SDS PODCAST EPISODE 863: TABPFN: DEEP LEARNING FOR TABULAR DATA (THAT ACTUALLY WORKS!) WITH PROF. FRANK HUTTER

Jon Krohn:	00:00:00	This is episode number 863 with Professor Frank Hutter, co-founder and CEO of Prior Labs. Today's episode is brought to you by ODSC, the Open Data Science Conference.
	00:00:15	Welcome to the SuperDataScience Podcast, the most listened to podcast in the data science industry. Each week we bring you fun and inspiring people and ideas exploring the cutting edge of machine learning, AI, and related technologies that are transforming our world for the better. I'm your host, Jon Krohn. Thanks for joining me today. And now let's make the complex simple.
	00:00:49	Welcome back to the SuperDataScience podcast. Today's episode is an excellent one with the renowned machine learning professor, Dr. Frank Hutter. Frank is a tenured professor of machine learning and head of the machine learning lab at the University of Freiburg, although he has been on leave since May to focus on his fellowship on AutoML and tabular foundation models at the ELLIS Institute in Tübingen, Germany, as well as becoming co-founder and CEO of Prior Labs, a German startup that provides a commercial counterpart to his tabular deep learning model research and open source projects. And the company has just announced a huge, nine million euro, so about nine million dollar pre-seed funding round. Wow. In addition to that, he holds a PhD in computer science from the University of British Columbia, and his research has been extremely impactful. It has been cited over 87,000 times.
	00:01:39	Today's episode is on the technical side and will largely appeal to hands-on practitioners like data scientists, AI or ML engineers, software developers or statisticians, especially Bayesian statisticians. So, for a bit of context on the topic of today's episode, pretty much everyone

works with tabular data, either primarily or occasionally. Tabular data, I'm sure you're familiar with them once I describe them, are data stored in a table format, tabular. So, they're structured into rows and columns, like in a spreadsheet where the columns might be different data types. Say some columns are numeric, some are categorical, and some are text.

00:02:14 For a decade, deep learning has ushered in the AI era by making huge advancements across many kinds of data, pixels from cameras, sounds from microphones, and, of course, natural language, but through all of this AI revolution, deep learning has struggled to be impactful on highly ubiquitous tabular data until now.

00:02:34 In today's episode, professor Frank Hutter details how his revolutionary transformer architecture TabPFN has finally cracked the code on using deep learning for tabular data and is outperforming traditionally leading approaches like gradient boosted trees on tabular data sets. In this episode, he talks about how version two of TabPFN, released last month to much fanfare thanks to its publication in the prestigious journal Nature, is a massive advancement allowing it to handle orders of magnitude more training data. He also talks about how embracing Bayesian principles allowed TabPFN version two to work out of the box on data it wasn't even trained on, like time series data, beating specialized models, and setting a new state of the art on the key time series analysis benchmark. And he also talks about the breadth of verticals that TabPFN has already been applied to and how you can now get started with this conveniently open source project on your tabular data today. All right. You ready for this horizon-expanding episode? Let's go.

00:03:36 Frank, welcome to the SuperDataScience podcast. It's awesome to have you on the show. Where are you calling in from today?

Frank Hutter:	00:03:42	Thanks so much for having me. I'm in Freiburg. It's a beautiful city in the south of Germany, close to France and Switzerland, the beautiful university town, and now it's actually sort of turned the new German center of foundation models. There is Black Forest Labs is here. We're here-
Jon Krohn:	00:04:00	Oh.
Frank Hutter:	00:04:00	... building TabPFN. So, yeah, I'm super excited to be on the show, but also about Freiburg being really on the rise.
Jon Krohn:	00:04:08	Nice. That is exciting. Do you end up having a lot of in-person interaction? Does your lab, does your company meet in person and all these people from Black Forest Labs? You actually rub shoulders with them in person?
Frank Hutter:	00:04:20	We do go for coffee every now and then, but we are both pretty busy.
Jon Krohn:	00:04:26	Yeah.
Frank Hutter:	00:04:27	And then, there's meetups and so on.
Jon Krohn:	00:04:28	Nice. It is great to have that kind of community. So, yeah, let's talk about TabPFN. You mentioned it just there, and that is ... TabPFN is something that's been exciting to me for a couple of years. So, when version one came out, I took notice of it as, really, the only tabular data deep learning framework that I've noticed, so it definitely made a splash. So, there's a few different things that I want to talk about. First of all, we're going to talk about what the name means, and we'll talk about-
Frank Hutter:	00:04:59	Everybody mispronounces it like "TabFPN."
Jon Krohn:	00:05:04	Yeah. So, it stands for Tabular Prior-Data Fitted Network. What does that mean? Break it down for us. Yeah. Tell us what it means to have prior-data fitted into the network.

