SDS PODCAST EPISODE 862: IN CASE YOU MISSED IT IN JANUARY 2025

Jon Krohn: 00:06 This is episode number 862 our In Case You Missed It in January episode.

00:19 Happy Valentine's day listeners and welcome back to the SuperDataScience Podcast I'm your amiable host John Krohn. This is an "In Case You Missed It" episode that highlights the best parts of conversations we had on the show over the past month. With 2025, a new year starting, in January, my conversations on the show were focused on glimpsing into what the years ahead may bring. Developments in AI continue to rock it forward, but is all that power sustainable?

00:45 In episode 855, I ask Azeem Azhar, the famed futurist, how exponential changes in tech could radically impact our future.

00:53 Following on from this idea of exponential growth humans seem to be, and maybe turkeys as well, seem to be poor at being able to imagine that they're on this exponential curve and so, Ray Kurzweil, for example another famous futurist, said that our intuition about the future is linear, but the reality of IT, as we've already been discussing in this episode, Azeem, is exponential. You similarly in your book, you talked about, in chapter three, how, for example, the COVID pandemic, when that was unfurling in 2020 around the world, it was experiencing exponential growth. I experienced that in real time, looking at... I was like many times every day, probably a hundred times a day, refreshing how much in New York state, how many more infections there were.

01:51 It was very difficult for me even as somebody with a lot of statistical background, been a data scientist for a decade. Even for me, it was difficult to process how this exponential change was happening. Given the difficulties that even experts face in predicting exponential growth or being able to have intuitions about exponential growth,

how can businesses, policymakers, our listeners better prepare for future technological shifts?

Azeem Azhar: 02:23 I agree, it's really difficult to normalize and rationalize in your head the speed of that change. I do think that it's quite commonplace. A very simple exponential process is compound interest. Virtually, all of us start saving for our pensions or 401(k)s or whatever it happens to be too late. The right time to start is when you are 23 and you just put 10 bucks a month away knowing it's going to compound. I think many of us are guilty of that. I am as well.

02:57 I think there are companies who have internalized this possibility and I think the technology industry as it comes out of the Bay Area has very much done that. They have relied on understanding that Moore's law keeps driving prices down and that you aren't really going to systemically run out of capacity or compute. You may have crunch periods where you can't onboard the machines or the hard drives or the storage fast enough, but in general you won't do that.

03:35 I mean, I think that one of the ways that you have to understand this is understand the processes and understand that these processes absolutely exist. I think it's really unhelpful for when you're trying to make sense of this world for people to think in linear terms. I still see it and I'm sure you may see it when you're helping clients or people at work and you see their business plan and it shows a fixed increment of growth and nothing grows that way. Everything follows a phase of a logistic S-curve where you have an exponential phase that tails off. Nothing is linear except for our birthdays, one to two to three to four.

04:16 I think a lot of the tools are all to hand but it is very difficult. And what you need to do at these moments is

perhaps go back to first principles thinking and perhaps say, "Look, the heuristics we've used were just that they were really helpful in a world that doesn't move as quickly, but in a world that moves this quickly we have to go back to heuristics." Sorry, pardon me, first principle thinking.

04:43 The thing that's so funny, Jon, is that most people who are listening to this podcast will have... Beyond their experience with COVID, they will have lived through exponential technologies because they will have lived through upgrading their iPhone or their Android phone every two years and getting twice as much compute for the dollar they spend. They will have lived through, if their data scientists, their data array or their data lake going from a gigabyte to 100 gigabytes to 10 terabytes to a petabyte and beyond. They've literally witnessed it and yet it still becomes quite difficult. I think going back to first principles is a really helpful way of doing that.

Jon Krohn: 05:26 Yeah, yeah, yeah. And so... In terms of something that people could be doing, this idea of first principles in this instance here, that's literally thinking about sketching for yourself those kinds of changes and thinking about how you adapt it to those changes and making projections based on that.

Azeem Azhar: 05:45 Yeah, I think that's a really good way of doing it. I mean, when I do my own planning and build models of where the business might go or where usage might go, and I've done this for more than 20 years, I've never put in linear increases like, "Oh, it'll go up by 20. It'll go up by 20." I've always gone in and put in a dynamic percentage because a percentage compounds. One of the things that drives the exponentials is feedback loops.

06:20 The reason something accelerates... I mean, let's think about silicon chips. Why did chips during the '80s and

the '90s and the 2000s get better and faster? It was because there was a feedback loop. When Intel came out with a new chip, it allowed Microsoft to deliver better tooling on Windows, which gave people an incentive to upgrade their computers, which put money in the system, which allowed Intel to develop a new chip, which allowed Microsoft to push out more features. That feedback loop accelerates.

