



## Questions to Ask the Interviewer

- What does success look like for this position? How will I know if I am accomplishing what is expected of me?
- What is the last project you shipped? What was the goal, how long did it take, what were the stumbling blocks, what tools did you use, etc.
- What will my first 90 days in this role look like? First 6 months?
- Who will I report to and how many people report to that person? Do they have regular 1:1 meetings with their team members?
- Why did the last person who quit this team leave? The company?
- If a startup, how long is your runway? How are financial decisions made?
- What would be my first project here? Has someone already been working on this or is this in the aspirational stage?
- What is the current state of the data infrastructure? How much work needs to be done on getting the infrastructure and pipeline into shape before we start analyzing that data?
- Which projects are you proud of? It'll set the bar for your work - even if the bar is way too high for you now. It'll give you an idea of what's expected of you eventually.
- What reading do you do on the side to improve your work? (This gives you an idea of which writers they look up to. Look them up between interviews and compare their preferences to yours.)
- Which projects do you have in mind for this role?
- Describe a typical day, an atypical day.
- How often do they have crunch time scenarios?
- How often do they have to dig into research articles?
- What kind of project would you be working on, starting on your first day?
- What is the relationship between the [team you're interviewing for] team and the rest of the company?
- What are the most used programming languages and corresponding modules?
- Is the data collected in-house or externally?
- Does the company use open-source data or technology? If so, how does the company give back to the open-source community?
- How do you think about ethical/legal issues with the product your team is working on, for example, XYZ?
- How do the other team members keep up to date with the field? How does your company support ongoing education?
- How will your first weeks be and will there be team members to support during your onboarding?
- Ask what tools they use. Specifically for build management, version control, documentation, and deployment. If they have poor documentation and version control you might consider that to be a red flag.

- Ask the same questions they ask you. What types of models do they prefer, what are their big successes/selling points; who are major clients; how many projects do they typically do in a given month/quarter/year?
  - What's the riskiest/most audacious project the team has taken on in the past year, and what was the result?
  - How many projects have you taken on that you know are going to be successful, and how many where success is uncertain?
  - Of the uncertain ones, why were you uncertain they'd be successful?
  - In your production system, what are you optimizing for? i.e. what are the various engineering tradeoffs (performance vs. scalability; latency vs. throughput etc) that you're making and how did you arrive at that decision?
  - What's one thing you've implemented from research advances that are not available as an open-source implementation yet?
  - What measurable business impact has your team had and how do you track that?
  - What metrics have changed because of your team's projects?
  - How much ROI has each of your teammates contributed in terms of measured business impact/salary?
  - What's the most exciting part of your job?
    - How much of your time do you get to spend on that?
  - If there was something you'd change about the company or your team, what would it be?
  - What percent of your time do you spend using different tools?
  - How much freedom/flexibility do you have to determine your methods?
  - What are some languages, frameworks or technologies you're excited about, but aren't using yet and why?
  - What has your team contributed back to the software engineering community?
  - Besides the software engineers, who are the most important stakeholders in a typical team project? How do they work with each other?
  - What does it take to get promoted?
  - How do other people in the company view the team?
  - What requests have you gotten from other groups within the company?
  - If someone on your team left today, how long would their impact last afterward?
  - To the team leader: What's your vision for this team?
  - If you had an unlimited headcount, how many people would you hire, and what roles would you hire them for?
  - How are responsibilities split between them?
  - What share of your time is spent on engineering/operations/ meetings?
  - When I come to work, how do I know what to work on?
  - How are tasks prioritized? When was the last time you had to circumvent that process?
- 
- How is code reviewed at the company? Are there formal code reviews?
  - How do you help less experienced engineers grow technically at the company?
  - What are some successful traits of engineers at the company?