1:40

Alexey

**This week, we'll talk about open source again. And we have a very special guest (second time), Vincent. So it's not the first time Vincent appears here. We already… I think you were one of the first guests ever on this podcast – that was more than three years ago.**

2:00

Vincent

Three to four. Yeah. Something like [that]. Yeah.

2:03

Alexey

**I think it was already… I was checking the podcast episode before we started and it was already season two. Season one was only five episodes, I think, and you were the one of the first recordings of season two. We didn't have transcriptions back then, so I had no idea what we talked about. But the topic was getting started with open source, so I assume we talked about that. Today we will talk again about open source. Actually, Vincent, you are… When I think about open source, you are the first person who I think about, because of 1000s of small libraries you have created and talked about.**

2:49

Vincent

It's not 1000s, just to be really clear. It's not 1000s.

2:52

Alexey

**Hundreds?**

2:52

Vincent

Nah, it's small… I mean, it's a small dozen, for sure. It is a bunch – that is definitely true. But thousands is a lot. A thousand really is a lot.

3:02

Alexey

**Well, compared to an average person in the industry… [Vincent agrees]**

3:09

Vincent

It's probably above average.

3:11

Alexey

**Yes. Like maybe the 99th percentile?**

3:14

Vincent

I actually… I don't know about… There's some sort of GitHub Actions thing that actually – per country (they check where you live – you can do that on GitHub). I did, at one point, learn that I'm in the top 10 people doing stuff on open source on GitHub in the Netherlands. In some year, I know that I was in the top 10. And I also noticed – I knew three other people that were also in that top 10. I know that I'm kind of up there. Not 1000s though. [chuckles]

3:46

Alexey

**Okay, fair enough. So that was the bio. I promised you I would improvise, because we don't have the bio in the show notes. I hope… You will tell us more.[Vincent agrees] So anyway.**

3:57

Vincent

Sure. Thanks for the intro. Lovely intro.

# Vincent’s Background

4:00

Alexey

**Thanks for being here. Before we start, I actually want to shout out to Johanna. The questions for today's interview were prepared by Johanna Bayer. Thanks, Johanna, for your help again. And the reason we're actually speaking here again, is because she met with you at a conference, right?**

4:19

Vincent

There was a PyLadies Code Sprint. There were a bunch of projects that were there, and on behalf of sort of SciKit Learning stuff, I was also there to help people get their first PR in. She was kind of in my SciKit Learn bubble doing docs work. Then she said, “Hey Vincent! Do you want to come on this podcast called Data Talks?” And I was like, “I think I've heard about that one before. Can I see a logo?” And I saw the logo and was like, “Oh! I've been on that thing before. But yeah, I don't mind coming back again.” One thing that is also kind of interesting with the intro you just said – I think a couple of years ago, we talked about SciKit Lego, probably. That was one of the projects that we ended up talking about.

That project actually had a role in getting me the job that I've got now. SciKit Lego are basically these Lego bricks that I've built back in the day with some SciKit Learn trick that I really like to use. It's kind of a project that I've been maintaining on the side. But five years later, it has like 30,000 downloads a month or so. Like, it's in production in a bunch of places. It's one of the reasons why, when some of the SciKit Learn core maintainers started thinking about having a company, they thought, “Well, having someone like Vincent around might be useful.” There was an interesting arc in that sense. Because back then I was just thinking, “Oh, this is just some sort of cute plugin.” But when it came to the job interview, there were these “Oh, but he's the guy behind SciKit Lego. Oh, okay.” So it's interesting, where five years later you start something, it can actually take you places as well that you can't predict – an interesting flashback.

5:48

Alexey

**So maybe what we should start with (and this is what we usually start with in our interviews) is your background. Can you talk about your career journey so far?**

6:03

Vincent

Sure. The short story is – I studied econometrics and operations research, pretty math-heavy stuff. When I was just around graduating, I learned about this machine learning thing and I was kind of eager to try it out. The goal I set for myself was “I'm gonna go backpacking for a bit, and I'm going to take some client work with programming with me.” And if I think clubbing is as fun as programming then that's a sign that I should pivot my career a little bit closer to techy stuff as opposed to, like, wearing suits and being a consultant stuff. Turned out, I actually thought programming was pretty neat, so that was a nice signal.

I did some tech consulting for a while. Then, at some point, I got a job offer from a company called Rasa, where they wanted me to be a Developer Advocate. I had never heard of that before. But it sounded pretty interesting because it was kind of a gateway to NLP for me. I really wanted to do NLP stuff and the consultancy company that I was working at… I mean, there are many reasons why I left, but I couldn't really do NLP stuff there at that time. So that was a good reason for me to switch.

Around that time, I was also a pretty big fan of spaCy, and two years after having worked at Rasa, the spaCy people – the Explosion AI people – they got around to funding and they could also use sort of a developer/advocate person and someone who could also be a core Dev. So I did Developer Relations stuff there and I was a Core Dev on their Prodigy product for two years. At some point, it was also time to leave there. But around the same time, there were these people from the SciKit maintainer group, who were starting this company called :probabl.. That ball got rolling pretty quickly as well.

So now I'm still doing open sourcey stuff there, but also Developer Relations stuff there. That's kind of the quick summary of all the stuff that happened. In between, I've also done… I've organized conferences, I've organized meetups, I built a bunch of open source projects. That's stuff you do on the side. But in short, that's kind of how the cookie crumbled.

8:01

Alexey

**How what?**

8:03

Vincent

That’s an American saying, “That's how the cookie crumbles.”

8:06

Alexey

**What does that mean? [chuckles]**

8:08

Vincent

Oh, I mean… It's kind of like, “Bob's your uncle,” like “That's the short story.” A cookie can only crumble in one way – the crumbs will fall down and they will never split in other directions. So yeah, that's the way the cookie crumbles. I've done NLP – let me say… Sayings are weird, just in general. [chuckles] But the cookie crumbles, I think… If I recall correctly, it is still an American saying.

# SciKit Learn’s History and Company Formation

8:33

Alexey

**I was wondering… So, there are companies behind open source products. Typically, these companies do not have the same name as the open source product. For example, Explosion AI – a spaCy. :probabl. – SciKit Learn. Then, there are a bunch of others…**

8:53

Vincent

Rasa has “Rasa” in their product, though. So some companies actually do that?

8:56

Alexey

**They do, yeah. So I'm wondering, why didn't they just call it SciKit Learn instead of :probabl.?**

9:04

Vincent

Well… Okay, in the case of :probabl., I can actually have a little bit of an explainer. SciKit is a huge project with a huge community. And some of its, I would say, core maintainers work at a company called :probabl., but it's not like the company :probabl. is SciKit Learn. There's a distinction there.