06:54 Sometimes when I do my planning, I will also try to put those types of feedback loops in, because an outcome of a feedback loop will often be a curve that ultimately has that quality of taking off. In a lot of places, you end up with these linear forecasts and if you're sitting there and you're thinking, "Listen, I need to put in my budget request for next year for storage on S3 and I also need to give some indication of what's going to happen the year after and the year after that and the year after that." If it's growing linearly, I think you're making incredibly extreme assumptions based on what evidence has shown us. So you have to go back and start to say, "How do I put in more realistic assumptions even if it's going to freak the CFO out?" Because that's what history has shown us.

Jon Krohn: 07:44 As Azeem says, it's super important to recognize change so that we can adapt to it. Encouraging feedback loops is one great way to make sure your models and algorithms are moving in the right direction. Now, quantum machine learning is opening up even more ways to solve computationally challenging problems and model the world. In episode 851, I talk to doctor Florian Neukart about how quantum computing can keep pushing the envelope.

Jon Krohn: 08:08 Now we have a bit of a sense of the theory and the special things that you can do with quantum computing. Can you provide an example of a practical, maybe optimization problem? That seems like the kind of thing that you guys

do at Terra Quantum a bit. Some kind of practical problem that is intractable for classical computers, but with some quantum computing as well. It sounds like typically a hybrid system. How we can have a real-world application that provides some value.

Florian Neukart: 08:38 Yes, there are so many. There's three branches that we look into. Everyone who does quantum computing does is machine learning and as you said, optimization and then simulation. One problem in optimization that sounds boring at first is scheduling. That is impossible to tackle with no matter how powerful a classical computer you have. The challenge is manifold. Scheduling appears in production. Scheduling appears in hospitals when you have to do plans for the nurses and the doctors. Scheduling appears in computers in electric vehicles when you want to optimize the subroutines for power consumption. One of the things that we did with an automotive company, with Volkswagen, in that case, was a scheduling problem for production. Imagine you have vehicles coming out of the production line, then all of these vehicles must undergo a couple of tests. Ideally, I can test every vehicle for everything, but the reality is you don't have enough time, you don't have enough people and not all of these people doing vehicle tests have the same skills.

Jon Krohn: 09:54 Especially if it's emissions testing. Then you've really got to skip a few cars.

Florian Neukart: 09:58 Yeah, that one. Some of these tests, of course, you can plan because you get reports, field errors, the workshops will report, "Well, I have these couple of customers complaining about water damage." So anytime it rains, get wet inside the vehicle. Then you do water tests. But then there are 250 something test classes and each of these test classes has subtests. The question now is given the staff, the personnel in production with the skills

available today, how can I maximize the number of tests for all of these vehicles that come out? And that is a very complex scheduling problem. But the same algorithm can be applied, as I said before, for scheduling subroutines in electric vehicles. You want to minimize power consumption, then maybe you have two subroutines that use the same data. So instead of loading it into a memory, deleting it and loading it again, maybe I can execute the subroutines in sequence and access the data in sequence before I delete it.

11:07 These are things where this can be applied, which does sound very exciting at the beginning, and you would wonder, "If there's really something where I would need quantum computing?" But you do, because in the end, with classical non-quantum algorithms, the only thing you can do is heuristics and make approximations. You can never be sure is this really the best solution I can find? I must admit, also with a quantum computer you cannot be sure, but what you can do then is you just compare the classical and the quantum algorithm. And if the quantum algorithm gives me a better solution, then that's the one that I take. Other problems are in logistics. We did many logistics optimization problems. Imagine you have a fleet of vehicles that have to transport goods through a network of hubs. For example food, which can decay, you have to have that vehicle number one at a certain hub between 1:00 and 3:00 PM, otherwise there is a problem with the food, for example.

12:08 How do you optimize the number of vehicles that I have in my transportation fleet? Minimize the number of vehicles that I need to transport all the goods efficiently through the network. Or in other ways, how do I reduce the empty miles? The empty miles meaning I have trucks that just go from A to B but don't have any load. How do I avoid that? This is also one of the things, one of the problems that we have solved with a customer. Then it ranges from

optimization of satellite constellations, which we did. Financial optimization, you want to predict market behavior. You want to do collateral optimization. You want to do exotic options pricing. You want to do machine learning. You want to do better image classification. All of these things benefit from hybrid quantum computing.

