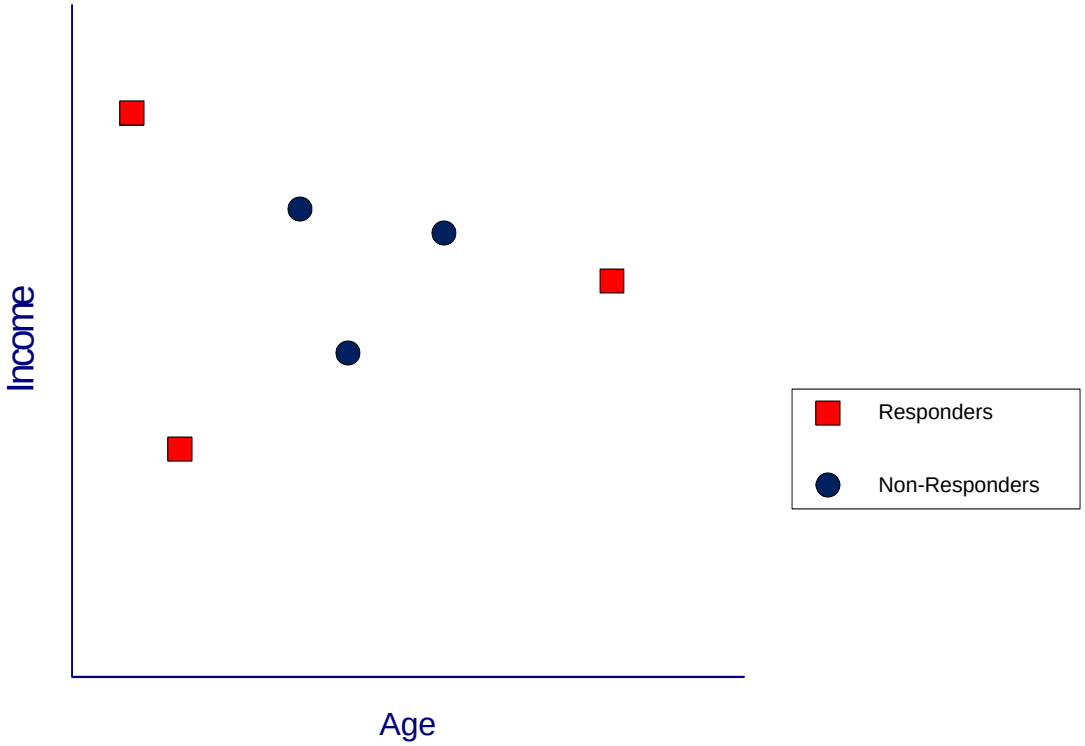

Figure 1 - Training data for the direct mail example

Figure 1 displays just the training data for our example. Red squares indicate persons who responded to a previous direct mail campaign, while the blue circles mark those who did not respond. For each person, we just have information about their age and income. The goal of data mining will be to build a mathematical model to predict, based on age and income, which other people are most likely to respond to a future mailing.

We'll use a common mathematical model that separates responders from non-responders by using straight lines. The three lines shown below in Figure 2 do a very nice job of dividing the age-income landscape into responder and non-responder regions, designated by the red "R" and the blue "N" markers, respectively. Notice that the model makes no mistakes

on the training data used to build it. See also that two parameters – slope and intercept - define each of the separating lines. Three lines, then, require 6 parameters (3 slopes and 3 intercepts).