And, I guess, something that I can even explain very easily is ... I mean, you can expand greater, but this tabular idea is that most deep learning models are optimized for dealing with data that have a lot of spatial patterns, so things like machine vision, natural language processing. But, I mean, I've been teaching deep learning for almost a decade now, and very frequently, I would have students that are, say, finance students who have some tabular data, and they think, "I'd love to train a deep learning model on this." And they would always find disappointing results, relative to things like boosted trees or sometimes often just plain old regression models. And so, yeah, tell us about what makes this TabPFN architecture different, why it made such a big splash a few years ago when version one came out, and then, now, with this brand new release of version two, what the differences are.

Frank Hutter: 00:06:15 All right. Yeah. That's a lot of questions to unpack.

Jon Krohn: 00:06:17 It is a lot of questions.

Frank Hutter: 00:06:18 So-

Jon Krohn: 00:06:18 I can remember them all.

Frank Hutter: 00:06:20 Maybe let's start with tabular.

Jon Krohn: 00:06:22 Yeah, yeah.

Frank Hutter: 00:06:22 So, what is tabular data, and why is it so different than, yeah, vision data or speech data or text data and so on? So, tabular data is super common in the enterprise. It's like tables. Think Excel sheets, relational databases. There's so much information stored in these tables, and you have applications in all kinds of domains like healthcare, finance, business analytics, insurance, retail, whatnot. And there's your typical classification and

regression problems, which you learn sort of in Machine Learning 101. That's the stuff you fit a Random Forest to and so on, and there's also time series data. There's recommender systems, and all of these really work with tabular data.

00:07:08 And one of the properties of tabular data is that, typically, actually, most data sets are relatively small, and there's a lot of these relatively small data sets, and each of these data sets is very different. So, if you have a data set from healthcare, let's say you want to predict, based on some omics blood work, whether a patient has early stage Alzheimer's. Then you collect some data. Maybe you have 5,000 patients that you had over the last couple of years, and you know what ... Did they have early stage Alzheimer's or not? And then you get a new one, and, well, you want to predict whether they have it or not. And, well, you can wait a couple years, and then you know whether they had it, but then it's too late to treat it, so you want to predict it. And so, there, your features are these omics blood values, and the prediction variable is whether they have early stage Alzheimer's or not.

00:08:08 And then take another data set from, I don't know, banking, fraud detection, or, yeah, let's say fraud detection. Then you have all kinds of different transactions as features that the person had before and then maybe how much money is this that is in the transaction. Who is it going to, et cetera? And that has just nothing to do in terms of the features with omics blood values. So, how are you going to learn a model that ... From all of these different tabular data sets, that is actually very tricky.

00:08:44 And, in particular, if you compare it to, for example, vision, there you have these spatial patterns. Regardless of what you're looking at in terms of an image, there is some spatial regularity that makes it actually an image

and rather than just some noisy thing to look at. Yeah. We had convolutional neural networks, et cetera, picking up the spatial structure and learning features from the data. That's what deep learning has been enormously strong at, learning successively abstract representations of your data, and then you have this high-level representation that you can just fit some sort of a very simple linear model in the final layer.

00:09:33 And tabular data, on the other hand, that is something where the features ... People actually, typically have put some thought into these features. What is this blood marker? What is this, the amount of money you spend? It doesn't get much more high-level than that, and so, you don't need to discover these features. You already have them.

00:09:55 And so, the power of deep learning hasn't really reached tabular because it wasn't needed there. You don't need to learn these features. You actually just have these features to start with. And then, rather than sort of these more low-level feature engineering methods or feature generation methods that you get from deep learning, you have higher-level feature engineering, like what data scientists are great at. You look at the particular application, and you're like, "Ah, we're in medicine. We have the height of the patient and the weight of the patient, and we want to classify some disease. Maybe it's useful to know whether they're obese, so let's compute BMI by using weight and height."

