SDS PODCAST EPISODE 1025: WORD GRAVITY: HOW TRANSFORMERS BEND SPACE, WITH DR. LUIS SERRANO

Jon Krohn:	00:00:00	What do planets orbiting the sun have in common with the words inside LLMs, according to my guest's new research, both, bend the space around them? Welcome to episode number 1025 of the Super Data Science Podcast. I'm your host, Jon Krohn. Today's exceptional guest is Dr. Luis Serrano, founder of the Serrano Academy, whose YouTube channel has over 200,000 subscribers hooked on his visual intuitive explanations of machine learning. Luis previously worked at Apple, Cohere, and a quantum computing startup, and the second edition of his bestselling book, Grokking Machine Learning, is out this very month. In today's episode, Luis explains his mind-bending new paper on how transformer architectures mirror Einstein's curved space time, why RAG isn't an agent, and how GRPO, the reinforcement learning technique behind DeepSeq's reasoning breakthrough really works. Enjoy. This episode of SuperDataScience is made possible by Anthropic Gurobi and the Open Data Science Conference.
	00:01:01	Luis, welcome back to the SuperDataScience podcast. Oh my goodness. So good to see you. How you doing, man?
Luis Serrano:	00:01:07	Good, Jon. Thank you so much for having me. Big fan of this podcast, so happy to be here.
Jon Krohn:	00:01:11	Yeah, we had you on the show a couple years ago, but I actually met you in person about a month ago at the time of recording. We watched the World Cup final from a bar in downtown Toronto. And boy, did I learn some new curse words in Spanish.
Luis Serrano:	00:01:26	Yeah, you got to see me in football mode, which is a different me. More rowdy than when I'm talking about machine learning, for sure.
Jon Krohn:	00:01:34	For sure. And I learned your beer preferences. It seems you love wheat beers in particular. I

Luis Serrano: 00:01:39 Do. I do, do.

Jon Krohn: 00:01:41 Yes, yes.

Luis Serrano: 00:01:43 It was nice to meet you in person and get to watch some soccer.

Jon Krohn: 00:01:47 Yeah, it was a fun day. So last time you were on the show, it was your final month at what was, at the time, a company that was talked about a lot, Cohere. It felt like Cohere was in the conversation next to OpenAI and Anthropic all the time a couple of years ago. But it's interesting because I don't hear about them as much and they're a Toronto based company, lots of people knew about them, but actually last time you were on the show was your final month at Cohere two years ago, and now you've been fully independent for two years. So what does a week look like for you these days? What are you doing? Oh yeah,

Luis Serrano: 00:02:20 I think it's similar to you, like things here, things there, a lot of things going on. So I think I remember asking you before leaving my job and you gave me an idea and I think it's pretty accurate. Sometimes super busy, sometimes not so busy when I've been making courses. Two big ones in particular with the Rack Pack and with the guys from the Neural Maze. And when I'm doing those, I'm super busy creating content because I have to create content at a rate that is much higher than what I normally create. But other than that, yeah, I've been creating a lot of videos. I sometimes do consulting meetings, sometimes teach live at other places. But yeah, basically I give my wild mind a place to sometimes hyper-focus on things and sometimes just be completely blank. So it's pretty random, but a lot of fun.

Jon Krohn: 00:03:17 That's nice. I'm really looking forward to. I don't think there's any way the viewers could tell this, but I'm recording a bunch of episodes this week so that I can

enjoy two weeks of vacation coming up in Europe. I think there probably hasn't been a period where I get two weeks without recording a podcast episode since I started doing this six years ago. So that is going to be nice. I think it's important, but back to you. You mentioned the Ragpack there and you mentioned Neural Maze. Tell us about those teams because the Rat Pack, famous Sinatra. Who else is in the original rat pack?

Luis Serrano: 00:03:55 They took our name. There's an ongoing debate. A

Jon Krohn: 00:04:00 Lawsuit with the Sinatra estate. The

Luis Serrano: 00:04:02 Sinatra. Yeah, that name, I was proud of that name because. Well, I always wanted to work with other people. That's my main thing. I love working in teams and I think we fill each other's gaps. I have the sort of conceptual teaching with cartoons and stuff like that, but other people teach with code, more technical ways or more formulaic. And we try to cover all those bases. So the first one is I started working with some of my best friends in this like Jay Lamar and Josh Starmer, and then they knew Martin Grotendorsk and Chris McCormick. And so we started making this group of five and we were just hanging out and we were just on Zoom calls thinking of if we could do something together. And then one day we said, okay, well let's just launch a course. And we needed a name and we thought about AI puns.

00:04:59 So there were a lot of AI Avengers, like this and that. And then the rag pack came up and I think we settled on that one trying to be fancy. So that was a lot of fun. Actually today we did our second cohort of the reinforcement learning for LLMs course. We started it live, totally live. And then I also was good friends with Miguel Otero from the Neural Maze. Their tour was with Antonios and Miguel Otero and we always wanted to make a course and their thing is agents. They're super expert on agents. So we thought let's make one on agents with a pretty intense

course, a six week course. We're all going to run it again. But that was a lot of fun. So yeah, I definitely enjoy working with others. Nice.

Jon Krohn: 00:05:47 Yeah, me too. It's the only way I can work. Don't love doing things on my own. But yeah, so Ragpack, Neural Maze, you guys are churning out fantastic courses. When you were talking about the Ragpack, you mentioned Jay Alamar, whom I'd love to have on the show. Oh yeah. Yeah, we should figure that out soon. He hasn't been on yet, but he is another rockstar like you. Oh my God. Amazing. He's got some great books, super bestsellers from O'Reilly. The way he

Luis Serrano: 00:06:13 Understands an LLM is on another level. You just open it and look inside. Yeah, it's been very rewarding for me to learn from.

Jon Krohn: 00:06:24 And then Josh Starmer, of course. Starmer's

Luis Serrano: 00:06:26 Amazing.

Jon Krohn: 00:06:27 Yeah, unbelievable.

Luis Serrano: 00:06:29 Yeah, he can really break down concepts and make them into a cartoon in a

Jon Krohn: 00:06:37 Way

Luis Serrano: 00:06:37 That's

Jon Krohn: 00:06:37 Amazing.

