Succeeding as a data scientist in small companies/startups

Randy Au Follow
Feb 8, 2019 · 9 min read ★

It's nothing like at a big mature company.

This'll probably be an unbounded series of posts that spawned from this question that came across the awesome community that is the data-nerd twitter cluster:

Sean J. Taylor
@seanjtaylor

Has anyone out there read/written something good about being valuable as a data scientist at an early stage company with a very small number of customers / small amount of data? People sometimes ask me about this and I have no experience to apply.

179 5:26 PM - Feb 7, 2019

69 people are talking about this

Some Background

I've spent almost 12 years now at companies sized between 15–150 wearing various hats of data analyst, engineer, and occasionally, scientist. Wandering into mega-corp Google Cloud as a UX researcher is a bit out of character, but with new products constantly being churned out, it feels like a startup in terms of questions and chaos, despite the billions of gross revenue involved.

I've worked in a mix of businesses from interior design (office design), ad-tech, social networks, link shortening, e-commerce, and now enterprise cloud. I come from a social

science background, with a dash of NLP, applied math, and business administration. I mostly live in back end systems, logs, and SQL.

In a word, I'm a generalist, the sort that people seem to recommend for startups, and that's where I've thrived my whole career.

Those are my biases, and the experiences I draw from. If you're in a startup where AI/ML is literally the core foundation of the business, I'm likely irrelevant to your needs. YMMV.

Being the 1st "Data Person"

I'll go ahead and say it, ***small companies don't need a data scientist, but they need a "data person".*** They might call the job "data scientist/engineer/analyst/ninja", whatever.

My experience is that somewhere between 20 and 60 employees, there's enough customers, accumulated data and role specialization that the need to bring in someone who can use data to give useful business insight starts to justify the cost of hiring someone. Until then, they make do with the skills available.

The job title can be almost anything, but the job descriptions tend to be various mixes of:

1. Make sense of the data we have
2. Help build out our data systems
3. Help us be data driven/ run experiments
4. Grow the business
5. Education/Certifications that may or may not be relevant to anything

It's usually quite likely that they don't really have a full understanding of what they need. There's just a generalized sense of "we have data, it seems useful, but we don't have anyone who has the skills to make it useful."

In practical terms, there are 2 big things someone taking this position needs to do in parallel:

- Help the company succeed TODAY
- Set up the company to be data driven TOMORROW

Helping the Company Succeed TODAY

Startups are surrounded by uncertainty. They're not sure who their customers are, production systems can be dodgy, they don't know what their customers are doing with the product, they don't know how to make decisions using data they have, they don't know if the data they have is useful.

Smart answers to the questions lead to smarter decisions and hopefully that mythical hockey-stick growth everyone dreams of. The problem is that most of those issues don't lend themselves to fancy methods. The useful ones are often a century old and/or based on qualitative methods instead of quant.

Most DS methods are most powerful when optimizing an existing process, they'll get you, 5%, 10%, even 25% growth on things like customer acquisition, conversion, retention and spend. A/B testing, recommendation systems, ML classifiers, all of them help to optimize. The gains are real, quantifiable, and can be significant, but early on there's likely bigger fish to fry.

The biggest impacts early on often involve *insights*. Insight changes what the company fundamentally does. They come from very pedestrian things like research into user preferences/behavior that uncovers a new marketing concept for sales folk, or helping product teams realize that the most vocally hated feature on Twitter is actually used by 90% of paying customers and they shouldn't drop it for no reason.

My view of this "help the company now" role is that the data person is a **force multiplier**. People within the business have problems, the job is to help them solve them.

Being the 1st "Data Person" = Being a "Scientist with data"