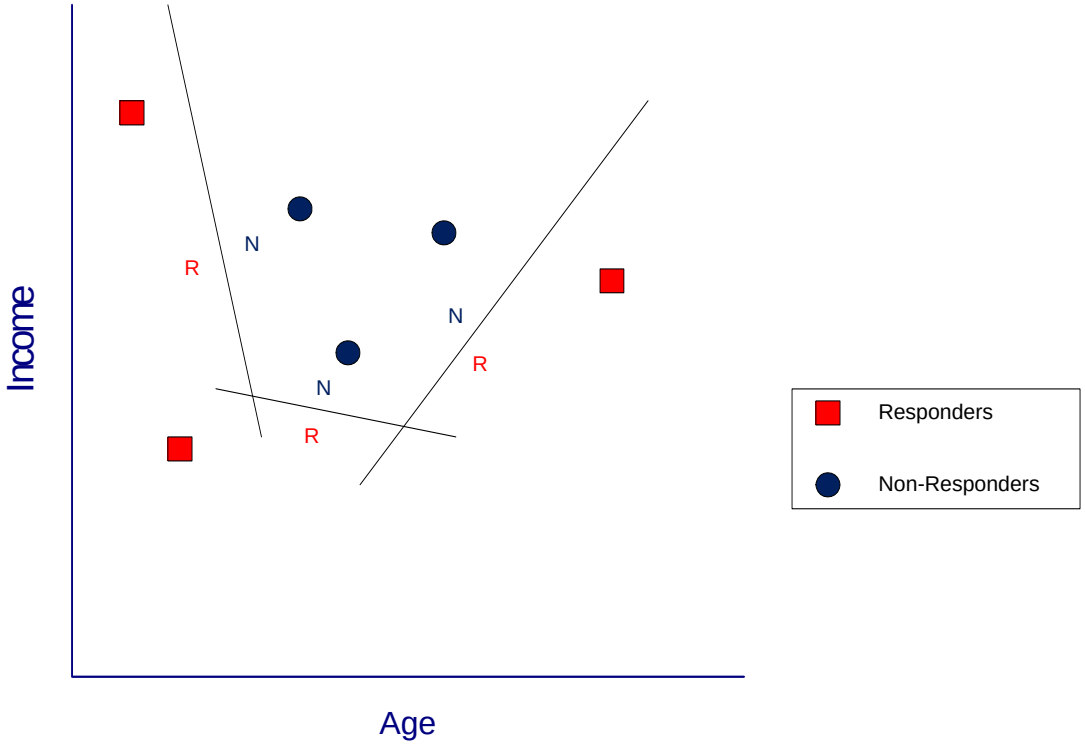

Figure 1 - Training data with straight lines separating responders and non-responders

Now comes the moment of truth – how does the model perform on the test data. In Figure 3 below, the model (the collection of separating lines) stays the same, but the data changes to be the testing set. The results are poor. Four of seven people are classified incorrectly (each is marked by an asterisk). Recall that no person in the training data was misclassified.

In this simple example, we detect a problem: poor performance on the testing data - which portends poor performance "in the real world". Without a testing set, we would be unaware that a problem exists until the model is put into service[3]. Fortunately, Pat and Liam detected a problem because they separated the data beforehand. However, they needed help starting with the next step, diagnosing the cause of the problem.

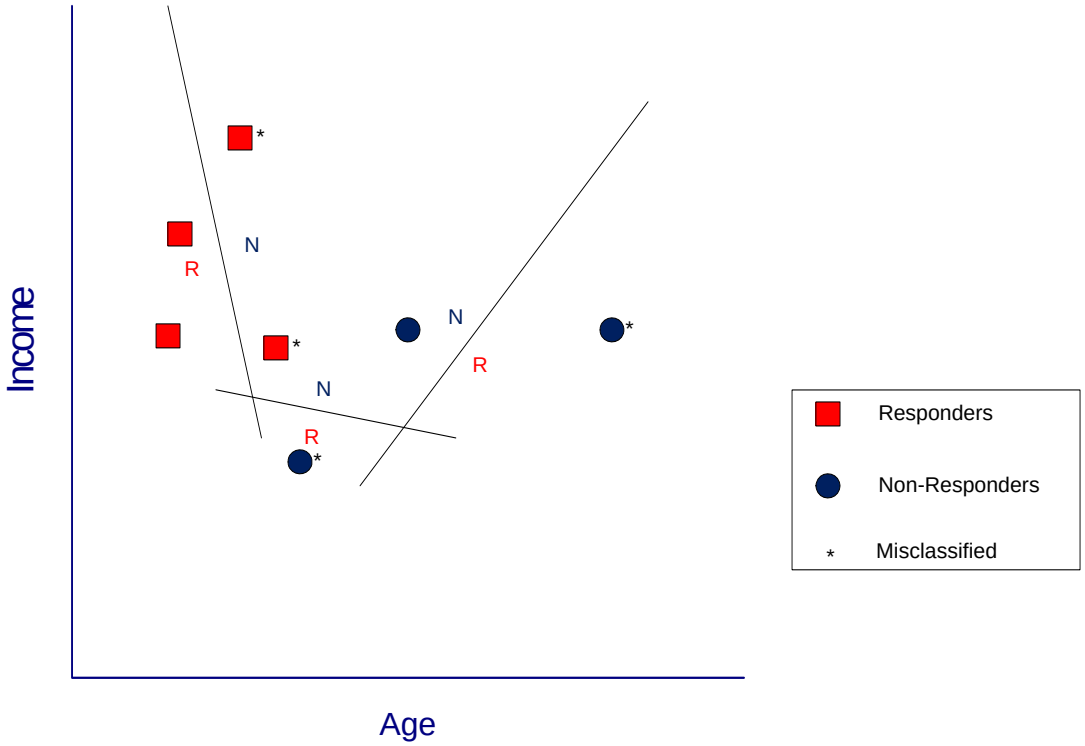

Figure 2 - Separating lines from the model applied to the testing data

Step Two: Diagnose

Diagnosing overfitting requires two key pieces of information:

