SDS PODCAST EPISODE 1047: NVIDIA Kumo: FOUNDATION MODELS FOR TABULAR DATA, WITH STANFORD PROF. JURE LESKOVEC

Jon Krohn:	00:00:00	What if a foundation model that has never seen a row of real data could beat your hand-built models with no feature engineering and with no training? Welcome to another episode of the Super Data Science Podcast. I'm your host, Jon Krohn. Today's guest is the legendary Dr. Jure Leskovec, a Stanford computer science professor who pioneered the field of graph neural networks. He also co-founded Kumo AI, which Nvidia recently acquired, and he now works there. And one of the things that Nvidia and Jure have recently done is they've released all of Kumo's foundation models open source. In this episode, Jure explains how these models make predictions across enterprise tables and databases, how his Stanford Group's AI co-scientist won over 10,000 scientists within weeks, and why he believes human judgment has never been more valuable. Enjoy this sensational technical episode. This episode of Super Data Science is made possible by Anthropic, Parallel, and Why Karat?
	00:00:55	Jure Leskovec, welcome to the Super Data Science Podcast. What a treat to have you on the show. How are you doing?
Jure Leskovec:	00:01:00	Great. Yeah, excited to be here.
Jon Krohn:	00:01:02	Yeah. You're a big wig that we've wanted to have on the show for a long time, so delighted that it all came together. You are a Stanford computer science professor and you pioneered the field of graph neural networks. We're going to talk about that, I mean, at points throughout the episode, but I also have some targeted questions about graph neural networks coming up. The thing that I want to focus on first though is that you were co-founder of a business called Kumo AI, K-U-M-O, and that had a big series A, 18 and a half million dollars, and then it was acquired by Nvidia. And so tell us about that journey a bit, and then I'm going to ask
Jure Leskovec:	00:01:38	You

Jon Krohn:	00:01:38	About the open weight releases of all these Nvidia Kumo models next.
Jure Leskovec:	00:01:42	Yeah, great. Excited to be here and give the story. So I'm known as a graph guy and we actually founded Kumo as a company from the inside that it seemed like that there was this huge excitement around AI on unstructured data, but these enterprises are collecting huge amounts of structured data, data in operational data, in data warehouses, databases and so on. And this type of data comes in a form of tables with rows and columns, and then the tables are connected through primary foreign key relations, a customer ID and a transaction maybe points to the customer and things like that. So that's this most valuable operational data that organizations have, but it seemed that that got completely missed, let's say by the AI revolution. So the reason why we founded the company was to bring AI to this structured data world. I can explain later how and why.
	00:02:51	And then we got acquired by NVIDIA about, I know at the beginning of summer, let's say this way, with the goal to basically take our technology and release it to the world as a set of open source, open weight models, libraries, training recipes with a super permissive open source license that allows free commercial use and so on to basically bring it to the community and lift up the data science, machine learning, structured data reasoning ecosystem.
Jon Krohn:	00:03:29	Yeah, interestingly, so the next week's episode, the one that airs after your episode is going to be with Gal Veroco, who also is a big specialist in relational data. You must know him. And in that episode, I know because we've recorded it before we recorded yours, that we talk about how the only truly open weight or open source projects in this structured data space are the models from them and the models from you guys, from Kumo.

Jure Leskovec: 00:04:03 Yeah, I think the idea here is to really support the ecosystem and provide basically models that are the easiest for the enterprises and everyone else to put in production, play with them, experiment with them and so on. So from that point of view, the more permissive, the more open it is, the easier it is for others to build upon and work on the top of it, the more we can help the ecosystem. And I think maybe one thing to explain is NVIDIA in some sense is a hardware, is a chip company, but what enables NVIDIA is this super rich, open, broad software ecosystem on top. So our effort in a sense is part of this broad open software ecosystem on top that is supported by the hardware, by the chips that the company is developing. So that's kind of the connection and in some sense why we are doing this because we want very good, let's say accelerated computing ecosystem and structured data is integral part of that and AI unstructured data is an important part of that.

Jon Krohn: 00:05:21 From CUDA to PyTorch to newer or more specialized things like you guys have been releasing with Kumo, that provides a lot of firepower for GPUs to be powering. Let's talk about the specific Nvidia Kumo models that have now been released. They were released at the end of October at the big Nvidia GTC conference in Berlin. And I've got it here that there are kind of four different kinds of models that were released as part of the Kumo structured collection. So there's Kumo tabular, which predicts outcomes from a single table like we were kind of just talking about. Kumo relational, which makes predictions across connected tables, which sounds pretty cool. We're going to have to dig into that. Kumo forecast for time series forecasting and Kumo anomaly for exploring unusual behavior and time series data. I guess we should start with Kumo tabular since we've already been talking about tabular data.

Jure Leskovec: 00:06:18 Yeah, I think Kumo tabular is the easiest to explain. Basically the problem is you have a table with some

business enterprise data and there is a target column that you would like to build a model to predict that target column. That target column could be a probability for a customer to churn, probability of a transaction being fraudulent, the forecast about the revenue forecast for next quarter, maybe we are imputing some data about users that is missing. All these can be kind of formulated in this tabular use case where you have a set of columns that are given and from the set of columns that are given, you want to kind of predict or impute a column that is not given. And the way these foundation models work is actually quite remarkable. The way we would solve this problem today is that we would train a machine learning model, I don't know, as some kind of three-based model or a neural network or a regression model or whatever the thing is that takes the set of given columns to predict let's call it the target column.