Jon Krohn: 12:59 From super cool, but still relatively niche today, quantum computing, we now turn to a challenge that lots of the show's listeners are facing immediately and on a daily basis because of all the new foundation models, such as large language models, that are being released every day. In episode 853, I sat down with my SuperDataScience colleagues Kirill Eremenko and Hadelin de Ponteves to run through a checklist they devised to help business owners choose the perfect AI models for their needs.

13:26 Earlier I talked about how large language models are a subset of all the foundation models out there. So it sounds like for that kind of medical application, unless it also needs to have vision to be able to read cancer scans, but let's just assume that it sounded like that initial application was just going to be natural language in and out of the foundation model. So in that case, we could be like, "Okay, I can use a language model." How do you choose... So maybe it's kind of vaguely you're within the space of all the possible foundation models you could select. There might be some kind of things like that where you can say, "Okay, if I want text in and text out, I want an LLM." But more specifically, how do you choose from all of the available foundation models out there? So, within the category of LLMs, there's thousands of possible options out there. How do you pick the right one for your application?

Kirill Eremenko: 14:15 Absolutely. You're right, Jon. So interesting how we're so spoiled for choice now, even though two and a half years ago there was no such thing, right? Even two years ago

there was no such thing as only just starting foundation models and LLMs and so on. Now there's thousands, as you said. Well, there's a lot of factors and we're going to highlight 12. You don't need to remember them off by heart, but see which ones you relate to as a listener, which ones you relate to the most, which ones will be most important for your business.

14:46 So, first factor that you probably need to think about is cost, because there is a cost associated with using these models and they have different pricing. So, you want to look at that as a starting point. Then there's a modality, which, Jon, you alluded to, what kind of data are we talking about? Are we talking about text data, video data, image data, and so on? So what outputs, what inputs do you want? What outputs do you want? Things like that. So, different models are designed for different things. You need to check that one off right away as well.

15:19 Customization options. So, we'll talk about customization further down in this session. You need to be, once you're aware of the customization options, once we've talked about them, you will know which ones you would need for your business, and then you would look at which one does the foundation model offer, its support. Inference options. Inference is basically once you've deployed the model, so there's training, which the first three steps, and then there's fine-tuning, which is also considered training, but then there's inference.

15:48 Once you've deployed the model, how is it used? Is it used right away? Instantly? If you're developing a gaming application, you want a foundation model to be integrated in your real-time game where users are playing with each other. For some user experience thing, you want it to be producing outputs right away. There cannot be even a second delay. So, that's one option. Then there's maybe a synchronous inference where you give the model some

data and then it gives you an answer back in five minutes, and maybe there's a batch transformation where it's done in the background later on. So, we'll talk more about that in this session as well. Basically, you need to be aware of inference options that are relevant to your use case.

16:33 Latency, generally speaking, it's kind of tied in with inference options, but basically what's the delay that the users will get and how the model responds, how quickly responds.

Jon Krohn: 16:49 With latency, if you want to be speaking in real time to the foundation model, it would need to have very low latency so that it feels like a natural conversation, for example.

Kirill Eremenko: 16:57 Yeah, yeah, exactly. That's a great example. Architecture is a bit more advanced. In some cases, you might need knowledge about the underlying architecture because that will affect how you're customizing the model or what performance you can get out of it. Usually that's a more technical consideration for more technical users. Performance benchmarks. So these models, there's lots of leaderboard scoreboards. Ed Donner was on the episode a few episodes ago and he was talking about a lot of-

Jon Krohn: 17:26 Yeah, 847.

Kirill Eremenko: 17:27 Yeah, he was talking a lot about leaderboards. What did he say? He's a leader bore. I laughed at that.

Jon Krohn: 17:34 That's right. Yeah.

Kirill Eremenko: 17:35 Yeah. So, there's lots of leaderboards and there's lots of benchmarks that these models are compared against even before you customize them. We're not talking about your evaluation of the fine-tune or customized model,

we're talking about the evaluation of that cake, that bottom layer of the cake. Even they have their own evaluations. How well do they perform on general language and general image tasks and things like that? So, you might want to consider those.

17:56 So, you might want really high performance, but that's going to cost you a lot of money. You might be okay in your use case with average performance because it's not critical, business critical or you don't need that super high level of accuracy, then you might be able to get a cheaper model because you don't require this super high accuracy.

18:19 You also need to consider language. If using a language model, what languages does it support, like human languages, the size and complexity. Also, how many parameters, small language models are becoming more popular these days? Can you use a small language model? Do you need to use a large language model? There's another consideration. It's a bit more technical as well. The ability to scale the model, that's an important consideration that probably I would imagine business users that are not technically savvy might overlook.