Luis Serrano: 00:06:38 So I love watching his content. And

Jon Krohn: 00:06:39 So he's been on the show. We did almost a two-hour long episode, which I never do that anymore. I don't know what I was thinking years ago. Sometimes I would just keep going and going and going, like the Rogan Show, but now I'm too tired. So that's episode number 553 if people

want to check it out. If the people don't know Josh Starmer, he's like, it's crazy. I mean, your channel has also grown a ton. We're going to talk about that in a second, but I think he's of anyone I know, I think he has the biggest YouTube following. He's coming up on two million subscribers on YouTube.

Luis Serrano: 00:07:15 Yeah, definitely. You walk with that guy in the street and people start recognizing him and stuff like that. You

Jon Krohn: 00:07:20 Go bam, double bam. Yeah,

Luis Serrano: 00:07:21 I know people do, bam.

Jon Krohn: 00:07:24 That's what Josh says in his videos for listeners who aren't aware. And yeah, so for lots of introductory statistical or machine learning topics, you can check out Josh's YouTube channel, but you can also check out Luis's channel for a lot of these same ideas. So the Serrano Academy, when we spoke in 2024, you had about 140,000 subscribers. Now you're over 200,000. You've got a full learning hub, you've got a Substack, you've got some of the paid cohort courses that you were talking about. Did you have a plan for all of this when you left Cohere a couple years ago or have you kind of just been following your whims and kind of like, oh, Jay and Josh reach out and you're like, "Cool. I'd love to do a course with you guys. Let's do it." Or is there a master plan?

Luis Serrano: 00:08:11 I never have a plan, a master plan. I just grade in the center life. I just go, "Okay, if I take one step in this direction, it might be..." When I left the job, I called it to myself, I called it a sabbatical just to not have pressure. And I started making my own more YouTube videos and free content and stuff like that. And then basically whatever comes up that seems like a good idea, I go for it and then I test it. So very machine learning approach, very RL. But yeah, no, I started the YouTube channel.

	00:08:48	I had a problem which you may have, which is a lot of people were like, "Oh, you should make a video on this." And I'm like, "Literally, the next recommended video is on that." And I feel like they had no order, but I do make the videos with some order in my head. I'm like, "This follows this." And so I put the learning hub, bunch of free courses in the page. The page actually for anybody is literally just serrano.academy. That's it. And then I organized the courses into what I think would be, and I also had some code labs and stuff that I did for the book or for blog posts. So I put them all together post, making all the material. I kind of frankenstained them into a bunch of courses and the stuff does follow. After this, this follows, et cetera. When people want to learn a particular topic, and definitely I point them to there first and I go, "Yeah, you may be able to find these videos in different places, but if you actually go there, you get the full thing for a particular topic." And just worked well.
	00:09:53	I mean, I think when I show it to people, it works more.
Jon Krohn:	00:09:56	It's a smart idea. When I navigated just now to serano.academy, serrano.academy, and I see everything there. One of the things that I see on the page, in addition to the stuff we've already talked about, like your YouTube channel, we've got your Grokking ML book, which is worth talking about now because so Grokking ML, it's an iconic book, but the first edition came out five years ago now.
Luis Serrano:	00:10:21	The
Jon Krohn:	00:10:22	Second edition is just about to drop. So I'm expecting this episode to be released in early, mid-September. And it looks like the release date for the second edition of Grokking Machine Learning is coming out on September 29th in the US and Canada, and I don't know all the dates around the world, but it's that kind of timeline. So tell us about the new version. I mean, a lot has changed

in machine learning since then. We didn't have ChatGPT when the last book came out. So what has survived from the first edition and what did you have to rethink completely?

Luis Serrano: 00:10:53 Yeah. So when I started writing the book, the only thing that I had in my mind or the most important thing is I wanted to do something as evergreen as possible because I didn't want that by the time the book comes out, there's a new package that kind of changed everything and the stuff is obsolete. So I went for the stuff that would be in a first year machine learning course regardless. So the linear regression, that's always going to be there. Neural network's always going to be there. Now they have a lot more stuff, but ChatGPT is a neural network that works with back propagation. So at some point you have to learn that stuff. And we focused a lot on the concepts. I wanted people to have a feel for what's happening inside an algorithm. And we went for sort of the quintessential algorithms, mostly for supervised machine learning.

00:11:49 So any tree-based algorithm, decision trees, boosting, grading boosting at your boost, anything regarding neural networks, logistic regression, linear regression. We had SVMs. Those are not as popular anymore, but I think they're beautiful. And then we had a fair amount of something I find very important, which is how to evaluate the models. It's not just train the model, but is it doing well? Do I just add more layers and that's it? Or do I need to check other things? Is it performing well in my data set, outside of my data set metrics? So we did cover a lot of metrics, overfitting, underfitting, all the keywords for a machine learning interview basically. And we had a whole chapter on using all these together in a data set. So I picked the most popular data set. Titanic data set is pretty popular. There's a lot done on it, but we kind of go over it and say, do all the possible things that what goes right, what goes wrong, how do you pick this algorithm, this model versus this other one?

00:13:00 So I was very excited about the book and definitely I wanted to last a while. So five years in machine learning is like dog years, right? It's like 20 years, eight years or something. But definitely every work was needed in several ways. The first thing is the package. I was using Psychic Learn, but I used a package that I really liked that was made at Apple called Touri. I was working at Apple and Turi Create. It's just the most beautiful package because it's so nice and it does a lot for you. So I wrote half of the book on that. That has been deprecated, so I needed to change that. And I went for Psychic Learn and surprisingly things worked actually really well on Psychic Learn. It actually didn't have to do very much and some things actually were even more intuitive. So that was a big change that I needed to do.

00:13:50 I changed it in GitHub, but I wanted to change it inside the book. Another thing that was enhanced was the boosting section. I mean, I did the gradient boosting and XG boost, but I think ever since I wrote it, it's just more clear in my head. There was the explanation, but then a few things clicked after that I was like, "Oh, I need that." And so I did that. But by far the biggest change is that we needed to add generative learning. So the editors reached out and said, "Yeah, we'd love to have. Obviously we're not going to make it a generative learning book, but if you have a rocking ML and you don't ever mention ChatGPT, you're not covering everything." So we have a new chapter, which is mostly text generation, the architecture of the transformer, the way I see it. I like to see it visually.