9:22

Alexey

**I see.**

9:22

Vincent

You could say that :probabl. could maybe be seen as a brand operator of sorts, because we do intend to hire lots of these open source maintainers to work on this – but it's a larger community. So claiming the name for a company would really not make sense.

9:37

Alexey

**I see. Makes sense, yeah.**

9:40

Vincent

And also, I guess, similar to the case of Explosion – Explosion does spaCy, sure, but they also do other stuff.

9:45

Alexey

**Prodigy, right?**

9:46

Vincent

Yeah. So they have a product and by really calling yourself the name of your open source thing, you're also limiting yourself to not doing anything else. Also :probabl., I mean… We will probably do more than just SciKit Learn. We're probably also going to do trainings, maybe a bit of consultancy, maybe also a product. There are many reasons not to call yourself SciKit Learn.

And I'm not even mentioning the legal aspect of it. Because I believe it's a name that already exists. Well, I don't know the full details there. Trademarks, I think, are for companies. But it is registered, let me put it that way. You cannot just call yourself SciKit Learn going forward. It is a brand that already exists.

10:28

Alexey

**I'm just checking the SciKit Learn website. I don't see any TM there. But it probably belongs to NumFOCUS or some other company.**

10:39

Vincent

To my knowledge it’s its own thing. Also, I believe… Okay, so I need to double check this. But NumFOCUS is like an umbrella that tries to do funding for some open source projects. But you've got the NumFOCUS projects, and you've got “associated” projects. And if I'm not mistaken, I believe SciKit Learn is an associated project. And also a project that’s… [cross-talk]

11:04

Alexey

**So NumPy is a NumFOCUS project, right? And SciKit Learn is only associated?**

11:10

Vincent

That is my understanding, yes. And these things can change over time. I'm not necessarily on top of that. But there are these subtle differences that, again, do matter, because SciKit Learn has been a large community for like decades at this point. So it's not necessarily [available] for a company to go out and claim. I do think that's important.

11:35

Alexey

**And before that, it was mainly… I know that the creator originally… Initially, it originated from Inria, right? The research lab in France. What's the story there? Maybe you know that?**

11:50

Vincent

I only know… I know parts of it. I do think it was something of an experimental thing during a Google Summer of Code project – maybe the most original version of that was. There was something about that in the story that I heard. But very quickly… It didn't become an Inria project fully, but a lot of the maintenance certainly did come from Inria, I think. I could be wrong, but I think four or five people, semi full-time, from Inria, were working on SciKit Learn and also writing papers about some of the stuff that they were doing via that work. It should also be said that, at some point, companies also sponsored developers. So Andreas Müller, for example – he's one of the core people who… I think he taught at NYU for a bit and that university, basically, also sponsored his open source work in SciKit Learn.

I believe Microsoft has a similar thing now. There are people over Quansight Labs that have a very similar deal. They do some consulting, but also have some time to add a PR or two. I think that's kind of how it naturally evolved. If you're a somewhat bigger company in the machine learning space, having a core developer be part of your team is kind of nice, right? [chuckles] So it's actually kind of an okay investment to go to that person to say, “Hey, you work for us, but one day a week, you're able to do the open source thing there. And all the lessons that you've got in your mind are just going to project onto the rest of the team and it can still make sense.” But, again, it's a huge effort – many different people from different vantage points. Also, there are tons of plugins in SciKit Learn as well. So…

13:24

Alexey

**Like SciKit Lego.**

13:26

Vincent

My experiment is a thing. But also, I don't know… UMAP. If you want to use the UMAP to the clustering visualization algorithm, that's technically seen as a SciKit Learn plugin – you have to install it by PIP installing UMAP–learn. And there are so many projects like it. Sure, you could argue that maybe that author who was working on UMAP didn't directly contribute lines of code to SciKit Learn when he was working on UMAP, but it is part of the ecosystem, and therefore still very valuable to the project, I would say.

# Maintaining and Transitioning Open Source Projects

14:01

Alexey

**One of the reasons I know that it's not really easy to contribute something new to SciKit Learn, because… Let's say there is a new method, and it's awesome, and they want to implement it in SciKit Learn. Most likely, my request will be rejected because the maintainers don't want to keep maintaining it, right? That's why it's easier to just create a plugin, and follow the API, and install it separately, rather than to put the load of maintaining this new thing on the maintainers of SciKit Learn.**

14:35

Vincent

Sort of. There are a couple of concerns. One, of course, is, “Well, I have a new fancy method!” But then, “Okay, did you benchmark it the right way?” Right? It can be seen as a career boost to have your algorithm move into SciKit Learn and people will acknowledge that – but it's not just the burden of maintaining it. It's also that people look at SciKit Learn as *the* example to copy. So the stuff that is in there should be like… good. There are a couple of algorithms you want to have around for historical reasons, like Naïve Bayes – I think it's somewhat safe to say that's not the state-of-the-art anymore. But it is one of the core algorithms so it has to be in SciKit Learn and that's a good reason to keep it in. But you can't really do that for every single paper out there. It would just get unmaintainable very quickly.

I could be wrong, but one reason why I think UMAP, for example, isn't in SciKit Learn, is related to the fact that UMAP relies on “numba”, which is this really cool LLVM compiler trick to get your Python code super quick. But SciKit Learn… I mean, then you're introducing a whole new backend for your compute as well. So sometimes it's also the dependencies that could be the issue. A fun thing, now, I would say is… SciKit Lego – the product I helped maintain – has a new maintainer now. And one thing that he really likes to do, which is awesome, is that he likes to look at the SciKit Learn issue list. And there are a couple of things where some of the SciKit Learn maintainers say, “Well, we're not going to implement this now, because… for lots of reasons.”

But a project like SciKit Lego can go, “Well, that sounds like a fun thing to implement. Let's do it!” The goal of SciKit Lego is to be fun for the maintainers and be useful. So some of the stuff that SciKit Learn cannot have, as long as it's fun to implement, we can go about it. But SciKit Lego is also something not to be taken as seriously as SciKit Learn – just to be really clear on that. [chuckles] But I think it's very fair for a project like SciKit Learn – if you want to be the stable core of everything, you’ve got to be a bit careful with what you let in. Otherwise, it might be detrimental to the project.

