00:00

Alexey

We had an amazing webinar about getting a data engineering job. It was a couple of weeks ago and it attracted a lot of attention. We had many questions and we didn't answer 12 of them. I thought it would be a pity just to throw away these questions. So I thought we should do a follow up and record the answers. I also want to release this as a podcast episode without video, just audio only.

## Summary of “Getting a Data Engineering Job” webinar

00:36

Alexey

For those who didn't listen to your webinar about getting a data engineering job, can you give a short summary of what you talked about there?

00:48

Jeff

The main point is that people really want to hire data engineers. If you look at the statistics, or even Slack channels or LinkedIn, you'll see lots of opportunities for data engineering jobs. But it's still challenging to get a data engineering job. The reason is you need to convince employers that you have the skills to start contributing.

01:20

Jeff

These skills are: backend engineering, cloud computing, and [building] data pipelines.

01:27

Jeff

With back end engineering, it means Python and SQL. With cloud computing, that's Docker and a cloud service, like AWS. Then data pipelines are like Airflow. Snowflake, or Redshift - for the data warehouse and DBT.

## Python and engineering skills

01:49

Jeff

You'll see that lots of people build projects that check these things off and post them on Reddit. But when I look at these projects, I see that there's not a lot of Python and SQL. And those are really the two main things you should be focusing on. This is what your project should display. I won't see 50 lines of Python and SQL. There really should be over a few hundred of Python and a few hundred of SQL.

02:22

Jeff

You also want to write cleaner code: small functions, like five fewer or fewer lines of code; use object oriented programming with 100 or fewer lines of code for each class; have descriptive naming; write tests. Show that this is a professional level project.

02:46

Jeff

You can showcase these skills in your own projects. But in addition to that, work on open source projects. This really enforces the quality of code. Your code needs to be reliable, to be tested - it should be close to professional level projects. Otherwise, your pull requests will not be accepted. Some of open-source projects are really on a professional level - they're built by and maintained by professional teams. You'll be forced to use CI/CD, Docker, Python and SQL. Also, a good resource for finding these is Code for America. It has a number of different chapters. They have remote meetups that you can join and make contributions.

03:38

Jeff

That was the first half of the talk - discussing the skills and ways to get them.

## Interview process

03:45

Jeff

Then the second half of the talk is the interview process. You can think of it as a funnel. The bottom of the funnel is sending in the application or getting views on your LinkedIn. The middle of the funnel is the behavioural interview. Then you have the technical interview, and the final round interviews.

04:12

Jeff

At the bottom of the funnel, improve your LinkedIn profile, have as many relevant skills listed as possible. Definitely have a picture. Have the about section relevant to data engineering. When you apply, they'll look at your LinkedIn. You'll get a lot more interviews that way.

04:47

Jeff

Then, a resume should be recrafted for a data engineering job. Make sure there are relevant skills. Even if you didn't have previous engineering experiences, highlight the skills that translate into tech. You can show this to somebody that works in a technical role, like an engineering role. Talk to them about your past experiences, that way you're able to highlight them. That's the bottom of the funnel.

## Behavioral interviews

05:15

Jeff

Then the next step is the behavioural interview. For this too, I recommend speaking with a peer - someone that you trust, someone that you're close to. You'll take real feedback from them. Give them this criteria. You're a positive professional. That means that you speak positively about all past experiences. This is something that people fail on. Even after you tell them, they still fail on this. In many jobs it can be an instant disqualifier. People want to make sure that you're easy to work with. This is something you have to do.

05:57

Jeff

The second thing is speaking clearly. You should talk in a way that's there's an outline when you're explaining things. You can think of a heading and then bullet points - so it looks like a pyramid. Another way to think about this is problem/solution. "In my last job, we had this issue. We had a great product, but it wasn't marketed properly. We marketed well in one city, but we needed to expand. So I needed to figure out what we were doing here and then replicate it across 30 different cities." I mentioned the problem, then went to the solution. And finally, answers first. If someone has a question in mind, or they explicitly ask a question, give them the answer. That should be your top line reply. You can always go into more details about it. This ensures that you're an effective communicator, which is really essential for an engineer.

07:04

Jeff

The last thing is to show that you're interested in the position. You're really passionate and motivated to work in that role. The key story that you should have walking into the interview is this: given my past experiences, and what I want to do in the future, this role is like the perfect next step for me.

