SDS PODCAST

EPISODE 1024: IN CASE YOU MISSED IT IN AUGUST 2026

Jon Krohn: 00:00 This is episode number 1024, our In Case You Missed It in August episode. Welcome back to the Super Data Science Podcast. I'm your host, Jon 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. My first clip is from episode number 1017 where I speak with MongoDB's field CTO of AI, Pete Johnson. Here's the key context. Four out of five organizations have AI steering committees and success metrics in place, and yet only one in five is seeing a return that matches. Pete has an unusually wide view of why, having made 19 stops across six countries this year, and he traces his answer back to an executive dinner in New York where every company seeing real ROI return on investment shared two common traits. In a recent article, you highlight that 83% of organizations have AI steering committees and success metrics, yet only 19%.

01:06 So roughly four out of five organizations, and we're probably talking bigger enterprises in this case, but four out of five of them have AI steering committees, success metrics, so they're trying to do something with AI, but only one in five, 19% see matching ROI. So there's this gap where three out of the five organizations seem to have organizational structures set up to succeed with AI, yet they aren't. So how does that AI ROI gap happen?

Pete Johnson: 01:36 It comes from a couple different places. What I take this back to is last fall, the mainstream media published a series of articles that asked the very fair question, where's the ROI for AI? Collectively, as an industry, we've spent all this CapEx developing the models, where's the benefits? And I was at an executive dinner in New York City the first week of January, the NRF, the big retail conference that's there every year. And it was the first time I heard customer talk about that they had agents deployed in production and they were getting ROI out of them. And they came with two caveats. Caveat number one was they were employee facing. And caveat number

two is that they were not autonomous, but they were human in the loop. The reason for choosing employee facing use cases was twofold. Number one, the data security and data quality bar is lower than it would be for customers.

02:35 It's terrible if you accidentally leak someone's salary to another employee. It's way worse if you leak some customer's data to the wrong customer. But the other thing has to do with the ROI part of your question, which is I know how I'm judging the effectiveness of an employee. If I take a agent and I put it in their workflow and I see those metrics jump, I can attribute that jump to the agents and therefore back of the envelope compute some ROI. So that's a very long way of saying picking the right problem is key here. You got to pick a problem that you have good data for because if you put bad data into an AI ecosystem, it's not going to solve that. And number two, you have to have some metrics of success. You can't just throw AI at a problem that you don't know how difficult it is or you don't know how to measure the success of it.

03:32 And that's why these employee facing use cases were so popular in the first half of 2026 is we already know what metrics we use to bonus people and to evaluate their performance. And like I said, if you put AI in their ecosystem within their workflow and notice the change in those metrics, that gives you an ability to then measure that ROI. So more and more companies are beginning to do this, but when those statistics came out in that survey, that was the problem is people were picking the wrong problems.

Jon Krohn: 04:06 Right. So instead of having your success metrics be related to how many employees have adopted AI, that kind of thing, instead, the success metrics should be how much does an existing process get accelerated by having some automation of workflows within the loop?

Pete Johnson:	04:26	Exactly. And the best example I've seen here in the second quarter of calendar 26 has been software delivery life cycles. So at the beginning of the year in January, we might have measured the success of something like Codex or Anti-Gravity or Clawed Code by the lines of code. How much code are you producing? And if you've been writing software as long as I have, and even if you haven't, you probably know, lines of code is a terrible metric to determine the effectiveness of an individual or in this case of some AI. But the maturity that I've seen really in the last eight to 12 weeks, really since the Uber Token Maxing story came out, was am I shipping code faster? That it's not just about the effectiveness of writing code or the effectiveness of an individual engineer, but the broader business metric is, have I increased the speed it takes me to go from idea or bug report to production solution getting deployed?
	05:29	When you measure it in terms of those business metrics, that's really where you see the improvement in overall ROI value.
Jon Krohn:	05:38	Makes a lot of sense. And yeah, the token maxing thing was so silly. I did an episode dedicated to that because I'm basically encouraging my listeners and managers not to be using lines of code or number of tokens as a way to measure productivity because it's so easy with agents to just have them spewing things out and be using tons and tons of tokens for sure. Yeah, I think we've seen, it's funny that, oh yeah, the token overrun that you're talking about, it was something like Uber's expected allotment for tokens for the entire year 2026 ended up being used in three or four months or something like that.
Pete Johnson:	06:18	13 weeks.
Jon Krohn:	06:19	13 weeks. Exactly. Yeah. Wow. Yeah. I'm glad that organizations are waking up to the dangers of having that

kind of silly metric, vanity metric to track. I think Meta had a similar kind of story.

