SDS PODCAST EPISODE 787: MLOPS: THE JOB AND THE KEY TOOLS, WITH DEMETRIOS BRINKMANN

Jon:	00:00	This is episode number 787 with Demetrios Brinkmann, CEO of the MLOps Community. Today's episode is brought to you by AWS Cloud Computing Services.
	00:14	Welcome to the Super Data Science Podcast, the most listened to podcast in the data science industry. Each week we bring you inspiring people and ideas to help you build a successful career in data science. I'm your host, Jon Krohn. Thanks for joining me today. And now let's make the complex simple.
	00:45	Welcome back to the Super Data Science Podcast. Today's guest, Demetrios Brinkmann is here today to deepen our understanding of all things MLOps, or to say it, expand it out, and nobody ever does, Machine Learning Operations. Demetrios is founder and CEO of thee MLOps Community, an organization dedicated to supporting MLOps professionals that has quickly grown to over 20,000 members. He was previously founder of the data on Kubernetes community and before that worked in public-facing roles at a number of European tech startups. Today's episode will be of interest to anyone who's keen to better understand the critical function of MLOps in bringing machine learning models to the real world. In today's episode, Demetrios details what exactly MLOps is and how it relates to other jobs like LLMOps, DevOps and AI engineer, the key MLOps tools and approaches and what it takes to build a thriving community of tens of thousands of professionals in just a few years. All right, you ready for this informative episode? Let's go.
	01:50	Demetrios, welcome to the Super Data Science Podcast. I am stoked to have you here on the show. Where are you calling in from today?
Demetrios:	01:58	Right outside of Frankfurt, Germany, and I've got to say this is a monumental podcast. I just heard that we are on

episode number 787, so congrats to you for making it this far.

Jon: 02:11 Thank you. Yeah, super grateful to obviously listeners, which are critical to enabling us to have the show. So thanks to everyone for listening. And maybe even more important though, I think these are all kind of necessary conditions is the team that works on the show is unbelievable. So we do two episodes a week every single week of the year. And so basically no one takes time off to make this show happen all year round. And so I'm so grateful to everyone and it's also amazing everyone who works on the show, we can trust them a hundred percent to execute on their piece to the highest level of professionalism. We don't need to be double checking things. And so it just makes everything for me, people are like, "How do you do this on top of your day job?" And I'm like, "Actually, I'm the fifth or sixth most important person working on the show."

Demetrios: 03:13 That's incredible, man. And that's so cool that you've got this well oiled system.

Jon: 03:18 Nice. Yeah, and I mean, I also can't take credit for that. That's Kirill Eremenko who founded the show. He put the system in place and the key people and I just slot in and talk.

Demetrios: 03:28 Yep. You get to draft off of his crazy abilities to create that system. I like it.

Jon: 03:36 Exactly. Yeah. Also, you are the leader of the MLOps Community with capital letters MLOps Community. It's not like a MLOps community, it's the MLOps Community, capital C. So you are the person to ask this question, what is exactly MLOps?

Demetrios:	03:56	Ooh, I thought you were going to go down the rabbit hole of what's a community, which is another fun one we could do also. And I think the best way that I've heard MLOps described is when you're, so basically you can put a machine learning model into production, but then when you want to put N plus 1 models into production, that is MLOps. And the whole ecosystem, processes everything around getting a sustainable and repeatable process to put models into production and then making sure that they're doing what they're supposed to be doing, that's MLOps.
Jon:	04:39	Nice. That was a really great concise description. I feel like when people ask questions like that, you often get really long answers and that was just crisp and perfect. I love it. There's a term that you and I were discussing just before we started recording, which I thought was cool and new and growing and you think actually was kind of a blip and is no longer cool, nobody's using it, which is LLMOps, so large language model ops, which was I thought kind of a specialized MLOps person. So you described MLOps there as being someone who puts models into production in a repeatable way, getting model weights updated in production, ensuring the models aren't drifting away from their original purpose. The data aren't drifting. And so LLMOps, I thought involved extra specialization in order to be able to handle the typically billions, sometimes now even trillions of model parameters that we have when we're training and deploying a large language model. So it seemed like there was kind of this extra specialization there in LLMOps, but yeah, you say that it's not really a term that's taking off anymore.
Demetrios:	05:51	Well, so let me be clear on the idea. I think there definitely is something, and there's a difference between working and doing platform work with LLMs and making sure that LLMs, when you're incorporating quote unquote

"AI" into your product, that it is doing all these things like we just said, it's doing what it's supposed to be doing. And it is different than if you're just using a linear regression model. What I think is, I haven't seen that much anymore. It was very, very hot back in the day was the term LLMOps, and this is where people can say, and maybe rightly so. "Well of course he's just talking his book." I've got the MLOps Community, I don't have the LLMOps community. That may be true, but I will say that some of the funniest thing, just a little sidebar, is that one of the people in the community, in the MLOps community went and got the website domain llmops.community and just redirected it to the MLOps community homepage, which I thought was hilarious.

