SDS PODCAST EPISODE 938: FRONTIER AI AGENTS FOR DATA SCIENCE, WITH SPHINX'S ROHAN KODIALAM

Jon Krohn:	00:00	Wouldn't it be amazing to have an AI agent working alongside you in Jupyter Notebooks, assisting you by automatically and fluently reasoning about data? Welcome to episode number 938 of the SuperDataScience podcast. I'm your host, Jon Krohn. Today's guest is Rohan Kodialam, co-founder and CEO of Sphinx, a startup that has raised $9.5 million in venture capital to finally bring to data science and data analysis. The same kind of agentic capabilities we've come to expect from LLMs with natural language and code. Rohan is an outstanding speaker building a revolutionary AI product. I'm confident you'll enjoy this one. Rohan, welcome to the SuperDataScience podcast. It's great to have you here. We are recording live in person at the beautiful Bessemer Venture Partners office in New York. People watching the video version can see the New York Public Library in Bryan Park at Sunset. It is a beautiful thing to see. Rohan, you spent years leading data science at prominent institutions like Citadel. What's the problem you noticed in data science that you're solving with your new startups fx?
Rohan Kodialam:	01:08	Yeah, absolutely. So thanks for having me. First of all, really in my time working on data, I think the most evident problem we found was that data and software engineering are often confused. They seem similar at first glance, both just writing code to the lay person. But it's almost like the difference between writing a poem and writing a technical paper. They're both English, they're very different from each other when working on data. I think the intuition that one needs to build is actually on the data itself. It's not necessarily on a code base or on any kind of well-documented knowledge, but rather on whatever information is encapsulated within the structure of that data. And that's really what syns with solving. We're trying to build an AI layer that can understand data at the same level of intuition as say a quant or a data scientist, and then deploy that

understanding of data, that treatment of data as its own modality, as a tool for AI models to then be able to genetically do data science work, to do quantitative research work to basically help people go from raw information to insights very quickly without making the kind of mistakes that we see, say the Claude Codes of the world doing when they're taken out of their natural regime of doing software engineering and just kind of slapped onto data kind of ad hoc.

Jon Krohn: 02:18 So is it in these kinds of age agentic processes applied to data science where you see a lot of failures or where are you seeing failures in AI being applied to data science and how does HINX mitigate those failures?

Rohan Kodialam: 02:30 Yeah, absolutely. So I'll start with a very simple example. If you imagine a linear aggression, which I think is the most basic thing one can do in data science, if your data's clean and you ask any AI model you want to make a linear aggression, it will probably work. If you imagine even the slightest deviation from the ideal state, you have some outliers, some of the data is invalid, it's not actually linear and it's kind of a different kind of shape, and you toss even a frontier coding agent at it, you tend to get very variable answers. Sometimes it's right, often it's wrong. Often it just kind of does some action. And what you'll find is that these models and AI in general are thinking about this is a coding problem. So the task is write some code, you write the code, the code runs successfully and produces an R squared or correlation or whatever it is, and mission accomplished. The failure I'm seeing here is that you're not actually interpreting the data at all, right? You're never looking at the data trying to understand the data, and this is just a mode of thinking that doesn't work. Human data scientists wouldn't act that way or they wouldn't get very far if they acted that way. And we want to kind of bridge that gap with sys technology.

Jon Krohn: 03:36 Nice. I like that. And so I understand as part of this, there was training of, or yeah. Is there training of bespoke models or is it about getting the context right?

Rohan Kodialam: 03:52 Yeah, yeah, absolutely. So right off the bat, we operate on data as a modality. That's our bread and butter. We don't want to compete with certain large players on text as a modality or images as a modality. They're very good at that, and we want to be able to leverage their advances as part of our product. So we only build a representation learning layer for data. So how do you turn data into context? And then if it's something else that's outside of data, we will rely on frontier models. I'll make this visual for you. I don't know if the camera can see it, but certainly people here can see it. So this is I think Walmart stock price over the last five years. I have it on 80 pages of just junk. And this is how your L LM is going to interpret data today.

04:33 It's just going to see a bunch of numbers. It's going to read these numbers as taxed and hope to make some sense of it right. Now, you as a human can probably say, okay, I'm not going to do that. Instead going to use a candlestick chart, right? I'm going to look at something like this. And by looking at this, instead of this stack of paper, you're going to immediately see this thing went up, it came down, you understand the trend, you understand the variation, you understand a whole bunch of information just by seeing this. This is what we're trying to do for ai. So you have data, you can encode it as text. That's what models do now. And then once you do that, you get barely any intuition from it. On the flip side, as a human, you can encode it with a variety of structures.