07:29

Jeff

It also benefits to do a mock interview with somebody that you're close with. It could be anyone.

## Technical interviews

07:46

Jeff

After you get through the behavioural interview, then you have the technical interview. The easiest way to sum this up is generally it's leetcode SQL problems from medium to hard; and Python leetcode problems easy level.

08:05

Jeff

Sometimes you can also see a take-home project. Let's say in about 1/3 interviews. They'll give you some raw data, like a CSV file. You need to load it into a database, query it and show some findings from the data. You'll need to present the findings with either matplotlib or just simply PowerPoint.

08:33

Jeff

From there, the last step is the final round interview. Again, speak clearly, show that you're motivated. It's primarily behavioural. They just want to see an understanding of the tech stack.

08:49

Alexey

That's a good summary. Of course, we missed a lot of details. I will include the link to the webinar, check it out. It's quite long, but it's packed with useful information.

09:14

Alexey

Maybe I'll just do a quick summary of your summary. You said: focus on four areas. It's Python and SQL; Cloud, Docker; Airflow; Data warehouses. Build a project that checks all these things off. But don't just do that. Focus on Python and SQL - your project should have a lot of Python and SQL. You want to show people that you really know how to use the things. You also need to write clean code, follow good object-oriented design and so on. This is the "how you build a good portfolio" part. You can also take part in open source projects to build this portfolio.

09:41

Alexey

Then you need to start applying. First, you send out applications. Then there's the behavioural interview. You need to show that you're a positive professional - you speak positively about things; you speak clearly; show that you're interested in this position. For technical interviews, you get leetcode SQL and Python problems. Then you get a take home assignment, which could be like building a small project.

10:05

Alexey

Sometimes you'll get database related questions. What's a view? What's a materialised view? Or what's the difference between an OLTP versus OLAP? Things like that.

10:31

Alexey

Now for those who didn't watch the webinar, I think we provided enough context. Please watch it if you haven't. Now I wanted to continue with the questions - where we stopped last time. We still have quite a lot of questions. I wanted to start with the first one.

## Learning Python and SQL from scratch

11:24

Alexey

If you could go back in time, how would you learn about Python and SQL from scratch?

11:32

Jeff

The hardest thing is getting started with a first programming language. I started with Ruby. It was very challenging to really understand those first few steps. I wrote an intro to Python. It's on jigsawlabs.io. I battled tested this - I mentioned it in our earlier podcasts - by teaching my mom and giving this material to my mom. She gave me lots of "Just get to the point" again and again. So it's really beginner friendly. I recommend it. I know, I'm biased, obviously. From there, I recommend two books. The first one is "Think Python", by Alan Downey. It's a short book, you can read it on the train. Then another one is "Automate the boring stuff with Python".

12:55

Jeff

Is it a book or a course?

13:03

Jeff

Book. But there's a Udemy course as well on the boring stuff. But yeah, both of them are good. Another resource I should mention for back in engineering is "The Flask Mega-Tutorial" by Miguel Grinberg. That's a nice project, if you want to get an overview of back-end engineering.

13:30

Jeff

For SQL, Khan Academy is excellent. Their interface is really good, you can just start typing, and it tells you when you're making a mistake, and it has an interactive console. It goes into pretty strong SQL concepts. I would start there. After that, Mode - the dashboarding company - has some nice lessons and tutorials on SQL as well.

14:05

Alexey

We will need all this links from you, I will put the link in the description.

14:11

Alexey

A question from Ilia. If I already work in BI and mostly use SQL and no-code tools, should I get the most from my current role to have a higher start in data engineering, or change without waiting?

14:33

Jeff

It's going to take time. Even if you land the first role, it's still going to take 4-6 weeks to actually move through the interview process and switch the roles. I would say start trying to make your job more technical. Take on those projects. It's the easiest thing you can do. Then you'll still need to supplement with Python. If you want a data engineering role, but don't have Python experience, look for analytics engineering roles. They generally don't require Python - at least too much Python.

15:15

Alexey

But it can be both, right? You can try to look for a job and try to make your work more technical. It can be a bit tricky to find a job and you learn new things at work at the same time. But that should be manageable.

15:38

Jeff

They'll figure out what you know and what you don't know. They are probably comfortable ramping you up.

