CHAPTER 15
The Students Speak

> Every algorithm is editorial.
>
> — Emily Bell (director of the Tow Center for Digital Journalism at Columbia's Graduate School of Journalism)

We invited the students who took Introduction to Data Science version 1.0 to contribute a chapter to the book. They chose to use their chapter to reflect on the course and describe how they experienced it. Contributors to this chapter are Alexandra Boghosian, Jed Dougherty, Eurry Kim, Albert Lee, Adam Obeng, and Kaz Sakamoto.

Process Thinking

When you're learning data science, you can't start anywhere except the cutting edge.

An introductory physics class will typically cover mechanics, electricity, and magnetism, and maybe move on to some more "modern" subjects like special relativity, presented broadly in order of ascending difficulty. But this presentation of accumulated and compounded ideas in an aggregated progression doesn't give any insight into, say, how Newton actually came up with a differential calculus. We are not taught about his process; how he got there. We don't learn about his tools or his thoughts. We don't learn which books he read or whether he took notes. Did he try to reproduce other people's proofs? Did he focus on problems that followed from previous writing? What exactly made him think, "I just can't do this without infinitesimals?" Did

Newton need scratch paper? Or did the ideas emerge somehow fully formed when he saw an apple drop? These things aren't taught, but they have to be learned; this process is what the fledgling scientists will actually have to do.

Rachel started the first Introduction to Data Science class with a hefty caveat. Data science is still being defined, in both industry and academia. In each subsequent lecture, we learned about substantive problems and how people decide what problems to study. Substantively, the weekly lectures covered the tools and techniques of a data scientist, but each lecturer had their own style, background, and approach to each problem. In almost every class, the speaker would say something like, "I don't know what you guys have covered so far, but…" The lectures were discretized in this way, and it was our job to interpolate a continuous narrative about data science. We had to create our own meaning from the course, just as data scientists continue to construct the field to which they belong.

That is not to say that Rachel left us in the dark. On the first day of class, she proposed a working definition of data science. A data scientist, she said, was a person whose aptitude was distributed over the following: mathematics, statistics, computer science, machine learning, visualization, communication, and domain expertise. We would soon find out that this was only a prior in our unfolding Bayesian understanding. All the students and each lecturer evaluated themselves in terms of this definition, providing at once a diverse picture of the data science community and a reference point throughout the course. The lecturers came from academia, finance, established tech companies, and startups. They were graduate school dropouts, Kaggle competition winners, and digital artists. Each provided us with a further likelihood ratio. The class itself became sort of an iterative definition of data science.

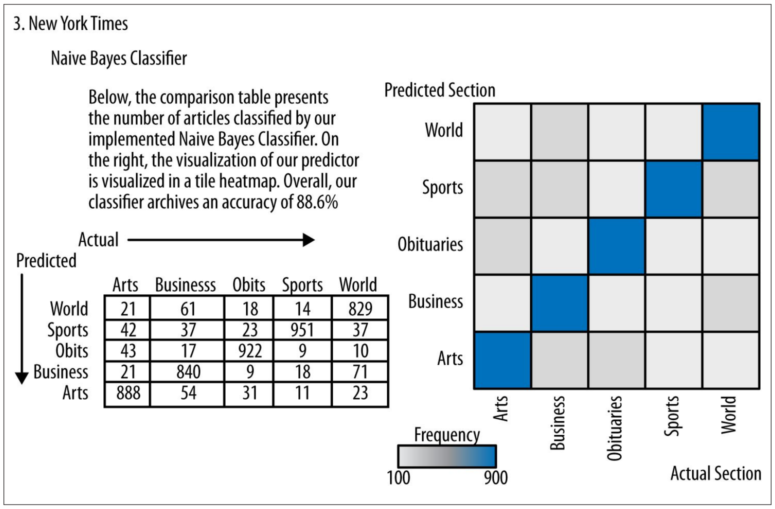
But we didn't just listen to people talk about their jobs week after week. We learned the tools of the trade with lots of difficult, head-to-table homework. Sometimes the assignments required us to implement the techniques and concepts we had discussed in the lectures. Other times we had to discover and use skills we didn't even know existed.

What's more, we were expected to deal with messy real-world data. We often worked on industry-related problems, and our homework was to be completed in the form of a clear and thoughtful report—something we could pride ourselves in presenting to an industry professional. Most importantly we often had little choice but to reach out beyond our comfort zones to one another to complete assignments. The social nature of data science was emphasized, and in addition to the formal groupings for assignments and projects, Rachel would often take us to the bar across the street along with whoever had presented that evening.[1] We worked and drank together throughout the course, learning from one another and building our skills together.