05:08 There's a whole slew of ways to visualize quantitative information. And then you as a human look at that and you in your mind can then do inference to understand that information. AI doesn't work the exact same way. It's

not great at interpreting things like say a scatter plot or a chart. It usually gets some very mixed signals from it. If you want to try, go make a scatter plot, put into chat GBT and say, what is the correlation of these points? And you'll get something kind of vague usually. But what we're finding is that our technology can actually help you contextualize data in a way that AI can understand. And then with that context combined with other people's innovations in terms of understanding code, understanding natural language, we're able to do data science much more effectively than just out of the box software agent.

Jon Krohn: 05:50 Nice. I love that. And so the analogy just to kind of repeat it and also maybe go into a tiny bit more detail for people listening in an audio only format to a podcast or even watching it at home, and this analogy is so perfect, Rohan, because it's so easy for me to understand even describe in audio because yeah, Amazon share price, if you represent it as text, it's just 80 pages where, so it's the closing price and the opening price on a bunch of days over a five-year period, and it creates an 80 page stack of text, and you can just imagine how easily that would fail as data to be interpreted by model, whereas those same data represented on a candlestick chart, on a plot

Rohan Kodialam: 06:39 Exactly

Jon Krohn: 06:39 On a line plot, basically for people who don't know the finance kind of candlestick look, and it is obvious. It is so much easier to understand. And so that makes a lot of sense. I can see why there's such an opportunity for you at Sphinx. So we kind of understand the value of what you're doing. What is the experience like for users? So if I'm a data scientist or I'm an AI engineer or a data analyst and I'm thinking, wow, this sounds great. I wish I had an LLM or I wish I had a product hinx that I could be using

data with, just like I can use natural language with one of the Frontier labs. What's that experience for users in Hinx?

Rohan Kodialam: 07:21 Yeah, yeah, absolutely. So we think Endang said this too, right? You want your LLMs or your agents to be able to have very varied levels of authenticity. So we adopt that philosophy. So for a user of syncs, you can go from, I don't want to make this plot, or I don't want to interpret this data, just do this one thing for me, all the way up to much more agentic flows like, here's a problem, here's my data warehouse, go solve it. We offer people that full range of experiences, and we also believe that AI models should fundamentally be highly configurable in natural language. So as a user, when you onboard the things we can onboard you in like five minutes and then you're on the product, you can really tell it to do whatever you want. Most of our users start small. They'll be say in a Jupiter notebook and they'll come across too annoying tables that they don't want to join. They say, okay, thanks. Can you join them for me? Andy will figure out, am I

Jon Krohn: 08:10 Understanding? So when you're in the Jupiter notebook and you say that, how are you doing it? You type it as a command, you

Rohan Kodialam: 08:16 Just type it in. It's a very, very familiar interface to anyone who's used any AI coding tools. You type it in as a command, and then that command gets contextualized. We figure out what data we need to answer it, whether it's already in your kernel, whether it's in your data warehouse, whatever it is. We'll go find it, get it, put it through our representation. Learning machinery. Once we understand the data, it's usually relatively obvious to say, oh, okay, you have these two columns that probably mesh together. Here's how we transform them. What's missing? Here's what I got to impute. It'll do that for you,

and it all runs on your side. So the other aspect for data is most people think of data as a crown jewel of their company or even of their own personal work. So we don't want to take anyone's data.

08:52 We actively don't do that. So the way Hinx runs is it figures out what to do and it runs it on your computer or your server or your kind of compute environment. So once FX figures out what to do, it executes on your side. You have a result in your environment with code you can run and then you can proceed from there. That's how people start. Once you see it work a few times and you're like, oh, it actually does work, and people need to do that because you've tried using cursor or cloud code to do it and it doesn't, so you don't really believe me at first, but then it works three times, four times and you're like, oh, maybe I can just help you the whole thing. And then you start to graduate to much more agentic flows, and that's really our happy path for users.

Jon Krohn: 09:28 I like that. And so you mentioned Jupyter Notebooks there, and so does Spinx operate inside of a Jupiter notebook or it feels like a Jupiter notebook?