00:07:36 And we would need to feature engineer and build the model and train the model and tune the model and then we could apply it to the new unseen data and make those predictions. Foundation models are fundamentally different in this respect. What they get is basically they get this historical labeled data as part of the input. They get the unlabeled examples as well as part of the input and then in a single forward pass of the architecture without any training, without any gradient updates, they give you those predictions. So commotabular is a family of purpose-built foundation models that are, and I can touch on that a bit, are kind of analogs to LLMs, but built to reason our structured data to give you accurate predictions. And on many benchmarks, capital competitions and so on, these models are getting more accurate than your traditional ML data science, purposefully built with sweat and tears kind of models.

00:08:41 So the idea is -

Jon Krohn: 00:08:43 The better lesson.

Jure Leskovec: 00:08:44 Yeah. We've all been there, right? It's very hard. It's very tedious to do, but now we have a pre-trained model that is basically data agnostic, domain agnostic, table structure agnostic that allows you basically to do this in context learning or predictions on the fly. Maybe the next question you could ask, okay, how is this different from large language models? Because you could textify a table and just throw those rows in and then say, "Hey, what do you think? Is this fraud or not?" And tell me. And it turns out that that doesn't really work. Even JAV-like or these decision LLM models don't really perform well. They're able to extract knowledge from their background, do well in zero shot, but as you give it more examples of what fraud is and what fraud is not, they are not able to improve their performance. While these, let's call them tabular foundation models, have been built purposefully to do this in context learning.

00:09:49 So you give it a set of examples, you say, "Here are examples of fraud. Here's examples of not fraud. Here's unlabeled, unlabeled row. Tell me what's the probability of fraud." And in a single forward pass of this neural network architecture kind of transformer based, you will get a well calibrated probability estimate out. So you truly get a statistically quantifiable, non-hallucinated in that respect probability.

Jon Krohn: 00:10:21 Yeah. Help me understand a little bit more what's going on under the hood. You gave a pretty good example there. I'm undersstanding that it's transformer based, it's pre-trained based on similar kinds of business data. If I were to do this in say quad, if I had a table, what I would be trying to do, I probably wouldn't be trying to convert the table into tokens for ingesting into the LLM as input tokens. What I would probably do is I would say something like, "Here's a table of data. I might have some understanding, some context that I can provide around what the columns mean, but it could also maybe be able to figure it out based on context." And then I would say

create a model that you think would be appropriate. And so it could go off and maybe create a regression model and iteratively add in features or move features, maybe adjust some of my features, normalize them, these kinds of things.

00:11:15 And so that's kind of like my small-minded, old school, all the way back in 2024 kind of thinking with how an LLM might be able to help me with analyzing tabular data. From everything you've said so far, it sounds like it's something completely different where the data are going through the foundation model. Yeah. So I guess maybe just maybe go one level deeper on what that's like, on what's happening behind the scenes when you use cumotabular or a model like that for analyzing your tables.

Jure Leskovec: 00:11:51 Yeah. And thanks for providing this contrast because I think it's very important, right? What you are talking in a sense is let's have a harness that automates the data scientist. What commotabular is something complementary, complementary in a sense that it's a neural network architecture through which you pass the data. So this neural network architecture is, as we said, it's transformer based, it's pre-trained, but the attention mechanism is different. So you have two types of specialized attention mechanisms. One is the attention mechanism across the column. So when you have a particular cell of interest, the attention mechanism attends to other values inside the same column to get an understanding of where does this value in a given cell sit with respect to other values of that same column, right? Is it at the extreme? Is it somewhere in the middle? And things like that. So that's the first type of attention you do.

00:12:54 And then the second type of attention you do is you do attention across the columns in a single row. So it's all the cells in a single row so that that attention mechanism

can understand and learn the dependencies between these different columns. And through basically stacking these two attention mechanisms and interleaving them, you are now basically able to pre-train a neural network that you could think of it as builds a model in its brain while doing inference to be able to give you a proper prediction. Of course, you can still go back to your agent harness doing feature engineering and all that. You can still do all that with this foundation model-like approach to basically have an outside harness that is trying to improve the inputs and the outputs of the foundation model. So the harness does not exclude the underlying foundation model, but these are two very different things.

00:13:58 One thing that's important, maybe we'll come to it, I assume later is that these tabular models, foundation models are trained purely on synthetic artificial data and that's important. And then maybe another important thing is in terms of the size of the context, the amount of data you need to attend over, we are attending, for example, over 500 million cells. So if you were to tokenize this and throw this into an LLM, you exceed the context limit far before you were able to textify or throw in 500 million cells worth of tokens. Because cells can have text, can have... There's a lot of different data that can be hidden in those cells. But these purpose-built models have this scalable attention infrastructure that allows you to attend over humongous tables.

Jon Krohn: 00:14:53 Yeah. It sounds like orders of magnitude more data than you could get with most of the cutting edge with the frontier LLMs. That is pretty interesting indeed. Tell us more about the synthetic data since that is such an important point. That's probably helpful from a privacy perspective. It also probably enables you to handle lots more edge cases that would have rare data. It allows you to continue scaling. You can just keep creating more and more data and improve your model more and more.

Jure Leskovec: 00:15:20 Yeah, those are great questions. So maybe before I get there, so you have the models. These models are relatively small. So we are compared to LLMs, right? Our largest model I think is 200 million parameters. So they are small, they're super efficient, they are super fast. We have the commotabular, which is the largest 200 million, and then we have also a medium and we have small size model that is like 60 million parameters. So these are small, in some sense, in this respect, small, efficient, compact models. Now, how are they being trained? They are trained on purely, I would say, artificial synthetic data. And why am I saying artificial? Because they are trained kind of on the data that is truly generated from scratch and kind of has no connection to the real world. And the way this data generation mechanism works is through this formalism of structured causal models where basically the way you can think of it is that you are randomly or you are instantiating for every column, you can think of it as a different variable.