00:14:39 Everything, I like to see some kind of visual analogy with little words and things like that. So the whole architecture of the transformer and we did a bit on image generation, so stable diffusion process less developed than the transformer parts form is a big one because it has so many nice kind of foundational things. It's a neural network, but it's got the attention mechanism. It's got all these bits and pieces, the soft max, it's got the

positional encoding. So we basically go through all of them and yeah, I'm very excited. It's coming out soon and I'll definitely send you a copy.

Jon Krohn: 00:15:13 Fantastic. I would love a copy. I'm sure I'll see you in Ontario soon and I can get a signed copy. That's really what - Perfect.

Luis Serrano: 00:15:22 That's

Jon Krohn: 00:15:22 Really what I would like, Louise. If it's not signed, I don't even want it.

Luis Serrano: 00:15:26 It'll be signed definitely.

Jon Krohn: 00:15:27 Yeah. Is it easy to. We actually haven't talked about the word grok, so we should maybe tell our listeners what grok means for those who aren't aware. And yeah, you could maybe give us an example of how do you grok a transformer?

Luis Serrano: 00:15:45 Yeah. I didn't know what grokking was first. They don't teach you that in ESL, so I didn't know what grokking

Jon Krohn: 00:15:53 Was. I don't think they teach you that in EFL either.

Luis Serrano: 00:15:56 Yeah. And so definitely when they said grok. I knew there was a grokking series because there was a grokking deep learning actually by Andrew Trask. I worked with him, he's a great guy. And so when they reached out and said, let's write groking. I knew grokking was a special term. I knew grokking was a series of books. And then they explained to me, yeah, it's like understanding something really well. It's kind of like really breaking down a concept and

00:16:27 Just making it super clear. And so I though, yeah, that's definitely the one thing that I like to do the most, which is taking a concept and really breaking it down. So it really made sense when we thought about writing the book. I

pretty much do that for everything. I don't really understand stuff if I see it in complicated terms. If I see a formula, if I'm relying on the formula, I feel like I haven't understood it. So I need a little story, like a little picture or something. So for my own self, even before I explained stuff publicly, I needed to understand my own work and my own courses and my research. And even in university or in high school, I needed to grok the whole thing to understand it. So yeah, it fit right there. When

Jon Krohn: 00:17:13 You were on the podcast two years ago, you said that some of the explanations that you come up, some of the visual cartoony ways or the grokking ways you come up with, they take a decade to think of. So yeah, were there any particular concepts in this second edition of grokking machine learning that. Were there some concepts that finally clicked that five years ago you weren't sure how to explain it, but now you were like, "Ah, I've got it."

Luis Serrano: 00:17:41 Yeah. I feel like a few of them, for example, like XG Boost, XG boost was something that I saw as a black box. There's a similarity score that never really clicked to me. So in the book I say there's something called a similarity score. If you have a group of a set of numbers that are very similar, then it's a high score and it's low if they're not or something like that. And it never really clicked. So I had it there and then you use that for XGBoost completely. And then later I realized that the similarity score is just a hidden difference between two things. They never tell you that. They just give you the number. It's the sum of things square divided by their number. So when you take an average of a bunch of numbers, you take their sum and divide it by the number by N, if there's N numbers.

00:18:32 This is the sum square divided by N. What makes no sense why? When I realized that that's the difference between the variance and the new variants. So when you have the variance of the original set, you have a lot of

variants because the numbers are all over the place, and then you break it into two and then you subtract those variances. So if you break it correctly and you have sets that are more homogeneous, then your difference is high because you went from a high variance to two sets of low variance. And so when I saw that, then I was like, "Okay, well, I need to completely rewrite XGBoost because that similarity score now needs to be explained as the difference between the original variance and the variance obtained by breaking the set." So that one was a big one. And then a few other things that came up, especially in the new chapter, then I have the attention mechanism that I like to see it as a kind of gravity thing of words flying around and then stable diffusion where I think of clip embeddings.

00:19:29 Embeddings are something that makes more sense every time to me. I remember at the beginning, you always have that example of king, queen, man, woman, and there was the parallelogram to me, but eventually what you start seeing embeddings is every dimension means something, even if you don't know what it means. I feel like that's the clearest way to see an embedding. It's like if I have 1,024 descriptors of a word and those are my 24 dimensions of the embedding, then the parallelograms can happen for free.

00:20:03 I like to see it as that. And obviously I like to see embeddings as words flying around in space and stuff. So yeah, definitely a lot of concepts in that book were sort of, as I was writing it, they came up more clear to me.

Jon Krohn: 00:20:16 Fantastic. Do you ever find yourself chatting with things like Claude or tools like that to help you understand concepts? All the time. All

Luis Serrano: 00:20:24 The time now. Yeah. Now I go to DeepSig for a while. DeepSig is pretty good. Claude two, I just ask them stuff because all these models started explaining to you the

stuff in the most formal way. So they go, "Well, this is that." It's like, "Thank you, Wikipedia." And then I just keep asking, asking, asking, and I go, "Okay, well, I didn't understand this part. Tell me more." And then eventually I start injecting my own examples. So I go, "Okay, I want you to see attention like this. Now, explain this to me." So it definitely. I find that by itself, maybe one day, I don't discard that possibility, but right now by themselves, they don't rock all the way there, but they definitely take you somewhere and then you help them cross a bridge and then they take you farther and then you help them jump over a wall and then they take you farther.

00:21:20 So I feel like the combination of human machine is taking me farther. So I understand stuff faster because before I had to just bug my friends and ask them my expert friends and ask them and ask them and ask them until their patience runs out and they're all very friendly and nice, but eventually people have a breaking point. But these models, you can go on forever. Worst can happen is that you have to upgrade and do the higher tier and then keep asking that people don't have that property.

Jon Krohn: 00:21:49 Why do you think listeners should still buy books? I'm writing a book, Engineering AI Agents right now. I have some reasons that I guess I could give, but why do you think people should buy Grokking Machine Learning, the second edition instead of talking to an LLM? I'll

Luis Serrano: 00:22:06 Definitely get your book. I'm excited about it.

00:22:09 Yeah. No, I think humans still. I wouldn't just take a book written by AI or I wouldn't just rely on AI. I mean, I think the human insight is still important. Even if it's just a human that. For example, when I look at what I input to a model, I think I am a way better, bad understander than any model. And what I mean by that is I know where to get lost. I know exactly what places I get lost as a student and I get more lost than the average student by

far. So I think one of my biggest strengths is I know what are the parts that somebody in the room are going to get lost. And I think models overlook that because they just know everything. So I think those kind of skills are what's important. The experience, the human has gone through difficulties at some point.