Rohan Kodialam: 09:37 Yeah. Yeah. This is a great question. So there are two relationships we have with the Jupiter notebook or the kind of interactive computing more broadly. The first more obvious one is that we use that as our choice of frontend data. Scientists are super familiar with it. It's kind of like the defacto standard. And so when we want to expose a way for a human to inspect SSA's work, to change shin's work to put in their own code, if they decide, I'm just going to do this myself manually because I have some strong bias and how it's supposed to be done, the Jupiter notebook is the right format to capture that in a way that's familiar but also quite powerful. The second deeper way is that I think a lot of coding agents

today kind of think of the bash terminal as their home base. They're running commands.

10:17 If you want to read something ULS the directory and you go cat the file, we actually use the Python kernel as our kind of home base for our agent. The reason for that is we are manipulating data so much that we want something as representative as Python to be able to take objects that live in memory ephemerally and transform them into something that our models can use. So because we're operating on the I Python kernel as kind of our fundamental building block of agentic steps, it's actually incredibly easy for us to expose Jupiter as the interface. So it's like a happy coincidence. So we can give people an interface they're familiar with that works almost anywhere with any type of compute, with any type of data, but also naturally meshes with how the agent is thinking about the problem internally.

Jon Krohn: 10:56 I love it. So then when I am in a Jupyter Notebook, I'm used to kind of having a markdown cell or a code cell. So is there a third type of cell that's like a S swing cell or

Rohan Kodialam: 11:07 We operate in markdown cells and code cells, so we don't want to build something that you're not familiar with. At the end of the day, we do this inference, we figure out the right way to do it, but the way it's implemented has to run on your system. We want it to be portable. We want it to be understandable, auditable, even if you trust us, you should always have the ability to go audit what we've done. So we operate in code cells, right? Sql, python, markdown, right?

Jon Krohn: 11:33 So you're in the code cell and then you just start writing a natural language.

Rohan Kodialam: 11:36 Yeah, we not exactly from the interface perspective, we want to kind of have syncs as a separate chat. So think of

it almost more like cloud code where you can ask sinks to take actions. The level of abstraction at which syncs operates is more at the action level than we're going to write one line of code for you. Because at the end of the day, data science, the code is just a means to an end. I see. You want to accomplish something with your data, you tell us what you want to accomplish and we'll go help you do it.

Jon Krohn: 12:02 So it's alongside me there as I code or

Rohan Kodialam: 12:05 Exactly.

Jon Krohn: 12:06 I see.

Rohan Kodialam: 12:06 That's right. I

Jon Krohn: 12:06 See. Perfect. Now I understand. Thank you. Thank you, Ryan. Alright, so you only founded the company this year, but I understand you've already been making some impact for clients. I also understand that those clients must remain anonymous, but using some anonymity, some obfuscation, I'd love to hear just one or two use cases of how phis has already made an impact for your clients.

Rohan Kodialam: 12:30 Okay, so most of our users already given how young the company is, are people who already value data a lot. So these are people as you can imagine, who have data teams invest a lot in their data teams and want their data teams to be successful. Honestly, it's not that exciting. It's not that revolutionary where we're doing that. We're taking something they want, we're making them five times faster at it and they're happy. Cool. What I think is more interesting, and where I see the kind of future trajectory of the company going longterm is in spaces where it's not like you have a huge data team and you're just starting to think about data as a concept. And so when you do

something like that, we have example. One of our early customers is in the CPG space and you see actually very transformative effects where they actually are now seeing, Hey, we should hire more data scientists because each individual data scientist can do so much more work and they have transformed part of our businesses and they're actually adding value.

13:21 And so that's really what we want to see. We want to see data science becoming part of the DNA of every institution, and the more value we're adding is in cases where it's not already the case and they realize we're sitting on a pile of information, we can monetize it. And so that's been quite transformative for us, seeing how the actual dynamics of the data team change. Where some people obviously are like, oh, do you need data science? And we're like, of course you need data scientist, the ones who are asking the right questions. And in fact, each data scientist can do so much more, and that profession just becomes more valuable with the right toolkit.

Jon Krohn: 13:53 So this is a common concern that people have over AI is that it replaces people in roles and of course some specific functions in roles end up being replaced. So there's very little reason today to be typing out every character of code that you write or that you use. And so then somebody might think, oh, well then maybe we only need 10 data scientists instead of 30 because they don't need to be doing all the coding. But you make a really great point there, which is it's actually, this is what we've seen with every automation over the past 200 years

Rohan Kodialam: 14:26 Exactly,

Jon Krohn: 14:27 Is that it actually creates more jobs because it allows people to be creating so much more value. You're sitting on top of more abstractions, you're able to work more rapidly, and each data scientist that CPG company hires

is now providing more ROI as opposed to being caught up in data ops or ML

Rohan Kodialam: 14:48 Ops

Jon Krohn: 14:49 Struggling with some simple low level coding.