00:16:37 And now you are instantiating a graphical model, like a structured causal model. How do these different columns relate to each other? And you generate that causal structure, and then you initiate the causal structure with random noise and see how the class variable pops out. And that's essentially what the data generation mechanism is. So now you can generate a different number of columns, you generate, let's say, random causal dependency structures, and then you push the data through to get the dependent variable out. And that's how this data is being generated. In reality, it's more interesting because you have different data distributions, you have different dependency structures, different functions that how parents affect the children variable. You can start playing with all kind of noise type things. You can start teaching the model to deal with missing data. So for example, one good feature is that we actually teach the model to automatically impute the data.

00:17:48 So the input to the model is not just a row with the columns given, but it's also you can do basically not available. You can say, "Okay, we don't know what is here." And then the model actually learns how to impute that without the data scientist needing to worry about that or to worry about extreme values and things like that. So the synthetic data generation mechanism allows you to be very strategic about what you want the model to learn, and you can basically engineer those capabilities inside the data, where with the real data, you don't have that possibility. The real data is given, it's static. We don't know how to make it more missing or less missing. We don't know how to make it more noisy or less noisy or the prediction to be more complicated or less complicated. We don't have those levers. But if you generate this synthetic data, you as a model developer, you have all those levers.

00:18:47 As we are training model through different stages, we can introduce different types of tables, different shapes of tables with different missing structures, with different heavy tailed extreme value type distributions and so on. So that flexibility turns out to be super useful because yeah, as you said, we can generate more and more data. We can keep scaling up these models as much as we like.

Jon Krohn: 00:19:18 Yeah. Getting all of these different kinds of permutations right around what kinds of simulations you're going to need must be critical, a big part of the secret sauce to having this work effectively. And then I guess you can kind of evaluate based on how they perform in say the Kegel competitions you mentioned earlier with real world data. So all these simulations, downstream you can say, okay, I know that even though I only worked with simulated data throughout my training process, we know that it applies to the real world because when we test it on real world data, it hits it out of the park.

Jure Leskovec: 00:19:48 That is an excellent point. One thing that is beneficial from having trained this with synthetic data is that

prediction is all about generalization. It doesn't matter. Making decisions about the past is easy. Making decisions about the future is much harder. When we use LLMs, we never know, is it generalization? Has it just read this information somewhere? Has it memorized this thing? We never really trust it, whether it's doing some real kind of inference or is it just kind of recalling from memory? But for this tabular models, because it's purely synthetic data trained, there is no real data in there, so there is no contamination. So when we apply this to any real world data set, we can guarantee that that real world data set is out of sample. The model has never seen it in any training, has never seen those numbers, has never seen those specific patterns that actually occur in the real world.

00:21:03 So that's great because then our, let's say accuracy estimates and so on either from Kegel competitions or things like that are true and real and they are really kind of out of sample and that's great. And on the other hand, I think what is also amazing and impressive is that through the synthetic data, you can actually generalize to the dependencies and the data shapes and structures and noise patterns and so on from the real world. I mean, maybe just to say for people who wonder, the way you evaluate is kind of as in classical machine learning or is in classical AI, you basically have this holdout, you make predictions, and then you compare them how close they are to the truth, to the reality.

Jon Krohn: 00:21:51 Something else that's interesting when comparing to say classical machine learning or classical AI approaches is that if we were to use say a simple logistic regression model or a linear regression model to try to explain how some features in a table are predicting some label in a table, the way that we would historically have come up with our confidence about the predictions would be using statistical approaches, like using beta weights, divided by variance, these kinds of things. And so it's kind of

interesting here where we are relying on a neural network to... I don't want to say magically, but I think you know what I mean when it kind of just based on all the training data, the neural network comes up with reasonable estimates that obviously hold up when tested with real world evaluations.

Jure Leskovec: 00:22:46 That I think is a good point and it's kind of the remarkable thing that happens here is that through this basically generating these random causal graphs, sampling, filling them with noise, getting the dependencies out, putting that into a big table and then telling the neural network to kind of learn the patterns that it's able to generalize so much that it allows us to transfer, again, with no post-training, with no fine tuning, nothing like that to a brand new data set, to a brand new tabular structure, to a brand new domain, being pre-trained only on synthetic data. What is also maybe a few things that are important to say, the first thing is that these models, they don't require any semantics. So they don't require column names, understanding what the columns are. It's like in classical machine learning or in a decision tree, you just care about the value and I know how to split on that variable.

00:23:46 So in that respect, it's about this pattern matching more than some kind of semantics. And then what is also important is that the training objective is not the next token prediction. There is no next token. It's really the logic, it's the output. It's a cross-entropy type loss. So the model is actually trained to give this accurate, let's say predictions over underlying numerical, categorical, textual data. And what we see is that basically the model learns to be very well calibrated, if you create these empirical QQ plots and things like that to understand the calibration of the model. We see that those probability estimates outputted by the model are nicely correct. And of course, what is also interesting is because you give it this set of in-context examples, you basically have this

mini labeled data set that you can use to almost do on the fly cross validation and things like that to get not only the probability estimate, but also uncertainty or kind of sense of the accuracy of the model of the prediction it's making.