00:10:40 And so, you have, what is it, weight divided by height squared, and that's a new feature. That would be pretty hard to learn for a network off of the bat. Of course, it can learn it, but it doesn't know that this would actually be a particularly good feature for this particular application because it doesn't know the context, et cetera, because, typically, in tabular data, all that is actually fed into

models like Random Forest, et cetera, is actually the features and the target variable, so the X and the Y. None of these typical machine learning methods like Random Forest, XGBoost, et cetera, even look at the column header. So, that's something that, well, for example, yeah, language models would be great at, at looking at the context and understanding what's going on and then understanding, "Ah, that's this column. I could actually generate something like BMI," but that's not what is sort of part of the problem description of standard tabular machine learning.

00:11:46 And therefore, it's really exciting if we can actually build a deep learning method that does do a good job at just the tabular core because when we have that, then we can actually combine it with all the power of deep learning, with language models, and so on and build something that's much greater. But the first step that we took is really to go for an apples-to-apples comparison on the problem that the traditional methods use and not use any of the column headers, et cetera, and still beat XGBoost, Random Forest, et cetera, on their own turf so that it's not just better because we use additional information, but it's already very strong to start with. And then we can, on top of that, include all of this other information. All right. That was a long-winded answer to tabular.

Jon Krohn: 00:12:42 Excited to announce, my friends, that the 10th annual ODSC East (Open Data Science Conference East), the one conference you don't want to miss in 2025, is returning to Boston from May 13-15! And I'll be there leading a hands-on workshop on Agentic AI! ODSC East is three days packed with hands-on sessions, and deep dives into cutting-edge AI topics, all taught by world-class AI experts. Plus, there will be many great networking opportunities. No matter your skill level, ODSC East will help you gain the AI expertise to take your career to the

next level. Don't miss out — the Early bird discount ends soon! Learn more at odsc.com/boston.

00:13:27 But it was such a great answer. That was an excellent ... You provide a great scope on this problem and how, with deep learning, we're typically concerned with extracting features from data. With tabular data, we don't typically need to be extracting those features from raw pixels or from raw sound files or from raw natural language. Instead, we typically already have some curated features, but there's a huge opportunity in those curated features to be, quote, unquote, "thinking" thoughtfully about how maybe those features could be recombined.

00:14:02 And so, it sounds like what you're saying is the ... and maybe this is the answer you're about to get into, the prior-data part, but it sounds like the prior-data part, the transformer architecture part of this model is it is able, unlike gradient-boosted trees or linear regression, to take into account the column header, to understand what that means, and automatically cook up something like, "Oh." The model then, quote, unquote, "knows" what height is, knows what weight is, and it can automatically calculate BMI. That's really, really cool. I almost swore. I almost said, "It's really effing cool."

Frank Hutter: 00:14:41 Yeah. And what's super cool, actually, is that we haven't even done that. And once we do that, it's going to be so much better, but what we have done, actually, so far is really only use the same information as XGBoosted, et cetera, and the X, Y, the raw numeric values, the raw category labels, et cetera. And we can put it together with all the power of language models, et cetera. And, of course, we're working on that and have some initial results, but yeah.

00:15:13 So, I mentioned that deep learning isn't really needed for tabular data, for generating these features because we

already have the features, but, well, we did actually come up with a deep neural network here. So, what is different? And what is different is that we actually use a transformer, very similar, in a sense, to GPT, to a standard language model in the sense that we actually can do in-context learning. So, in-context learning is a term that was introduced in the GPT-2 paper, and it's sort of this phenomenon where you can tell GPT something in the prompt, and you can tell it sort of in the prompt what it should be doing. So, you can say, for example ... basically prompt it to do a translation task without telling it that it should translate, but just say to languages, "Dog is [speaking German]. [speaking German] is cat. Mother is question mark."

Jon Krohn: 00:16:23 [speaking German].