15:53

Alexey

It's always worth keeping in mind that the job description describes the ideal candidate. This candidate usually does not exist. If you do not tick off the boxes, it doesn't mean that you shouldn't apply. If it's 60%, or 50%, you can still apply. You shouldn't reject yourself before they reject you. Let them do this. You just apply, and then let them worry about you being a good fit or not.

16:23

Jeff

Just to reinforce that. When I think back to hiring candidates. If everyone who applied to the job actually had all the skills we listed, we would have filled the job in an hour. But it takes weeks and weeks. And when we ultimately make a hire, generally they rarely have all the skills. And we're totally comfortable with that.

## Is having non-coding experience a disadvantage?

16:48

Alexey

Another question. I have almost 10 years of working experience, but non-coding related. Does that negatively impact the career path?

17:07

Jeff

I think you would just start as a junior engineer. Otherwise, it doesn't negatively impact the career path as long as you adapt to the required skills. We see that in people who come from a corporate background. They're a little bit concerned about asking questions. They need to adapt to the startup engineering culture. But as long as you do that, that's totally fine, it has no negative impact. Many of my students have had 10-20 years experience in something else.

17:49

Alexey

And, on the contrary, it can have a positive impact, right? Let's say, you're a lawyer or a bookkeeper. If you apply to a company where your domain expertise is valuable, then all of a sudden, you're ahead of other candidates - just because of this domain expertise. So, maybe, for your first jump, if possible, try to find areas where your domain expertise and your previous experience will put you ahead of other people.

18:30

Jeff

Right. One thing that people don't expect, but it's true - musicians make very good engineers. They're very detail oriented. They get into the work, they focus on little steps to make it better. It's a real craft. So a lot of my students who were musicians became exceptional engineers. Aslo a lot of hiring managers have had that experience as well.

19:04

Alexey

You said something like "if you're good at one thing, it's likely that you're good at other things".

19:16

Jeff

Exactly. This was something the CEO of my old company told me. One of our students was an NCAA basketball player. He was on the championship team. He was getting tons of interviews. The CEO said "I'm not surprised. Winners are winners. We always want to hire winners." So if you're really strong in something, it shows dedication, passion and being detail-oriented, self-learning. All those skills translate.

## Analyst or engineer?

19:57

Alexey

How to decide between data analysts or engineers?

20:03

Jeff

There's a couple of ways to think through it. Data analyst, depending on the role, can be a little bit more entry level. It doesn't always require technical skills. It's also more of actually doing analytics - meaning using data to make informed decisions. While data engineering is more on the engineering skill set, meaning SQL and Python.

20:37

Jeff

The reason why I hesitated is because data analyst is a pretty fluid definition. You'll see descriptions that require zero coding, while some descriptions look like a data engineering role and some descriptions look like a data science role. So it depends on what that term means. But in general, data analytics is more about extracting insights from data. And the engineering is Python, SQL, cloud computing - to collect and organise that data.

21:15

Alexey

And perhaps in data analytics your previous 10 years of experience can be a bigger advantage. In data engineering, engineers don't tend to work closely with domain experts, but data analysts are a lot closer to the product, to the end users. So having the domain expertise and all this experience is more important.

## Do you need certificates?

21:56

Alexey

Another question. I am just in my first year of an IT job in data integration. I monitor pipelines and solve data quality issues. I have a Python certificate from IBM and a data engineering certificate. Am I eligible to switch to a data engineering role? Or do I opt for post-grad in big data data engineering? So I guess what the question asks is "am I ready to work or I need to keep learning?"

22:36

Jeff

Do you have the skills mentioned? Like, do you have Python and SQL skills? That's the main thing. Can you contribute to an open-source project? Or an ETL Project? If you look at a code base, can you make contributions to this? Do you know Git and GitHub? So really, it's all about the skill sets. We want data engineers, and here are the skills that we need. So if we hired you, could you start contributing, and helping us organise and clean our data?

## Do I need a master’s degree?

23:13

Alexey

And also it doesn't mean that if you go for a postgraduate degree in big data, you will automatically learn the skills. You'll spend two years learning interesting stuff. But after these two years, with a master's degree, you could be back to square one. You still might not have the experience you actually need for the job. So you will need to learn how to do this anyways.