00:25:05 So you always know what you are getting out. The model will give you the certainty or uncertainty as well as the estimate in its own accuracy.

Jon Krohn: 00:25:15 It's very cool stuff. What a world we live in these days. I couldn't have imagined these kinds of things. You wouldn't know this about me because there's no reason why you would know anything about me, but I did a PhD focused on machine learning and AI, finished that in 2012. And this kind of stuff that we're doing today, it's like I would never have expected in my life to be having neural networks doing this kind of work with tabular data.

Jure Leskovec: 00:25:43 It's a very different world today. Yeah, I agree.

Jon Krohn: 00:25:45 For sure. All right. So that's QMO tabular. I said that there were these four open way models and we probably spent more time on Qumotabular than the others that we'll talk about now, but let's talk about Kumo relational next, which can make predictions across multiple different tables. So where Kumo Tabular was designed for working with a single two-dimensional grid of information, a matrix, with Kumo relational, we're able to kind of add a dimension of combining information across connected tables. And so that sounds pretty powerful. And I imagine the simulations, the data simulations must get a lot more complex in that scenario.

Jure Leskovec: 00:26:28 Yeah, I can explain. I think the tabular models are good, but kind of no real world data comes in a single table, right? And usually the data scientist has to take this raw multitabular data, relational data, and then feature

engineer aggregate into this one nice training table or table. So we were very interested to say, okay, how could we now generalize these tabular models to just be able to attend over the entire database? And what do I mean by connected tables? You can imagine, I know I have a customer table where for every customer ID, I have a customer ID, I have their first name, last name, address, I don't know, and so on. Then maybe I have a product catalog where again, I have a product ID column and then I have product description and name and brand and weight and price and so on, and that's maybe in my product catalog table.

00:27:22 And then maybe I have another table in between called the transactions table who just says what is the time, what is the customer ID, what is the product ID, and what was the method of payment, for example. And this would mean that now particular user ID, particular customer ID, bought a particular product ID. And now this is a set of three different tables that are interconnected. In the old days, imagine if you wanted to build maybe a customer churn model, you could say, "Okay, I'm just going to predict churn from whatever I know about the customer from their demographics and so on." And that model won't be very accurate because it's not capturing customer behavior. So then how do you bring in customer behavior? You need to take your customer table and your transactions table, join the two, and then aggregate information because now every row in that table is on a transaction level.

00:28:22 You have to aggregate transactions to get some feature, some variable that is per customer. So you could say, "Let's count the transactions. Let's maybe count them by the hour of the day. Let's count them based on the weekday. Is it over the weekend? Is it on a holiday?" And that's only the count. And then it could be, "Okay, let's consider the price. Let's consider the method of payment and things like that." And this feature engineering step

gets super manual and super kind of arbitrary in a sense because the space you are exploring is super huge. So our idea was to say, "Can we take this super successful attention is all you need, and can we find an attention mechanism that allows me to attend across tables that will just automatically grow from the user and just attend over all their transactions?" And it won't stop there.

00:29:19 It can go a step further and now attend over all the products that were part of those transactions and all their properties and images and descriptions and so on. And that's what relational models are doing. They are learning directly on top of the database where you don't even have to do feature engineering, but you let the attention mechanism kind of do the attention and aggregation, which is kind of a core part of feature engineering for you. So similarly, these models are now called Kumo relational. They are also fully trained on synthetic data. The data generation process is kind of similar to a single table, but more general because now you first have to generate the schema and dependency between tables. And now you can use the structured causal models to generate the data. And then in terms of attention mechanisms, we talked, you have a column attention, you have a row attention, and now you also need the cross table attention.

00:30:25 And this cross table attention is essentially a graph based attention mechanism. You can think of this schema and the entities inside it as a graph of relationships. So that's the kumo relational.

Jon Krohn: 00:30:41 Cumorlational. Yeah. It kind of brings everything together given your pioneering of the field of graph neural networks. You ended it nicely there talking about chumorelational, how you can think of the relational entities as a graph of these connected items, but it also, it seems like more broadly, whether we're talking about chumotabular or kumo relational, being able to simulate

graphs effectively and realistically is a big part of any of these models working out in the end.

Jure Leskovec: 00:31:13 That's a great point. I think all our prior work on graph generation, scale free networks, small world networks and all that work, plus then the work on graph neural networks that are basically a way to bring this deep learning to the graph structured data now naturally came together in this foundation model era of the world where now essentially the attention mechanism is the attention across the graph. And I think why this is interesting is early on we were just saying, oh, the graph is given. And maybe here the insight is, and it's a well known fact, is that any database can be basically thought of as a graph, of a graph where every entity is a node and then there are this primary foreign key. Those are the relationships between the entities. And the intuition why we started doing this was basically to say, look, the way we built computer vision models today is nobody does feature engineering in computer vision.

00:32:26 Even though in the past they did, there were sift features, Gabor filters, all those things. Nobody in the right mind does that. Today we just say, here's the root pixels, here's the attention mechanism, convolutional operator, vision transformer, whatever it is, and let it figure out how to add these little pixels together to tell me what's in the image. And we were saying, okay, if I can now take any tabular structured database type object, represent it as a graph, I can now start applying the attention mechanism over these graph of nodes and let the neural network learn how to combine this information into a prediction. And I think the effect here is first, you don't have to do the manual feature engineering yourself. And then the second thing is that this neural network combining things is so much more expressive than any manual feature engineered in SQL will ever do.