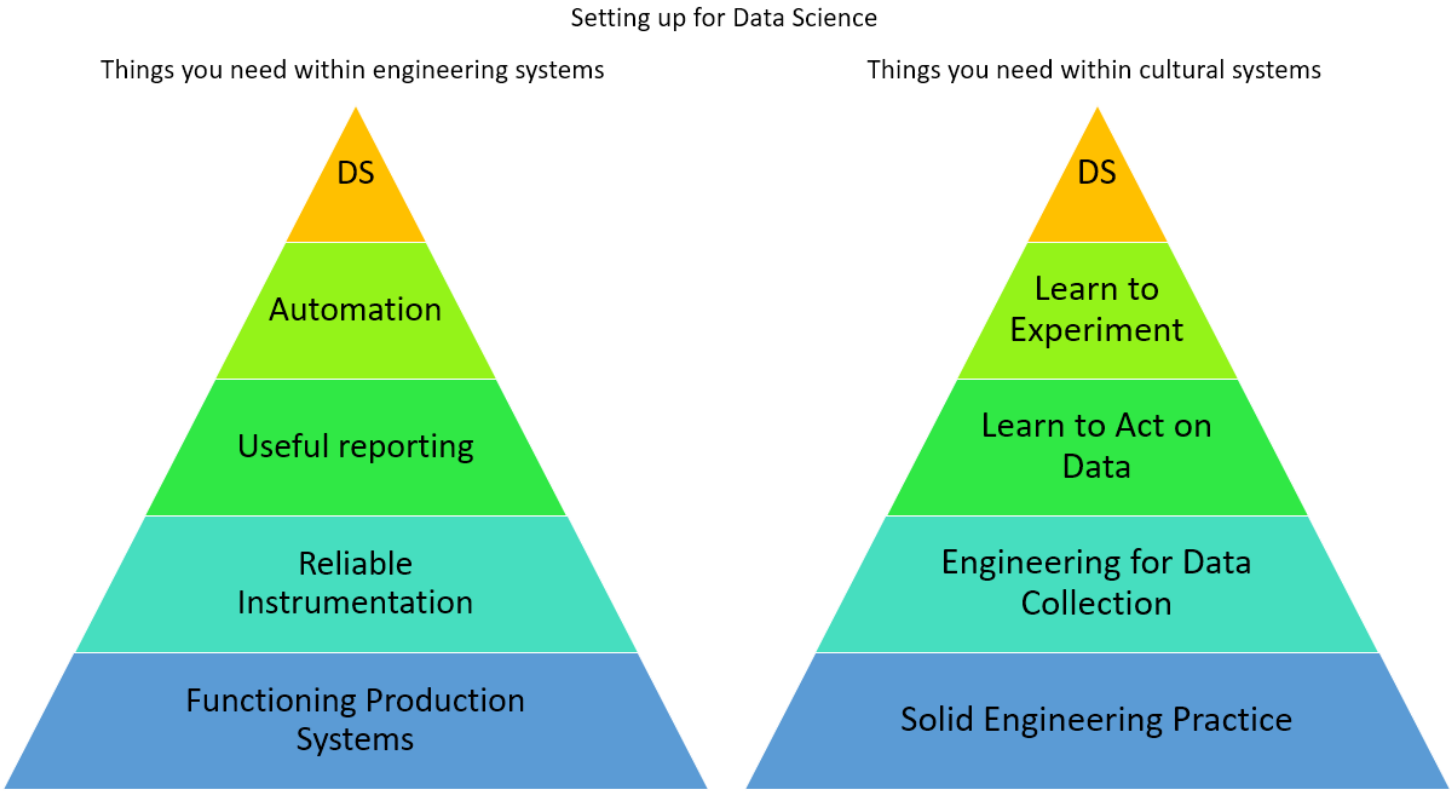
Being a scientist to me means that you have a problem, a research question, and you use whatever methods you can to come to a solid answer to that question.

As data scientists, we tend answer questions using quantitative methods and data collected from systems, but that's not the only path to insight. Sometimes you flat out observe or ask users (qualitative methods), or you go out and collect data (experiments and surveys), or you watch others (competitive analysis). A good scientist doesn't define themselves by their methods, and neither should the 1st data person (or any subsequent data person).

The goal is to answer the pressing business needs: "Why is no one using our product?" "How come our returns are so high?" Should we be running this expensive sale or not?" "What drives customer churn?" "What's a customer's lifetime value, what drives that?"

Becoming Data Driven TOMORROW

A common trap I see are people who come out of Data Science programs joining these positions expecting to be using sexy things like Spark and applying RNNs to their work. But sadly, they want to live on top of a mountain of foundation work that needs to be done first, **both** from an engineering standpoint *and* from cultural standpoint. The mismatch is brutal.



Fancy "Data Science" methods rely on a ton of things, DO NOT expect each layer to be "done" before moving to the next. Think of the colors as "time spent".

Being the first person specifically hired to handle data, it's very unlikely any pieces of the pyramids are sturdy. It's a multi-year, cross-functional, full company effort to get all the pieces in place. Nurturing those **all** those pieces **in parallel** is a big part of the job.

Note, in a typical business, you'll be attempting to do things up and down the pyramid at the same time, regardless of the stability of underlying layers. I've built plenty of dashboards and classifiers against fragile new systems, and you will too.

There's room for tons of posts about aspects of all this stuff, but here's a rough overview of my thoughts.

Solid Production Systems and Engineering Practice

This is a thing that's firmly in engineering's wheelhouse, but you can't really measure the behavior of a system if the system is buggy and broken, so expect to help out here as needed.

The more of a "data engineer" hat you wind up wearing, the bigger a role you can play in helping build solid systems. People will naturally ask your input for questions like PostgresSQL vs MySQL, AWS vs GCP, Spark vs Redshift, etc. Helping with those decisions adds lasting value. Expect to have to set up systems and run them yourself if there's not enough Eng resources.