Jon: 07:02 And that makes perfect sense to me because it seems to me like LLMOps, it's just a subspecialization of MLOps anyway, and it would be weird to. Yeah.

Demetrios: 07:12 What I think is happening is more of the term and more of what you're seeing is who is actually doing the work and what are they calling themselves. And this has been, you've known this for years, a data scientist at a FinTech company versus a data scientist at a bank versus a data scientist at a five person startup. The job description and what they're expected to be doing is universes apart. And so now you have that same thing happening. It's just with the new term AI engineer, and we've had all these different terms come up over the past four or five years, whether it's a data scientist or a full stack data scientist or a ML engineer or an applied AI scientist, or I've even heard deep learning engineer all of these different terms. And now it seems like there's a lot of consolidation or people are agreeing on the idea of an AI engineer.

08:23 And an AI engineer I think is starting to be like, well, is it someone that just prompts a model and then can add some front end to that? That person can call themselves

an AI engineer, or is it a platform engineer that typically is looked at as someone that's doing MLOps work and they're putting not only linear regression models into production, they're also able to take a Mistral or a Llama and then add that to the platform. And so there's a huge spectrum of what it is. And coming back to the question of like, oh, is LLMOps a thing? I just haven't heard it that much as before. And especially a year ago, it was very hot. It was so hot that I was like, do I need to rebrand? What's going on here? And now I'm not too worried about the whole rebrand thing.

Jon: 09:17 Nice. Wow, that's wild. Yeah, so I guess I'm living in the past, a year ago, back in the day, as you said. So this is an interesting thing because you also highlighted, I thought that LLMOps would clearly be a subset of MLOps, which it sounds like it is. And to me, some of the other terms that you were describing, like ML engineer, AI engineer, deep learning engineer, before you said it, I kind of thought of those things as a separate role, but it sounds like there might be a lot of overlap or in fact, they are the same thing after all.

Demetrios: 09:55 I think they're separate roles and it's just that it's really hard to be clear on what exactly you are looking for and what you're trying to get. And the way I should probably say that it's when people are putting jobs up and putting in the job description what they expect from this role, that's where you're getting all these different job descriptions and you're getting these different roles. And so if they're two separate jobs, I would say yeah, you're looking at a bit of a subset LLMOps can be viewed as a subset of MLOps. What happens is, and what I think is a great unlock is that you can have people that are front end engineers or full stack developers that now all of a sudden can leverage, even product managers. I've been seeing that a ton. Someone who is a product engineer, they can now leverage AI and they can use a few of these

tools like LlamaIndex or Ollama, and they can incorporate AI features into their product really quickly.

11:14 And they do a little bit of orchestration, they do a little bit of evaluation, and that is what I think could be called LLMOps. And so there's that huge unlock that has happened now because I think I saw a slide from the CEO of Vercel, what's his name? Guillermo, Guillerme or something. And he was talking about how because of this whole new paradigm shift, you now have 25 million developers that can leverage AI. And when it was just ML, you had maybe less than a million, maybe a million people that were leveraging it. So that's where the paradigm shift is, and that's why I think with LLMOps, there was a lot of attention and interest there because of the democratization.

Jon: 12:05 This is really interesting. So LLMOps, despite sounding like a subspecialization, because it is leveraging LLMs which make using code or interacting with systems suddenly now so much more intuitive for users, for developers because now you have natural language interfaces to code or machine capabilities that that greatly broadened the number of people, the number of developers that could be making impactful products with AI.

Demetrios: 12:40 Yeah, I look at the story of Philip, my friend Phillip Carter from Honeycomb, and he wrote a great blog post all about the hard part of LLMs that nobody's talking about. And basically he was saying-

Jon: 12:54 Yes, that was super popular.

Demetrios: 12:56 Yeah, it was incredible. I loved it. I think it struck a chord with a lot of people because they were feeling that pain and he just put words to the pain that everyone was feeling. And something that I noticed is that he's a

product engineer and he never messed around with ML before LLMs came out, but then all of a sudden he's like, wait, we can add this into our product, and now people can, they're an observability tool, now people can just type in natural language what they want done and we can try and make that happen. And so he added LLMs or AI capabilities into the product, him as a product engineer. And so that type of stuff doesn't really happen on the ML side, I don't think it's as common.