	00:33:26	You know what I mean? It's very hard to write this as a query and say, now here is a single number. Neurons are this non-discreet continuous type things that can just combine the data in so many different ways that is at the end much better for the final prediction.
Jon Krohn:	00:33:44	Yeah. Very cool. Thanks for taking us that one layer deeper on the graphs and data simulation. Let's move on now to Kumo Forecast and Kumo Anomaly, which are both designed for dealing with time series forecasting. And my understanding is that they're useful for different kinds of scenarios. So Kumo Forecast is useful for, I guess, when we kind of have more balanced events happening or continuous inputs where we have different amounts of ads being... Ad money spend is going up and down kind of gradually different quantifiable levels. So we could do things like demand planning, capacity management with Kumo Forecast. And then Kumo Anomaly is designed for handling situations where you have rare events. So things like operational issues, manufacturing issues, infrastructure issues where 99.999% of the time there's no anomaly and then all of a sudden boom, something weird happens that we need to be notified about right away.
Jure Leskovec:	00:34:52	Yeah, good. It is true that there are these two quite different use cases. One is really about forecasting the future values, the future set of values from one time series or from multidimensional or multiple connected time series. And then many times what we see when working with our partners and collaborators is that they actually would like to also have this notion of for anomaly detection is something abnormal, is something that we should pay extra attention to. And even though anomaly is something that is far away from the predicted state or from the expected state, the two are actually so different that actually they require a bit different approach, a bit different model.

Jon Krohn: 00:35:45 It's interesting that you say different model because it also, it is an interesting decision to have four different open weigh models for different kinds of scenarios. How did you end up coming into that situation? Yo might think maybe Kumo Forecast and Kumu Anomaly, they're both dealing with time series data. We should have them as one model. Or maybe all four of these models, tabular, relational forecast and anomaly are all dealing with tabular data in one way or another. Let's just have one big model.

Jure Leskovec: 00:36:14 There are several different answers I can give to you, right? Yes, we can go and build one big model and that's very doable, but then the model becomes big and big things are expensive to run and slow to run. So whenever you then say, "Oh, I want to put this in production," what you want to have in production is small and fast model because you will use a bit less compute. So you want something maybe a bit smaller and more specialized. On the other hand, there might be some benefit in the big model because it learns from all these different tasks and all these different types of patterns is that the big model will be able to generalize better and so on. So our goal is to really support the entire, I would say, range in this ecosystem so that people who want cutting edge, the biggest, maybe are not making so many predictions can use big things and people who want something that is fast, efficient, easy to run can use smaller versions of the models.

Jon Krohn: 00:37:20 Cool. That makes a lot of sense. That makes a huge amount of sense, especially because the vast majority of our compute costs come at inference time, so we should want to have these smaller models. And it is impressive how small these models are, hundreds of millions. I think your biggest models were like hundreds of millions of parameters, right? Which in this day and age, that's unusual to say that's the biggest one in the family.

Jure Leskovec: 00:37:48 It's a good question. So our biggest is a bit over 200 million parameters. There is no fundamental reason why the model couldn't be better. We felt that it gives kind of the best throughput, latency, computational cost type balance with the accuracy, but we plan to build as the field moves forward also bigger models and further push up the accuracy.

Jon Krohn: 00:38:23 That's cool. I'm not saying that they should be bigger. I think small is great. So let's talk about how these are being used in the real world. At GTC in Berlin with a big announcement, it mentioned that a number of well known names are already using Kuma models. So Databricks, DoorDash, Coinbase, and Snowflake customers are using them in a broad range of areas with clients that have advertising, energy, financial services, retail data. For our listeners, when should they decide to pick up a model like this and use zero shot predictions and be confident that the zero shot predictions provided by Kuma models or models like this are going to be good enough? When could they be confident that zero shot's going to be great out of the box and when should they maybe be thinking about fine tuning, if ever?

Jure Leskovec: 00:39:14 That's a good point. Maybe when you say zero shot, we really mean this kind of in context. You still have a little kind of training label data set, but it all happens kind of at inference time. Yeah, that's a good point. So right now these models can attend over 500 million cells. So if your data is on the smaller side, then no need to fine tune. If it's less than 500 million cells, you can just use these models, especially with smaller amounts of data because these models have been pre-trained, they're so much easier. They will have better generalization properties. They won't kind of overfit and things like that, which is the danger when you start building custom models. What we also see is that when you have a very specific task and a large amount of data, then maybe you just want to specialize the model for that single specific task.

00:40:14 Because the point is, the question is, are you doing this for a very specific task that you know it's a specific question that you'll be kind of always asking? Or is it more like, "Hey, I can apply this to any predictive task, any type of question and just use it immediately." And both have their merits. If it goes for questions that are more like one off data science, something figure something out, just use a foundation model, very quick, super. Now if we are talking about, I know predicting the click-through rate of an ad, which is kind of the most valuable prediction out there because all the web advertising depends on estimating how likely somebody to click on an ad and every 0.01% difference in accuracy means a lot, then there I think you want to explore all kinds of fine tuning, you have huge amount of data and so on.

00:41:12 So I think again, it's kind of the spectrum, depends what you are doing, but for iteration and for understanding, I would definitely start with these foundation models. And today Kuma Tabular is essentially a drop in replacement for XGBoost and all these tabular models, classical machine learning models. And we see all kinds of customers being very successful in just deploying them, ripping out their custom models, throwing in foundation models and getting improvement in accuracy. It doesn't happen always. I think also us getting feedback where are these models lacking is very useful because then we can go to our synthetic data generator and just tell it, Hey, we need more data like this and we can generate infinite amounts of that type of data to improve the model. So the synthetic data generation loop becomes super attractive because we can generate more data than any enterprise in the world has.