23:46

Jeff

When evaluating these programmes, check what percentage of the students graduate from the program? And of those graduates, what percentage get jobs? When you multiply those two numbers, you see your probability of getting a job after the program. If 60% graduate, and 60% of graduates get jobs, that's really a low percentage. It's around 36% of students that enrol in the course ultimately get hired. You want to see what those jobs are and how long it takes to get those jobs. That will show you the success rate of enrolling in the program.

24:36

Alexey

It's true but maybe it shouldn't completely discourage you. This is statistics, but if you're motivated, you'll do it. But do you really want to spend two years getting a master's degree? Because, in my experience, it focuses more on research rather than on applied stuff. If you want to learn the applied stuff, I'm not sure a master's degree is what you need.

25:09

Jeff

One thing I see is the courses aren't really integrated. Sometimes they'll teach the same topic again and again. If you do data science, you'll learn regression so many times through the course. But instead they could ensure that students learn it and then move on to other topics. And some courses may teach Python and some might go into Weka or R, which doesn't really reinforce the first skill. That's what I see sometimes with master's courses - they have individual teachers developing each course. And they don't really mesh too well.

26:03

Alexey

It sounds like we're trying to convince you not to do masters. While it's partially true - I think I was trying to do it to some extent. But, on the other hand, it also depends on what you want. Because for example, in my master's, I studied with the people behind Apache Flink. They actually learned all these fundamentals - a lot of hardcore stuff - to be able to work on Apache Flink. This is not something you can easily learn at work. If you want to come up with a new way of doing stream processing, you probably need to have some research foundation. But if all you want is to work as a data engineer, at an internet company, maybe you can take a shortcut.

27:01

Jeff

I saw that some students will first land the job as an engineer, and then, after a year or two, take a master's degree course. Because they think it can be really useful to their skills. If they feel like "I want to go deeper into algorithms" or maybe it's just something I feel like I need on my resume in order to progress. I think it could be useful, after you're more informed, and you're around people that are really informed to then get a master's degree.

## Fully remote data engineering jobs

27:46

Alexey

Okay, moving on to the next one. What do you think about fully remote - i.e. work from anywhere - data engineering jobs? I found many remote jobs, but all of them are based in the same country.

27:59

Alexey

I also saw them - it's "Remote" and then - in parentheses - US only, or EU only. So, what do you think about this?

28:11

Jeff

I tend to see the same thing - US, Canada, or willing to work in this time zone. I guess the question is, how do you get jobs from a different country? I think that's challenging. It feels the bar would be even higher. You'd need to be a standout candidate.

28:42

Jeff

The theme of what I keep saying is "to get the job, just do the job", "do the job before you get the job". And show that. That's why I keep saying "open-source", but it could be any kind of real project that shows, "I'm already doing this, I'll be able to contribute. There's really no risk with me". If you can eliminate the company's risk, they'd love to hire you.

29:16

Alexey

I did come across a few companies who hire fully remotely. There are fewer of them compared to those who hire in specific geographic locations. But I think the reason for this - if a company has a legal entity in the US, maybe they don't want to deal with all the paperwork that comes with hiring people from Europe, and vice versa. But some companies are willing to invest in this, they believe in fully remote work, work from anywhere. These are the companies you need to try to look for. They are harder to find, but they do exist. And to find them, I guess you need to do what Jeff said.

## Should I include teaching on my resume?

30:06

Alexey

Okay, moving on. Should teaching and coaching on data science related classes be included in your CV in the past experience section?

30:19

Jeff

Of course. It's definitely relevant. It involves coding and it develops analytical thinking, and involves communication. All those things are great to highlight.

30:34

Alexey

To me teaching experience shows that a person can communicate. A person can not just learn something, but also pass on this knowledge. In teamwork, this is a very important skill, especially if you talk about senior roles, when it's expected from senior people to coach and teach more junior team members. This is very helpful. And also, to me, if you can teach something, it means that you also learn quite well. Because in order to teach, you need to know the thing quite well. There are only pluses, I cannot think of any negative sides here.

31:21

Jeff

I agree. I'm thinking of all the teachers that I've worked with in the past. A lot of them moved on from teaching, and they immediately got hired to engineering roles - to Netflix or Facebook, Microsoft, like name your company. But another reason why they're so successful is their mastery of the fundamentals. That's what you're teaching lots of times. That's essential in interviews and on the job.