16:43

Alexey

**Can you tell us more about SciKit Lego? How did it appear? How did maintaining this library, and working on this library, actually lead to working at :probabl.?**

16:53

Vincent

When I was a consultant, at some point I noticed that you go to a client, and [they say], “Oh, we need to have something that can select columns from pandas. I'll write a custom thing that can do that.” You go to the next client, and – back then, definitely still pandas, definitely still need that component. At some point, I was just kind of done with re-implementing the same thing over and over and over again. There were just these components that are something, kind of… I get that SciKit Learn wouldn't want to have it, but it just made sense for this one use case. And I figured, “Well, if I just have a collection of Lego bricks that I've made myself, then maybe building out in the open just makes a whole lot of sense.” And I had a colleague, Matthijs Brouns, who was also really into it – both of us gave a lot of training. And then at some point, I also noticed that, if I want to give a course on how to do open source, I have a merge button! And it's in the public library!

So I can tell students during a course, “If you want to learn open source, we're just going to do a bunch of Git today.” So it's just kind of a nice utility. If you're giving corporate trainings to have an open source project with lots of these Lego bricks… And then it also helped me get a ton of contributors. Because if you go to a bunch of students for a traineeship [and say], “Would you like to spend a day committing to open source and that will be the lesson for the day?” They tend to say “Yes.” [chuckles]

18:09

Alexey

**That’s smart. [chuckles]**

18:11

Vincent

Yeah, it's one of the least boring things you could do, and it's kind of a nice win-win, because you actively… You help people get started with their next contribution this way and they actually learn something – and my library becomes a bit better. Now, fast forward a few years, I started working at Rasa, then I started working at Explosion. As time moved forward, I was using the library less and less and the same thing happened with my other maintainer.

We were still maintaining it, but there was this awkward moment when we were maintaining a library that we ourselves were not really using. So PyData Amsterdam happened again last year and I just started telling people, “Hey, if you're interested in learning how to be a maintainer, we are looking for a new one.” [video cuts out]

18:56

Alexey

**I think you’re back.**

18:57

Vincent

Oh, okay. We had a hiccup there. Where did you lose me?

19:02

Alexey

**So you said you were at PyData Amsterdam and you were telling people, “Hey, do you want to learn how to maintain a library because I have a library I don't want to maintain. Maybe you can do this?” [chuckles]**

19:13

Vincent

Well, it's not so much that I don't want to maintain it. It's just that, in fairness, I was just looking for someone who actually uses it. That's more of the thing. Again, back then, I wasn't working at :probabl. yet. The prospect was that I might have been still doing NLP for the next year or so. Then, SciKit Lego is tabular data instead, so it helps to have someone around – and someone volunteered, Francesco.

And it's been great. He's picked up stuff that I wouldn't. And he's added things to the library that I also wouldn't do, but it's better because of it. But moreover, he has fun doing this as well. So we're both on the same Slack and we just have fun maintaining it together – way more than I did before. So that's also just kind of nice. There's fresh blood in the project, I guess you could say.

20:02

Alexey

**So again, the way you did this is – you were just randomly approaching people at the conference, you were talking to them, and then you started asking, “Hey, I have this library. I'm looking for maintainers because I'm no longer using it. It needs…”**

20:17

Vincent

Well… It's kind of the other way around. Part of it was that people would walk up to me about SciKit Lego stuff, because I was a maintainer and I do know some people. So every time that someone would mention that to me, I would say, “By the way, we are looking for a new maintainer.” That was something I would do. In the case of Francesco, though – we met before because he actually made a PR to SciKit Lego. Before… I did tell him, “Hey, thanks. It was…”

He made some solid PRs and I did say, “Look, if this is fun, let's talk about that at PyData Amsterdam. So in the case of Francesco we kind of met virtually before, and he already told me that he was somewhat interested. It's just that it also helps to meet in person. So having that moment at the PyData event was also just great.

21:03

Alexey

**And he uses SciKit Lego at work, right?**

21:08

Vincent

Yeah. He's not at liberty to confirm to what extent [chuckles] because some of that is a bit private. But he did confirm to me that there were some tricks in there that he definitely used, if only to explore stuff in a notebook. And he found it [to be] a fun little library. One thing I do imagine that's appealing, if this is a first project that you maintain… I'm kind of explicit that I would really prefer the library just to be fun to maintain, and be useful as a second goal. We are critical at what we accept, but there's also this realization that if it's not fun for the maintainers, it will just not be maintained anymore – and that's the biggest risk at this point.

# Teaching and Learning Through Open Source

21:49

Alexey

**How do you make a library fun to maintain?**

21:51

Vincent

Part of that is just to celebrate, “Well, we're all doing this as volunteer work.” Right? If something is fundamentally broken, then you kind of feel obliged to have a look, but I'm not going to pressure anyone into implementing anything. In fact, I would prefer people consider implementing things if it's either in their domain, or if it sounds like a fun experiment. I do have a reasonable requirement that, if you're going to add something to our library, we want to have some sort of benchmark that confirms that it works better than the standard way of doing it. Because I don't want to maintain… We have a couple of silly components, but having the benchmark is also great for documentation – it helps to motivate people to explore it. But really, that's about it. I'm not going to tell any maintainer that they *must* do something.

Also, for myself, I've got a kid now who's a handful, and I'm not gonna spend my precious evenings [chuckles] implementing stuff that I don't think is fun. I don't think that that's reasonable. I like to think that that also helps the appeal. But in the case of Francesco, from my perspective, it's just fun to see how he picks stuff up and does stuff that I wouldn't do. I also think that that's just kind of nice for the project. So yeah, have him as a podcast guest, maybe – because he can tell you his perspective. But that's the cool thing in this case about having the extra maintainer who hasn't been around for a while. He really has a fresh perspective, and also ideas of his own. And that's kind of a nice combo, because it keeps it fun for both of us, I like to think.

# Role of Developer Relations and Content Creation

23:29

Alexey

**Okay. I asked you multiple questions in one question. First, you told us about SciKit Lego. How did it actually lead to your current job position?**

23:44

Vincent

You would have to ask the people on the other side of the fence for their part of the story, but one thing I will [say is that] it was more of a thing where it's a SciKit Learn maintainer group, basically. There are some very senior people from the SciKit Learn project. And you can kind of imagine – if you go to YouTube, and you type “SciKit Learn tutorial,” some of that stuff is going to be really hypey. Because data science is just so hot right now, right? That kind of a thing.

My impression was that they were really looking for someone, if we're gonna do Dev Rel stuff, there was a very strong preference for someone who could still explain very clearly – just not the hypey thing. If you want to do data science and machine learning, you have to take benchmarking seriously. There's a lot of tomfoolery, we'd like to not have that.

Then, when people started interviewing me, one thing that I do think really helped was “Oh! Looking at Vincent's resume – he wrote SciKit Lego! That is actually one of the larger plugins. Oh! He takes testing seriously. There's all these things that we are usually worried about, but okay. That seems to check out quite kind of nicely.” Honestly, I think that if I didn't maintain SciKit Lego, there was still a pretty good chance that I would get in, but the interviews would have gone way differently.