Jon Krohn: 00:42:18 Yeah, it's really cool. Speaking of real world applications now, as we talk about how we can be using these models in the real world, we can't have an episode today seemingly without talking about agents and we haven't

said the word agent yet 45 minutes into this episode. Over the past year, graphs have been a big topic in part because of the popularization of graph reg and graph ontologies for agentic AI. And Nvidia has joined in. So Nvidia's agent knowledge foundation includes something called a new auto ontology library that translates raw company data into business concepts like customers, products, orders, and maps how they relate in a graph. And so as an example, Databricks pairs Kumo Anomaly, which we've been talking about earlier in the episode, with Databricks genie ontology to power context aware agents. What do you think about this? After this, I'm going to ask you some questions about some pretty amazing automations that you're doing in the biomedical space with your research.

00:43:26 But I feel like this, just talking about agents a bit in general will be a nice way to get there.

Jure Leskovec: 00:43:32 The way I would think of this is that enterprise data comes in many modalities, many forms, from documents, databases, transactions, business events and so on. And the first thing you kind of need is to build an ontology, like a catalog of what these data are and what do they mean. And you can think of that as some kind of knowledge graph or a semantic model or something like this. And then you basically have two components in this world I think that are important. One is this retrieval component to find facts and extract evidence. It's a question about what do we know? And NVIDIA has an open source solution that people can use called NIMO Retriever. And then for decision making for predicting outcomes from structured data, anticipating what might happen or what the evidence shows may happen is done through this NVIDIA KUMA family of models.

00:44:34 And that's the way I would think of it. Another place where we see a lot of use cases for KUMA is in terms of agent safety and understanding and having basically

systems that are essentially predictive that are saying based on the activity of the agent, what's the safety concern? What's the probability that something goes wrong? And especially long running agents are generating so much context that you cannot have another LLM just process that context. You need to be smarter. So this structured data, tabular data, foundation models that have a much more scalable attention mechanism are able to help in those use cases as well.

Jon Krohn: 00:45:25 Brilliant. Thanks for walking us through those various agentic use cases with these open weight tabular models. Now let's dig into some of your applied research in biomedical AI. So your research over the years has applied machine learning to interconnected systems at huge scales from human societies to biological networks with computational biomedicine and drug discovery becoming a major part of the research that you've done. To which, you led the development of something called Biomni. You can tell me if I'm mispronouncing that, but that's Bio, B-I-O-M-N-I, Biomni, which is an autonomous AI co-scientist that orchestrates hundreds of tools and databases to streamline scientific workflows. And it seems to be doing pretty well because it recently won a million dollars from, it's called the Alzheimer's Insights AI Prize at the ADPD conference in Copenhagen, which I'm guessing is Alzheimer's and Parkinson's maybe conference. Yeah. So tell us about Biomni.

Jure Leskovec: 00:46:27 Yeah. So I'm a graphs guy in the old days. I was building people you may know algorithms at Facebook and things like that and studying information virality on what was called Twitter and so on. But if you think about biology, human cell, life, it's all about interactions. It's a giant humongous graph of interactions of molecules, proteins at the level of the cell, where are these things and how do they work together? Then cells organize into tissues and then an organism emerges and you have a phenotype and then there are diseases and things like that. So I've been

very, very interested in my research group at Stanford to thinking about how does AI push forward these disciplines and what can we do to make, let's say, drug development, our understanding of biology, life, medicine improve. So one of the projects we developed here at Stanford, it's an agent called Biomni, Bio and Omni.

00:47:45 Omni means everything, right? So that's kind of the name. And we think of it as a co-scientist. It's basically a companion to a human biologist scientist who is executing daily tasks and this agent can help a human with all kinds of, let's say, bioinformatics types, experimental data analysis, reasoning and so on. And what's interesting or how is this different than ChatGPT or something like that is that the agent becomes useful when it's able to use tools. And let's say in biology, bioinformatics and so on, you have these super specialized tools. AlphaFold won a Nobel Prize, but it's just one of many tools you can use. And this expertise about what tools to use and how to use them is very narrow and very specialized. So what we decided to do was to first build an environment for the agent that it can operate in. And the way you can think of this environment is just think of it as a very beefy machine or a compute cluster where all these tools are pre-installed and have been modified so that they've been accessible to agents.

00:49:03 A lot of tools are built for humans and the kind of ergonomics of the way humans use things is different than the ergonomics, the way agents use these tools. So what we basically did is we built another agent who is reading archive and bio archive papers, figuring out what tools are human scientists using. And those are the tools, those are the databases, ontologies, knowledge basis, resources, software tools, software packages that this agent has access to. And we implemented this at Stanford, launched it between basically using Anthropic Claude basically as an orchestrator, as an LLM, and then

having access through this big compute environment where these tools can be used by basically Claude writing code, having a knowledge base and things like that. And it worked amazingly well. And the project exploded and in a few weeks we had over 10,000 scientists using this practically every day for their daily tasks.

00:50:17 And what was amazing is that the agent was able to do basically the work of a well-trained senior PhD student in a sense of helping with some cloning design protocols or doing some single cell data analysis or looking at some multiomics data and so on. So it was huge boosts in productivity of human scientists because now you can just tell it what to do and it does it for you.