31:51

Alexey

Yeah. If you taught a course about Python, then probably answering the question on the interview about some stuff you were teaching shouldn't be too difficult. Of course, if they go outside of what you are teaching, that might be trickier. But at least if you were teaching a course about something, then it gives you an advantage.

## Object-oriented programming for data engineering

32:22

Alexey

Moving on. Is object oriented programming a must for data engineering roles?

32:31

Jeff

You can probably find data engineering roles that don't require it. But I think of data engineering as of a software engineer that also knows cloud computing and data pipelines. And a lot of tools like Airflow use an object-oriented way to create DAGs and workflows. So yeah, I think that you should know object oriented programming.

## Python vs Java/Scala

33:03

Jeff

As for Java and Scala... I've had some students that didn't know Java but got hired in Java roles because they know object-oriented programming. So you can learn Java, but most of the roles that I see are Python. So that's what I recommend. Scala also comes up sometimes. That's more different than Python - it's more functional. It's more of a challenge to learn for beginners. It would probably disqualify you from some jobs that are more senior level. But if you're starting to get that first role, focus on Python. Ignore the jobs that want Java or Scala.

34:00

Alexey

Or maybe not ignore, apply anyways. Let them reject you, don't reject yourself.

34:05

Alexey

I think it comes back to the discussion we had previously about Spark. The point you made was that often you see Spark on job descriptions for more senior people. I think these are the same jobs where you see Java and Scala. Because Spark is written in Scala. So it's natural that you might also need to write some of your Spark jobs in Scala at some point. Or maybe at that company, they write everything in Scala. That also happens.

34:50

Alexey

If you already have some experience with PySpark, then they can hire you without Java. Just because you already have all the foundation. Scala and Java are different, but not so different that you cannot learn this on the job.

## SQL and Python technical interview questions

35:09

Alexey

A question from Christian. How many technical questions are you given in the interview - SQL and Python? I guess this is a question about the technical interviews where we were talking about all the leetcode SQL and leetcode Python kind of questions.

35:29

Jeff

You generally get 5-8 questions. It could be a take home, in which case, they'll list 5-8 questions that are Python or SQL related. Or it could be one of these live coding interviews. It could also be a timed interview.

36:03

Alexey

They are relatively small, right? Because I cannot imagine if somebody asks you to solve eight leetcode medium algorithmic challenges.

36:12

Jeff

That would be a lot. But they would be relatively small. And around a couple of different rounds. You do get these marathon interviews that are four hours long.

36:34

Alexey

I think I saw these questions. You're given a database, and the first question is to do a GROUP BY. The second question is about the HAVING clause. And another one. You still use the same database. You still write the same kind of queries, so you don't need to switch context every time, you know the data already. So it's about tweaking a bit the previous answer, adding the HAVING clause or the ORDER BY clause, or - God forbid - window functions that I have to google every time.

37:24

Jeff

I'd definitely prefer those interviews. Sometimes you will see one of these hackerrank style quizzes. You finish one problem, and then you click and it gives you a different prompt. I find them pretty challenging, because they're timed. And you have to just get used to the interface.

## GCP certificates

37:49

Alexey

I think we already talked a bit about certificates. So good. The question is, "Is GCP data engineer certification valuable for getting your first data engineering job?"

37:59

Jeff

I think we're in agreement. It's a skill set that matters, not the certificate. Maybe there will be some recruiters that will be looking for that and you'll get into that first round. But ultimately, you'll talk to a hiring manager. They want to know if you know these topics and if you can code. In my experience, candidates haven't had too much difficulty getting to the interview for data engineering positions. Because of that, I'd just focus on the skill sets.

38:38

Alexey

Preparing for the certificate if you study the fundamentals. If you don't grind for the format of the certificate.

39:10

Jeff

I've used some of those certificate books to ramp up on different features of AWS and some of them are really well written.

39:20

Alexey

In AWS there's IAM roles. This was blowing my mind all the time, until I took a preparation course for AWS certification. I just took the part about the IAM roles. I cannot say I got this completely, but at least now I have some order in my head, so it's not a chaotic mess anymore. So they can be helpful.