I also find that pushing engineering to instrument systems for business purposes has a nice side effect of finding interesting bugs. Once, I found duplicated IDs in a critical table that shouldn't have dupes, that set of a bug hunt that fixed some things in a revenue-generating system. Poking at production tables often can find really weird bugs.

Reliable Instrumentation

Garbage in, garbage out.

Getting reliable instrumentation is **THE** most critical thing a 1st data person does. It is an endless journey, from picking frameworks (multiple) to collect system and user data, making sure engineers learn how to implement things without counting errors (which are so easy to make), making sure the databases and logs are doing the Right Thing(tm), and making sure you're counting what you think you're counting.

At the same time, systems will be added specifically to collect and report on data. Those will need to be put together and managed by someone, maybe you.

On the culture front you'll constantly have everyone, from the new guy to the CEO, asking how reliable the data is and what to do with the information. They'll ask for clarification, explanation, deep dives as they themselves learn about the system they've built.

This cultural training alone is a long journey in itself as you make reports and dashboards, and people find inconsistencies vs other systems and get results that don't jive with their view of reality. Sometimes their reality is wrong, oftentimes they're correct. Just having these conversations makes everyone involved smarter.

I honestly don't think this phase is ever "complete", it just gets to a point where you only have to worry about it when a new feature or big change goes out.

Reporting and Acting on Data

Dashboarding and reporting is not a sexy job, but unfortunately, it's often the first/only way for people throughout the company to observe the health of the company on a daily basis so the investment is necessary. The goal is to put (potentially) actionable information in the hands of people who can take initiative to act on it.

In the beginning, most dashboards and reports will be manual. It takes a lot of iteration to hit upon metrics that people care enough to see multiple times. Automation is a nice-to-have, but getting insights out takes priority.

The technical aspect isn't super complex. There are many services and platforms for generating reports and dashboards, you can even just do it with custom code. The trick is to get all the data systems to play together (Ha), have good performance, and minimize the (significant!) overhead in maintaining dashboards and reports as the business grows.

The cultural side is where things are interesting. You're training people to become more data driven here. This also can take years of work and practice.

There is a ton of education involved here. You'll be teaching people how to read the results of an A/B test, what significant difference means, explaining what a

confidence/prediction interval is, explaining why that chart "is only an estimate because we rely on a 3rd party report and they're flaky". You'll field questions about sample size (never enough), and constantly need to teach people good methodologies.

And what should people do if they're concerned about a number on a dashboard? Personally, I tell them to come talk to me. Their concern is potentially a research question (or a bug), and that's literally research GOLD. These people are domain experts in that part of the business, while I'm just a nerd slinging SQL.

Automation and Experimentation

Over time, you'll set up metrics and dashboards for when new features go out. Inevitably people will be disappointed in how a feature is doing and want to know why. If you haven't picked up pre/post analysis methods and memorized all the national holidays of your primary market, now's a good time to do so.

Once people get used to having information and maybe run a few A/B tests to disappointing results, expect to spend a ton of time verifying that the numbers are correct when test data comes back contrary to people's assumptions.

Despite those painful moments, we're now finally doing Real Science(tm). Having a hypothesis and collecting data to test things out.

By now, useful dashboards should be fairly automated and people are used to using data to make decisions. You know you've arrived when a new feature is proposed and "how will we judge the success of this?" comes up without you pushing for it.

One other thing interesting I've noticed is that around this phase, a company can get TOO comfortable running experiments. They learn to design tests that they know they'll succeed at (low risk taking), or they'll "test" a thing that they 100% know they will launch regardless of the results ("we'll optimize it later").

It's now your job as the scientist to call people out on this behavior. They can totally launch regardless of test results, radical changes often test poorly, but that should be explicitly state that intention.

Finally, Data Science

After that the long, long, long journey above and with many detours, the company itself has transitioned to being data driven. They have hypotheses, can reliably collect data, and make deliberate decisions based on the outcomes. They're also more self sufficient, can read (and maybe create) dashboards with some guidance, and have learned when to worry and how to raise issues.

There'll always be things to do further down the pyramid, but at least now things aren't on fire all the time. The business has hopefully become better at understanding its place in the market and the world, and interest shifts to optimizing exiting processes for incremental (but significant) gains.

Now you can think about breaking out the fancy algorithms…

Or maybe there's a data warehouse that needs building because you've got too many systems now and analytic queries can't be sped up any further, oops.

Comments/Feedback Welcome

I wrote this whole piece in one extended sitting because the content just needed an outlet. I've glossed over a ton of things for length reasons. So any feedback and topic requests for future posts would be most welcome. Hit me up on Twitter.