00:23:12 They had an absolute disaster and got fired from a job because they made a huge mistake. The model doesn't have that, and maybe it can read it from someone, but there's a big chunk that the human still inputs, especially in education. Education requires a great deal of empathy, a great deal of human connection, a lot of things that the model doesn't have, at least yet.

00:23:40 I still rely a lot on humans, and even though I prompt these models like crazy trying to understand stuff, I still rely on books. I still rely on YouTube channels and courses because I think education is a group endeavor. It's a very human thing still. Even if it's the most technical subjects, there's a human connection that you feel with someone teaching you or someone learning from you that I think is hard to replace for a machine. Yeah.

Jon Krohn: 00:24:14 I think another one of the great advantages of a book is that it's a convenient collection of everything at the right time and you can just know that in a well-written book, you can go from the beginning to the end and you're going to get this great arc. It's going to cover all the key topics you need to know. There was an editorial team. There were people who advised on what should be in the syllabus of the book in the first place. I think you typically end up with a really great product and at least not at the time of recording. We can't yet completely have the book generated by an LLM, maybe by the time this is published.

Luis Serrano: 00:24:52 Maybe. Maybe it'll generate podcasts too.

Jon Krohn:	00:24:56	For sure.Maybe this podcast right now is generated. How do we know? Just had the hands,
Luis Serrano:	00:25:05	The hands of five fingers, five fingers. Yeah. Still us.
Jon Krohn:	00:25:09	All right. So in addition to your new book coming out is something you've been writing, something you've been working on. Something else that you published is actually a new paper. So you published last November with, I'm going to try not to butcher their names, Ricardo DeCipio and Jairo Diaz Rodriguez. Yes. Very
Luis Serrano:	00:25:31	Good. Yeah.
Jon Krohn:	00:25:33	I really put a lot of effort and thought into that. You guys wrote a paper together about the curved space time of transformer architectures. Yes.
Luis Serrano:	00:25:43	And
Jon Krohn:	00:25:45	That is pretty mind blowing. I think we're going to spend a bunch of time on that right now because we'll learn about transformer architectures in a way, but I think we're also going to learn about space time and relativity and these kinds of concepts. Yes.
Luis Serrano:	00:25:59	This was definitely very exciting, definitely very exciting to work on. And yeah, definitely for the physicists listening, we use the word relativity, but in a very loose way. It's basically a spacetime curvature analogy, weak analogy of what's happening inside a transformer. But I think it opens the door to what's happening underneath and the fact that physics related things start appearing, I found it mind blowing. The story of that is that in order to understand attention, it never clicked to me with the. People say it's like a search table with a query and a key never made sense to me. Then they said, "Oh, the words pay attention to other words never made sense to me." Then they gave me the formula, made even less sense. Soft max of KQ divided by square root of DK times V.

00:26:54 Said nothing to me. So I started looking at videos and looking at other things and looking at just writing and watching videos. Somewhere in the process, somebody, which I bless his soul, I don't remember which channel was this. I think it was kind of an underrated channel that didn't have very described, but this person just kind of made a. This is a beautiful description where at some point they did a linear combination of the words and they said this word becomes more like that one. And the linear combination added to one, the coefficients added to one because there's a soft max. So you turn your word apple into 70% of apple and 30% of orange, if you said the two words consecutively, say orange, apple, you know what I mean? So the word becomes a percentage of itself and the rest of percentage of another word.

00:27:48 And to me, that's moving in a line. If I have two points and then I take a percentage of the position of one point and the other, I'm moving in the line between them. And so I though maybe words are moving in a line. And I started rewriting all the equations as in words are in a position in space because embedding words are in a position in space and then attention just moves them in a line toward each other. And I immediately though, oh my God, that's gravity or magnetism, right? Words pull each other. I thought that makes a lot of sense because if I'm saying the quintessential example is the river bank. Bank is a bank in the financial sector of the embedding around stocks and bonds. Then you say river bank and the bank just becomes a nature thing. So the word river just pulled it towards itself into the nature region of the embedding because in the nature region of the embedding lives a river and tree and stream and sea and all that stuff.

00:28:51 And so it just pulled it. It infused itself with it, right? It infused some properties of it. For example, the nature property. It didn't infuse all the properties. There are some that don't, but the nature property, it moved in. The moving actually works in different directions. It's not

towards it because of the value matrix. But anyway, the fact is words. I started calling it word gravity. And as I said, any physics words that I say is a very loose analogy, but I started calling it word gravity, word gravity, word gravity. And then one day I gave a talk in the Toronto Machine Learning Summit and Ricardo de Cipio, physicist. I also, by the way, I hope I'm pronouncing it well because I don't know Italian, but. And Ricardo's a physicist who was sitting in the first row, and then he came to me and said, "Hey, what you have is actually..." When Newton was talking about gravitation and the Einstein came, which is a force between objects, he died and he had no idea why this force happened.

00:29:51 And then Einstein came hundreds of years later and he said, "It's not a force. The masses are bending the space. The reason you fall Towards the earth or the earth falls towards the sun is not because the sun exerts a force, it's because the sun bends space and all of a sudden the line in which the earth should be flying in a straight line, it's curved because of the sun and it happens to be curved around the sun and that's why we're there. So he said, "I think it's the same concept. You're talking about word gravity as like a force between words. I think it's more like it's probably a space-time curvature thing. Probably words are bending space in a way that they just pull towards each other." And so we started working on that and he worked out the math a lot. He knows the physics a lot more than me.

00:30:37 So he actually worked out the geodesics and very much like the matrix that appear in the geodesics appear in are the key query and value matrices. And then we started working with another friend, hire the professor at York University in data science and statistics to run a lot of experiments. So these two guys are wonderful. They actually know a lot more about that than me and worked out both the physics and a bunch of experiments that really study this analogy. And so we're very excited

actually of this. I mean, it provides an analogy. I think it's more of a visual work. We meant it as a visual work. We've got notices of labs that are working with it for something else. So I'd love to see applications of it. But as of right now, we though of it as a fun analogy, like physics analogy of what's happening inside ChatGPT or inside the brain of these models.

Jon Krohn: 00:31:33 In that analogy, what plays the role of gravity in the same way that you described the sun bending space so that the earth comes toward it? Yeah. What's gravity in a transformer?