Jon: 13:45 This episode of Super Data Science is brought to you by AWS Trainium and Inferentia, the ideal accelerators for generative AI. AWS Trainium and Inferentia chips are purpose-built by AWS to train and deploy large-scale models. Whether you are building with large language models or latent diffusion models, you no longer have to choose between optimizing performance or lowering costs. Learn more about how you can save up to 50% on training costs and up to 40% on inference costs with these high-performance accelerators. We have all the links for getting started right away in the show notes. Awesome. Now back to our show.

14:25 Yeah, really great blog post by Phillip Carter there called All the Hard Stuff Nobody Talks About when Building Products With LLMs. We'll be sure to include a link to that blog post in the show notes, but I also did an episode of this podcast exclusively on the content from that post. So that was Super Data Science, episode number 688, Six Reasons Why Building LLM Products is Tricky. So I'm glad that you highlighted that again for us. Demetrios, that was a great dive deeper into this LLMOps, MLOps distinction, and I think it provided a lot of insight into how LLMs are changing software development and getting models deployed and product development changed in general. But I feel like the thing that I was trying to distinguish there, which I'm not sure we got much insight

into, was this difference between ML engineer and MLOps. What do you think about that one?

Demetrios: 15:25 Yeah, my fault on being muddy in my answers and going on total tangents, I digress.

Jon: 15:31 No, that was super interesting. I'm glad we got all that out of you.

Demetrios: 15:36 So I would say that is again, the job title. It's so funny that you mentioned that because an ML engineer can be someone that works on the platform and does MLOps, and it feels like that's more of what is expected from ML engineers, but it's not always. And sometimes ML engineers, you'll see a job posting for an ML engineer, but really it's a data scientist who does machine learning.

Jon: 16:08 So yeah, so the ML engineer might in a larger organization with enough people might sit between the data scientist and the MLOps person. Where, so the data scientist might be working completely offline and training a model, training the model ways offline. Then the ML engineer is tinkering with that model, maybe writing it, using a completely different library to make it perform it in production. And then the MLOps person takes that production code that the ML engineer put together and puts it into production and make sure that the way that happens is repeatable, that there's great operations around it. And a key term here, which I think to you as the leader of thee MLOps Community, there's a term that you've used a few times that I think is so obvious to you, but a bit nebulous even to me is you said, "Someone in MLOps is working on the platform." What does that mean working on the platform?

Demetrios: 17:12 That's just the velocity of being able to take an idea and put it into production. That is that platform that is there for you to go from exploration mode in a Jupyter

Notebook to battle hardened code that is being monitored and can be retrained. And when it starts to drift, it can be retrained. So that is what I consider the platform, platform engineers. ML platform engineer is another term that you'll hear thrown around. So you get a little bit of everything, I think, and that's because it is still so new that people are trying to figure out, okay, what do we need and what kind of person is going to help us get there? Let's try and create a job title for that.

Jon: 18:07 So platform is kind of like the system. It's setting up the system, setting up the operations. Cool. Cool, cool. Nice. Actually, that is a bit of mind expansion on my part there. A couple of tools that you mentioned as you were describing particularly LLMOps people are tools that I've heard of but haven't used much myself, LlamaIndex and Ollama. What are those and why should our listeners know about them?

Demetrios: 18:35 So there is what feels like a stack that's happening in the LLM space, and so Ollama lets you do a lot of large language model stuff on your own computer. And LlamaIndex is like an orc hestration tool, so it allows you to take all of your data and do stuff with that. It's on the same layer or kind of plays in the same space as what I imagine a lot of people have heard of LlamaIndex, sorry, we're talking about LlamaIndex, LangChain. And then another one that is in that space, which I think is gaining a lot of popularity right now is DSPy. And so those will help you chain prompts together and get the right prompt templates and then send it out. And then you've got the tools that will help you either host the large language model or you just hit an API that is the OpenAI API or some third party API to make your life easier.

19:47 And really it's about abstracting away all that hard engineering so that you just have to be like, all right, I'm going to write some code and I can get to product and

MVP much faster. I don't need to worry about how to set this up so that I can serve it to hundreds of thousands of requests per second. I just need to hit this API and then I'm good.

Jon: 20:13 Very cool. Love that. So yeah, so LlamaIndex, Ollama, LangChain, and I hadn't heard of DSPy yet, so it's great. I mean this is why it's so great having a show like this because we can hear from people like you who are right on the front line, and now yeah, know about these tools that make working with LLMs easier so that we don't have to worry about scaling as much. Awesome. Before we started recording, related to you being on the front lines and having a real finger on the pulse of what's going on in MLOps, you run a big MLOps survey. Do you want to tell us about some of the big insights from that or where people can access the survey, either your analysis or even the raw data?

