Goodbye, Data Science

Posted on November 27, 2022 by W.D.

This is more of a personal post than something intended to be profound. If you are looking for a point, you will not find one here. Frankly I am not even sure who the target audience is for this (probably "data scientists who hate themselves"?).

I had been a data scientist for the past few years, but in 2022, I got a new job as a data engineer, and it's been pretty good to me so far.

I'm still working alongside "data scientists," and do a little bit of that myself still, but most of my "data science" work is directing and consulting on others' work. I've be focusing more on implementation of data science ("MLops") and data engineering.

The main reason I soured on data science is that the work felt like it didn't matter, in *multiple* senses of the words "didn't matter":

- The work is downstream of engineering, product, and office politics, meaning the work was only often as good as the weakest link in that chain.
- Nobody knew or even cared what the difference was between good and bad data science work. Meaning you could absolutely suck at your job or be incredible at it and you'd get nearly the same regards in either case.
- The work was often very low value-add to the business (often compensating for incompetence up the management chain).
- When the work's value-add exceeded the labor costs, it was often personally unfulfilling (e.g. tuning a parameter to make the business extra money).

Shitty management & insane projects

Management was by far my biggest gripe. I am completely exhausted by the absolute insanity that was the tech industry up to 2021. Companies all over were consistently pursuing things that could be reasoned about *a priori* as being insane ideas– ideas any decently smart person should know wouldn't work before they're tried. Some projects could have saved whole person-years of labor had anyone possessed a better understanding of the business, customers, broader economic / social landscape, financial accounting, and (far too underrated in tech) any relevant subject matter areas.

Those who have seen my Twitter posts know that I believe the role of the data scientist in a scenario of insane management is not to provide real, honest consultation, but to launder these insane ideas as having some sort of basis in objective reality even if they don't. Managers will say they want to make data-driven decisions, but they really want decision-driven data. If you strayed from this role– e.g. by warning people not to pursue stupid ideas– your reward was their disdain, then they'd do it anyway, then it wouldn't work (what a shocker). The only way to win is to become a stooge.

The reason managers pursued these insane ideas is partly because they are hired despite not having any subject matter expertise in business or the company's operations, and partly because VC firms had the strange idea that ballooning costs well in excess of revenue was "growth" and therefore good in all cases; the business equivalent of the Flat Earth Society. It was also beneficial for one's personal career growth to manage an insane project (résumé lines such as "managed $10 million in top-line revenue," failing to disclose that their COGS was $30 million). Basically, there's a decent reward for succeeding and no downside for failing, and sometimes you will even be rewarded for your failures! So why not do something insane?

Also, it seems that VC firms like companies to run the same way their

portfolios do– they want companies to try 100 different things, and if only 5 out of those 100 things work, then the VCs will consider that a success. On the ground floor, this creates a lot of misery, since the median employee at the company is almost certainly working on a product that is not destined to perform well, but the shareholders are happy, which is of course all that matters.

Shitty code & shitty data science

The median data scientist is horrible at coding and engineering in general. The few who are remotely decent at coding are often not good at engineering in the sense that they tend to over-engineer solutions, have a sense of self-grandeur, and want to waste time building their own platform stuff (folks, do not do this).

This leads to two feelings on my end:

1. It got annoying not having some amount of authority over code and infra decisions. Working with data scientists without having control over infra feels like wading through piles of immutable shit.
2. It was obvious that there is a general industry-wide need for people who are good at both data science and coding to oversee firms' data science practices in a technical capacity.

Poor mentorship

I don't want to be *too* snooty: in a sense, it's fine for data scientists to suck at coding! Especially if they bring other valuable skills to the table, or if they're starting out. And in another sense, bad code getting into production is a symptom of bad team design and management, more than any individual contributors' faults! By describing the median data scientist's coding skills as shitty, I'm just trying to be honest, not scornful.

The problem is that the median data scientist works at a small to medium-

sized company that doesn't build their data science practices around a conceit that the data scientists' code will suck. They'd rather let a 23 year old who knows how to `pip install jupyterlab` run loose and self-manage, or manage alongside other similarly situated 23 year-olds. Where is the adult in charge?

23 year-old data scientists should probably not work in start-ups, frankly; they should be working at companies that have actual capacity to on-board and delegate work to data folks fresh out of college. So many careers are being ruined before they've even started because data science kids went straight from undergrad to being the third data science hire at a series C company where the first two hires either provide no mentorship, or provide shitty mentorship because they too started their careers in the same way.

Poor self-directed education

On the other hand, it's not just the companies' and managers' faults; individual data scientists are also to blame for being really bad at career growth. This is not contemptible for people who are just starting out their careers, but at some point folks' résumés starts to outpace actual accumulation of skills and I cannot help but to find that a teeny bit embarrassing.

It seems like the main career growth data scientists subject themselves to is learning the API of some gradient boosting tool or consuming superficial + shallow + irrelevant knowledge. I don't really sympathize with this learning trajectory because I've never felt the main bottleneck to my work was that I needed some gradient boosting tool. Rather the main bottlenecks I've faced were always crappy infrastructure and lacking (quality) data, so it has always felt natural to focus my efforts toward learning that stuff to unblock myself.

My knowledge gaps have also historically been less grandiose than learning how some state of the art language model works or pretending I understand some arXiv white paper ornate with LaTeX notation of advanced calculus.