Frank Hutter: 00:16:24 And then it's the German, "[speaking German]." And then, it basically, from just these two, three different examples, it figures out, "Ah, I'm supposed to translate. Let's do that." And so, it, basically, GPT has learned to encode an algorithm that first figures out what the problem is and then solves it. And just like that, actually, we have learned an algorithm that can do tabular learning.

00:16:52 And so, what we do in our architecture is we feed in the entire X train, Y train, and X test as part of the prompt, and the output is going to be the Y test. And so, one data set is basically a data point for training our model. So, we take the X train, Y train, X test, feed it in. The network outputs something, and whatever it outputs, how similar is it to the true Y test? We take the gradient, the loss between these, take the simple cross-entropy loss, and optimize for the outputs of this network to be as similar as possible to the true Y test. Does it make sense? Okay.

Jon Krohn: 00:17:42 For sure.

Frank Hutter: 00:17:44 And so, this, I mentioned, a data set is a data point. So, if we had trillions of data sets, just like GPT is trained on trillions of tokens from the internet, then we could just say, "Well, we have trillions of data sets from the real world. We just fit a foundation model that does precisely this machine learning task like classification, for example, on all of these data sets, and we're done." So, once we have learned to do that on a trillion data sets, then we can do it on the next data set. That makes a lot of sense. It's very much like a standard language model. You can predict the next word. You've just learned to predict the next word, but we don't have a trillion data sets. In contrast to language models, there is really very few high quality data sets that are on the internet.

00:18:40 So, there's a bunch of tables. For example, if you go to Wikipedia, there's tables of, "Well, this basketball player has this number on their back." That's not a machine learning task. It's maybe a retrieval task, but you can't learn anything from that. You can't learn to, yeah, do a classification or regression, but what you need is really these properly formatted data sets in order to actually train your algorithm on. And when there's lots of noise and missing values and garbage, then yeah. It's pretty hard to actually learn on that.

00:19:21 And what we did instead is to actually generate all of our training data synthetically. It's a trend that also is happening in language models to partially train on synthetic data, but I believe our paper is the first one that really succeeded in only using synthetic data and coming up with a state-of-the-art model. And so, the key is, really, we needed to generate a prior over what we believe data sets might look like and what the types of data sets are that we want to work well at. And so, that's where we're getting to prior-data fitted networks.

00:20:06 There we'll take a step back now and first explain what these prior-data fitted networks are, and then, again, come back to TabPFN, which is a prior-data fitted network on tabular data. So, basically, I'll first explain the theory of PFNs, prior-data fitted networks, and then the step to TabPFN is just actually creating a prior that generates tabular data.

00:20:35 Yeah. So, the PFNs, the prior-data fitted networks, there was actually a paper already from 2022. It was called Transformers Can Do Bayesian Inference. So, there, we basically showed that if you have a prior that you can sample from, then you can take many ... draw many data sets from this prior and draw many data points from each of these data sets and fit them, just like we just explained with TabPFN. And the resulting model will actually have learned to encapsulate the prior so that, when it's actually fed real data, it will give you a approximation of the posterior predictive distribution under that prior for that data.

00:21:34 And then, when you train this sort of with an arbitrarily large transformer, with an arbitrarily good, so the cross-entropy loss really goes as far down as potentially possible, then you are actually exact, and your posterior prediction is exactly what the true posterior prediction should be. So, if you, for example, take a Gaussian process prior or a linear regression, then you get the true basal linear regression out or the true posterior Gaussian process. Maybe I should pause here for some-

Jon Krohn: 00:22:08 Yeah. So, I'm just going to quickly pop in with a couple hopefully short questions and clarifications. So, something that I didn't realize from my initial research, maybe this isn't the case, that it is the case with all PFNs, with prior fitted networks, with all prior-data fitted networks, but it sounds like fundamental to prior-data

fitted networks is Bayesian inference. That's always the case?

Frank Hutter: 00:22:35 Mm-hmm.

Jon Krohn: 00:22:35 Okay, okay. So-

Frank Hutter: 00:22:35 Yes. Yeah. So, they do compute the Bayesian posterior predictive distribution. Basically, that's what the optimization objective is.

Jon Krohn: 00:22:44 Nice. So, there's a Bayesian process, a Bayesian learning process happening on, in this case, a transformer architecture. That is very cool. And because we come across sometimes hearing that Bayesian inference is going to be useful with deep learning architectures like transformer architectures, but this sounds like a very concrete use case of it, and, yeah, it sounds like a powerful application of it.