Luis Serrano: 00:31:47 Yeah, that's a great question. So it's not the same gravity as in mass one, mass two times R squared in particular, because mass doesn't make sense. If you have a big mass, it attracts every word. But here it's not the case. For example, river really attracts bank, but it doesn't attract other words. If I say river tree, it's still the same tree. So it's not a matter of how close you are because tree is close to river, doesn't get pulled as much as bank, which is farther away. So it's not that words independently have an information of how much you pull and they pull everything based on the distance and the mass of the word. For every pair, there's an actual quantity of pull and it's not symmetric because for example, bank doesn't pull river. If I say the river and the word bank comes before, it's still a river.

00:32:45 Unless they name something bank, a river of something, then yes. But as of right now, the pull is not symmetric. A word can pull another one, but the other one can't pull. So it's weirder than gravity in many ways. And the formula is not GM1 and two divided by R squared, it's soft max of KQ, blah, blah. And it's based on the dot product of the words, but when you multiply them by K and Q. So K and Q, what they do, the key and the query matrix, is they extract all the features of a word that would influence other words for key. And the query

extracts all the features of a word that get influenced by other words. So in river bank, then the key of river extracts the nature part and says, "You know what? If a word comes with some nature, the river will pull it." And the query does that for bank.

00:33:40 And then you multiply those. So it's very dependent on the key and the query matrices that basically transform your space into something else where the dot product says more than just similarity. So it's a similarity, it's a two directional similarity. And then the V matrix comes out and says, "You know what? You want to move in this direction, but I actually want to move you in a different direction because some properties are movable and some are not. I can add nature to the word bank and turn it into a more nature word, but there's a bunch of properties that are not. For example, bank is a noun. I can't change that no matter how much I pull it." So for example, in the book, I have the example canoeing bank. So canoeing is a verb and it pulls bank, but it's never going to bring it to become a verb.

00:34:31 It's going to bring it towards nature. So out of the all day descriptors of a word, if there are 1,024 numbers in your embedding, a bunch of them are movable, a bunch of them are not. So V does that. And interestingly enough, in the analog for relativity, K and Q make a lot of sense, but the V just came out of nowhere. So it even has extra stuff, but I'm getting ahead of myself.

Jon Krohn: 00:34:55 Luis, you're talking about the bank bending towards river. It reminds me of an analogy that happens in the paper that I found fascinating. And this isn't really that related to data science or AI. It's I guess more an astronomical thing, but maybe it'll help people understand this same kind of transformer idea as it was helpful for you. I'd love you to tell us more about Eddington's 1919 eclipse experiment.

Luis Serrano: 00:35:22 Yeah, that was one of my favorite experiments because what happened is you can't really see space time curvature. We feel like we live in a giant 3D cube, but we don't know it's bent in different ways. You can see bending in 2D because we can sort of see the earth being round, but you can't really see the universe being curved if all we have is 3D vision. So what Eddington did was he though about a star that was sort of behind the sun and in a certain eclipse and you would see it not behind, but in another place. You would see it sort of in a place where it's not. And the reason is because the light of the star came in and bent thanks to the sun and then got to our eyes, but we could see it right behind. We could see it like it was above the sun.

00:36:18 And so we have an analog of that experiment, which is we have the word, and it's always with bank and river, but basically bank takes the place of the star and river is the place of the sun. A fun story, I actually wanted to find the right example where the word would be sun and the word light would be bent, something like maybe there was so much sun that I had to pick up the suitcase and it was light, something like that. Anyway, I couldn't work it out. So if it ever happens, I'll have an experiment with the word sun and star or light or something, but we had to do bank and river. And basically we just looked at space is the embedding, right? And the analog for time is the layers in the neural network. So the first layer is the beginning of time and each layer is a snapshot in time.

00:37:08 And obviously then that's a problem because time is discreet in our analog, but we can't really mimic the continuity of time. We can only take snapshots. That's a limitation, but actually we had what's the opposite of limitation? We had actually something that was easier, which is that you could see the entire trajectory, whereas Eddington could only see the end of it. Where do I see the star? He doesn't know where the light went because he can't look at them. We could actually take the position of

river and bank in every single layer. So there's a pro and a con. But basically, yeah, I mean, the word bank flies through the embedding in every snapshot, in every layer. And it's like bank is here, then it's there, then it's there, then it's there. And eventually it appears in another place at the final layer. And when we added the word river in the sentence, then it would basically go around river in a different path.

00:38:07 So we would see the path of bank is this, but when you add river, then the path of river is this. So we had no choice then to say, well, river must have been the space around it for it to take a different route.

Jon Krohn: 00:38:21 It's interesting to be able to see that happening through all the layers of the neural network in that transformer. That is very cool. One of the counterintuitive findings of doing that kind of analysis was that I think you found that function words like two, like T-O, which are like, they don't seem like they have much semantic meaning, but they actually showed some of the sharpest curvature in your experiments as opposed to semantically heavy words. What do you make of that?

Luis Serrano: 00:38:51 Yeah, that was a really interesting experiment because some words are pretty flat. For example, the word the doesn't get influenced very much. The is the. On the other hand, two can be influenced a lot because it can mean different things based on what's before. Two can be to something or there's a lot of meanings of it. So in some layers, it really showed a lot of curvature because it was easily influential. So that's something interesting we found. We also found that each layer takes care of certain things. So sometimes words had a big curvature around them in some layer and in another one not so much. And that just lets to think that different layers are taking care of semantics or meaning or things like that.

Jon Krohn:	00:39:40	For sure. That's something that we've known about deep neural networks forever where you have different layers that come to represent different kinds of meaning. And so yeah, in some layers, two doesn't need to change, but in others it needs to change wildly in order to be able to make sense of some sentence or document or what have you. Very cool. Well, thanks for talking us through that paper, The Curved Space Time of Transformer Architectures, Decipio, Diaz Rodriguez and Serrano. I'll have a link to that in the show notes for sure. Let's move on to agents. Agents are, I don't think we've talked about them at all yet in this episode, which is pretty amazing because it's hard to go very far without talking about them. You actually just wrapped a course called, it's your first paid cohort course. You can tell us what that means, what the cohort course is all about, but it's called Grokking Agents in Production and it's with some more names that I'm going to start with.
	00:40:46	I'm
Luis Serrano:	00:40:46	Giving you
Jon Krohn:	00:40:46	So many
Luis Serrano:	00:40:46	Times. Miguel Otero
Jon Krohn:	00:40:47	Pedrido and Antonio Zaros.
Luis Serrano:	00:40:51	Saraus. Although if I was Spain, I wouldn't say Saraus. I would say Tarous. Taraus Moreno.
Jon Krohn:	00:40:58	There you
Luis Serrano:	00:40:59	Go. I was saying it wrong actually. I was saying Miguel Otero Pedrido and it's Pedrido. Yeah. I didn't even hear a difference between those two. Pedrido versus Pedrido. Okay.
Jon Krohn:	00:41:10	Okay. So