Pete Johnson: 06:33 They did. Meta had their own token scoreboard that was measuring the number of tokens that individual engineers were consuming. And like you said, that turned out to be a bad idea. And among the things that's interesting about this AI wave of technology compared to others that I've seen in my career is how fast the cycles are. Anthropic dropped model context protocol on the Monday of Thanksgiving week in 2024, and by March, all of their competitors had embraced it. And here, I mean, Tokenmaxing really got its shining moment in March at GTC when Jensen mentioned it on stage. And by the time the Uber story came out eight weeks later, nobody was talking about Tokenmaxing anymore. So the life cycle of these stories and how quickly we move on to new things is unlike anything I've ever seen in my career.

Jon Krohn: 07:28 For sure. It seems like being in the AI space, we're going to soon have a 24-hour news cycle just for what the popular packages and approaches are in our discipline.

Pete Johnson: 07:38 We're getting there. And as an engineer, that's the hard part is how do I keep up? How do I know where I should invest my learning? And how do I know when to sort of cut bait and move on to the next thing? That's the hard part about being an engineer in this ecosystem right now.

Jon Krohn: 07:53 Pete, this isn't a question that I had planned for you, but I think it's, given what we've just been talking about, I think it might be interesting for listeners. It kind of begs this question, what you were just talking about. How do you personally, given that you always need to be right on the cutting edge of what's happening with AI, how do you stay up to date?

Pete Johnson: 08:11 Well, I'm fortunate in my job that I get to talk to so many different people. I've been calling it my AI world tour this

year because I've visited multiple cities. So I've made 19 stops on my world tour in the first six months of 2026 across six countries. I've probably talked to a hundred different companies. I get to listen. I mean, I spend a good amount of my time telling them about how MongoDB can help them, but I get to listen. What are you doing? What are you seeing? So because of the diversity of the different organizations that I get to talk to, I get to hear not just what one person thinks, I get to hear what a couple dozen people think and use that as a guide for where I go spend my time. And it's things like I think we'll get into a little bit later, things like better agentic memory, things like maybe not always calling the generative LLM and being a little bit more creative about how you use rag pipelines and why retrieval quality is important.

09:15 I get to talk to people about that kind of stuff and let that be the guide for where I spend my time.

Jon Krohn: 09:20 Pete's answer is about picking problems you can measure. My next clip asks a harder question, which decisions should you be handing to a language model in the first place? In episode number 1015, Jerry Yurchisin, manager of decision intelligence strategy at Gurobi Optimization, makes the case for mathematical optimization in the agentic era. An agent will tell you with confidence that it has optimized your business while quietly ignoring the one constraint that could cost you millions. Jerry explains what optimization does differently and where he thinks the division of labor between agents and mathematical solvers ought to fall. Speaking of the times changing rapidly, since you were last on the show last October, we're now constantly talking about agentic AI obviously on a data science podcast that focuses on AI, which is more and more what data science is all about, I think, certainly in the way that I've been curating content on the show and I've been experiencing the world.

	10:18	And so yeah, where does mathematical optimization fit into this new agentic world that we're in?
Jerry Yurchisin:	10:26	From our perspective, that's the million dollar question, billion dollar question. Yeah,
Jon Krohn:	10:29	I bet it's more than a million.
Jerry Yurchisin:	10:32	Yeah. Million dollar question because that's the term that people used to use a lot. I don't know, but yeah, it's billion, it's massive now. But anyways, when you think about what, and I'm going to be a little bit sort of high level and not absolutely correct, but if you think about what an LLM does, an LLM takes all the input tokens and then just produces more output tokens about text or something like that, let's say if you're purely natural language type of stuff like input tokens and your output tokens, that's it. That's all it really cares about is providing the tokens that give you a really good response, a highly likely response, something that is all about just that sort of input output flow. That really doesn't jive with what I was talking about, the types of decisions that optimization can make and what it does and the rigor that it provides is it provides these.
	11:32	The constraints are what we call hard constraints. These are things that cannot be violated. It's not like, oh, my context, I mentioned my constraints just outside of a context window type of thing and now the LM's sort of forgetting this.
Jon Krohn:	11:47	Or even if it's in the context window, it's still a very frequent occurrence that some piece of information that you say. There's an example, Sinan Osdimer, do you know that guy?
Jerry Yurchisin:	11:58	No, I don't think so.
Jon Krohn:	11:59	Sinan Osdimer, I think he's been on this podcast more than anybody else and he's a crazy prolific author of data

science and AI books. I think he's younger than me. He might be in his mid - 30s and he's written at least 10 books and he's created tons of online content. Recently he started working at Fireworks AI. But the point that I'm getting to is that Sinan Ostomer, I've seen him do a talk a couple of times where he shows surprising issues with even frontier LLMs where he would do something like have a tool available for an agent to call. And in his prompt, he would say, "You must use this tool." And it was like single digit percentages, but some single digit percentage of the time that very simple, very specific instruction, the LLM controlling the agent just wouldn't do it. It wouldn't call the tool.