Basically, the technical interview bit was really lightweight, because they just looked at SciKit Lego and just kind of went, “Yep! Legit.” [chuckles] So that's, I think, how it helps me in that sense. But other stuff helped me as well. It's not just SciKit Lego. It's also the conference talks that I've done. I gave a keynote at PyData Amsterdam that some of them really liked. Stuff like that also really helped.

25:27

Alexey

**I was talking about thousands of small open source libraries and you said, “It's not true. It’s not thousands.”. But when it comes to the talks… I was just doing this the other day – I typed your name in the YouTube search…**

25:41

Vincent

Yeah?

25:42

Alexey

**I couldn't finish scrolling. [chuckles]**

# Teaching Through Calm Code and The Importance of Content Creation

25:46

Vincent

I mean, yeah. I am a frequent speaker at PyData. That is definitely true. I mean, it's something that I enjoy doing and whenever I send in a proposal, so far, they’ve accepted. So it's easy for me to hit repeat on that. I guess another thing that also plays in with this is, during COVID, I started this project called Calm Code, which is kind of a tutorial website, if you will. I kind of designed it to be an alternative to DataCamp, because there's some stuff I didn't like about them.

I like the way they teach. There's this… I'm sure the company has changed, but there was a time when data camp said, “Oh, you *must* learn these skills in order to be future-proof.” And I always just thought that was just nonsense. So I figured, “Well, what about a learning platform where you just learn ‘Well, here's the trick.’? And if you know that, that's great. That's enough.” Kind of have that vibe a bit more.

26:37

Alexey

**That’s why it’s “calm” right? No pressure to learn things. It's like, “You can learn whatever you want.”**

26:44

Vincent

Yeah. Here's just some stuff to make your day-to-day nicer. Like, that's it. The thing about that is, if you're the [kind of] person who, like me, took that seriously and had been making that for over a year, you kind of gain a skill of, “Okay, how do I communicate things clearly?” Because it's something you actively practice. It's a problem you're exposed to. Part of it is also, of course, if you've done lots of presentations, the next presentation is going to be less scary, but… I've made over 700 videos for that platform, and making a video now for me is like a “takes me an hour” kind of exercise. It's like writing a FOR loop in a way.

27:20

Alexey

**Thousands is actually not too far from the truth, right?**

27:24

Vincent

If we're talking about videos, then yes. I do think if you take all of them together, then we might be close to 1000.

27:30

Alexey

**Also, counting all your work as a Dev Advocate at Rasa, Explosion, right?**

27:39

Vincent

Yeah. At Rasa, that's like 100 videos or so – an education platform, as well as plugins. Then at Explosion… I mean, an Explosion, we did have a preference for quality over quantity, so I think I made way fewer videos, but each video I made was more polished. But it's basically like, when you learn Python for the first time, it is super scary because there's like, “Oh my God, how does pip… Pip and virtual ends? What is this? But after a year or two, you’re just, “Oh, it's a pip install,” and you just don't think about it anymore. Very similarly, it's like, “Oh, how does this recording software work? God!” And then at some point [it’s like], “Oh, wait. This is the shortcut.” Click, click, click – done. I'm a little bit more worried about “Do I have a fun example or a good insight?” Because if I have one, then recording it is just not the hard part anymore. That's kind of the easy bit.

28:28

Alexey

**Coming up with examples is the most difficult part, right?**

28:32

Vincent

Yes, yes. I mean, one thing that does help – I do have access to some of the SciKit Learn core maintainers. So I can @ poke them a bit like, “Hey, what's the most annoying question you've seen pop up on the GitHub Issues?” That could be a good thing to pick up. A lot of these maintainers are also happy nerds on the inside, right? There are some crazy experiments that I can do that they'd still be interested in the exercise, but they wouldn't do themselves because they're too busy and that's stuff that I can pick up. There are some crazy benchmarks that I'm running right now – like 100 tabular datasets, just to sort of benchmark our hyperparameters because I'm curious. If we make a change, can we measure that? So there is enough inspiration to go around if you have a team. There's also kind of a lesson [there] I would say

# Current Projects and Future Plans for Calm Code

29:25

Alexey

**When it comes to Calm Code, do you still actively put things out there?**

29:30

Vincent

Yeah, so for Calm Code, I came to a similar conclusion as with SciKit Lego. I could do this thing on my own but, from a sustainability perspective – on one hand, there are also limits to my knowledge, so having a collaborator around just for that would make sense. Someone who's a bit more in-depth into databases and stuff. That would be useful, I think. I could learn, I could teach myself, but it would be more effective if I have someone around who just knows stuff that I don't know. Also, for the motivation, I just noticed it'd be more fun to collaborate with someone. What has been happening is, there is now a collaborator. We were actually building a proper platform for Calm Code.

Because up until now, it was like two FOR loops and a bunch of markdown files. That's amazing when you are working on your own, but it's not as amazing when you want to have… [Alexey inaudible] Well, at some point, maybe you want to have a login and a payment thing, and stuff like that. So the thinking here is – to make a thing sustainable, I would like that to be a hobby project that just gives a trickle of income so that this collaborator and I can hire someone externally, who can give us notebooks that we can turn into videos? Let's say something like that. I think [that] would be a really cool end game – something we could do longer term. So we're building this right now. The Django app is live. People have not noticed it going down or anything, even though we made tons and tons of change.

30:53

Alexey

**So right now it’s Django?**

30:54

Vincent

Yeah. So it's a Django – full Django… [cross-talk]

30:56

Alexey

**[inaudible]**

30:58

Vincent

Yeah. We're adding all sorts of stuff to it right now, but… We're learning how to do payments, which is a huge pain. [chuckles] But the goal will be that this will just be… I would like to do more content on that thing. I have been adding a new short course like once a month. So I'm subtly still doing stuff. But the hope will be that this will still be a calm place, but maybe we can do more about databases. And maybe we can do a little bit more data analytics. And maybe a little bit more cloud, a little bit more Docker and Kubernetes. Not just have Vincent do his Vincent thing, but just a more general, “Here's just Calm Code.” That's kind of the next phase of the project.

31:38

Alexey

**I see that there is already some Docker stuff, right?**

31:42

Vincent

Um, barely. I have in my mind what I would like to add. There are a few mini-Docker things on there. But there's way more that we would want to do. One of the things I'm interested in doing is – I would love to have a sort of “insights and benchmarks” section where it's more about the demo and an idea [and] it's not about being optimal. One thing I am interested in doing there with Docker is… So you know GitHub Actions, right? When does it make sense to have a custom runner? You're not going to give your compute costs to GitHub, you're going to give it to your own VM. Like, when does it make sense to do something [there]?