Demetrios: 20:57 Yeah, so the survey we started when we started doing these virtual conferences and it was all around using LLMs in production. It's like I had been hearing a bunch about the advent of, all right, cool, now we've got OpenAI and GPT-3, a 3.5 is out, ChatGPT is changing the world. But I wanted to see who's actually doing stuff with this and how, because I couldn't wrap my head around it. I was trying as hard as I could and I had on one hand people saying, "Wow, AI is going to change everything. This is the invention of the internet all over again." And so I had played with it and I would ask it questions like, is the earth flat? And get answers like, yeah, it is. Some people say it is. So this was back in the early days and then I'm like, who's actually doing stuff with this in their products? How are they doing it?

21:55 And so we started having these virtual conferences to just get people that were creating products and putting out information on it. And through that, since we would get a

lot of people to these conferences, we ended up started trying to do stuff with the attention that we had on those days. So it would be like during the virtual conference we would run surveys and the first survey that we did was all about LLMs in production, and that was almost a year ago now. And it really was just trying to figure out what's going on? How are people using these models, what's the use cases, what models are you using, what's the challenging parts about it? What are you not like? What is easy, why do you like it? All that kind of stuff and the questions. And we made sure to open source all the data.

22:45 So I think there was 200, 250 people that filled out the survey. Which by the way, if anybody's ever done a survey before getting people to actually sit through 15, 20 minutes of filling out surveys. And a lot of these questions were open-ended answers. That's hard work, man. But I digress. The next survey that we did on the next conference was all about evaluation because it became very clear that one of the biggest and most painful pieces of working with LLMs is how do you trust them and how do you evaluate them and how do you know if they're doing what they should be doing? And so again, we created the survey all around evaluation, how you evaluate your system, not just the model, because I think a lot of people probably know this, your listeners I imagine are smart and they've not been disillusioned or they are disillusioned with seeing on social media how every time a new model comes out, it's like it beats all the benchmarks.

23:52 You're like, these benchmarks mean absolutely nothing to me anymore because every single model, no matter who comes out with what, it beats the benchmarks and it's now SOTA. And so I'm like, all right, SOTA, everything is SOTA now. So the evaluation survey was to try and put that into perspective and see how people are evaluating and is there anything that's working? Is there stuff out

there that is knowledge that we don't know about and we can share it with the greater community? And so we did the survey, and again, we made sure to make all of the data open source. And so you can see it's very rudimentary. It's just in an Excel or a Google Sheets spreadsheet with all the answers, but you get a really clear picture of why it's difficult, what people are doing, where the old standards and old ways of evaluating things, old in quotations, the ways of evaluating machine learning. We're trying to do that with AI, but it's not quite as simple as that.

25:03 And so it was eye-opening and we'll leave the links to all this in there, but it is been great. I take, on each of these surveys that we did, I just go through it. I spend a lot of time trying to understand what people are saying, and then I'll usually write a blog post. On the first one, I wrote a whole report about it. And so it's a lot of fun and it helps me understand where we're at as a community and as practitioners.

Jon: 25:32 Nice. Very cool. Yeah, we look forward to sharing those results with everyone and allowing all of our data analysts, data scientists listening and curious people be able to analyze the raw data themselves. And thank you for the key insights from there. It is a tricky situation with this. Every LLM coming out, breaking the benchmarks and becoming the state-of-the-art, the SOTA, it's so tricky because how can we do better? You see interesting things like, okay, there was a recent Grok release from X.ai, an Elon Musk outfit associated with Twitter and using Twitter data and this new Grok model, it has visual capabilities like GPT-4V. And so they created a new benchmark and they're like, "Wow, look on our new benchmark, this is the best model in the world." And in some ways I'm like, okay, cool. That's actually one good way of doing it. One good way of having these benchmarks is literally inventing new benchmarks every

time so that you can be somewhat confident that people aren't training their models specifically to excel, let's say MMLU.

26:46 But on the other hand, the obvious hack in that situation is if you're working at X.ai, you then you can create a training validation split and have the most perfect training data for fine-tuning just that task that no one else does because you made the evaluation. So it's tricky. All right. So it sounds like you are trying to do everything you can to answer this question. You are organizing The AI Quality Conference, which is coming up shortly on June 25th in San Francisco, AIQCon. Can you expand on what's going on with this conference? Yeah, it sounds like this problem that we're describing maybe is kind of the genesis of having the whole conference.