12:58 It would find some other way of doing the approach. And so yeah, Sinan's done lots of. He did a whole book on agentic AI where these kinds of experiments that he was running, he published them in there. While you're talking, I'll look up the name of that book.

13:16 Yeah, go ahead.

Jerry Yurchisin: 13:17 Yeah, if you think about exactly what you said right there, when it comes to, again, the fate of my business, do I want to trust decision making to something where that can happen, where it can forget a. Let's say you have some sort of environmental constraint where if you violate that, then you're going to be fined millions of dollars, something like that. And then you output a solution that is the one thing, all of these agentic tools and everything, the confidence is so high. It's like, "I got you, boss, exactly just what you're looking for. We're 100% good to go, but it misses this environmental constraint and then all of a sudden you put into production a solution that is not good and then the bad things happen and then all because of just forgetting, just doing all something that happening. And the contrast to mathematical optimization is if that is a constraint in your model saying that here's my..." Again, I'm going to do the arm thing

again for everyone listening, just purely video, "Here's my constraint and all my decisions are in here.

14:31 I cannot go past this. I cannot break this environmental constraint." You are guaranteed that. So that's what we think is a differentiator. First off, is that you have the trust of the model to actually do what it says. And what's really nice about it is what this line represents is something that you talked about. It is a constraint that the people who are designing the problem, who are talking about it, have hopefully agreed upon as an actual constraint. So it's not just some generated business rule type or something. It is something that as if you and I were working on a problem, I'd be like, "Hey, I think this is an environmental constraint." You're like, "Yes, it is, but it should look more like this." And then we agree what that is, and then it's represented in there mathematically. So it's just a very different decision-making framework, but where I see this all fitting together is you mentioned that, "Okay, I have an agent that's going to call a tool." Okay, an agent should be able to.

15:32 Well, what an agent can do is help you develop the problem statement that you're really trying to solve, help you understand all the other bits and pieces, all the other regulations say, "Hey, I have this environmental regulation," and an LM or an agent can do like, "Hey, these are other things that you may want to consider." And you might be like, "Holy crap, yeah, I want to consider these. I forgot about them." So it can really help there. It can help you identify the problem, help you actually write the code, help you to come up with the mathematical formulation, do all of that kind of stuff, but it can't do the solving. It can't give you the optimal solution. It can't give you a solution that's close to optimal and have the defendability, the explainability, all that sort of stuff that comes with these high stakes decisions.

16:19 So how we see it as agents should be able to develop all those things and then call an optimization engine like Gurobi saying, "Here is the problem statement that we have. Here is the model they're trying to solve." And then Gurobi runs, gives you the output, gives the solution, and then you can then dive deeper into why. Why is this happening? Why did I decide to build a new production facility in Atlanta as opposed to Baltimore or something like that? Those are actual questions that you can get answers to with mathematical optimization because essentially what can happen in that situation is it'll resolve with the Baltimore production facility there and say, "This is the difference. It is the difference because the cost is going to be this much higher or you're going to have this much less demand or whatever it may be. You can actually sort of figure those things out and get to be able to answer questions that people are going to have when it comes to business problems and decision making is why this?

17:22 Why not that?" Those are all things that can happen with mathematical optimization because of the structure and the rigor that's there.

Jon Krohn: 17:30 Those first two clips came at AI adoption from the technology side. My next guest comes at it from the people side. In episode number 1019, the cloud girl, Priyanka Vergadia, bestselling author, former senior director of AI transformation at Microsoft, and head of North America developer relations at Google, walks me through how she structures Claude skills so that her output stops being slop. She then lays out the 10 / 20 / 70 framework she uses to guide AI budgets a split that will make any CFO wins. There's a specific thing now that we're back on Claude, way back in the content creation section that we were in 15, 20 minutes ago, you were talking about how a big part of your success with using GenAI tools with Claude specifically is having skills set up. And so that's something that I meant to, at that time,

we ended up going off and talking about something else, but I'd love to come back because I feel like that's something really practical and technical and useful for our listeners.

18:28 Could you explain for listeners who don't already know about skills, what those are and how they can make the most of them in Claude?