18:53 And that basically means, okay, you will deploy a model now and you can use it for your 10,000 users, but what if your business grows to 100,000? How are you going to scale it? Are you going to scale it by spending more money? Are you going to on the size of the underlying server or is there a way to scale it by fine-tuning it and changing the underlying architecture somehow? And that's a very technical consideration, but it can be like a bottleneck for growth for businesses.

19:21 And the final two are, last but not least, compliance and licensing agreements. Very important as well. In certain jurisdictions, there's certain compliance requirements for

data or how data is processed, or even AI, there's more and more regulations coming out around AI. And licensing, of course. These models come with licenses. How are you going to use to make sure that your is aligned with the license that you're getting from the provider? And the final consideration is environmental considerations. It might sound strange, but if you think about it, these models to pre-train them, there's a lot of computers required.

19:59 A lot of energy is used up training these models. So you might want to look into, okay, well, am I supporting an organization that is environmentally conscious? Are they using the right chips? By the way, we'll have some comments on chips later down in the course. Even inference of this model. Is this model efficient during inference? Am I going to be using a lot of electricity or not as much electricity as I could be of another model?

29:28 So, there we go. Those are the 12 considerations that... maybe not all of them are applicable in your business, your use case, but those are the main ones that businesses tend to look out for when selecting a foundation model.

Jon Krohn: 20:39 Thanks, Kirill. At the end there, you let slip again later on in this course because I think you've been recording so many courses lately. But, yeah, later in this episode, in fact, we'll be talking about chips. Yeah. So to recap those 12 criteria for foundation model selection, you had cost, modality, customization, inference options, latency architecture, performance benchmarks, language, sizing complexity, ability to scale, compliance and licensing agreements, and finally the environmental considerations at the end.

21:06 There's a ton there. Hadelin, I'd love to hear your thoughts on this, and particularly if there's some way

across all of these dimensions. I mean, where do you start? How do you start to narrow down the world? I mean, I feel like now that I know these 12 criteria for making selections, I feel like I'm even more lost in the woods than before.

Hadelin de P.: 21:30 Yeah, that's right. I was feeling the same first when I was starting building a new application of generative AI and I had to pick a foundation model. In my experience, it had a lot to do with the dataset format because different foundation models expect different dataset formats, especially when you fine tune them.

21:51 So, for example, I'll tell you about my recent experience. I did another fine-tuning experiment. I think it was on one of the Amazon Titan models. Yes. So it's one of the foundation models by Amazon, which by the way just released their brand new foundation models called Nova. So I can't wait to test them out. But yes, at the time, I chose the Amazon Titan foundation models because the dataset that I used to augment, once again, the knowledge of the foundation model was fitting perfectly to the Amazon Titan model.

22:32 So, I chose this one. It could have been a different one if it was a different dataset format. But, yes, it really depends on the experiment that you're working on, it depends on the goal. So that's kind of an extra criteria that you need to consider, take into account. And when I created this chatbot doctor, this time, yes, as I said before, it was a Llama model and I chose this one once again for a format concern. So, yeah, in my experience on a practical experience, it will have a lot to do with the dataset that you're using to augment the knowledge with to do fine-tuning or even RAG, which we'll talk about later in this episode.

Jon Krohn:	23:11	And this will sound like I am giving you guys a boost, and I am giving you guys a boost, but I'm not doing it just because of this. But this kind of difficult decision, trying to figure out what kind of foundation model you should be using, making that selection effectively could depend a lot on people like you, the two of you, who are staying abreast of all the latest in foundation models. And so, it's the perfect kind of opportunity to be working with your new company, with BravoTech, to be able to, that three hours, for example, that you were offering up front at the top of the episode, a lot of that could be spent on just figuring out what kind of foundation model to be using for this particular use case.
Hadelin de P.:	23:51	Definitely.
Kirill Eremenko:	23:53	Fantastic. Yeah. Thanks, Jon.
Jon Krohn:	23:54	Choosing the right model is at the core of how to solve a problem with AI. And in episode 857, engineer and entrepreneur Brooke Hopkins walked me through how her company Coval helps users to reach these decisions. The Coval platform simulations and evaluates voice and chat agents and in this clip we talk about the custom metrics we can apply to get a full picture of how our systems behave.
	24:17	Something else that Coval offers is custom metrics. So there could be complex scenarios where standard metrics, just plain old accuracy aren't useful. I mean actually that would be something. In a scenario where this isn't like a math test, scoring a conversation isn't like a math test where there's a correct answer, you just get to some integer or some float and you're like, "Okay, that is the correct answer, nice work, algorithm." When you have an agent handling a complex task, there's an effectively infinite amount of variability where there's an infinite number of ways that it could be right not even including

the infinite number of ways that it could also be wrong. So what kinds of metrics do you use to evaluate whether an agent is performing correctly and then maybe building on that, what kinds of custom metrics might your clients need?

