**A Day in the Life of a Data Scientist**

**Spoiler alert — I don’t create any fancy machine learning models**



Photo by [Annie Spratt](https://unsplash.com/@anniespratt?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Lately, I’ve been meeting a lot of people who are interested in making a career shift into data science. One of the first things they always ask me is, “what does a typical day look like?”. I’ve seen a lot of articles that give an overview of the skills and tools Data Scientists use, but I don’t see very many that provide real examples of daily tasks.

While every day is different, these tasks represent a typical day for me as a Senior Data Scientist at a large financial institution.

**Day at a Glance**

* 8:30–9:00 — Starting My Day
* 9:00–10:00 — Pair Programming
* 10:00–10:30 — Scrum
* 10:30–11:00 — Prep for a Presentation
* 11:30–12:00 — 1–1 with Manager
* 12:00–1:00 — Get Feedback from Lead Data Scientist
* 1:00–4:30 — Code!

**Starting My Day**

I typically start my work day around 8:30 am after I roll out of bed at 8:20. I’ve worked remotely since March 2020, and it has been a game-changer for me. I love the peace and quiet of my home office and being able to work in my pajamas and get a few loads of laundry done while I wait for some code to run.

First on my to-do list is reviewing emails and Teams chats that I may have missed from the day before. One email I get every day includes the status of one of my team’s production machine learning models. I check to make sure there are no errors in the model itself or the related processes that extract and load data into our data warehouse.

If there are no errors, I finish checking and replying to various messages/requests and then open up a project/task tracking tool called Jira to update the status of the tickets I’m working on for the next three weeks — in the agile software development world, this is called a sprint. From here, I prioritize my tasks for the day.

**Pair Programming**

Throughout my time as a Data Scientist over the past five years, I’ve noticed a shift in the kind of work I do. While my first few years focused on doing the work myself, I now spend a large share of my time helping and educating less experienced team members.

Once a week, I meet with a Junior Data Scientist on my team for a pair programming session. In agile, pair programming consists of two developers working together on the same task by either sitting at the same computer or sharing their screen and programming together on the fly.



Photo by [KOBU Agency](https://unsplash.com/es/@kobuagency?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

I’ll admit, a few years ago when a Senior Data Scientist suggested the idea of pair programming to me, I was skeptical. I thought it would be a waste of time having two people work on the exact same task, and I was terrified I’d look dumb when I didn’t know how to code something. But what I found is that pair programming is a great way to learn from other developers. In fact, I almost always learn something new from junior developers when I work with them.

In this pair programming session from 9:00–10:00 am, we started with an issue I was having. Both of us were learning a new-to-us graph database tool, and the Junior Data Scientist mentioned a plugin he installed that helped him load data more easily. Unfortunately, I wasn’t able to install it.

Through our pair programming session, we found that our IT department installed my version of the tool without access to the internet. After the Junior Data Scientist showed me his settings, I was able to install the plugin!

Next, we reviewed a script I wrote to create the nodes and relationships in my graph. The Junior Data Scientist gave me some suggestions to follow best practices such as using all caps to identify relationships. I was so focused on getting my script to work that I lost sight of making it more readable for others.

After helping me out, we then switched over to a challenge he was facing. He needed to write a SQL query that would calculate current and previous metrics for several business lines.

He showed me a mockup of the final output in Excel to help me better understand the problem. From there, we moved over to SQL Server where I suggested we break down the problem by writing a simple query to return the current result for a single metric and single business line. After successfully retrieving that piece, we discussed how he could write multiple small queries and union the results together to get closer to the final output.

**Scrum**

When 10:00 am rolls around, it’s time for my team’s daily scrum.



Photo by [Jason Goodman](https://unsplash.com/@jasongoodman_youxventures?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Traditionally, the person leading the meeting, the scrum master, asks everyone three questions.

1. What did you do yesterday?
2. What will you do today?
3. Do you have any blockers?

I don’t find much value in status updates, so I’ve pushed my team to take a different approach and emphasize learning. Rather than *saying* what we did yesterday, we *show* it.

Let’s say yesterday I was working on some Python code to determine if the day before a date in my data set was a holiday. Instead of verbally giving that update, I’ll literally share my screen and walk through the code.