# Data Processing Tricks and The Importance of Innovation

32:18

Alexey

**A runner is when the action is executed on your environment [Alexey agrees] not on GitHub’s environment?**

32:26

Vincent

Yes. But the thing is, if you do that, there's actually a trick you can pull off. Because if it's a VM that you own, and you have Docker on it, it does the caching for you on that machine. So if you have insane Docker containers with sentence transformers… big-ass PyTorch models, etc. Assuming you don't make any big changes to the core library, that entire thing is going to be cached – no download needed. And sure, you could do something like that with a GitHub registry, and you could be hooked up to the Docker registry to your GitHub Actions. You could go there. Oh, but hang on! If you just do this on the VM, you just kind of get this for free – that's kind of a nice little benchmark and insight.

33:07

Alexey

**But the VM is not free.**

33:09

Vincent

Well, sure, but you can pick your own VM. It could be something that you've paid ahead of time with a good rebate. It can also be something that you have in your Kubernetes cluster anyway. And this is the thing I kind of want to pitch there – it can also be a VM from a cloud provider that has carbon negative compute, and they do exist. The big cloud providers don't do this, but there's a startup here in the Netherlands called Leaf.cloud – and I do want to give them a shout out, because it's a badass idea.

They put server racks in basements of apartment buildings, there's a glass fiber connection to a data center, so they have all the disk there. But the compute happens in the basement and all the heat that comes from that is used to preheat water, therefore saving gas use. And when you do the math, it turns out that that's actually carbon negative because it reduces the gas usage. Another benefit is that you only pay for the compute and dealing with the heat is now paid for by another party, so it's actually economically competitive as well.

When you put all of that together, then you kind of go like, “Okay, there's all sorts of cool stuff you can do if you just own it yourself a bit.” And I would like to have more of those examples, where you don't need a cloud provider for everything. Sometimes you just need to be aware of the fact that you can really still do everything. It's code – it's yours. That intellectual freedom is something I would like to celebrate more on Calm Code going forward.

34:29

Alexey

**Yeah, speaking of Docker – you mentioned that for us (for people who have been around the Python ecosystem for some time) Pip install is the everyday thing you do. You don't even realize it – it's muscle memory. Create a virtual environment, do pip install. The same thing with Docker, but for people who’ve just started [Vincent agrees] Docker is super difficult. This is what I see in our courses. In our data engineering course, the first module is about Docker and this is the most problematic module in the entire course.**

35:07

Vincent

Yeah. I think there are three things, usually, that for someone who’s fresh out of college is a stumbling block. So Pip is one, Docker is one, and Git. I think those are kind of the big three.

35:19

Alexey

**You want to cover all three, right?**

35:21

Vincent

Eventually, yes. But designing a good course is something I want to do in my life only once. [chuckles] If that makes sense.

35:31

Alexey

**You already have. I think I saw a logo of GitHub on your...**

# Learning the Fundamentals and Changing the Way You See a Problem

35:36

Vincent

Yeah, I use GitHub in a bunch of courses, but I have no... Calm Code, at the moment, doesn't assume that you're a full-on beginner. That's not the platform for you. You need to have read your first programming book and done your first thing.

35:48

Alexey

**That’s what we do, too. [Vincent agrees] I'm wondering where, in the end, people actually learn all that stuff. [chuckles]**

35:54

Vincent

Well. One thing… This is a hypothesis of mine when it comes to teaching, but sometimes I wonder, maybe it's not so much the tool that I gotta teach you, but the way you should think about the tool. I can teach you the commands of Git, but what I can also teach you is just how you should think about it conceptually. Or something like, “If something goes wrong, it's probably this.” And that's not really a syntax thing, it's more of a “Does it click in your mind?” kind of thing. Also [things] like, “Hey, don't use Git for data, maybe? [chuckles] Use a different system for that.”

If you have huge CSV files, and you want to do versioning, yeah, you can do *some* of that in Git, maybe – if the data is not big enough – but don't use it for everything and here's why. That context, I think, is something I also want to try to focus on with Calm Code things. Especially because a lot of these YouTube tutorials, they just regurgitate the docs anyway, and I do want to do something more than just that.

36:50

Alexey

**Yeah. Interestingly, while we were talking about that, I realized that I never took a Git course, I never took a Docker course, I never took a Python course… It was just, “I need to do something. Okay, how do I do this?” I do this and I learn. I guess following this approach, like, “Okay, I can assume that they know Docker. If they don't, then they will learn.” Maybe that's also a good way of [going about it]?**

37:16

Vincent

This is just thinking out loud – I'm kind of curious about your opinions while [we’re] here. So, okay, what can I do to make it as easy or enjoyable as possible to learn Docker? I mean, I can teach you the basic syntax, and it's *a* way to do it. Maybe what will be nicest, is if I can figure out 20 minutes that you have to spend watching my things – but after that, you kind of understand the basics and you're in a position where you're able to tinker around. I think part of me is wondering if maybe that's the key. In the beginning, you're not comfortable tinkering around yet, because you don't know how the Lego bricks click together.

But once you're there, then I actually really want to encourage you to tinker as much as possible. That's also a kind of a philosophy I'm playing with here. I can imagine getting to a “minimum viable tinkerability” or something like that. Something about that feels appealing to me – I want to give that more of a spin. I don't know if that's something you're also considering, but…

38:14

Alexey

**That's a nice idea. I realized that today’s topic is actually not education, even though this is really great stuff to talk about. [chuckles]**

38:22

Vincent

Sure – segue back to SciKit Learn stuff, I guess. [chuckles] Yeah.

# Dev Rel and Core Dev in One

38:26

Alexey

**One of the things you mentioned is… So you work at Rasa as a developer advocate, and you worked at Explosion as a Dev Advocate and also a core Dev. This is what you also do now at :probabl., right? You wear two hats – one is the educator hat, where you talk about the product, you promote it, you educate about this product. But also, at the same time, you actually work on implementing core features. Can you tell us how that actually works? How do you manage to do both of these things?**

39:07

Vincent

I will say, back at Explosion, at least the way I experienced it – the company experimented a little bit with a Dev Rel team, but at some point, we also just knew that “Okay, but maybe we should just assume that everyone's a machine learning engineer?” So my title also wasn't Dev Rel or anything. It was just Machine Learning Engineer. And, if you're a good Dev Rel, you also know your stuff, so you should also be a machine learning engineer anyway. It's just that you're also more comfortable and well-equipped to make content and to educate. It's more like an extra.