Brooke Hopkins: 25:21 I think you're exactly right that it's really hard to find the line between this is objectively a good conversation and this is objectively a failing conversation, but rather it's a spectrum. And so what we find works really well is layering metrics. So being able to run a whole suite of metrics and then looking at trends within those metrics. This allows you to make trade-offs as well. So maybe you're a little bit worse at instruction following, but you get the cases that you care about most a hundred percent correct. Because the distribution of how well you do on all these cases isn't like machine learning where you just care about getting 99% of examples right. Because if you're getting the one most oftenly used case wrong, it doesn't matter if you get the other 99% right, because when someone tries to book an appointment, they fail.

26:09 And so we see that these patterns of what matters is correct is different than other traditional software applications or machine learning applications or even robotics. The other piece of this is being able to show by having a variety of metrics, you can create a whole picture of how the system is behaving. So for example, a short conversation isn't inherently bad, but a short conversation where the goal wasn't achieved and the steps that the agent was supposed to take were not executed, that's an objectively bad conversation. So you can filter down by whether potentially true failures or false positives or false failures, et cetera. You can basically figure out which ones are ones worth looking into through filtering by these metrics.

	27:02	So I think while we aim to provide all automated metrics for things like did it follow the workflow, was the conversation successfully completed? Were all the right function called with the right arguments? There's also always going to be space, I think for human review and really diving into those examples. And the question is how can you use that time most effectively? So it's not that you never look at all these examples, but you're looking at the most interesting examples.
Jon Krohn:	27:28	Nice. Very cool. That's a great example of what to prioritize. Are you able to give concrete examples of metrics? What are the most common metrics for evaluating performance?
Brooke Hopkins:	27:40	Yeah, so we have a metric that allows you to determine if you're following a workflow. So for a given workflow described in JSON, which is pretty common in a lot of these different voice platforms, can you determine if you're following these steps outlined in that workflow and determine when you're not meeting those in the conversation. This is super useful I think especially for objective-oriented agents where they're trying to complete a task. Often if they miss a step in that workflow, it's a really good indicator that the task wasn't completed correctly. So for example, if you're booking an appointment, just to use a consistent example, if you're booking the appointment and it asks for the email and the day that they want to book the appointment for, but they forget to ask for the phone number, that task has been completed technically but hasn't been completed correctly because it missed this key step in the workflow.
	28:36	Another interesting metric that we do, and then we also dynamically create these workflows in monitoring so that you can see what workflows your agents are actually going through in production and see how often if that matches with your expectations or where you're seeing

new use cases or new patterns of user behavior. We also have metrics around function calling, so where the right arguments called for these different tool calls, and that's all custom configurable. What's interesting here is I think we try to make all of our metrics reference free. So there's two types of metrics. There's reference-based and reference-free. Reference-based is metrics where you have an expected output and you must curate that expected output with a golden data set and maintain that as your agent behavior changes. Reference-free, we infer what the correct answer should be based on the context of the conversation.

29:39 I think for LLMs in general, reference-free evaluation is really helpful because of the non-deterministic nature, whereas traditional unit testing and software is all reference-based, right? It's easy to make some assertions about what an API call should look like, but even more so with voice and chat agents, the conversations can go so many different ways. And this changes when you change your prompt, when you change the models, when you change your infrastructure. So having reference-free metrics, or at least a strong subset and test sets that rely on those is really important for being able to iterate really quickly. So we try to do function calling, create a reference-free evaluation for function calling. So we say, for example, if we're taking the order, can we confirm that the right function call was made based on what was described in the order from the user? It should match. Those two things should match based on a prompt and inside of heuristics. So this gives you, the users a lot more flexibility.

30:44 Those are just two examples we actually will build. We've been building out a lot of metrics for new use cases and kind of pulling them from all over the map of using off-the-shelf models, drawing from inspiration in self-driving of can we measure, for example, the agent

performance against the human performance? If it took the agent longer to perform a task or shorter to perform a task, that's interesting intel. It's not necessarily good or bad when it stands alone, but if the agent takes significantly longer to perform a task and then ultimately doesn't or is repeating itself a lot, it's a good indication that your agent is going in circles.

Jon Krohn: 31:27 All right, that's it for today's In Case You Missed It episode. To be sure not to miss any of our exciting upcoming episodes, subscribe to this podcast if you haven't already, but most importantly, just keep on listening. Until next time, keep on rockin' it out there, and I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.