Jon Krohn: 00:50:46 With agents being able to handle more and more of what I do and what my listeners do as data scientists and AI practitioners in general, and with tools like Biomni being able to handle more and more of biomedical science and streamlining that, looking at a lot of charts, it seems like in 2027 are not that far after we might have recursive self-improvements, AI models improving themselves without humans in the loop. And some people think that that will very quickly trigger broad artificial superintelligence and things like the singularity, I guess. What do you make of all that? Where do you think we're going and how fast are we getting there? Where can we as practitioners find solid ground in this fast moving environment to be confident that the skills we're investing in are still going to be relevant years from now?

Jure Leskovec: 00:51:47 Yeah, that's a great question. I teach students here at Stanford, so this question is very important to me because I need to prepare my students for the future. My view is maybe this is kind of what I would say. I actually think that human knowledge and human judgment has never been so valuable in the history. And the reason for this is that humans have this unique ability to judge and to make decisions. And you can command an army of

hundred agents or thousand agents, whatever it is, they will generate slop unless you have this human who is making the right decisions, setting them on the right path and so on. And yes, maybe in some domains where the objective function is super clear and it's called a verifiable reward, I can always get the exact estimate how well did the agent solve the task. And then of course the agent can kind of keep iterating and looping.

00:53:02 But if you think about this harder, more open-ended problems, it's not like that. You need that human judgment. You need that human component of identifying, okay, what is correct? What is incorrect? Where do we

Jon Krohn: 00:53:18 Want

Jure Leskovec: 00:53:18 To go deeper? Where did the agent steer off and so on. And I mean, even in coding, in just writing code, we see that the best people are able to command these agents in a great way and just multiply their skills while to somebody else is just kind of generating slop over slop over slop. So I think that rigor, super strong fundamentals is what is not getting obsolete. It's actually getting more valuable and much more important.

Jon Krohn: 00:53:54 I like that a lot. And it makes a lot of sense to me with the intuition that I have and something, if people listen to the show every episode, they will have heard me say this before, but one of the places that I find I provide a lot of value in this increasingly automated world is thinking about what all the right context is and what the right questions are for setting an agentic process off in the right direction. Like you're saying, kind of getting it going in the right direction. I think if we just provide a generic little bit of information, we don't provide much specialized context, that AI slop scenario is very likely because we're just going to end up with an average response. Whereas if we can be really thoughtful about what call transcripts,

what data, what information that I had on the phone that wasn't recorded that I've got to make sure these points get in, they're critical context for these agents to be able to do their research and come to the answer that I'm looking for.

00:54:54 I think I've said on the podcast before, I haven't tested this definitively, but I think the plurality of my time, I think the single biggest category of anything that I do in my work week now is coming up with the right context.

00:55:12 And I think that's related to what you're saying. And then of course, when some output comes back from the agentic process running, then my human judgment can still provide some value. I guess when you're talking about areas where you can definitively know a right answer, so now we're seeing in 2026 lots of long unsolved math problems being solved. And so in that kind of area, to some extent in computer science where either the code runs or it doesn't run, you kind of have these ways of verifying. In those kinds of domains, I'm just kind of repeating what you said in a different way, it does seem like, okay, a lot can potentially be automated. A lot can be simulated, going back to our conversation earlier in the episode, and a lot can be automated. Whereas with real, you call them hard problems, but it's almost kind of like messy problems.

00:56:01 Real world problems don't have a right answer and real world problems never have all of the context that we'd like to have. And we still have to make some kind of decision. We have to put one foot in front of the other and make a business decision. And it does seem like humans might continue to provide a lot of value there, even in an artificially super intelligent future.

Jure Leskovec: 00:56:25 I agree. I agree with you. I think in that respect, I think it's kind of I said, I think the value of human judgment is getting higher and higher. That ability to manage

messiness, context, and being able to direct and judge the output of these agents I think is very important and that is very valuable. But in some sense, I feel like these agents feel like candy or feel like sugar. It requires a lot of self-discipline to not say, "Oh, it's good enough. Yes, let me send it. I'm done." It's actually, no, no, no. Let's think through it. Let's analyze it. Let's be rigorous. And being rigorous and thinking, as you said, it's super hard. Nobody wants to do that. My brains are lazy. I don't want them to think too much. So I mean, I'm joking a bit. So that's why I think it's in some sense so important to validate, to bring in this, let's say, additional context, think about the messiness, think what are all the inputs, let's say the agent needs to be able to do its work and then guide it, judge where it's going, stop it when it's going off track and so on.

00:57:43 And if you connect now to Biomni, human scientists love it because it's not built to say, "Hey, I will replace you." It's built to say, "I am here to be super transparent. You can stop me. You can talk to me." So now all of a sudden, science gets more digitized. Science does not leave in this notebooks and pieces of code and Excel spreadsheets and so on. It lives kind of inside this agentic environment. Everything is traceable. Everything is auditable. Everything in some sense is documented. And I think another benefit is that now a scientist kind of gets access to an entire specialized team, which they may not have at their home institution or in their home team. So that means also that the speed of science at which we can be making is accelerating.

Jon Krohn: 00:58:42 Yeah. Yeah. From that perspective, it does seem like we're going into an even more exciting period in history where we can accelerate discovery and we can live longer, healthier, better lives with better nutrition. Yeah. And hopefully more peace comes along with all that prosperity. I think it's an exciting future ahead and I'm glad that we'll get to see it happening and you get to play

a big part in realizing all of it. One last question for you just related to your teaching work. I want to highlight for people that you have a very popular course, CS224W from Stanford, and it's machine learning with graphs. It's a very popular course. You can watch it online for free. It's got open source code associated with it. And yeah, obviously we talked about Kumo and how it's open source. You mentor top tier AI researchers. You continue to shape how the next generation learns, skills and applies machine learning expertise through your teaching and all this mentorship.