Personally, I've benefited a ton from reading the first couple chapters out of advanced textbooks (while ignoring the last 75% of the textbook), and refreshing on embarrassingly pedestrian math knowledge like "how do logarithms work." Yeah I admit it, I'm an old man and I've had to refresh on my high school pre-calc. Maybe it's because I have 30k Twitter followers, but I live in constant anxiety that someone will pop quiz me with questions like "what is the formula for an F-statistic," and that by failing to get it right I will vanish in a puff of smoke. So my brain tells me that I must always refresh myself on the basics. I admit this is perhaps a strange way to live one's life, but it worked for me: after having gauged my eyes out on linear regression and basic math, it's shockingly apparent to me how much people merely pretend to understand this stuff, and how much ostensible interest in more advanced topics is pure sophistry.

For the life of me I cannot see how reading a blog post that has sentences in it such as "DALL-E is a diffusion model with billions of parameters" would ever be relevant to my work. The median guy who is into this sort of superficial content consumption hasn't actually gone through chapters in an advanced textbook in years if ever. Don't take them at their word that they've actually grinded through the math because people lie about how well-read they are *all the time* (and it's easy to tell when people are lying). Like bro, you want to do stuff with "diffusion models"? You don't even know how to add two normal distributions together! You ain't diffusing shit!

I don't want to blame people for spending their free-time doing things other than learning how to code or doing math exercises out of grad school textbooks. To actually become experts in multiple things is oppressively time-consuming, and leaves little time for other stuff. There's more to life than your dang job or the subject matters that may be relevant to your career. One of the main sins of "data scientist" jobs is that it expects far too much from people.

But there's also a part of me that's just like, how can you not be curious?

How can you write Python for 5 years of your life and never look at a bit of source code and try to understand how it works, why it was designed a certain way, and why a particular file in the repo is there? How can you fit a dozen regressions and not try to understand where those coefficients come from and the linear algebra behind it? I dunno, man.

Ultimately nobody really knows what they are doing, and that's OK. But between companies not building around this observation, and individuals not self-directing their educations around this observation, it is just a bit maddening to feel stuck in stupid hell.

Data engineering has been relaxing

These are the things I've been enjoying about data engineering:

- Sense of autonomy and control.
    - By virtue of what my job is, I have tons of control over the infrastructure.
    - Data engineering feels significantly less subject to the whims and direction of insane management.
- Less need for sophistry.
    - My work is judged based on how good the data pipelines are, not based on how good looking my slide decks are or how many buzzwords I can use in a sentence. Not to say data engineering doesn't have buzzwords and trends, but that's peddled by SaaS vendors more than actual engineers.
- More free time.
    - I dunno, it feels like becoming a data engineer cured my imposter syndrome? I feel like I have more ability to dick around in my spare time without feeling inadequate about some aspect of my job or expertise. But this is probably highly collinear with not being a lackey for product managers.
- Obvious and immediate value that is not tied to a KPI.

    - I like being valued, what can I say.
    - Ultimately the data scientists need me more than I need them; I'm the reason their stuff is in production and runs smoothly.
    - I have a sense that, if my current place of business needed to chop employees, that it would be a dumb decision to chop me over any data scientist.
- Frankly, I feel really good at what I do.
    - As someone who has worked a variety of downstream data-related jobs, I have both a very strong sense of what my downstream consumers want, as well as the chops to QC/QA my own work with relative ease the way a good analyst would.
    - At my last company I had a lot of "I could totally do a better job at designing this" feelings regarding our data stack, and it has immensely fed my ego to have confirmed all of these suspicions myself.
    - This role gets to leverage basically everything I've learned in my career so far.

By far the most important thing here is the sense of independence. At this point it feels like the main person I should be complaining about is myself. (And self-loathing is so much healthier than hating a random product manager.) As long as my company's data scientists are dependent on me to get code into production, I have veto power over a lot of bad code. So if they are putting bad code in production, that ultimately ends up being my fault.

I think my career trajectory made sense– there was no way I was hopping straight into data engineering and doing a good job of it without having done the following first:

- See a few data stacks and form opinions about them as a downstream consumer.
- Get better at coding.
- Pick up on the lingo that data engineers use to describe data (which is

distinct from how social scientists, financial professionals, data scientists, etc. describe data), like “entity,” “normalization,” “slowly-changing dimension type 2,” “CAP theorem,” “upsert,” “association table,” so on and so on.

So, ultimately I have no regrets having done data science, but I am also enjoying the transition to data engineering. I continue to do data science in the sense that these roles are murkily defined (at both my “data scientist” and “data engineer” jobs I spend like 40% of the time writing downstream SQL transformations), I get to critique and provide feedback on data science work, and, hey, I actually *did* deploy some math heavy code recently. Hell, you could argue I’m just a data scientist who manages the data science infra at a company.

Anyway, great transition, would recommend for anyone who is good at coding and currently hates their data science job. My advice is, if you want to get into this from the data science angle, make sure you are actively blurring the lines between data science and data engineering at your current job to prepare for the transition. Contribute code to your company’s workflow management repo; put stuff into production (both live APIs and batch jobs); learn how to do CI/CD, Docker, Terraform; and form opinions about the stuff your upstream engineers are feeding you (what you do and don’t like and why). In fact it is very likely this work is higher value and more fun than tuning hyperparameters anyway, so why not start now?

Sorry, this post has no point, so it’s ending rather anticlimactically.