# Poker Player to Data Scientist

I played poker for a living for seven years. About three years ago, I decided to quit and pursue a career in machine learning and data science. I’m now working full-time as a data scientist at [reserved.ai](http://reserved.ai).

This post is about the most important lessons from that journey. Since there’s already a lot of great resources out there about coding and math, I focused more on how to get a job without a standard technical background, and specific lessons from poker that carry over into data science.

## People skills

In the course of playing poker a living for seven years I had to learn about various fields related to ML - statistics, probability, game theory - but I think the most relevant thing for getting a job in data science was actually just spending a lot of time sitting around a table with a diverse group of people and learning to get along with them. In poker, getting invited to the biggest games is often more important than being the best player.

I’m really grossed out by the term “networking” - it feels like reducing social interaction to something inauthentic and manipulative. Nonetheless, if you’re looking for a job, it pays to be a little strategic about forming social connections.

If I could pick just one thing that I think really helped me find a job, it would be doing informational interviews. I learned about this by reading [Designing Your Life](https://designingyour.life/the-book/) by Bill Burnett and Dave Evans.

I started reaching out to people I knew were doing interesting things in data science or machine learning and asking if they’d meet for coffee or lunch. I was kind of shocked how easy it was. Almost everyone I reached out to said yes, and often introduced me to new connections.

For example, I met a program manager at Sonos at a one-day data visualization class. From him I learned a lot about the day-to-day reality of working on data in a business setting. He introduced me to an engineer at Amazon who helped me prepare for coding interviews.

Some tips:

* If you ask someone for help, remember that “No” is a valid answer. Be considerate and polite - apart from just being a good way to carry yourself in general, you may run into them again later.
* If someone doesn’t respond the first time, wait a week or so and try again. Many times people just don’t see an email or forget to respond. If they don’t respond a second time, it’s probably time to move on.
* Be respectful of other people’s time. I think 30 minutes is about what’s expected for a meeting like this.

## Books aren’t where it’s at

A few weeks ago I was thinking about which resources I wished I knew about when I started learning about ML and I realized all my top choices are presented as videos: [fast.ai](https://course.fast.ai/), [Weights & Biases](https://www.wandb.com/tutorials), and [3Blue1Brown](https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi).

Part of this may be my personal learning style - I often learn better by watching than by reading. But even so, I was surprised I couldn’t find a book I was really happy with. I’ve read, or at least looked through, a lot of the most famous introductory ML books, but I found they were outdated, or poorly organized, or the explanations didn’t make sense to me, or they assumed prior knowledge I thought many readers wouldn’t have. One way or another, I couldn’t find a book I was really happy with.

ML and poker are really similar in this regard. Cutting-edge information is passed through videos, forums, or groups of people working together. I think it’s because both fields are moving so fast: the theory completely changes every year or two. By the time you finish writing a book, it’s already out of date.

There are exceptions, of course. [Play Optimal Poker](https://www.amazon.com/Play-Optimal-Poker-Practical-Theory-ebook/dp/B07SGGC53Q) is a new poker book that’s really well done. And if you’re reading this you probably know there’s a [fast.ai book](https://github.com/fastai/fastbook) now. I haven’t had a chance to read it yet, but I’m very excited about it.

## Keep it simple

We have a term in poker called FPS (“Fancy Play Syndrome”). This is when you reject the obvious play in favor of something fancy and complicated. It rarely works out well.

One amazing thing about poker right now is you can watch some of the best players in the world play and explain what they’re doing in real time. When I watch players like Ben Sulsky, Doug Polk, or Phil Galfond I’m often struck by how simple their thought process seems. Once you hear it, it seems obvious. What sets them apart is an ability to focus with laser-like precision on the most important factors for the current decision.

The ML version of FPS would be skipping linear regression and going straight to an RNN. The academic literature, in particular, is full of insanely complicated and hard-to-understand methodologies. In my experience so far, this is very far removed from working with data in a business context. For the most part you figure out a simple solution that works pretty well, then move on to the next urgent problem.

## Dealing with failure

Maybe the biggest thing poker and ML have in common is this: you can make good choices, yet fail, over and over and over and over again. How you deal with this situation will have more to do with whether you ultimately succeed than anything else.

In poker someone can make a bad play and still beat you by hitting a lucky card. If this happens once, it’s easy enough to shrug off. But when you “run bad” for a long time - when the losing sessions extend into losing weeks or months - it’s hard to deal with, no matter how well you understand variance on a conceptual level.

Likewise, dealing with failure was what I found most challenging when I started learning about ML. Even simple things like setting up my environment and getting all the appropriate packages installed could take hours or days, where nothing I did seemed to work. I felt like an idiot and had no idea when it was going to end.

The obvious response is to persevere and tough it out. I think this is good advice, up to a point, but it needs to be balanced with a certain amount of levity. One of my poker coaches once gave me very good advice: “When you lose, the first thing you need to get back isn’t the money you lost, it’s your love of the game.”

While ML isn’t a game per se, it’s still helpful to have a playful mindset. This encourages creativity, which is important not just because it’s more fun, but because it will allow you to find better solutions more quickly. Try to see everything you do as an “experiment” where failure is normal, healthy, and productive.

I learned a lot by watching [Rachel Tatman’s live coding streams](https://www.twitch.tv/rctatman). She makes a lot of mistakes, but iterates on them quickly and cheerfully. As they say in Designing Your Life, “Fail fast and fail forward.”

## Conclusion

When I was doing informational interviews, one of the most common questions I asked was, “How big of a deal is it that I don’t have an academic background in this field?” What I heard back was pretty consistent: It might make it hard for me to get hired at some companies, but it wouldn’t prevent me from doing the work effectively.

For those trying to get a job in machine learning or data science, especially if you’re coming from a non-traditional background, I hope you’ve found something helpful here.