00:59:56 In Slovenian podcasts, we translated the transcripts. To be able to get a little bit more context for our questions here, you said that you have to be very meticulous preparing for your students, realizing that creating a single homework question might take a hundred hours of work and treating every lecture like a rock concert while you simultaneously, you put all that work in on the front end and then you give it all away for free. And so I guess I wanted to thank you for that. I don't know if you have anything to add. There isn't really a question here at the end. I just wanted to make sure that people were aware that you're doing all this and that if they want to be learning more about graph neural networks, machine learning with graphs in general, the CS224W course is available for them online.

Jure Leskovec: 01:00:38 Yeah. Thanks for saying this. I think what's the most amazing to me is that when you basically, I come to the students and I say, "Hey, look, this will be hard, but it'll be good learning experience." And the most amazing thing is when basically students say, "Okay, I'm willing to put in the work because I'm going to learn a lot." And then the happiest student is the student who kind of excels beyond what they though they can do. That's why I think we are putting in all this work and that's why we are also making all these teaching materials, slides, videos, homeworks and everything publicly available. And I

encourage others to take my slides, take my courses and build on them because it helps the world. It helps spread the excitement about AI, computing, machine learning, the graphs and all this super cool stuff.

Jon Krohn: 01:01:27 Yeah, it's enormously helpful. 10 years ago, so having finished a PhD focused on machine learning and AI in 2012, that was right when neural networks and deep learning took off. And so I didn't have that as part of my formal education. And so to pick that up on my own as a professional a decade ago, it was these various Stanford 200 level CS courses that were useful to me. Andre Karpathi, Richard Socher, Christopher Manning, those kinds of people, Jure Leskovec, you guys were key to being able to help me understand what the heck is going on here and build some real world applications and talk about them on air. So thank you. Before I let my guests go, and I usually remember to warn them, but I forgot to warn you before we started recording, I always ask my guests for a book recommendation. Do you happen to have any book recommendations that come to mind for you?

01:02:18 They don't need to be technical. It can be just anything you've enjoyed recently that our listeners might enjoy as well.

Jure Leskovec: 01:02:26 The books I read, I have three little kids, so I can give you book recommendations. I like Julia Donaldson. I think those books are very nice. Those are my book recommendations. There is I think a German illustrator, Alex Schaeffler or something like this. And then yeah, Julia Donaldson, they nicely rhyme and they're fun to read. That's what I recommend if you have small kids.

Jon Krohn: 01:02:54 Love it. Thanks for that recommendation. And for people who want to follow you after this episode, to continue to get your thoughts, how should they follow you? How should they follow Kumo? How should they follow Nvidia

or whatever you think is important here for us to ensure we have in the show notes for people to be able to follow along after the episode.

Jure Leskovec: 01:03:14 Yeah, they can find me on LinkedIn. They can find me on X. I'm @ure, so it's very easy. Those are basically the two places where I talk. And then all our work, all our papers, software is open source for people to see and play with both at NVIDIA and also what we do at Stanford.

Jon Krohn: 01:03:41 Brilliant. Thank you for taking the time, Yure. This has been such a fun episode for me. And something that's pretty wild is we did this without any cuts. There were absolutely no breaks in conversation. This was just one continuous flow. So thanks for doing that with me. Thank you. And yeah, maybe we can check in in a few years and see how all of your various journeys are coming along and the big impacts that you're making.

Jure Leskovec: 01:04:07 Excited to chat again about whatever is next, but I'm sure I'm already working on it, as they say.

Jon Krohn: 01:04:16 Fantastic episode today. In it, Jure Leskovech detailed how Kumotabular takes labeled and unlabeled rows as its input and returns well calibrated predictions in a single neural network forward pass with no training or gradient updates required. He talked about how these models learn entirely from synthetic data generated with structured causally linked graphs, how Kumo relational treats a whole database as a graph, how Biomni gives scientists an agent with access to hundreds of specialized tools and databases doing work on par with a well-trained senior PhD student, and why strong fundamentals are becoming more valuable in the age of agents, even though those agents can feel like candy, tempting us to accept output that seems good enough. All right. 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 your social media

profiles as well as my own at superdatascience.com/1047.

01:05:09 Yes, this was episode number 1047. And if you'd like to engage with me in person as opposed to just through social media, I'd love to meet you in real life next week at the Open Data Science Conference's AI for Work Summit running from December 2nd to 3rd in Midtown Manhattan. It's a conference focused on building AI agents and applications for your everyday work. There will be workshops for hands-on practitioners as well as no-code workshops for more business-oriented folks. I'll be delivering the opening keynote on how to thrive in the Agentic AI era. I hope to see you there. Thanks to everyone on the SuperData Science Podcast team, podcast manager Sonya Braevich, media editor Mario Pombo, partnerships manager Natalie Jayski, researcher Serge Masis, and our founder, Kiral Eromenko. Thanks to all of them for producing this sensational episode for us today for enabling that great team to create this free podcast for you.

01:05:58 We are deeply grateful to our sponsors. You can support the show by checking out our sponsor's links 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, share this episode with someone who loves tabular data and would love to be doing something new with it, getting more efficiencies in their business. Review the podcast on your favorite podcasting platform, subscribe if you're not a subscriber, but most importantly, I just hope you'll 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. 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.