Luis Serrano: 00:41:12 Yeah, even I mess up the Spanish. But yes, yes, this course was very fun. It was a six week course. It was dense. It was really more of an eight week course. I'm rocking with Miguel Otero and Antonio Sarahus from. They make the Neural Maze, which by the way, this is an amazing channel. They have so much Substack subscribers and they make so much content. If anyone wants to learn agents, definitely the Neural Maze, you should check it out. And so I worked with them on this course and I was doing. It was the same breakdown as I normally do. They have the technical side and the deployment side. And I was doing the sort of conceptual explanations on agents and it was a lot of fun. I learned a lot about agents. For example, I had a conversation with Miguel where actually we're recording a video because it was such an interesting thing.

00:42:12 I used to think that RAG was an agent. So I would say, RAG is an agent. Whenever you have an LLM doing things, that's an agent. And then I quickly found out, he corrected me and said, no, Rag is not an agent because it's an LLM workflow. So one thing is an LLM that does stuff and another thing is an agent. An LLM workflow is when you tell the LLM what to do. You go, okay, you do this and if this happens, you do this and then you nod, you close and that's it. An agent is when the LM makes a decision. When the LLM says, oh, I need to look for more information, I go look for information. Now I feel like I can answer this question, but I'm missing something. When the LLM is in charge, it's an agent. So that's one of the interesting things I learned.

00:43:06 And yeah, other than that, it was a lot of new stuff and basically a zero to 100 on agents with deployment.

Jon Krohn: 00:43:15 Yeah. Lots of the unglamorous parts, containerization, IM, observability, CICD. And it seems like the tagline for the course is demo agents are trivially easy to create, but production agents get hired.

Luis Serrano: 00:43:31 Yeah. And that's what the course does. It's end to end. The project was to build a PDF reader that can read everything on your PDF using different techniques, chunking, OCR, all that kind of stuff. And they combine them and then answer questions on it. And every module had its own sort of deployment part. We use Google Cloud and then Cohere for the LLM. So it's pretty interesting. I learned a lot.

Jon Krohn: 00:44:02 In 2024, when you were on this episode previously, you predicted agents and tool use as the next big wave right after multi-modality. And you got both of those right. Are there things that happened faster over the past two years than you expected? Or do you think there's parts of agents, multi-modality that are still over-hyped?

Luis Serrano: 00:44:21 I'm so glad my prediction went right. Thank you. And you're so kind that if I had done something wrong, you would have erased it, right? But I'm glad it worked out. Yeah, no. I mean, definitely that was the vibe at Cohere when I was leaving. The vibe was tool use, tool use. It wasn't even called agents. It was tool usage. You got the thing to talk, you now have to make it do things. It's the next thing, because it's the next thing with everything, with the internet. First, it was like text coming in and coming out, and then it started doing things like a bank transaction or something. And it looked weird at a time. It looks weird to get the model to do things for you, but that was definitely the trend. No, I mean, I think it went. I didn't predict any speed for it.

00:45:12 I knew it was going to be fast that we're going to start getting this to do things. Something that I found interesting is that in my head, the LLM already talks and now you make it do things. Go open your calendar, go write an email. What actually didn't occur to me, and I found it really interesting. I mean, it was there, but it just clicked, is that you can use agents to make the LLM do better things. You can have an agent. If you want to write

a book, you don't just tell the LLM to write a book. You have a multi-agent system where one writes the syllabus and the other one makes the chapters and the other one checks it and the other checks the grammar and there's a loop and there's a manager that decides what part you need more. And so definitely the fact that you can make the LLM talk better by making it do things, that was a nice addition.

Jon Krohn: 00:46:06 Yeah. Worth checking out for folks who want to get into getting agents into production, could be grokking agents in production course. Definitely. You can of course, an easy place, of course I'll have a link to that in the show notes, but again, you can find all of this at serrano.academy in one convenient location.

Luis Serrano: 00:46:23 Yeah, definitely check it out. We're going to run another cohort. We had a lot of students and it was a lot of work, a lot of interaction, a lot of office hours. And then we're definitely going to run it again later in the year. So keep an eye if you want to learn agents.

Jon Krohn: 00:46:38 Fantastic. Thank you for all the great courses, Luis. Thank you. Two of your six agent course modules were on evaluation. So things like golden test sets, trajectory scoring, LLM is judge, eval gated deployments. Why is agent evaluation so much harder than say reg evaluation?

Luis Serrano: 00:46:58 Yeah. I mean, I feel like ML evaluation gets harder and harder, right? 10 years ago it was like evaluating a multiple choice test when it was predictive machine learning, right? Then LLMs came in and it's like evaluating an essay much harder. So you have to evaluate conceptual writing. And then what's one step even harder than that is evaluating doing stuff, right? How do you evaluate someone who's doing a job? That's even harder than grading an essay. So definitely agent evaluation is one step harder than LLM evaluation. And the reason is

because there's so many things to take care of, right? If you only look at the final result, that has to be good, but that doesn't tell the whole story. And then you have to see every single step along the way. Now, every single step is, did you do this step or did you do the step right?

00:47:51 Huge difference, but you have to check if the step was done. And sometimes it's hard because let's say something simple. In an LLM doing RAG, you ask it something and it just answers out of memory and it answers correctly. So you just go, good. But no, it didn't use RAG. And if I'm going to ask it something like, what's the weather today? It has to use RAG. It has to use the tool. So I can't just rely on did it do the thing right? So basically the modules on that were something as simple as check if all the tools were called, then check if the result was good, then check every tool separately using different ways, root score, just basically comparing words, using an LLM as a judge is also quite interesting. And those two are basically need to be done because there are cases where yeah, you just check if the right words appear in the answer, but other times you need to check if you need an LLM to check if the question was answered.