We’ve found several benefits to doing this. Many times when I share, someone else on the team will have an idea for a better, faster, or simpler way to solve the problem. While things like this could be caught in a code review, it’s usually much easier to catch them when you’ve written five lines of code instead of 100. Other times, I find out that someone on my team is doing something very similar and can save time by reusing my code and not reinventing the wheel. Lastly, it’s honestly just really cool to see incremental progress every day instead of just the final product.

**Prep for a Presentation**

After scrum, I’ve got about 30 minutes until my next meeting. I’ve found that’s usually not enough time to dive into any true “data science” tasks like cleaning data or modeling, so I’ll use small amounts of time like this to answer more emails that have come in throughout the day or prepare for any upcoming presentations.



Photo by [Nghia Nguyen](https://unsplash.com/@nghiaagent?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

A good chunk of my role as a Data Scientist includes creating presentations to educate others about what data science is and isn’t. A lot of executives hear the buzz words artificial intelligence and machine learning and then say, “we should be doing that!” But the truth is, machine learning isn’t always the answer. Oftentimes, basic reporting or simple automation will solve the majority of teams’ problems, and we shouldn’t overcomplicate it with machine learning just to say we’re doing it.

In addition to educational presentations, I also may present the results of an initial model to my business stakeholders. In my first year as a Data Scientist, I was obsessed with including as much jargon as possible in these presentations because I wanted to sound smart. This was a huge mistake!

None of my stakeholders had a math degree and didn’t understand or care what my model’s F1 score was. Over time, I learned the importance of knowing your audience and speaking in layman’s terms. Now when I discuss model performance, even if I’m talking about an F1 score, I’ll call it accuracy. Is that technically precise, no? But the business stakeholders understand accuracy better than the harmonic mean between precision and recall.

If you can’t explain it simply, you don’t understand it well enough — Albert Einstein

**1–1 with Manager**

After I wrap up my presentation edits, I’m off to a one-on-one (1–1) with my manager. If you’re new to the business world, these meetings are an opportunity for you to meet with your manager and discuss your career goals, recent successes, and/or any challenges you may be facing.



Photo by [LinkedIn Sales Solutions](https://unsplash.com/@linkedinsalesnavigator?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

On the agenda for today’s meeting is a challenge I’ve had setting up a tool in my team’s development environment. After spending a few weeks navigating permissions with another team, we reached a point where none of us knew how to configure security settings for the tool properly.

Because my manager has a wider network across my organization than I do, he was able to give a few suggestions of people who might be able to help when I thought I was out of options.

**Get Feedback From a Lead Data Scientist**

Next up, I get to meet with the Lead Data Scientist on my team to get some feedback on a proof of concept I’ve been working on. For the past month, I’ve been exploring a data set for a business area using the graph database tool I mentioned in the pair programming section.

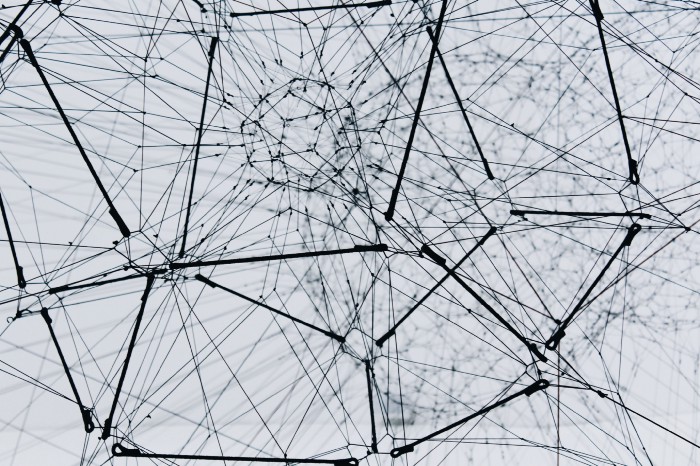


Photo by [Alina Grubnyak](https://unsplash.com/@alinnnaaaa?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Up until this project, most of my experience as a Data Scientist had been in natural language processing, so I had to do a lot of research on graph databases in general and learn a new tool. Learning new tools and techniques is one of my favorite things about being a Data Scientist, so this has been a fun project.