Priyanka Vergad...: 18:35 Skill is something that you would define your task to be. So break down your task into small sub-tasks and you define how you do that. When I write the blog, I do a research. This is how I would do it as a human, right? So think about this. When you're writing a skill, think about it like a human. Now, the way Claude helps you do it is amazing, but before you even get into it and start setting up a skill, think about your task explicitly as. I'll give you an example because it'll be more material that way. I'll do a research first if I'm about to write a blog. And the research prompt has to be really good as well where I want only high quality content written by researchers and scientific research from schools and universities and organizations like this, only look for that stuff around this topic and then create a report.

19:40 Let's say that is the prompt, but that becomes part of my skill as step one. Then the next part of that skill is after I do the research, so the prompt is the how, right? So I've put the how in there as well. Research is the task, how you do it is that prompt. Then the next step would be to synthesize that research. I usually do a human in the loop thing in there. I don't trust it to make decisions beyond that. So I would do a human in the loop. It sends me a text message. I've got all this set up in Hermes. So it sends me a text message saying, "I've done the research, here's the doc." I like to read because I'm trying to do that intentionally so that I don't lose the skill with AI. I've also done cases where it would send the audio to me and I can

just hear what the research was while I'm running or while I'm on a workout, which is super handy.

20:42 And then I would give it instructions on, "Okay, I want to change this or that." And then the next step is kickoff writing the first outline.

20:55 Most people would just go in, "Write a blog on this topic. You're going to get slop." The whole idea here is how would you approach a blog? I approach with research, then I would go in and do my analogy and storytelling on top of it. That's my next step. Then I would synthesize after the synthesis, the storytelling, and then putting it into a format that I usually used to write blogs when I was doing it all on my own, which is I need to have three images in this blog and they need to be developer focused, which usually talk about the flow of movement of tokens or query. And I have some of these examples in there, and I need to have one practical example in there and that can come from. There's prompts in there where Claude would ask me of a practical example from my experiences, and so that it can take those and do that.

21:54 So I'm going into too much detail, but the idea of a skill is how do you do the task, define it into sub-task. Those are your bullets that go into the skill, and also some example prompts that go in there. That way you will get a much more personalized, the type of outcome you would. It's never perfect, but at least 80% there and now you can start editing it from there.

Jon Krohn: 22:22 Perfect. Yeah, that did have a lot of examples, a lot of detail, but hopefully it helps us understand just like a lot of your cartoons, your illustrations go into practical examples. And so we got lots of examples there of how to build effective skills in Claude. So thank you for indulging us with that. Back to the enterprise stuff with your Google Cloud experience, your GitHub Copilot experience at Microsoft. With your work bridging the gap between

high level boardroom strategy and real world enterprise AI execution, you have something that you call the 10 - 20-70 framework to help guide budgets for AI success. Could you tell us about that 10 - 20-70 framework?

Priyanka Vergad...: 23:07 So 10% on tools, 20% on execution with those tools, and 70% on education and skilling and upskilling. And I know this sounds crazy, but I have work with enterprises that have large number of large teams and have bought the tools and don't see ROI. This is exactly the Stanford report that just came out, the impact report is a great example. It has eight or 9% of the actual AI use cases in production, in real production use cases are about 8% to 9%. Everything else is just like experimentation, and this is exactly the reason. You're not going to see ROI, you're not going to see real use cases that are leading to revenue or cost reduction or savings because you've bought the tools. AI is a habit and habits don't form in days. They form in an extended period of time. So this whole concept of token maxing and all of these things are just natural evolutions as well.

24:28 Yes, the idea of token maxing is super weird because we went from, okay, we've got a tool, now the AI officer in the company is like, "Nobody's using this. We got to make them use this." So you get into this whole problem of now everybody's using it for writing emails or the dumbest tasks, right? And now you're in this token maxing situation, which you never thought of where it's like, "Oh my God, now we are spending so much money on this tool and we are not seeing ROI." So you got them to use it, not effectively, and you're in a different problem. But I think it's all a good problem because you at least got them to touch it. When you look at 10, 20, 70 rule, if you spend that 70% of your budget and time on upskilling your employees, which means showing them effective ways of using the tool, not just telling them use it, showing them effective ways of using the tool, not just giving them training, but actually giving them real use

cases of the thing they can do in their job, and that requires time and effort and energy.