00:49:00 So there's a lot of little moving pieces that you somehow have to combine into a big evaluation step. But we made a big deal on that because that is less common to see. A lot of the work on agencies do it and deploy it and we're like, no, you have to make sure it works well. I mean, you need to be very responsible on this thing. So we add a lot of material on evaluation like I do with everything. I think it's very important.

Jon Krohn: 00:49:29 Yeah. Super important. This is something that we run into a lot with my consulting firm, Y Carrot. We have enterprise clients, we have government clients, and it is essential for them that if agents are deployed, they can be trusted. And it's these kinds of evals that allow that to

happen. So thank you for creating a course like this. In addition to the agents course, you've also built a whole free course and video series on reinforcement learning for LLMs. So you cover RLHF, PPO, DPO, GRPO. If we had all the time in the world, I would have you go in and explain all those different things. But so instead, I'm just going to skip to GRPO because that seems to be a reinforcement learning approach that has really mattered a lot, particularly with the deep seq explosion from last year.

Luis Serrano: 00:50:20 Yeah. I'm very interested in GRPO and I love it. And yeah, I work on that and the video series on GRPO was a lot of fun for me because it was just learning stuff from scratch. And we talk about that in the course a lot in the course I have with the rag pack. And it's basically, here's the interesting thing of GRPO versus PPO, right? Basically GRPO is for reasoning models. It's for making the model not just talk, but do math or write code or do logic, which are much harder. But when you look at GRPO, it's not an actor critic model. It's really just an actor. Whereas PPO is actor critic model. So I'm going to sort of overgeneralize, but in my head, PPO is more meant to make the model talk well and GRPO is make the model do math well. And there's one difficulty and one thing that is easier, one thing is that's harder, which makes GRPO what it is.

00:51:23 If you want to make the model talk, that's easier than making the model doing math. I'll tell you why in a minute. But if you want to evaluate a model talking, that's harder than evaluating a model doing math for the simple reason that if you write an essay, that's hard to grade. But if you do a math test whose answers are numbers, that's easy to grade. So we kind of have a two by two square where we say, okay, talking is easy, doing math is hard, but evaluating talking is hard and evaluating math is easy. So PPO has an okay actor that talks and a really good evaluator that evaluates. And GRPO has a really strong talker that does math. And an

evaluator that is not as strong, obviously it's strong, but where does it come that the evaluator in GRPO is not as sophisticated as the evaluator in PPO?

00:52:29 The evaluator in PPO is a model that gets trained. So you're training the talker and the evaluator at the same time. The model talks, the other one evaluates, the one talks, the one evaluates, and they both get better. On GRPO, you're only training the talker. Your evaluator is fixed and the evaluator is simple.

00:52:51 It's a formula that grades your responses and it doesn't get any better. It just gets trained. It uses LLMs, it uses other stuff. It uses compilers for compiling the code. It uses math engines to check the math. It uses a lot of stuff, but it's not a model that gets trained. So all you have is the actor and the evaluator is sort of this big formula they have in the paper where you have. It's basically a big loss formula. But that's what I find interesting of GRPO because you would have thought in order to make the model the math, I need to make everything better, but it's not. I need to make the model stronger, but the evaluator actually, there's a lot of stuff I can use in the fact that the model is reasoning instead of talking that I can take advantage on. And at the end of the day, I did say another thing that I want to clarify.

00:53:44 I said talking is easy and doing math is hard. Obviously talking is not easy for a model, but what I mean is that probabilistically talking is easier for a model because the model doesn't go for the right answer. The model throws a dart in the vicinity of the answers because it's a probabilistic model and it gets some answer that is close to what you meant to say. If the perfect answer is here and you throw a dart, you get something close. But in text, that's no problem because if I live in text space and I pick a sentence that's close enough, I'm likely to say the right thing worded differently. You try that in math, that don't work because if the answer is two plus two equals

four and I move one millimeter, I get two plus two equals 4.1 and that's not correct. So math is a hostile space where a right answer is surrounded by wrong answers.

00:54:35 Code is the same. A right line of code is surrounded by wrong answers where you change a colon into a semicolon. Whereas text, for the most part, if I land very close to what I wanted to say, I'm okay. So in some way, then we have this two by two matrix where we have top left, we have talking is easy. Top right, we have evaluating talking is hard. Bottom left we have doing math is hard and bottom right we have evaluating math is easy. And the top row is PPO and the bottom row is GRPO basically.

Jon Krohn: 00:55:08 That was a really fantastic explanation of GRPO, which I don't think I even said is called group relative policy optimization. That what GRPO stands for. And I'll have a link to materials on that of course in the show notes as well, but fantastic explanation there and particularly for why it's important, that mathematical reasoning. And it seems like that's a key part of this. GRPO was actually introduced in a paper called Deep Seek Math.

Luis Serrano: 00:55:37 Yes. Yes, yes, yes, exactly.

Jon Krohn: 00:55:39 Pushing the limits of mathematical reasoning in open language models. So aligns very nicely with what you're saying there. Luis, you used to work on quantum computers at Zapata Computing. Yeah. What's been happening in the world of quantum as it relates to ML or AI? Have there been any big innovations in recent years?

Luis Serrano: 00:56:03 I think there's been hardware things happening. The computers have been getting bigger. There was this topological qubit thing happening. So I think in terms of that, yes. I don't know, and maybe it's just that I haven't been sort of that up to date, but I don't know in terms of algorithms, at least in machine learning or something. But yeah, I've been out of the world a bit, but definitely

hardware advantages have been happening and hopefully we'll have a quantum computer sometime. That I don't know if I can predict, but you know what? I'll say the two things and we'll have both recordings and you're so nice that in two years you'll put in the podcast the one that was right. I'll say that the quantum computers are never going to work and then I'll say we're a year away from a quantum computer and then we'll make sure the right video appears.

00:56:55 How's that?

Jon Krohn: 00:56:56 Nice. Sounds great, Luis. Last technical question for you here. It's actually not that technical, but it's related to a technical topic. So the International Math Olympiad, you're very familiar with it because you won the Broadens Medal in the International Math Olympiad years ago. And in 2025, AI models hit a gold medal standard for the IMO, for the International Math Olympiad. And it seems like in 2026, they did even better. We haven't had full reports out yet. What goes through your mind when you see a machine being able to act at that level on math? Is it just exciting for you?