Demetrios: 27:34 So this one is the first time that we're doing it in person, which if you would've told me beforehand, I may not have actually decided to do it. Knowing what I know now, in-person stuff, wow. I'm used to the online version and having a six, eight hour live stream is way different than the in-person. And I knew it was going to be hard. I did not realize it was going to be this hard, but it is fun, man. Learning a ton just every day getting in standup with the team and figuring out how we're going to make this the best event possible. And so one thing that we're doing, which I'm very excited about, and as you said kind of coming off the back of this, realizing that evaluation is one piece, evaluating models is one thing, but also evaluating the system, not just the model is another thing.

28:25 And so how can we make sure that AI is doing what it's supposed to be doing? And that is what AI quality means to us. At the end of the day, AI quality is, does the AI work as intended? And so we are planning on not only having this conference to talk about ways that people are

doing this and get together with a bunch of people that, this is top of mind for everyone at the conference. And I think that's one thing that's really cool is that you see there are a lot of people thinking through this and making sure that, okay, I'm using AI, but I need to make sure that it's doing what it says it's doing. Or how do I know that if I'm going to put something into production, I can trust that it's not going to do some things that could potentially put my company into hot water.

29:23 And one way that we're going to do that is on the day of the conference, we're going to kick off a whole project to create these gold standards. And so you have the industry-wide standards, but then you have the use cases. And what we want to do is say, "All right, you're in finance and you have a chatbot that's using AI. You need to have certain gold standards and quality standards around that chatbot." And so can we try and just see what the use cases are and what the industries are and create working groups and nine months later come out with some standards that at least make it so people understand, all right, we've got a chatbot, let's make sure that we're hitting these standards that have been set out there. And so leaders from their team or leaders can tell their teams, "Here's some standards we want to try and work towards."

Jon: 30:26 Data science and machine learning jobs increasingly demand cloud skills, with over 30% of job postings listing cloud skills as a requirement today, and that percentage set to continue growing. Thankfully, Kirill and Hadelin, who have taught machine learning to millions of students, have now launched CloudWolf to efficiently provide you with the essential cloud computing skills. With CloudWolf, commit just 30 minutes a day for 30 days, and you can obtain your official AWS certification badge. Secure your career's future, join now at cloudwolf.com/SDS for a whopping 30% membership

discount. Again, that's cloudwolf.com/SDS to start your cloud journey today.

31:07 Nice. Yeah, it makes a lot of sense and I can't imagine how difficult it is to organize an in-person conference, especially the first time. I think maybe after you've done it the first time you're like, okay, we're going to go back to the same venue. We know some of the people over there, we know which food vendors we're going to work with or whatever. And so there's some pieces. But the first time, yeah, it's got to be a nightmare. Our researcher, Serg Masis pulled out a quote from you recently that you said that, "Organizing this conference has taken years off of your life."

Demetrios: 31:37 Great on Serg. He definitely, it was not wrong. I was not lying when I said that. I feel like I am happy that I never was a smoker because the effect of organizing this conference is basically as if I were smoking two packs a day.

Jon: 31:55 Wow. Crazy. You're organizing it with Mohamed Elgendy, the CEO of Kolena.

Demetrios: 32:03 Yeah.

Jon: 32:04 Actually, they sponsored the show in the past.

Demetrios: 32:06 Nice. It's very cool. And Mohamed is the one that back in 2020 even, he came on our podcast and he was talking about how you need testing for ML. And now I think it's very common for people to understand, okay, you need some kind of change management solutions when you're playing around with data, when you're doing things in ML, you need to make sure that if you're going to put something into production, you test it in every way possible before it goes out there. Otherwise, you are left with that situation that can wind you up in the headlines.

And you don't want that. Nobody wants to be in the head headlines for bad AI uses.

Jon: 32:53 Yeah, yeah, for sure. So we've already talked a lot about MLOps, LLMOps. In terms of other kind of related titles that I didn't get into. So we got into ML engineering, but speaking of Serg, he also brought up for me here that another title that we see related to this, and maybe this is kind of one of the easier ways of explaining what MLOps is, is through the lens of DevOps, which we haven't mentioned here. So is it fair to say that software development is to DevOps as machine learning is to MLOps?

Demetrios: 33:32 Ooh, I like that. I hadn't thought about it that way, but yeah, and I also think that MLOps is a subset of DevOps, if we want to put it that way. Because the thing that it's easy to get caught up in is thinking that MLOps is just implementing some tools and then you're good. But really it's that organizational level and having the reliability, having the ability to put things into production quickly and roll them back if you need to, but recognizing it's not just like a tooling aspect, it is an organizational aspect. And so I think DevOps has a lot of that organizational culture. And when you're looking at DevOps too, there was another thing that I was going to say. The whole idea, I'm just big on this because I was just thinking about it today. DevOps has a lot of change management. So when you make changes to code, you have a process and it's very mature process that goes into that, right?

