Data Mining Lessons from "Moneyball"

By Tim Graettinger

If you are a data miner, there's a good chance you saw the movie, *Moneyball*, starring Brad Pitt and Jonah Hill, based on the book by Michael Lewis. (If you missed it, the DVD is due to be released January 10, 2012). Pitt plays Billy Beane, the general manager of the Oakland Athletics baseball team. Beane is a former major league ballplayer himself, now tasked with building a competitive team on a shoestring budget.

Hill plays Peter Brand, Beane's assistant and "stats guy". You and I would immediately recognize him as a serious data miner. He totes his laptop everywhere, helping Beane make personnel decisions based on new, key metrics of a ballplayer performance. This is in stark contrast to the traditional, anecdote-based scouting approach used by the other major league teams.

For me, the movie was entertaining, but the book was really enlightening. I found it to be a source of numerous and familiar "lessons" about data mining - but couched in the less familiar context of major league baseball. In this article, I'd like to share some of these lessons with you.

Lessons 1, 2 & 3: Ask Questions, Pay Attention to Mistakes, and Don't Be Defensive

Early in my career as a data miner, I was conditioned to believe that my role was to answer questions. Someone would hand me a data set and say, "Tell me who our best customers are", or "Find out why fewer customers are renewing." I would trundle off to my cube and dig into the data to try to provide an answer. Have you had a similar experience?

As I worked on more and more projects, however, I found myself turning around to those clients and colleagues to ask questions – about the meaning of the data, about the population underlying the data, about the time frame of interest, about the outcome of interest, about the measures of success, etc. And I asked myself, "Am I asking too many questions?"

Luckily, I came across an article[1] by Kevin Kelly that made me feel good about asking so many questions. He wrote:

> *"Some day answers (correct answers!) will be so cheap that the really valuable things will be questions. A really good question will be worth a thousand correct answers."*

The story of *Moneyball* perfectly illustrates the value of asking pesky questions like:

> *"How did one of the [financially] poorest teams in baseball, the Oakland Athletics, win so many games?"* [2]

This question was pesky because it challenged the conventional wisdom, the mental-if-not-mathematical model that said:

> *"the [teams] with the most money often win."*[3]

The Oakland Athletics were a "bright spot" counter-example to this model. That is, they won many more games than expected, given their spending on players. There were also exceptions of a different type, what we might call "dark spots".

> *"The bottom of each division was littered with teams (Rangers, Orioles, Dodgers, and Mets) that had spent huge sums and failed."*[4]

Perhaps surprisingly, these big-spending failures were practically ignored. Instead, people in positions of power, like Major League Baseball Commissioner Bud Selig, were chafing about Oakland and:

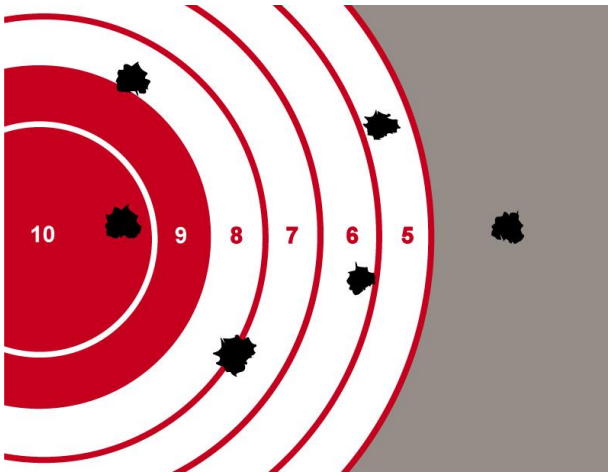
> *"were calling the Oakland A's success 'an aberration' - but that was less an explanation than an excuse not to grapple with the question."*[5]

The lesson from this portion of the book that resonates with me is to <u>pay attention to mistakes</u>[6] made by a model (mental or mathematical, simple or sophisticated, built by others or by me). We all need to grapple with questions like, "Is the model performing poorly on certain cases?" or "Is there any pattern to the errors?"

Here's a related example from my own work. Recently, I collaborated with a client who has a large contingent of field technicians, called "techs". These techs perform services on-site for both residential and commercial customers. My task was to build a model to estimate the time required for a tech to complete a given job, given data about the type of service requested and about the nature and location of the site.

I presented the model and some initial performance results to my client. She was politely happy with them, but she was really enthusiastic about digging deeper into the model errors – the jobs that were <u>not</u> predicted well. She wanted to understand those errors, to see if we could make the model even better.

Her enthusiasm infected me as well. As we drilled into the errors, we found something incredibly interesting: the biggest source of variation seemed to stem from the particular tech that was assigned to the job. (By the way, the tenure of the tech was already factored into the model. So the variation we saw was not due simply to differences in experience). For jobs with the same estimated time for completion - which we equated with "degree of difficulty" – one tech might take 15 minutes longer than predicted while another might take 10 minutes less. Further, these patterns were consistent across the full spectrum of job difficulty. That is, Fred was generally 20% slower across all of his jobs, adjusted for difficulty, while Sally was 25% quicker across her jobs.



Despite the model's imperfections, it served a valuable purpose as a consistent, benchmark arbiter of job time (analogous to the function of the more- money-equals-more-wins mental model in baseball). By paying attention to the model's mistakes, we identified people like Fred who might need more training. We also found "bright spots"[7], like Sally, who probably should be training Fred.