Luis Serrano: 00:57:40 I find it very exciting. I mean, I think there's always a tiny bit of nostalgia saying, "Oh, I worked so hard to get this and a machine can do it." But I think that's 0.1% because I'm 99.9% very excited to see models being able to solve these problems. These problems at the end require a lot of creativity, but a lot of it is using a lot of tools that exist, which is not that different from a game of go or a game of chess where it requires tons of creativity. But the fact that you have a finite set of moves and that a computer can search in a space so humongous, then definitely, I mean, Olympiad Mathematics is a game like that. And I actually was really hoping that a model would perfect it. I think it doesn't take away from the competition in the same way that we still watch people run, even though cars are faster.

Jon Krohn: 00:58:42 And even in intellectual pursuits like chess that you already mentioned.

Luis Serrano: 00:58:47 Yeah. We still watch chess. Go is still interesting even though a computer can be there. So computers, we can watch the Olympics with machines and it'll be less fun, but they will win. We could watch two models playing chess, but we still watch two humans playing chess because I think there's still the fact that we do it with our limited computer in our head and we use different creativity and things like that, I think it's exciting. So I still follow IMO results very excited and I think it's not going to go away. And I think that's the same though that I talk with a lot of IMO people and I see nothing but excitement. I mean, I think it's great to see models that can build and it's a good benchmark. And I mean, research mathematics is already in reach and I think I'm excited about it because at the end of the day, the human is going to pass to a higher level of saying, "Okay, what do I want the direction of this field to go?

00:59:51 What I want are the big problems I want to solve?" And for things like solving the problems, I mean, mathematics at the end of the day is a game. A finite number of axioms. And from the axioms you build lemmas, from the lemmas, you build theorems. And it's no different than walking in a very high dimensional world trying to find a needle in a haystack. So models can do that very well. I mean, Go has more positions in the board than atoms in the known universe. And it's able to navigate that space very well and find the tiniest needle in the biggest haystack. So I'm excited. I'm excited of seeing it solve mathematics. And people would work like Terry Tau, for example, is the best mathematician. He's done a lot of work on AI and mathematics and there's a lot of other famous mathematicians doing that work.

01:00:53 Timothy Gowers, another field analyst is also working on that. So yeah, there's definitely interest there.

Jon Krohn:	01:00:57	Well, fascinating conversation today, Luis, across the board, reinforcement learning, agents, curved space time of transformers. And of course, Groking Machine Learning, the second edition, which will be available to listeners very soon. In addition to your own books, do you have any book recommendations for us?
Luis Serrano:	01:01:19	Yeah. I think I was checking that I gave you some of this last time, so I wanted to not repeat myself. But there's one that actually someone has been on your channel. There's on your podcast, Kathy O'Neill: Weapons of Math Destruction, Math X M-A-T-H. That's a really good one. So that one I recommend. What else is good? I mean, good friends of mine, Jay Lamar and Martin Grotendors are getting one out on agents pretty soon. So that's one to definitely check out.
Jon Krohn:	01:01:52	It's pretty funny that, am I correct in thinking this when you were giving the Spanish, the way that someone would say something in Spanish, in Spanish from Spain, would they say weapons of math destruction? They would say weapons of math destruction. Weapons of math.
Luis Serrano:	01:02:08	I think so. Yeah. Oh, that's the perfect way to teach the Spanish set. If you say math versus math, that's how you would say the Zet and the C in Spain, whereas in Latin America, we just say S, Set and C in the same ways.
Jon Krohn:	01:02:28	Yeah, yeah, yeah. Really cool. That's funny. Good to have a laugh with you again. If people want to follow you after this show, which I encourage them to do, obviously we know they can go to serrano.academy. Is there anywhere else they should follow you on social media or something like that?
Luis Serrano:	01:02:45	Yeah, the pageserano.academy, the channel, Serrano Academy. The YouTube channel is where I put all the stuff I know goes there. LinkedIn, I'm also quite active on

LinkedIn. So I'm just Luis Serrano on LinkedIn. Those are the main places.

Jon Krohn: 01:03:00 Perfect. We'll have links to all of that in the show notes. Luis, I always enjoy spending time with you. You are a treat, so friendly, so fun, so intelligent, and you tell it like it is. So I hope we'll have the honor of having you back on the show again sometime soon. Thank you so much for taking the time out of all the work that you're doing these days to join us.

Luis Serrano: 01:03:20 Thank you so much, Jon. I'm a great fan of your podcast and you're a wonderful person to talk to now. And I know that in person and in virtually, so I definitely look forward to our next conversations. So thank you so much for having me. So it's a treat for me to be here.

Jon Krohn: 01:03:35 What a fun episode with Luis Serrano. In it, he detailed his new paper on the curved space time of transformers, how his team recreated Eddington's famous 1919 eclipse experiment inside a transformer, tracing the word bank through every layer of the network and watching its path curve around the word river. He talked about the crucial distinction between an LLM workflow where you tell the model exactly what to do and an agent where the model itself decides what it needs to do next. He talked about why agent evaluation is so much harder than evaluating a chatbot, his elegant explanation of why GRPO powers reasoning models for a probabilistic model. Talking is easy, but hard to evaluate while math is hard but easy to evaluate. And he talked about why as an international math Olympiad metalist himself, he's thrilled rather than threatened by AI hitting gold medal standard on the IMO, comparing it to how we still watch humans race even though cars are faster.

01:04:32 As always, you can get all the show notes, including the transcript of this episode, the video recording, any materials mentioned on the show, the URLs for Luis's

social media profiles as well as my own at superdatascience.com/1025.

01:04:50 Thanks to everyone on the SuperDataScience podcast team for another great episode. We've got our podcast manager, Sonja Brajovic, media editor, Mario Pombo, our partnerships team Natalie Ziajski, our researcher, Serg Masís and our founder Kirill Eremenko. Thanks to all of them for producing such a fascinating and fun episode for us today. For enabling that super team to create this free podcast for you, we are deeply grateful to our sponsors. You can support this show by checking out our sponsor's links, which you can find in the show notes. And if you'd ever like to sponsor an episode yourself, you can get the details on how by making your way to Jonkrohn.com/podcast. Otherwise, please help us out by sharing this episode with other nerds that would enjoy this deep dive that Louise took us on. Review this podcast on your favorite podcasting platform or the YouTube video that you're watching.

01:05:46 If you write in an Apple podcast review that is particularly helpful to us and I will eventually actually read it on the show if you do that. Subscribe obviously if you're not already a subscriber, but most importantly, I hope you'll just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Till 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.