- the number of records in the training set that belong to the minority group
- the number of parameters in the model

From these numbers we can compute the "fitting ratio", FR, as:

FR = 2 x number of minority records / number of model parameters

In the training data for our simple example, the minority group has 3 members[4]. As noted above, the number of parameters in the model is 6. Thus, the fitting ratio is 1. Small fitting ratios – less than 10 - are often red flags for overfitting. Small ratios mean that enough parameters may be available to dice up the landscape too finely, or overfit, the training data.

For comparison, Pat and Liam's real-world model was a complex, mathematical neural network. It consisted of about 2000 parameters. From my question about response rate, I could mentally estimate that the data set contained about 1000 responders. From these two numbers, I realized that their fitting ratio was close to 1. So, despite the fact that their overall data set contained around a million records, the real limiting factor

was the small number of responders, the minority group. Having diagnosed the problem, the next step was to fix it.

Step Three: Treat

Thoreau said it best, "Simplify, simplify, simplify!" The preferred treatment for overfitting is simplifying the model. This means reducing the number of parameters – which is the portion of the fitting ratio that is easiest to impact[5]. The simplified model is much more likely to be robust and perform similarly on training data and testing data. And this performance will be a good indicator of the actual performance under real-world conditions.

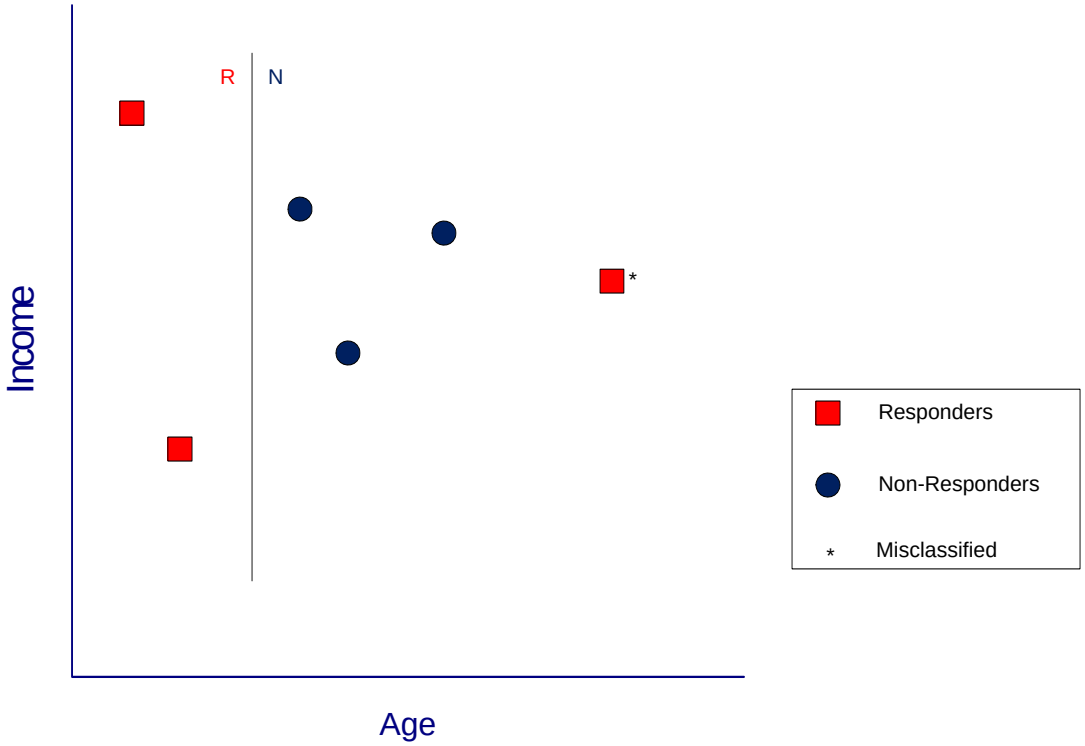

Figure 3 - New, simpler model built with the training data

For the example, I constructed the new, simpler model shown in Figure 4. It ignores income and classifies people just based on age (the result: a one-parameter model and a fitting ratio of 6). Notice that this model misclassifies one person in the training data. When applied to the testing data in Figure 5, the new model also makes only a single error – now that's more like it.