But it's not that different from “Oh, yeah. I'm a full-stack developer, and I do a fair bit of Kubernetes well.” It's like the Dungeons & Dragons character that has more points into Kubernetes, but it's not like you're not a Full-stack Dev. Similarly, I still like to think about myself as a machine learning engineer, it's just that I’m a bit more comfortable and a bit more skilled, perhaps, and making some of that content stuff. So, in that sense, at some point within Explosion, there was this more need for me to do Prodigy stuff. And I'm still the same person, so I would just do more Prodigy stuff and do less content.

Something similar is happening here over :probabl., I think. It's just that, right now, I'm the main developer advocate “thing” within the company, so it makes sense that most of my time right now in this phase is spent bootstrapping a whole bunch of things. I have a whiteboarding playlist on our YouTube thing, I have a live stream – we also just set up a podcast, and we're gonna do a second one at some point. That's not going to be just me, it's gonna be me with a colleague. But right now, the company is in a “bootstrapping the Dev Rel” practice phase.

Once that ball kind of gets rolling, I will probably do more open-sourcey stuff that I'm doing now. Right now, I am helping out with the Skrub effort – doing benchmarks and some of those things and sharing ideas. There's an interest to port some stuff from SciKit Lego to Skrub, maybe. So it makes sense that I think along there. But I really wouldn't see it as “Oh, I have two different hats that I'm juggling.” I guess it's way more “Well, my name is Vincent. I'm a Senior person and I work on stuff that matters to the company right now.” That's, I think, the best way to also think about it.

41:17

Alexey

**Yet your title is Dev Advocate, right? Or what's your title?**

41:21

Vincent

It's Developer Relations Engineer. When they hired me, I made a joke that I would really prefer Senior Person as the title, but they did say, [chuckles] “Well, if you have to pick something better than that,” which I do think is fair. [chuckles] But I do still appreciate the way that Explosion just went about it, “Just call everyone a Machine Learning Engineer.” Also [they do] not fuss too much around with “senior” or “junior” or anything. Because you are just a person in the end, and you just try to fix problems for the company. In the end, that's usually what you end up doing. Constraining yourself to one and only one role just feels very counterproductive. But sure, I do a whole bunch of Dev Rel stuff now – so Senior Relations Engineer, I guess, is also totally fine by me. It’s just that I would have had a preference to be a “Senior Person” instead. I’m fully aware of the fact that that’s also a jokey title. [Alexey chuckles]Totally cool with them not going with it, by the way.

# Why :probabl. Needs a Dev Rel

42:13

Alexey

**Why does SciKit Learn actually need a Dev Rel? Is there… [cross-talk]**

42:20

Vincent

This takes me pack to :probabl.. The company :probabl. has hired me – it's not SciKit Learn, to be clear. :probabl. is not the same entity, right? And but you are right in the sense that, SciKit Learn is a pretty good project when you look at the documentation. It is one of the…

42:39

Alexey

**It has the best documentation that I can think of. For me, this is a good example of how documentation should look like – how you should document API.**

42:49

Vincent

It's funny you mention that. There's a person that's a colleague of mine, Arturo. So he is actually… not necessarily “in charge” but he's kind of a Lead on the docs – that's one of his main concerns in his day-to-day. And yeah, the docs are amazing. But imagine being on the other side and thinking, “Oh, but there's so much more stuff that we could do.” [Alexey agrees] “It will be amazing if we could just run the code examples interactively in the browser.” We're not there yet but we would like to. Similarly, the podcast is a nice example, too, because maybe there's just the experience from the maintainers that we just want to have archived and documented as well.

You're definitely correct in saying the docs are very good, and we should celebrate that. But there are also more things that we could do. That's kind of the idea first and foremost. YouTube videos, I think, do contribute a bunch of things. We just reached 10,000 views on our YouTube channel, where I just explained some algorithmic details that people aren't necessarily aware of, and it's still very valuable to the SciKit Learn project. You could argue that some of this is in the “nice to have” department, but I would still argue that it adds value. If something that's very confusing, or [something that is a] best practice is relatively not known, that's a good thing to still have tutorials for and sometimes video is a nice medium for it.

44:07

Alexey

**Because people like me, and other educators – they have courses, they have content, they have articles about SciKit Learn. You don't need to do anything. There is already a ton of content that explains it. But I guess you said that we need to make a distinction between :probabl. and SciKit Learn right? So your job is to promote :probabl.**

44:30

Vincent

Via :probabl., I will promote SciKit Learn, yes. But I do think it's an important distinction because SciKit Learn has a MOOC, for example, right? I had no part in that. That's something that was done by community members. It wasn't something me or :probabl. did. But I guess like one thing that I do try to do, which is… This is kind of hard to do on the docs page, but I do think it's very valuable. So you know how in SciKit Learn we have scalers?

44:57

Alexey

**Like min/max normalization, right?**

44:59

Vincent

Yeah. We’ve got the min/max scaler, we've got the standard scaler – and the idea is you've got your tabular data frame, one column has super high values, the other column has super low values, and there's numerical stuff that could go wrong when you give that to a linear regression or a KNN classifier or something like that. You want that to be standardized in some way or form. So a lot of people like to use a standard scaler that takes the mean for a column, subtracts it, and also scales the variants such that the variance is always equal to one – or standard deviation, I should say.

Okay, fine, great. But the thing with that standard scaler is that it's actually not standard at all when you think of all the stuff it has to do. What if the data that's going in – what if that's a sparse matrix instead of a normal NumPy one? Oh, well, we can't subtract the mean anymore because that would turn that sparse matrix into something non-sparse. But we can normalize the variants. Oh, okay. So that should be a feature, we should be able to do that, actually. That'd be super nice. All right. What if it's a data frame instead of a NumPy array? Okay, we gotta handle that, obviously. And not just pandas, by the way, but also polars these days. All right, all right, we’ve got to be able to account for that. Oh, but there's also this other thing where it has a partial fit method, because technically, from a mathematical standpoint, nothing has stopped…

Usually, you calculate these things in the batch, and you take the mean of the column. But what if the user is interested in doing microbatching because the data set doesn't fit in memory? Well, SciKit Learn has a partial fit API that allows you to take those batches and do it as a sort of an “online learning” way. Oh, okay. So that has to be implemented on the standard scaler. Are there many ways of doing it? Yeah. And a bunch of methods are numerically super unstable, so you have to be really careful about the method that you pick, because otherwise, the whole thing is just going to explode.

This is a video that is coming out this week and there are so many of those little details that are kind of hard to put in the SciKit Learn tutorial, because as a user, the whole point of SciKit Learn is that you're not worried about that. But I do think it is valuable to the project that I'm able to spend at least a little bit of time making a good video about it so people appreciate it.