Naive No Longer

Our reckoning came on the third subsection of our second homework assignment, "Jake's Exercise: Naive Bayes for Article Classification" on page 109. It required us to download 2,000 articles from the *New York Times*—which only allowed a pull of 20 articles at a time—and train a simple Naive Bayes classifier to sort them by the section of the newspaper in which they appeared. Acquiring the articles was only half the challenge. While there are packages for many languages that implement Naive Bayes, we had to write our own code, with nothing more than a few equations for guidance. Not that using an existing package would not have helped us. Unlike them, our version had to include tunable regularization hyperparameters. It also demanded that we classify the articles across five categories instead of two. As we were told, simple Naive Bayes isn't particularly naive, nor very Bayesian. Turns out it's not that simple, either. Still, those of us who stayed in the class through the pain of spending 40 hours "polishing" 300 lines of disgustingly hacky R code got the pleasure of seeing our models graze 90% predictive accuracy. We were instantly hooked. Well, maybe it was the sunk cost. But still, we were hooked. Figure 15-1 shows one student's solution to the homework.

[1] Rachel's note: it was a graduate-level course. I made sure students were of age and that there were non-alcoholic options.



3. New York Times

Naive Bayes Classifier

Below, the comparison table presents the number of articles classified by our implemented Naive Bayes Classifier. On the right, the visualization of our predictor is visualized in a tile heatmap. Overall, our classifier archives an accuracy of 88.6%

Actual →

Predicted ↓

	Arts	Businesss	Obits	Sports	World
World	21	61	18	14	829
Sports	42	37	23	951	37
Obits	43	17	922	9	10
Business	21	840	9	18	71
Arts	888	54	31	11	23

Figure 15-1. Part of a student's solution to a homework assignment

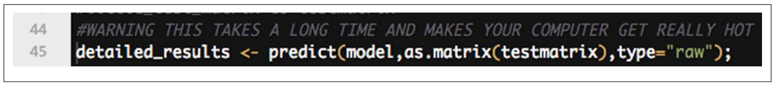
The Kaggle competition that made up half of our final project was a particular opportunity to go off the beaten track. The final was designed as a competition between the members of the class to create an essay-grading algorithm. The homework often simulated the data scientist's experience in industry, but Kaggle competitions could be described as the dick-measuring contests of data science. It encouraged all of the best data science practices we had learned while putting us in the thick of a quintessential data science activity. One of the present authors' solutions ended up including as features (among others) the number of misspellings, the number of words on the Dale-Chall list of words a fourth-grader should understand, TF-IDF vectors of the top 50 words, and the fourth root of the number of words in the essay. Don't ask why. The model used both random forest and gradient boosted models, and implemented stochastic hyperparameter optimization, training fifty thousand models across thousands of hours on Amazon EC2 instances. It worked OK.

Helping Hands

In one of the first classes, we met guest lecturer Jake Hofman. Do you remember seeing your first magic trick? Jake's lecture? Yeah, it was like that—as prestidigitatory as any close-up act. By using a combination of simple Unix utilities and basic data, he built a Naive Bayes spam classifier before our very eyes. After writing a few math equations on the board, he demonstrated some of his “command-line-fu” to parse the publicly available Enron emails he downloaded live.

In the weekly lectures, we were presented with standing-ovation-inducing performers. But it was with the help of Jared Lander and Ben Reddy, who led our supplementary lab sessions, that we stayed afloat in the fast-paced class. They presented to us the mechanics of data science. We covered the gamut: statistical concepts ranging from linear regression to the mechanics behind the random forest algorithm. And many of us were introduced to new tools: regular expressions, LaTeX, SQL, R, the shell, git, and Python. We were given the keys to new sources of data through APIs and web-scraping.

Generally, the computer scientists in the class had to quickly pick up the basics of theory-based feature selection and using R. Meanwhile, the social scientists had to understand the mechanics of a database and the differences between globally and locally scoped variables, and the finance majors had to learn ethics. We all had our burdens. But as we all methodically yet mistakenly constructed for loops in R and considered the striking inefficiencies of our code, our bags of tricks became a little heavier. Figure 15-2 shows one of the lessons or pitfalls of this method:



```
#WARNING THIS TAKES A LONG TIME AND MAKES YOUR COMPUTER GET REALLY HOT
detailed_results <- predict(model,as.matrix(testmatrix),type="raw");
```

Figure 15-2. Lesson learned

And as our skills improved, we were increasingly able to focus on the analysis of the data. Eventually, we didn’t even see the code—just the ideas behind it.

But how could we figure these things out on our own? Could any one of us do it all?

It took hours of frustrating errors and climbing learning curves to appreciate the possibilities of data science. But to actually finish our homework on time, we had to be willing to learn from the different backgrounds of the students in the class.