Rohan Kodialam: 14:52 That's absolutely right. And we see this as data as a super unsaturated space. There are so many companies which have a lot of data are not monetizing at all. You see these Harvard Business School statistics of like 80% of CEOs say data is a top priority. 20% of CEOs actually invest in data. Okay, there's clearly some problem here. Maybe it's too expensive, maybe it's too complicated, but syncs lowers all those barriers and lets data teams actually deliver to their full potential. Then that's why we see for some of our early customers, their data teams have doubled and that's great. So amazing to see that.

Jon Krohn: 15:24 Very cool. Nice soundbites in there as well. Alright. All right. So Hinx obviously sounds like a fantastic product. How can people here sitting at Bessemer Venture Partners in real life or our listeners at home, how can they access Sphinx and is there a free tier?

Rohan Kodialam: 15:42 Yes. So there absolutely is a free tier. So our website is sphinx.ai. You can get our free tier there. It's a giant button on the top, syncs runs on basically whatever compute you want, it runs on whatever database you want. If you don't have data, it will run on CSVs and things like that too. If you're going to use it at home on a pet project, we sit on top of Jupiter as our interface so that it's quite familiar to anyone who works with data to just jump in and start working. And really, again, we believe deeply that things should be configured in natural language. So that means you don't need to do any setup, you don't need to do any integrations. You just download the thing, sign up for an account, off you go. Our feature

is pretty generous. You can generally do, you can do almost whatever you want. You can do probably 10 or 20 analysis before it runs out. And then if you are doing something sufficiently interesting with your free tier, please just reach out to me, we'll just give you more credits. We are much more interested in seeing people do cool things than in nickel and diming anyone.

Jon Krohn: 16:34 Nice. That sounds great, Rohan, thank you for offering that. So that is really the end of my technical questions for you, but as regular listeners to this podcast, well know not necessarily everyone here at Bessemer today, but I always ask my guests for a book recommendation. What do you have for us Rohan?

Rohan Kodialam: 16:53 Okay, so I'm going to give you a book that is very related to Spanx, maybe not so fun, but I obviously spend most of my time working on Spanx. There's a book called, and I'm sure someone's recommended this before on your podcast. It's called The Visual Display of Quantitative Information. Tuft. Yes, that's the one. I like that book a lot because it kind of goes through the history of how humans in their minds built the equivalent of Sphinx. Everything from the famous graph of Napoleon's army in Russia dwindling down to almost no one to John Snow's plot of cholera in London, which by the way, if you read our blog, you'll find that AI cannot replicate that analysis. So another example of how it's about a data science, but regardless of that, there's just this whole history of hundreds of years of humans trying to figure out data is big, data is complicated, how do we stuff it into our heads? It's kind of inspirational to see how people have done that and the kind of work that's come out of it, whether it's analytical or public health outcomes or other kinds of outcomes that are beneficial to the community. And so it's a super interesting book. Definitely explains what Hinx is, but also it's just a fun read and it has a lot

of pretty pictures, so it's a easy reading if you're also coding 18 hours a day.

Jon Krohn: 17:57 Fantastic. Thanks Rohan. So yes, fx.ai for hinx, and then for following you for getting some more of your insights. Where should people follow you?

Rohan Kodialam: 18:05 Yeah, so I'm on Cody Ro on x. I'm also on LinkedIn, of course, most of my content is AI data science related, as you might imagine. So if you're interested in that, definitely take a look. Sync's posts, reasonably interesting blog posts pretty often, so would love to have you read them. And yeah, we are always looking for comments from the data community. We build this for the data community. We're from the data community. Most of our team have experience working in data science or in quantitative research, so we would love to hear from you, especially criticisms that's much more interesting and useful for our team than anything else. Yeah,

Jon Krohn: 18:39 Nice. Thank you. What an exceptional episode with Rohan Kodialam who's revolutionizing and accelerating data science with Sphinx ai. I hope you enjoyed the conversation to be sure not to miss any of our exciting upcoming episodes. Subscribe to this podcast if you haven't already, but most importantly, I just hope you'll keep on listening. Until next time, keep on rocking it out there, and I'm looking forward to enjoying another round of the SuperDataScience podcast with you very soon.