47:03

Alexey

**Ah. Because I was going to ask, “Okay, yes. This is a very difficult problem. Why would I actually need to learn the details?”**

47:13

Vincent

So the beauty of it is, you don't. But it might still be useful for you to appreciate some of these details. That’s kind of the larger point.

47:20

Alexey

**Yes, that is the case. I have never realized that such a simple thing could be that difficult.**

47:25

Vincent

The Standard Scaler is Not Standard is the title of the video that's coming out this week. It is not standard. [chuckles]

47:29

Alexey

**I see. Because like… it's too mathematical. “Two, three, compute the mean, then you subtract the mean…”**

47:38

Vincent

On the spectrum of math, this is still relatively lightweight. But still, the fact that it's a lightweight mathematical, not-too-complex thing – so much can still go wrong.

47:46

Alexey

**Yeah, I never realized that.**

47:48

Vincent

So again, the SciKit Lego thing definitely helps you. Because I've looked at the source code of SciKit Learn sometimes to understand how I should implement it in SciKit Lego. So I'm kind of well-equipped to know which parts are worth diving into a bit. [There’s] definitely stuff for me to do there. But I should also mention… Probably [chuckles] it is very :probabl. that in, let's say a month or two, when we have the new quarter in the summer, that the ball for the content is rolling and rolling. Then I can dive a bit more deeper into like Skrub stuff. And I probably will.

# Exploration of Skrub and Advanced Data Processing

48:27

Alexey

**What is Skrub? You mentioned it like three-four times. What is that?**

48:31

Vincent

Yeah! It's a pretty cool library. So Skrub, at the moment, is a SciKit Learn plugin. It's still in an experimental phase. But Gaël Varoquaux, who is one of the maintainers of it – there's a bunch of other people's.

48:42

Alexey

**So this is how you pronounce his last name? Okay**

48:44

Vincent

I don't know. One thing to remember is that Gaël is French and, for example, Vincent is pronounced “Van-sant” in France, right? So don't take my pronunciation as accurate. [chuckles] But I believe that's how it's… I don't recall being corrected. Yeah, let's just call him Gaël, because that also is his name. Now, so Gaël… The dream of that project is that… Let me put it this way – there's one feature in that library right now, that is a sort of guiding arrow towards the dream we're trying to chase. So there's this thing called a table vectorizer. You give it your data frame, and it just figures out what it should probably do. It's kind of the 80/20 thing where, Okay, there's only three categories in this column so we want to one-hot encode that.” “Oh, there's over 100 in this one. Okay, let's do a topic model thing instead, because we probably don't want to have that sparse of an input.”

There are a couple of tricks like that. And the idea is, we just have enough features for you to handle the tabular stuff, such that your pipeline is just a few components without having to be very elaborate. This should give you a very reasonable benchmark. The dream is kind of… If you just use a table vectorizer, let's say, and then histogram gradient boosted regressor or classifier (depending on your use case) that you're kind of 80% of the way there. Like those components.

50:12

Alexey

**[inaudible] almost,**

50:13

Vincent

Well, these components are definitely a little bit too experimental to go into SciKit Learn directly right now. But you can imagine the… There's stuff to be gained here that will be very pragmatic and useful. So that's kind of the overarching goal of that effort.

50:25

Alexey

**Sounds quite cool, yeah.**

50:27

Vincent

One thing I do think is nice – there's one trick in there that I do think is maybe nice to mention. There's a thing in there called a GAP encoder.

50:36

Alexey

**Gap… encoder? For [inaudible]**

50:40

Vincent

Yeah, GAP actually stands for gamma… It's a distribution name – it doesn't have anything to do with a gap in the literal sense, but it's more of a topic modeling thing that's happening in there. So let's say that you've got a column with a dirty category. So there are typos in there. The example we have in the docs is job titles. So you're a Senior Programmer, or Senior Fireman Expert – and then there's typos in there as well. Well, we could one-off encode all of that. We could. But you can also look at that and say, “Well, ‘senior’ has meaning.” Almost as if we're modeling this as text, right? Maybe there are clusters in there.

So instead of one-hot encoding it, let's just tell the algorithm, “Well, try to figure out 20 topics or so and return that as a dense array, where probably all the Senior Firemen or Police Officers there [are] in one topic,” and etc. And this is kind of a nice way to prevent that explosion of one-hot encoding from happening. Tools like that are also being used under the hood here and there's a little bit of research behind this as well. Again, like before, we aren't expecting you to be aware of all the algorithms under the hood – we would like it if you appreciated them – but we're really trying to make it easier for you to just dunk a data frame in there and have a solid benchmark. From there, you can fiddle, but have that solid benchmark there, also to compare approaches from academic articles.

52:09

Alexey

**Yeah, that does sound awesome. It also answers… [For example,] in our courses, we have questions like, “Okay, I realize that with 10 categories, I can use that. But, what if we have 1000 categories?” And then there are a million ways… Well, there are *many* ways you can deal with thousands of categories, right? You can do topic modeling, you can do min average encoding, or whatever. There are many, many different ways.**

52:34

Vincent

You can model it as text. There’s lots of stuff that you could do. You can use embeddings even. But the hope of Skrub is that we aren't necessarily going to be perfect for every use case, but it should at least be a reasonable place to start with very little effort.

52:46

Alexey

**That's what I was thinking. Instead of saying, “Okay, look. There are these and these and these options.” The answer instead could be, “Try this library and see what it comes up with and try to understand what encoders it uses for encoding your table. Then you will understand how it works.”**

53:06

Vincent

Yeah. The thing to also remember here is, we will never be able to do it perfectly, because suddenly “Oh, this is a string. Yeah, but in the string, it's an IP address.” [chuckles] Not sure if we can detect that, right? Some of that stuff still needs to be custom. But another avenue that I'm helping out with there is just the time series stuff. So how do you encode a date or a date/time? Can we make that distinction? What are sensible defaults there? That's also something I've got an opinion on.

# Personal Insights on SciKit Learn and Industry Trends

53:36

Alexey

**So SciKit Learn existed for quite some time without a company behind it. Why, all of a sudden…**

53:47

Vincent

So what I'm about to do is give a perspective. For many people in the company, they might have a slightly different perspective on “Why would you start a company?” But I can imagine that when they started this, there were a couple of concerns. One of them was – well, it is an open source project, and a lot of it is sort of funded via… I do think Inria definitely did a lot of good stuff with funding it. But you can wonder, “Do we really want the open source project SciKit Learn (which is at the center of everything) to be dependent on academic funding models?” I mean, it's worked out pretty okay, from what I can gather (not being a maintainer).