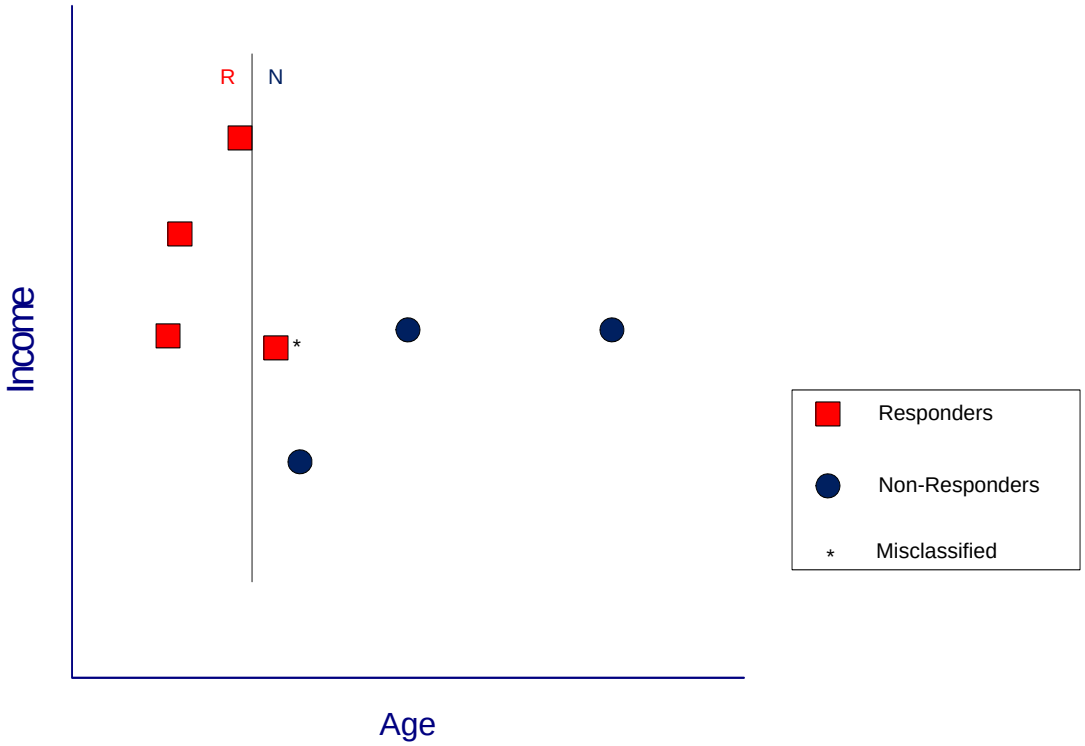

Figure 4 - New, simpler model applied to the testing data

A Hollywood Ending

Simplifying the model also solved the problem for Pat and Liam. By reducing the number of predictive factors (like age, income, etc.) and eliminating some complexity within the model, the fitting ratio was increased from 1 to about 100. The performance on testing data became nearly identical to that on training data. Most importantly, the model then did a terrific job of selecting prospects for the next mailing campaign.

The question of how much data you need for data mining is very complicated. But here's the good news: often you can use what you have, and then detect, diagnose, and treat problems if any develop. And you can often avoid many problems in the first place by building conservative, robust models with high fitting ratios.

Tim Graettinger, Ph.D., is the President of Discovery Corps, Inc. (http://www.discoverycorpsinc.com), a Pittsburgh-area company specializing in data mining, visualization, and predictive analytics.

Your comments and questions about this article are welcome. Please contact Tim at (724)-743-3642 or tgraettinger@discoverycorpsinc.com

---

[1] Spider-Man is a comic book superhero. Like me, his spider-sense tingles when he senses trouble.
[2] More sophisticated cross-validation schemes are often used in practice. But the simple train-test scheme with 50% of the data in each segment is adequate for discussion.
[3] Finding a problem after a model is put into service is no fun – and it's expensive.
[4] Since we have equal numbers of responders and non-responders in the example training data, we can choose either one as the minority group.
[5] You can go out and get more data, too, but that's not always an easy option. But it should be considered.