34:46 You change code and then you have unit tests and you have integration tests, and you have somebody that's merging the branch and maybe you look over it and so there's a human in the loop and then you roll it out slowly and so you can do some kind of feature flags so that this new code goes into production and you make sure that it doesn't take down the whole website or take

down the whole app, whatever it is. Then you have other tools that can monitor if that new feature or that piece of code is actually being used by users. None of that exists when it comes to data. Where is all that? So when it comes to data, data's changing all the time and people are making changes to data all the time, whether it's way upstream or downstream. And so what is happening, and I just had a conversation with Chad Sanderson about this, and this is why it's top of mind for me, is because that all affects your quality downstream.

35:44 Data's changing continuously, but there's no integration tests, there's no unit tests, there's no type of feature flag on rolling out these data changes. There's no type of monitoring if the data is actually being used later on or the new data streams, it just can break things. And you see that in a broken ML model that now is making bad predictions or it's going insanely slow for some reason, or it's just not hitting the mark. Or you see it in a broken dashboard, you see it at the end product. And so it's funny to me that, going back to DevOps and that whole idea of change management and having these processes in place so that when you do change something, you can still have the reliability that you are going to be able to push out this change and you don't have to get a call at 03:00 AM.

Jon: 36:46 I think one of the really tricky things here for MLOps relative to DevOps keying in a little bit more on what you're saying, I think is that with DevOps, when you have some production issue, it can be reliably traced back to something deterministic in code. But when you've got a machine learning model in production, you update the model weights and there can be really subtle failure modes in really specific circumstances based on some specific kind of prompt or types of prompts that your users are providing or some upstream process is providing. And so it's really tricky you can't, if you're

working with a large language model with billions of parameters, you can't be like, oh, here's the parameter, the changed since that model weight update and that's why we're getting this issue. It's a black box.

Demetrios: 37:41 Exactly. That brings up a little tangent, but it's a conversation that I had with Verena Weber and she was working on Amazon Alexa and she was talking about how when they would update, basically they would fine tune the Alexa model, and when that would happen, they wanted to make sure that they didn't have what she was calling negative flips. And so she explained the story of how, imagine what you're used to saying to Alexa, now there's an update that got pushed to the on air and you're used to asking, "All right, play Taylor Swift." And then out of nowhere for 90% of the people, that phrase no longer works because you push some update in and oh, now it catches it's much better at X amount of phrases, but now it doesn't do the thing that a lot of people really want it to do. So she mentioned how they combated that, and it is very fascinating to think about how you just can't dig into those gigantic models and say, oh, here's the root cause of the problem, right?

Jon: 38:58 Yeah, exactly. It's tricky. And certainly you don't want to feel the wrath of the Swifties out there. That's one you got to be careful for.

Demetrios: 39:09 Oh, imagine how fast, yeah, the internet would explode if your Amazon Alexa stopped recognizing Taylor Swift keyword.

Jon: 39:19 And so what kind of guidance do you have in terms of processes or tools in order to try to combat these things? So we want our systems, our software systems, including those involving ML, that MLOps people are overseeing. We want them to be redundant, it's critical. We need some robustness. And you in conversation recently with Neil

Leiser, you talked about how we can manage these risks, especially when we're not even always in control of the model ourselves. So in that case, in that example that you just gave with Amazon Alexa, Amazon is responsible for the model weights, but oftentimes, and you talked about this earlier, when we're using tools like Ollama or LlamaIndex, we can be calling third party APIs like the OpenAI API, and in that case we have even less control. So what is your guidance for our listeners when they're trying to use third party APIs in particular, but maybe their own and trying to have some redundancy, some robustness?

Demetrios: 40:29 Yeah, I think there's almost like a spectrum of how hard and how difficult do you want to make this and what's your maturity level, what's your capability? What can your team realistically take on? Because if you are just trying to get something, and how fast? Is this something that you're just testing or is this something that you know works, you know has a business value and you want to take it a lot deeper down the rabbit hole and bring it in-house? And so I think I usually will say, and it kind of sucks because I don't like this world where the majority of people are just using some outsourced or some managed service from the clouds, like the three providers. But if you're on step zero of your journey, then that's probably going to be the easiest route is just grab the managed service from one of the three big cloud providers go with either Bedrock or SageMaker or Vertex or Azure ML.

41:39 And once you want to go deeper down the rabbit hole, then you can look at where are we having the most trouble or where would we like to customize the most? Because as they say, the best part about using SageMaker is that it's a fully managed service. The worst part about using SageMaker is that it's a fully managed service, so it is a blessing and a curse in that way.