Frank Hutter: 00:23:12 Yeah. It is super powerful because, I mean, something like Bayesian linear regression, you can do in closed form. Maybe I should still explain it to set the scene. So, there, the prior would just be ... The data is just a line. Line has some axis, a line on ... and some slope. And so, you basically just want a posterior over these two parameters and yeah. If you, yeah, you do the math and you can compute this Bayesian posterior predictive distribution in closed form.

00:23:46 For lines, this is fine. For Gaussian processes, for example, it is also fine, but once you go a step further, for example, a Gaussian process where you don't know the hyperparameters, then you have to do Markov chain Monte Carlo or variational inference. Or once you have a neural network and you want to be Bayesian over the weights of your neural network, then it becomes extremely complicated. And you can do all kinds of

approximations or SGLD or Hamiltonian Monte Carlo or whatnot. The math is really hairy and yeah. Markov chain Monte Carlo typically takes a long time to convert. There's a saying, "While my chain is smoothly sampling, it takes a week," or something like that until you get going.

00:24:38 The other possibility is variational inference, which is also often quite hairy in the mouth and also has approximation errors, typically, and in contrast to that, prior-fitted networks are just so incredibly simple. All you do is you sample from your prior. You get a bunch of lines, and then you feed these lines. That's data points from a line and another data point from the line as a test distribution, as a test example, and you want to output the true predictive distribution for that data point. That is the optimization objective for that one line, and you sample millions of these lines and just learn over millions of these lines by just standard supervised learning to predict the missing values.

00:25:35 And naturally, you learn to actually compute the predictive distribution because, well, sometimes, typically, there is noise in this data, et cetera. And since we optimize with cross-entropy loss, if we're far off and certain, then that's bad, and that's penalized. So, you actually automatically learn something that is, yeah, regularized to be exactly the right thing to be predicted.

00:26:05 If you sample arbitrarily many curves from this prior that look like your … that, in the data sample, look like your sample integrated over all of that, you want to get the best predictive distribution. And that's precisely the right predictive distribution. So, just by sampling from the prior and then running supervised learning, you learn to actually approximate Bayesian inference to an arbitrarily strong degree, and that's really powerful.

00:26:39 And so, there's a lot of applications that are outside from tabular data where this is also really cool. For a neural network, you could say typical Bayesian neural networks give you a posterior predictive distribution or, sorry, not posterior predictive distribution, posterior distribution over the weights of the neural network and then integrate over that in order to give you a posterior predictive distribution.

00:27:06 But what they don't do, for example, is consider, well, which is the right architecture. They just say, "Well, what are the right weights of this particular architecture?" But you don't know which is the right architecture to explain this data. So, with PFNs, you can just say, "Well, I have this distribution of architectures, and then, for each of the weights of the architecture, I have this distribution." You sample from it. It's trivial, and then you just get the posterior predictive distribution, which is the right architecture for this data. So, you kind of do some Bayesian neural architecture search in a forward pass, which is just really cool.

Jon Krohn: 00:27:36 Hmm.

Frank Hutter: 00:27:36 There are some limitations.

Jon Krohn: 00:27:37 That is really cool. So, we've been talking a lot, obviously, about priors, posteriors, posterior predictive distributions. I want to quickly break down, for our listeners who aren't Bayesian, what these kinds of things broadly mean. And so, I'm going to give a really simple toy example, and with your expertise, you can tell me what I get wrong in my explanation, but basically, if I have some ... So, I could have a simple linear model where all that I have is the slope of the distribution and some Y-intercept. So, this is kind of the simplest kind of regression model. With a Bayesian approach, I could assign some kind of prior distribution to both of those variables, to the Y-intercept

and to the slope of the line. So, I could say, "Based on my experience with these kinds of data, with this kind of problem, I think that there's going to be a slight slope, and I think the y-intercept will be around zero."