Since I’ve been working on this by myself for about a month, I needed to start from the beginning by showing the Lead Data Scientist a sample of the data and what it looks like in a network graph. Next, I covered some basic queries that I wrote, like determining which node has the most relationships. After that, I discussed some of the built-in algorithms that I tried like determining the similarity between two nodes in the network.

While discussing some of the built-in algorithms, I noted that I wasn’t able to try a lot of them because my graph had relationships between two types of nodes (a bipartite graph), and the algorithms only worked with graphs with one node type. He suggested that I restructure my graph so that I could test out some of the other algorithms, and we walked through what the new data model might look like.

While I was a bit embarrassed that I hadn’t thought of that myself, that’s one of the great things about getting feedback from others! Sometimes you get so stuck on solving one problem, and it’s incredibly helpful to get a fresh perspective. We ended the meeting by brainstorming ideas for my upcoming demo to the business area.

**Code!**

Finally, to close out my day, I actually get to code — yay! I ended up switching gears from network graphs to a document classification project that I’m working on for another business area. My team acts kind of like contractors and works on a variety of projects across multiple business areas. One thing I love about this is the variety of work. If I get tired or frustrated with one project, I can switch to another one and let my brain relax. However, one of the biggest challenges is that I constantly need to learn new business areas and processes.

For this project, we don’t have labeled data and are trying a new-to-us technique called active learning which essentially allows you to create better machine learning models with fewer labeled data points. We selected a sample of documents for five people internal to our company to label, and my task for today is to review all of our labelers’ annotations and determine agreement.

Before reviewing annotations, I needed to load the data from five separate excel files into a single data frame and transform it so that I had a column for each labeler’s result.

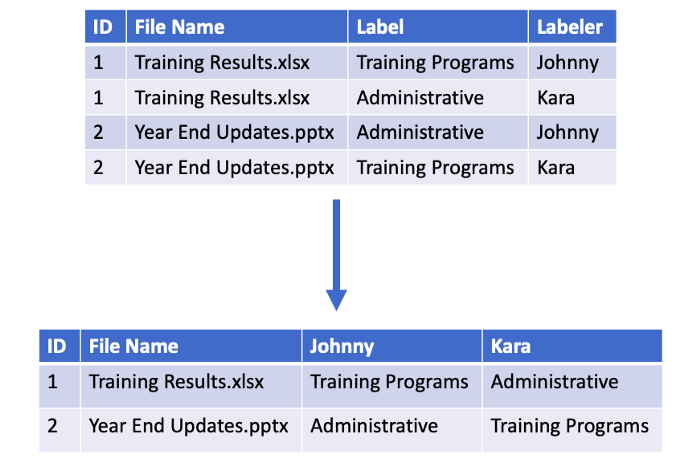


Image by author

After getting the data in the right format (which always takes longer than I expect), I then pondered how to determine agreement amongst labelers. For some documents, all five annotators agreed on the label which made my decision easy, but for others, three or fewer out of five agreed. There were even a few where all five labelers chose a different answer!

I decided to start simple and take a majority-rule approach. If three or more labelers selected the same answer, that was used as the final answer and would ultimately be used to train our model. This ended up taking me a while to code up because I was also interested in seeing if there were any scenarios where three people agreed with each other but the other two both agreed on a different answer. I ended my day by making some notes of my decisions and jotting down some tasks for tomorrow.

**Conclusion**

You might be surprised that a large portion of my day as a Data Scientist isn’t spent coding. And even when I was coding, I wasn’t making machine learning models — I was cleaning and analyzing data. When I got my first job as a Data Scientist, I thought I’d spend all day writing algorithms and creating complex machine learning models. In practice, I’ve found that most of my time is actually spent preparing/cleaning data for modeling and understanding the people/processes generating that data. Data Scientists are often viewed as having a super cool and exciting job, but the reality is that it’s not as glamorous as it may seem. This isn’t necessarily a bad thing, I just find that new Data Scientists don’t realize what they’re getting themselves into.

If you enjoyed this article and are a new Data Scientist who wants to learn what it’s like to do data science outside of an academic setting, check out my [workshop](https://www.datasciencerebalanced.com/the-data-science-workshop) where I teach you the skills you don’t learn in school.