Jon:	42:05	Cool. Yeah, that's great. And it's nice to be able to get those specific tool names out there for people to check out. Thank you. Over the next few years, what do you think are going to be key trends or maybe key technologies, key tools that develop in MLOps? Maybe we got a hint there already from you around how you were describing how there are so many fewer data quality evaluation tools relative to the kinds of DevOps tools that are out there.
Demetrios:	42:34	Yeah, I think that is one piece. I also think one thing that you see, and it's fascinating to me is how the data engineer plays such a pivotal role in the machine learning life cycle and how you really need to be, when you are creating machine learning or AI products, you really should be speaking with all your stakeholders, but data engineers are one of those ones. It's like I consider them the unsung heroes of this whole AI revolution. And so that's one piece. If you don't know anything about data engineering, that would be a great place to start and get involved there. Start, just figuring out pipelines. I think that is one thing, especially data preparation pipelines, that's huge and that's going to take you so far.
	43:35	And obviously you've got all the, if you want to play on what I would call the front, I almost see it as, I don't know if you see this too, man, but this is before API call or after API call or before the model or after the model because you've got, and that's not the best way of looking at it, but it's almost like you've got, I consider everything after the model, like the prompt templates and the orchestration and the vector databases and all of that stack that you can look at. And then before the model is those data pipelines and potentially if you're doing stuff in the ML world, you've got the Jupyter Notebooks that you're exploring and you're just trying to see is there something valuable here? Can we do that?

	44:24	So coming back to the question, very long-winded answer as far as where to go, it depends on where you want to play and where you think you are most excited because I could say, all right, well go learn some Kubernetes and learn about how to write YAML files because that's going to be very useful for a platform engineer. Go and figure out SQL because that's very useful for data engineering, but I don't think you can go wrong if you start to understand the whole data and ML lifecycle and then you start to really work on getting pipelines working like they should, data prep pipelines and just making sure that you have an understanding there. Deployment is always a key piece, but it's not as key if you're just planning on hitting an API. And so there's always these little caveats on what do you think you're going to be doing and I guess what it comes down to is just get good at learning and expect to have to totally revamp and relearn everything maybe in a year, maybe in six months.
Jon:	45:42	Nice. Makes a lot of sense. Do you have any particular resources that you recommend to people who are just getting started in MLOps?
Demetrios:	45:48	If you're trying to get into the other side of things like figuring out that AI engineering or the way to work with large language models, I would also recommend the blogs of, LlamaIndex blog is a great place to start. LangChain blog is also really useful because they show you all of the learnings that they've had. And really people are, we're kind of all in the same boat. If there's somebody that's more advanced or more experienced at this, the cool thing to know is that they're only probably six months more experienced at it than you are. It's very hard to be an expert of 10 years in this because it all came out just so recently.
Jon:	46:44	Nice. That's great. And for people beyond just people who are interested in MLOps. Growing the community, as you

have done, thee MLOps Community from obviously nothing to over 21,000 members today, lots of diverse backgrounds. Do you have any recommendations for people who would like to build their own community and whatever their area of interest is?

Demetrios: 47:12 Yes, so first off, communities are incredible. I think they're very special and they pay dividends many times over. It is hard work to get a community that is bought in, and I liken it to catching lightning in a bottle almost because you have to get people interested and spending time and hanging out in the community, helping each other. And community for those who are wondering like, oh, what is a community? Right? Let's define that first. I would consider a community where you have many to many interactions, so it's not just one person to many people like what a podcast or a video would be, right? A community is where you have a lot of people speaking with a lot of people and anyone can join in, and so that's why you traditionally see them on forums, but we also have our in-person meetups that you can go to and you can have those kind of interactions-

Jon: 48:22 And you have a podcast too.

Demetrios: 48:25 We also have a podcast, but the podcast, the fun thing is that I don't think people join the MLOps Community because of me, so it's not like a creator driven community, which is really good in a way, despite my ego really wanting it to be that people like me so much, they're like, let's go to his community. A lot of other creators have that. I don't have that at all. I'm just some guy that's part of the community and trying to help out, making sure there's not spam being shared all over the place, or you don't get approached by people trying to sell you tools in these places where the many to many conversations are happening.