In fact, it became crucial for us to find fellow students with complementary skills to complete our assignments. Rachel forced group work upon us, not by requiring it directly, but by assigning massive yet discretizable assignments. It turned out that she meant for us to know that practicing data science is inherently a collective endeavor. In the beginning of the course, Rachel showed us a hub-and-spoke network diagram. She had brought us all together and so was at the center. The spokes connected each of us to her. It became her hope that new friendships/ideas/projects/connections would form during the course.

It's perhaps more important in an emergent field than in any other to be part of a community. For data science in particular, it's not just useful to your career—it's essential to your practice. If you don't read the blogs, or follow people on Twitter, or attend meetups, how can you find out about the latest distributed computing software, or a refutation of the statistical approach of a prominent article? The community is so tight-knit that when Cathy was speaking about MapReduce at a meetup in April, she was able to refer a question to an audience member, Nick Avteniev—easy and immediate references to the experts of the field is the norm. Data science's body of knowledge is changing and distributed, to the extent that the only way of finding out what you should know is by looking at what other people know. Having a bunch of different lecturers kickstarted this process for us. All of them answered our questions. All gave us their email addresses. Some even gave us jobs.

Having listened to and conversed with these experts, we formed more questions. How can we create a time series object in R? Why do we keep getting errors in our plotting of our confusion matrix? What the heck is a random forest? Here, we not only looked to our fellow students for answers, but we went to online social communities such as Stack Overflow, Google Groups, and R bloggers. It turns out that there is a rich support community out there for budding data scientists like us trying to make our code run. And we weren't just getting answers from others who had run into the same problems before us. No, these questions were being answered by the pioneers of the methods. People like Hadley Wickham, Wes McKinney, and Mike Bostock were providing support for the packages they themselves wrote. Amazing.

Your Mileage May Vary

It's not as if there's some platonic repository of perfect data science knowledge that you can absorb by osmosis. There are various good practices from various disciplines, and different vocabularies and interpretations for the same method (is the regularization parameter a prior, or just there for smoothing? Should we choose it on principled grounds or in order to maximize fit?) There is no institutionalized knowledge because there are no institutions, and that's why the structure of interactions matters: you can can create your own institutions. You choose who you are influenced by, as Gabriel Tarde put it (via Bruno Latour, via Mark Hansen):

> When a young farmer, facing the sunset, does not know if he should believe his school master asserting that the fall of the day is due to the movement of the earth and not of the sun, or if he should accept as witness his senses that tell him the opposite, in this case, there is one imitative ray, which, through his school master, ties him to Galileo.
>
> — Gabriel Tarde

Standing on the shoulders of giants is all well and good, but before jumping on someone's back you might want to make sure that they can take the weight. There is a focus in the business world to use data science to sell advertisements. You may have access to the best dataset in the world, but if the people employing you only want you to find out how to best sell shoes with it, is it really worth it?

As we worked on assignments and compared solutions, it became clear that the results of our analyses could vary widely based on just a few decisions. Even if you've learned all the steps in the process from hypothesis-building to results, there are so many ways to do each step that the number of possible combinations is huge. Even then, it's not as simple as piping the output of one command into the next. Algorithms are editorial, and the decision of which algorithm and variables to use is even more so.

Claudia Perlich from Media 6 Degrees (M6D) was a winner of the prestigious KDD Cup in 2003, 2007, 2008, 2009, and now can be seen on the coordinating side of these competitions. She was generous enough to share with us the ins and outs of the data science circuit and the different approaches that you can take when making these editorial decisions. In one competition to predict hospital treatment outcomes, she had noticed that patient identifiers had been assigned sequentially, such that all the patients from particular clinics had sequential numbers. Because different clinics treated patients with different severities of condition, the patient ID turned out to be a great predictor for the outcome in question. Obviously, the inclusion of this data leakage was unintentional. It made the competition trivial. But in the real world, perhaps it should actually be used in models; after all, the clinic that better? Actually, who cares how well we perform on what is, essentially, our training data?