But there is something to be said like, “Okay, there's still a funding concern no matter what. So maybe a company could make sense there.” Then there's also the matter of the fact that it adds tremendous value – and there might be companies willing to pay for it, right? Some companies have a lot of time and not a lot of money. Some companies do not have a lot of time, but a lot of money. It will be kind of nice if we can figure out a company structure such that that money is actually spent on maintaining SciKit Learn, and it's a little ecosystem. So that definitely is part of it.

Another idea is also that – it would be cool if maybe there are just more European companies doing this sort of stuff. France is trying to position itself into this a bit – there’s Hugging Face, there’s Mistral. There's a bunch. But :probabl. also kind of fits that story like, “Let's have more European-centric tech companies as well.” Because it feels like all the cloud providers are from the US these days, and it would be kind of nice to have more stuff from Europe. Then there's also, if you're a company… This still needs to be proven, but I can also imagine that by putting some of this stuff in a company, you are also exposing yourself a little bit more to industry problems.

Because… I don't know for sure, but I do think it's fair to say that a large chunk of the SciKit Learn maintainers do have an academic background. Some of them will definitely also have industry exposure. But there is also something about the fact that like, “Yeah, by actually being a company, you are going to be doing company stuff. Just being exposed to that it's probably also going to be good for the project.” Again, these are just reasons… Everyone within the company favors one reason a bit more than the other, maybe, but this is the general vibe. Then at some point, you do kind of go, “Yeah, okay. A company makes sense. This is something that should be tried.” That's at least the vibe check of the origin story as I understand it.

56:15

Alexey

**And the business model is still yet to be determined? The exact business model.**

56:19

Vincent

Um, there's some stuff that I *can* share. You can definitely imagine us doing training and consulting. That's going to be a thing.

56:28

Alexey

**Yeah. You kind of mentioned that.**

56:30

Vincent

It's on the website. I mean, it's on the website. There are also these collaborations that may or may not happen, right? There might be a cloud provider that sort of integrates with us? There's all sorts of ideas that could happen there. But it's definitely early. So there's not an official announcement I can do or anything like that. But you can… There's a TechCrunch article where our CEO says a bunch of things. There are things you can check out.

56:58

Alexey

**[The article] was published on February 1st?**

57:03

Vincent

Uh, February-January, earlier this year. Something around that, I think. That's when all the official announcements kind of start.

57:10

Alexey

**Yeah. So “What we do. Product: Open source services – provide training, certification, and expert solutions for enterprise AI challenges.”**

57:19

Vincent

Yeah, that just about covers it. [chuckles]

# Vincent’s Upcoming Projects

57:24

Alexey

**Yeah. [chuckles] Well, we don't have a lot of time [left]. And I still want to ask you this. What is your next personal project?**

57:34

Vincent

Oh, I can't [say]. So there's one that… There's. What *can* I talk about there? There's one that I can't talk about yet. Oh! There's one that I can't talk about, actually. Calm Code will have a book.

57:48

Alexey

**Calm Code will have a book?**

57:50

Vincent

Yeah. So we're writing a book. I can't share the title just yet. But it's about expectations clashing with reality in the field of data – it’s gonna be the title of the book. There's just been too many of these stories where… Remember, back in the day, when people told us “Data science is going to be the sexiest profession of all time!”?

58:11

Alexey

**Yes, I guess I [laughs][inaudible] was doing. I want it.**

58:17

Vincent

But then you look back at the last 10 years, and you kind of go, “Gee, a bunch of things were overpromised.” And there are lots of these failure stories that people tell me when they're drunk. And maybe I should be the one that writes some of them down. Because I also think that maybe there's an overemphasis on tools, and not enough about culture, and “How do humans work? How can we actually prevent failures?” I think just having a little bit of a book that just has a couple of these anecdotes and stories, are going to do a bunch of good. That's a project that is definitely going to happen. But I should also say, the way that my side projects work these days is – I do a live stream over at :probabl.

Basically, every week, I just pick a new technology where I could do something interesting. In two weeks’ time, I'm going to experiment with gradient boosted machines. You know that tree based models – you can stack them together? A tree based model could be turned into a SQL query. A whole stack of them can be turned into a SQL query. Yeah, the SQL query will be big, but these boosted machines are some of the most performant models. So being able to SQL-ize that will be cool. And I wonder – for large batch jobs, is this quicker?

59:31

Alexey

**Quicker then loading the data, turning in into the… [cross-talk]**

59:36

Vincent

It’s a Python process. Maybe this can be handled better inside of a database? Or a query engine that’s optimized for this? Sending it to a Python process feels expensive.

59:44

Alexey

**Or data warehouses, like for BigQuery, for example, or other alternatives – you pay for the data scanned, right?**

59:51

Vincent

Yeah, well, not… You have to move it into Python and back out. There's all sorts of stuff there. And I have no idea if this will work. But that's what the live stream is for. [chuckles]

60:03

Alexey

**So you're going to figure this out on the live stream?**

60:06

Vincent

The way it works – every Wednesday, I have an afternoon where I prepare the live stream. And then during the live stream, part of it is live coding, but part of it is just “Okay, here's some stuff I've seen – some lessons I've learned just from doing this.” Then the conversation starts with people and then either I pick it up again next week or I pick a different topic.

60:24

Alexey

**So you come prepared. It's not like complete exploration.**

60:27

Vincent

No, I think it's reasonable… If people are going to spend their lunch watching me, I do think it's fair that I do prepare at least something.

60:32

Alexey

**Yeah, yeah. Yeah, I was curious how exactly it works. I would be nervous to be unprepared and do a stream.**

60:40

Vincent

I've also done a few streams unprepared but it was about Embetter – a library that I wrote. So I can do that. Yeah. That's an easy demo. But also, another thing, I do want to do my homework because… Let's say, last week, I did something on IBus and… You know, I do want to give a fair representation of what IBus can or can't do. I shouldn't say nonsense. Also, because I'm sometimes demoing other projects that aren't from within my company, I also think it's fair that I at least do some of my homework – and also post an Issue on GitHub and get their feedback and that sort of thing.

61:15

Alexey

**Makes sense. Okay, I think that's all we have time for today. It was amazing having you again. Actually, this is our second interview with you, but you also did the demo as you mentioned.**

61:29

Vincent

Doubtlab and Embetter, I guess?

61:30

Alexey

**Yeah, yeah. I'm not sure about SciKit Lego. But Embetter, certainly. Okay. So thanks, again, for joining us today – for sharing your experience with us, for sharing your opinion on things, for talking about your future plans. And thanks, everyone, too, for joining us today and listening in. Yeah, that was fun.**

61:53

Vincent

Have a good one!

61:55

Alexey

**Yeah, you too. Have a great week!**