	49:08	And it just so happens that I also really enjoy podcasting, and so I have a podcast and we have a blog post where anybody from the community can go and write on our blog, but in the forums and getting people bought into hanging out in a place and fostering that sense of community, I just think you really have to put in an absurd amount of effort, especially in the beginning to get it off the ground, because if you don't put in that effort and you don't nurture it in the beginning, it's almost like the snake eating its tail. You get people that might show up, but then there's not really anything happening. It's not really adding that much value, and so then they may go dormant, and then the next group or the next person that shows up doesn't see much value happening, and so it never really gets off the ground, and I've seen that happen with a lot of community efforts, and so that's what I would say if you're trying to start a community, you have to put in an unreasonable amount of effort, and especially in the beginning.
Jon:	50:30	Yeah. Yeah. I hear that. It is a tough thing to do, but it sounds like the way that you have this set up where you're trying to minimize the ego and have it be less about the content creator, more about the community supporting the community and you ensuring that spam is minimized, that selling is minimized, and probably on occasion other unwelcome behaviors are minimized. Yeah, that sounds like the way to go.
Demetrios:	50:59	And I would say, sorry, just to double click on how to put in that effort or how to get people involved early. One thing that I realize I didn't mention it, and it me a ton, I'm not sure if it always works for everybody in every scenario, but for me, one thing that I noticed got people more involved and more active in the community was instead of me answering everything or me going around and giving my thoughts on things, because I was in a really nice situation, I didn't know the answers to a lot of

stuff, but when people would come into the community, I would get to know them, and then if I had a cool conversation with them and then a question around whatever these people's expertise were, it came up in the community, I would try and get that person to answer.

51:52 And so it would either be me going to that person and saying, "Hey, I think you might know the answer to this. Can you chime in here?" Or I would just tag him in a thread and say, "Oh, yeah, I was just having a conversation with Jon about this." And that way others felt like, all right, it's not just me asking questions and then Demetrios answering.

Jon: 52:15 Nice. Very cool. Yeah, great guidance there. Demetrios, this has been a great episode, eye-opening for me across the spectrum of MLOps and of course community building, which we've just discussed. Before I let my guests go, I always ask for a book recommendation. What do you have for us?

Demetrios: 52:32 I love reading man, and I do it in every form possible, audio or visual reading or Kindle or iPad, give me whatever. I think that the book that I continuously read and I read it every day, it's more of a personal thing and personal growth type thing is A Course in Miracles. I love that book, and so I was trying to think about, there's probably a lot of recency bias. I could tell you the last books that I read or the books that I'm reading right now. It's a lot to do with parenting because I have two kids, but what I always come back to and I try and start my day with is A Course of Miracles.

Jon: 53:16 That's cool. That is a great recommendation. I love that. Thank you so much for bringing that, and it does seem like you have a lot of peace. It seems like something that maybe you've worked on, but yeah, great to have you bringing that into the world and into the community. Very

		last question Demetrios, we know some ways of following you already. Obviously we've got the podcast, you've got thee MLOps Community that people can get linked into. What are other ways that people should be following you?
Demetrios:	53:48	Yeah, I'm probably the most active on LinkedIn, but we do have Twitter and we have a newsletter too, if anyone is not on social media.
Jon:	53:59	Nice. All right. Thank you so much, man. It has been a wonderful experience and I look forward to catching up with you again soon. Good luck at the conference coming up.
Demetrios:	54:07	Thank you. I appreciate that and I appreciate you having me on here. This is awesome, man.
Jon:	54:17	What a deep dude Demetrios is. He's already accomplished so much, but I have a feeling he's just getting started. In today's episode, Demetrios filled us in on how LLMOps is a subset of MLOps, which itself is a subset of DevOps. He talked about how MLOps focuses on developing the ML platform that allows ML models to be put into production repeatedly and efficiently. He talked about how LlamaIndex, Ollama, LangChain and DSPy make it easy to work with and to scale up LLMs and how learning to engineer data pipelines such as through Kubernetes and YAML files can be a great starting point for getting going in MLOps yourself. As always, you can get all the show notes including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Demetrios social media profiles, as well as my own at superdatascience.com/787.
	55:05	Thanks to my colleagues at Nebula for supporting me while I create content like this Super Data Science episode for you. And thanks of course, as always, to Ivana, Mario, Natalie, Serg, Sylvia, Zara, and Kirill on the

Super Data Science team for producing another informative episode for us today. For enabling that super team to create this free podcast for you, we are also very grateful to our sponsors. You can support this show by checking out our sponsors links, which are in the show notes, and if you yourself would like to sponsor this podcast, you can get the details on how you can do that by going to jonkrohn.com/podcast.

55:39 Otherwise, share this episode with people who you think would love it. Review it on your favorite podcasting app or on YouTube, subscribe if you're not already a subscriber, of course. And yeah, you can do any of those things. But most importantly, all I really care about is that you just keep on tuning in. I'm so grateful to have you listening. I hope I can continue to make episodes you love for years and years. Until next time, keep on rocking it out there, and I'm looking forward to enjoying another round of the Super Data Science Podcast with you very soon.