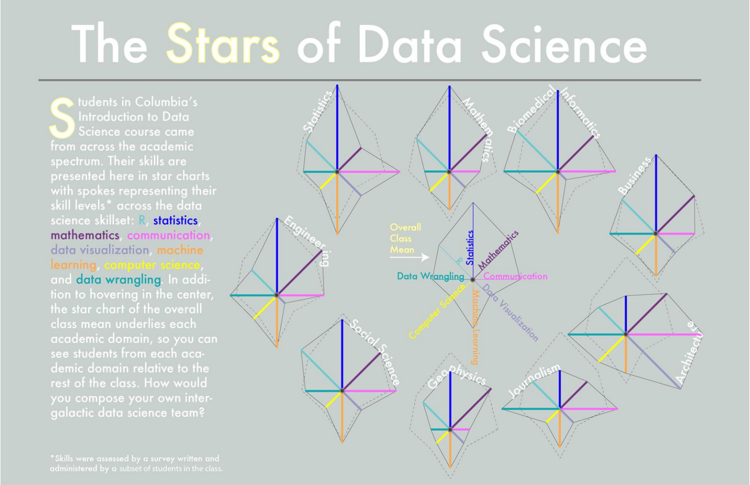
This is not completely fair to Hastie and his coauthors. They would probably argue that if students wanted to learn about data scraping and organization, they should get a different book that covers those topics—the difference in the problems shows the stark contrast in approach that this class took from normal academic introductory courses. The philosophy that was repeatedly pushed on us was that understanding the statistical tools of data science without the context of the larger decisions and processes surrounding them strips them of much of their meaning. Also, you can't just be told that real data is messy and a pain to deal with, or that in the real world no one is going to tell you exactly which regression model to use. These issues—and the intuition gained from working through them—can only be understood through experience.

Bridging Tunnels

As fledgling data scientists, we're not—with all due respect to Michael Driscoll—exactly civil engineers. We don't necessarily have a grand vision for what we're doing; there aren't always blueprints. Data scientists are adventurers, we know what we're questing for, we've some tools in the toolkit, and maybe a map, and a couple of friends. When we get to the castle, our princess might be elsewhere, but what matters is that along the way we stomped a bunch of Goombas and ate a bunch of mushrooms, and we're still spitting hot fire. If science is a series of pipes, we're not plumbers. We're the freaking Mario Brothers.

Some of Our Work

The students improved on the original data science profile from back in Chapter 1 in Figure 15-3 and created an infographic for the growing popularity of data science in universities in Figure 15-4, based on information available to them at the end of 2012.

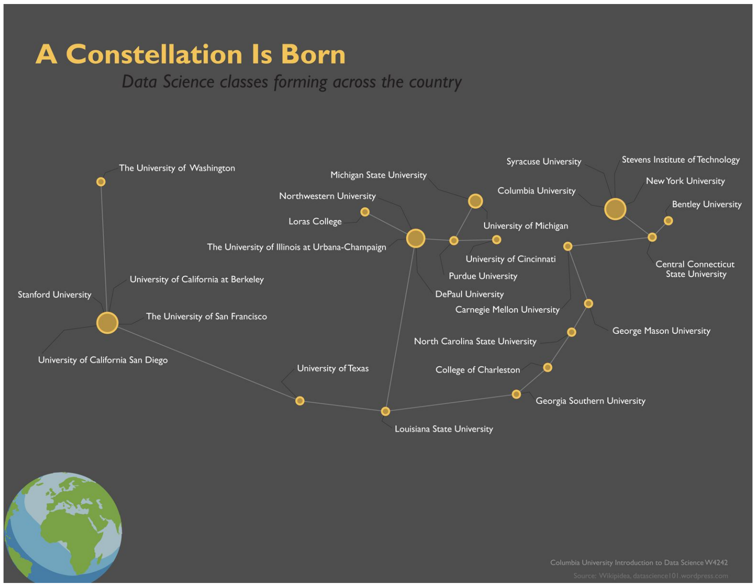


The Stars of Data Science

Students in Columbia's Introduction to Data Science course came from across the academic spectrum. Their skills are presented here in star charts with spokes representing their skill levels* across the data science skillset: R, statistics, mathematics, communication, data visualization, machine learning, computer science, and data wrangling. In addition to hovering in the center, the star chart of the overall class mean underlies each academic domain, so you can see students from each academic domain relative to the rest of the class. How would you compose your own intergalactic data science team?

*Skills were assessed by a survey written and administered by a subset of students in the class.

Figure 15-3. The Stars of Data Science (a collaborative effort among many students including Adam Obeng, Eurry Kim, Christina Gutierrez, Kaz Sakamoto, and Vaibhav Bhandari)



A Constellation Is Born

Data Science classes forming across the country

The University of Washington
Michigan State University
Syracuse University
Stevens Institute of Technology
New York University
Northwestern University
Columbia University
Bentley University
Loras College
University of Michigan
The University of Illinois at Urbana-Champaign
University of Cincinnati
Central Connecticut State University
University of California at Berkeley
Purdue University
Stanford University
DePaul University
The University of San Francisco
Carnegie Mellon University
George Mason University
North Carolina State University
University of California San Diego
University of Texas
College of Charleston
Georgia Southern University
Louisiana State University

Columbia University Introduction to Data Science W4242

Source: Wikipidea, datascience101.wordpress.com

Figure 15-4. A Constellation is Born (Kaz Sakamoto, Eurry Kim and Vaibhav Bhandari created this as part of a larger class collaboration)