25:43 My DevRel hat on, that requires building a community and saying, "I tried this thing today. Let me share it with you all." If you are a testing team, if you're a coding team, if you're a team that's product managers, got to share those experiences with each other and you have to bring space for that as a leadership team to allow people to build and form these communities. This is a big J curve, and after six, eight months, you start to see effective use of tools actually helping them be productive. And then you get to a point where it's like, "Now we can write test cases with this. That's looking really good. How do I write them faster, improve my prompts a little bit more? How do I take an entire process and make that an agent?" Now you have agents in each of the different business units and you can form that into a repository of agents and now an entire company is becoming efficient.

26:51 But this is a trajectory and the curve, you have to see the vision for a year or so and pour into it, which is why I say if you spend 70% on some of this stuff, which is going to be very costly and hard for a CFO to agree to, but that's the only way to build a habit and an effective habit.

Jon Krohn: 27:14 We're rounding up a great month with episode number 1021 in which DBT Labs founder and CEO Tristan Handy explains why the semantic layer matters more, not less, now that analytics agents are the ones asking the questions. You mentioned there in your most recent response how DBT adds meaning. So the Fivetran is like pipes and DBT adds meaning. There's a term that came up a lot in our research for DBT labs, which is semantic layer. Do you want to explain how DBT acts as a semantic layer for your data?

Tristan Handy: 27:47 This is a topic that is particularly hot right now as analytics agents are very in view. The problem that the

semantic layer solves is not a problem for small organizations. So if you imagine that you're a part of a whatever, a 20 person company, a 50 person company, you probably don't need a semantic layer. But now imagine that you are Siemens, you're a global company, you have 2,000 data engineers that are serving 300,000 employees globally. It is not possible to just know the answer to random questions that you might need to know the answer to without getting meetings together of people that you search for in your Outlook phone book. You get everyone together in a room and you say, "How should we be measuring this thing?" And there's a lot of conversation and everybody's got to figure it out and which table should we be using and all this stuff.

28:55 And literally that's how big companies have for the past, whatever, 30 years, tried to answer questions like this, that's the process. And the semantic layer is tooling that allows those types of decisions to be made and then stored so that successive people when they ask those questions can confidently measure things in the same way twice and they don't have to reconvene the whole group. It is a technically complex problem area, but the problem that it solves is really an organizational problem. It is how do you scale knowledge to increasingly large groups of people? And that is honestly a tale as old as civilization. I mean, we don't have to go too deep on this, but as long as people have been organizing together to figure out how to do stuff, there's been this question of, well, how do we make sure that we know things?

30:00 Everybody in this organization knows a consistent set of things. And so the semantic layer is an attempt to do that. And it's particularly relevant today because absent these types of cues, how do you measure X thing? AI agents have to try to re-derive that for themselves at every turn. And oftentimes they do one of two things, or almost always they do one of two things. One is that they, in re-deriving how do you measure something, they just take

a tremendous number of tokens to do that. They just have to look into a lot of stuff and think a lot and that becomes slow and expensive. And then the other outcome is that they just get it wrong or they come up with an answer that may be kind of reasonable, but it's actually not the way that your organization measures these things. So the semantic layer extends very, very nicely into a world of agents.

Jon Krohn: 30:54 Yeah, really cool. It seems like one of the key use cases, and if people aren't aware, that word semantic basically just means understanding, just means the underlying meaning of some data. And so by having this common playing ground or this common lingua franca, this common agreement on what meaning is across data sets, across agents, there's efficiencies, especially across the large organizations that you were describing. They're like 200,000 person companies with 2,000 data engineers, that kind of thing.

Tristan Handy: 31:29 Yes, totally. And there are these successive layers of meaning where you start off with raw data, you go to modeled data, then you go to semantic layers, which are typically, they help you understand how to join these tables together and how to measure certain metrics. But then you can even go one step further, and this is not an area of particular expertise for me, but it's a very interesting conversation happening in the industry, is ontologies. So ontologies are another layer of meaning making on top of data that are not just how do you measure a thing, which is typically how we think about the semantic layer, but it's also how do you model business processes? How do you model causation? And I think that most of us do not operate in an environment where it's appropriate to say we need ontologies for this because the world changes quickly and sometimes it's hard to keep up.

	32:36	But in very specific high value domains, think like genetic research or famously ontologies are employed a lot in defense. So I think about these four layers as kind of like the knowledge or meaning hierarchy.
Jon Krohn:	32:53	All right, that's it for today's in case you missed it episode, be sure not to miss any of our exciting upcoming episodes. Subscribe to this podcast if you haven't already. But most importantly, I hope you'll just keep on listening. Until next time, keep on rocking it out there and I'm looking forward to enjoying another round of the SuperDataScience podcast with you very soon.