## Is commercial experience really necessary?

39:49

Alexey

Another question. I find that open-source contributions still fall short as they ask for commercial experience. I think "they" refer to companies. You have open-source contributions, but companies want to have commercial experience from you. How do you compensate for that? How do you apply for a job that requires commercial experience when all you have is open-source contributions?

40:17

Jeff

I would still apply. Like I said, people don't have difficulty getting to the interview round. So I wonder if it's just the way that it's spoken about. Also if you're making contributions to Prefect or something like that, it's hard for me to think that people would poopoo that.

40:45

Jeff

And if you're making contributions to a nonprofit with a team of engineers, you're essentially doing free work for the company at that point. So if you're doing that for some nonprofits, you can parlay that into an internship experience. You ask to meet with an engineer once every couple of weeks. I think there are companies that will mentor you.

41:27

Jeff

The other thing is maybe there are some gigs - contractor type roles - that you can take on at that point, do them and add that to your resume.

41:38

Jeff

But I often see that people misdiagnose where they're getting stuck. You think through which part of the funnel you're getting stuck in. If you're getting interviews, then your resume is good enough. That's the purpose of the resume - to get you to the interview. So you want to think if you're communicating your skills. Is there something you're not doing in the behavioural interview? Maybe you're not being as positive as you could be. Are you showing that this is really the job for you?

42:23

Jeff

But they want to hire people that have the skills. So as long as you can demonstrate that, there should be enough roles for you. There are going to be companies that say, "We only want people with 2-3 years experience". That happens every time I reach out to companies. But I also reach out to companies, I don't claim my students have any experience, and people are happy to talk and interview them.

42:55

Alexey

Another question is what actually qualifies as commercial experience? Is it when you get paid for what you do? Or is it when you work in a team with senior engineers and stakeholders and whatnot? You probably can have the second thing in data for social good kinds of projects. As for getting paid, I don't think companies really care if you were paid or not. All they care about is whether you can solve their problems. I guess you need to demonstrate that in the behavioural interviews.

## From sales to engineering

43:31

Alexey

Okay. I am currently 40 and work in sales. How can I convince recruiters to give me a chance to do a career change? I'm currently pursuing a computer science degree.

43:47

Jeff

I've had a lot of students that are 40+. I have a student currently, he is 40+ and he gets plenty of interviews. I'm trying to think when it would be an issue. In general, I don't find that to be a barrier. A good amount of people would see that as an asset. People want to hire adults. It's great to have someone who will show up for work, do their job, and be a great professional on their team. It can be an issue if a startup is looking to hire like a 23-28 year old to work 70 hours a week. That's simply not a good fit for you at this point in your life. You know it and they know it. That happens too. But there's generally enough roles. You should see where you are getting stuck in the interview process. In my experience, it really won't be an issue. I'm asked a lot about it when students enrol in a course. When they actually go on interviews and start the application process, sometimes they're the first people to get hired - at least my last couple of students that fit that demographic.

45:22

Alexey

Also thinking about sales. I know what salespeople need to do. They're pretty good communicators. You should use these skills because tech people usually do not have them. Engineers are pretty bad at convincing people, but salespeople know it well - because you sell. And here, you need to sell yourself and you've been doing this for quite some time. So you need to know how to package your skills to sell yourself. And you're in a good position for this. Even though this area is not super technical, you can use the skills to your advantage.

## Solution engineers

46:16

Jeff

One other thing is finding solutions for problems. You find solutions for the customer, given a problem. We have students that are solutions engineers, and a lot of that involves sales. That's a good background: it's understanding the client needs, seeing and understanding the issue, and then recommending how the product can fit into that. That's a big part of engineering - choosing the right product, understanding the issue, and taking the right steps forward.

46:51

Alexey

I saw solution engineers or solution architects kind of roles. So it's like pre-sales or post-sales.

47:02

Jeff

Exactly. They know a lot about the product, and they can understand the client's needs. They work with them to get on-boarded, and provide ongoing support.

47:14

Alexey

So maybe you're not working on the core part of the tool, but you know how to use the tool very well to sell it to the client. That's actually a good thing to try.

## Wrapping up

47:26

Alexey

I think that is the last question. Amazing. So we covered all of these questions. It took a bit longer than I anticipated. But thanks a lot, Jeff, for being available and for taking some time to finish this.

48:02

Jeff

Thanks, Alexey. Bye