Another *Moneyball* lesson[8] for me related to this project was: don't be too defensive about the models that I build. Do you have this defensive tendency, too? (It sure seems like the major league baseball power brokers were extremely defensive about the more-money-equals-more-wins mental model). You spend a lot of time building a robust model. You sweat over the data representation, the transformations, the choice of model type, and a hundred more tiny-but-crucial details. Then, a client or colleague has the nerve, the unmitigated gall, to ask, "Why did you do that?" or "Can't the model predict any better?"

It's human nature, I think, to defend the choices you make and the effort you put into building a model. But, a model can be better, and more useful, and more valuable if you seriously scrutinize its flaws as well as its strengths. We need to get feedback and insight from multiple objective sources - not our pals. From this effort, we can really build understanding, which is our next topic.

Lessons 4 & 5: Keep an Outsider's Perspective and Build Transparent Models to Achieve Understanding

Bill James is a pivotal figure in the *Moneyball* saga, despite not being directly involved as an employee or consultant to any major league baseball team. For a number of years preceding the *Moneyball* timeframe, James had written (and

sometimes self-published) his annual *Baseball Abstract*. His *Abstracts* had become bibles of enlightened baseball thought. James was first and foremost a writer and thinker, even more than he was a "stats guy". He was an outsider with an outsider's perspective. But his writing influenced insiders, like Billy Beane and Paul DePodesta of the Oakland A's, to think and act differently from their peers.

> *"The whole point of James was [to] think for yourself along rational lines. Hypothesize, test against the evidence, never accept that a question has been answered as well as it ever will be. Don't believe a thing is true just because some [famous or high-ranking person] says that it is true."*[9]

Do you find, in your data mining work, that it can be an advantage not to be an expert in the business domain you're modeling? I do. I think it helps to keep an outsider's perspective. I think it helps to clearly define a hypothesis (ask a question!) and then let the data disprove it – or not[10]. That's the scientific method. And I think it helps to build models.

Bill James built a lot of models. In particular, he built a "run-created" model to predict the number of runs a baseball team would score in a season, given the collective offensive statistics of the players on the team.

> *"His model came far closer … to describing the run totals of every big league baseball team than anything the teams themselves had come up with. That, in turn, implied that professional baseball people had a false view of their offenses. It implied, specifically, that they didn't place enough value on walks and extra base hits, [factors present] in the runs-created model, and placed too much value on batting average and stolen bases [factors which didn't appear in the model]."*[11]

An important aspect of James' model is that it is "transparent". You can inspect the factors that are involved (like walks and extra-base hits) and not involved (like batting average and stolen bases) in his model. You can also study the relative importance, or weight, of the factors. The ability to inspect the factors and their weight can provide valuable insight – insight for making decisions about how to operate. For example, insight from the runs-created model influenced the Oakland A's to acquire undervalued players who walked and got extra base hits and to trade overvalued players who had high batting averages and/or who stole a lot of bases.

For a recent insurance project, I built a model to predict churn – that is, to predict who would renew their policy and who would defect to another carrier. Because the model was transparent, my client and I saw clearly that calls to customer service were key indicators of future churn. In particular, calls classified as complaints indicated that a customer had "one foot out the door". This knowledge influenced my client to define and implement a new business rule – even before the model was deployed for general use. The rule states: a retention specialist must make a follow-up call within 24 hours to any customer calling with a complaint. The reduction in churn has been immediate and significant.

To summarize, as in the case above, a model is more than a statistic or a formula. It encapsulates a hard-won nugget of knowledge and understanding about how things work.

> *"What James's wider audience had failed to understand was that statistics were beside the point. The point was understanding. 'I wonder', James wrote, 'if we haven't become so numbed by all these numbers that we are no longer capable of truly assimilating any knowledge which might result from them.'"*[12]

Keeping an outsider's perspective and building transparent models will help you and your clients achieve understanding. 'Nuff said.

Play Ball!

In this article, we discussed a handful of lessons gleaned from *Moneyball*, the popular book and movie. I hope both you and I will remember to:

- ask questions
- pay attention to mistakes
- not be defensive about our models
- keep an outsider's perspective
- build transparent models

If we do these things, maybe Brad Pitt will play one of us in a future big-screen production.

Tim Graettinger, Ph.D., is the President of Discovery Corps, Incorporated (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] Kevin Kelly is a "big thinker" who is well worth reading whenever you can. See http://www.kk.org/thetechnium/archives/2004/11/when_answers_ar.php for more details about "When Answers Are Cheap."

[2] "Moneyball", by Michael Lewis, *p. xi. Note that this citation, and those that follow, refer to the paperback edition of the book. Page numbers in the hardcover edition will be different.*

[3] Ibid, *p. xii.*

[4] Ibid, *p. xii.*

[5] Ibid, *p. xii.*

[6] These "mistakes" might be called by different names in different contexts. They might be called outliers, or exceptions, or anomalies, or errors, or worst of all, failures. Most people don't like to use any of these words.

[7] For an illuminating discussion of the value of looking for bright spots, see "Switch: How to Change Things When Change Is Hard" by Chip Heath and Dan Heath (http://www.heathbrothers.com/switch/).

[8] This is a tough lesson for me, one that I've had to wrestle with many times.

[9] "Moneyball", by Michael Lewis, *p.98*

[10] Remember, we can't prove anything with data. We can only reject or disprove a hypothesis.

[11] "Moneyball", by Michael Lewis, *p.77*

[12] Ibid, p. 95