00:28:44 If I'm very confident I can assign a really narrow variance to my distribution, or if I want to ... If I don't have much kind of prior understanding of what this process should be like, what this regression model all should be like, I could have a very wide distribution which will allow these kinds of learning approaches, like you said there, Markov chain Monte Carlo, some kind of Hamiltonian process. There's lots of different kinds of solvers for Bayesian approaches that allow me to search gradually, like you said. It was kind of like While My Guitar Gently Weeps, while my Markov chain gently converges. What did you say?

Frank Hutter: 00:29:31 Yeah, it relatively makes sense. Probably the right thing to say.

Jon Krohn: 00:29:39 A big advantage of this kind of approach, of this Bayesian approach, is it allows you to incorporate prior information. You don't have to have your model be learning from scratch necessarily, although you could have it for some parameters or maybe even all the parameters, basically learn from scratch. And then, after this learning process happens for a while, like a Markov chain Monte Carlo, like a Hamiltonian, we end up in the end with posterior distributions that represent kind of what we've learned from the data. So, we start with a prior distribution that could be a highly informed prior distribution, or it could be relatively uninformed prior distribution. And then we use the training data that we have to converge upon some posterior distributions that give us ... Yeah. So, they incorporate the prior information as well as the data that we have trained on. And you were just giving lots of cool examples there where we can use

posterior distributions to be finding the weights of a deep learning network, for example. How did I do?

Frank Hutter: 00:30:35 Yeah, no, actually very good. The one thing I wanted to say, we do have a limitation. Actually, we don't get the posterior over the weights of the neural network. That is what you get with standard methods like MCMC and variational inference. We bypass that step. We just directly go to the posterior predictive distribution, so the Y given the X.

Jon Krohn: 00:31:00 Oh.

Frank Hutter: 00:31:01 And we don't have a posterior over the W, the weights. With MCMC and this variational inference, you actually integrate out over all of the weights in order to get your predictive distribution, but we bypass that. We can't tell you which is the right architecture, actually. We just tell you the Bayesian integral over all the possible architectures that might have explained the data.

Jon Krohn: 00:31:23 Nice. Okay. So, I think we've now covered what prior fitted networks are, what PFNs are. So, now, I think we're probably at a point where we can move to TabPFN, so a PFN specifically designed for tabular data.

Frank Hutter: 00:31:38 Yeah, exactly. That is what TabPFN is, and it's a ... We've talked about PFNs. You need the prior to, yeah, explain what type of different assumptions you have on the data. So, we would have a prior that creates tabular data sets in order to express our assumptions on what data sets we might be facing.

00:32:05 So, the first author of the paper, Noah Hollmann, he came up with this pretty ingenious method to sample structural causal models. A structural causal model is basically model that samples a graph, and then the features are nodes in that graph, and the target variable is also a node

in that graph, and then your sample connections in this graph, and you don't quite know. Does a target variable cause some of the features? Do the features cause the target? Do some of the features jointly cause a target? Does a feature cause a target, and that, in turn, causes another feature? There's this huge set of possible structural causal models that could explain the data, and if you could identify the right structural causal model for your data at hand, then you would get much better predictions, but you don't ... You just get the data. You don't actually get the structural causal model.

00:33:07 So, what we actually, with TabPFN, do in the end is to build the Bayesian posterior over all the possible structural causal models that could be explaining the data. And so, you could have one structural causal model that's completely wrong for the data, that gets a very low probability. And so, the predictions from that model would be really low weighted, and then you can have a model that matches really well with this data that gets a higher probability and is weighted more highly. So, that's what the true Bayesian posterior would do, but, of course, the TabPFN, that doesn't get to store all the 130 million structural causal models that we used to generate it. It just gets the raw data, and it has learned to actually interpolate over all these possible models and has learned to actually approximate this Bayesian posterior in a forward path.

Jon Krohn: 00:34:06 A big strength here with the TabPFN approach is using generated data. So, it sounded like you said, "Over a hundred million generated data sets," because we don't have ... Unlike, say, natural language data when you're training something like a GPT kind of architecture, you have trillions of tokens that you can train your model over, but we don't have that kind of scale, anywhere near that scale in terms of high quality, well-structured

tabular data sets. And so, you've gone ahead and simulated over a hundred million of them.

Frank Hutter: 00:34:37 Yes, exactly. And so, we can actually really control what's going in. So, we have no data leakage because we actually didn't put any real data in, so there's not any possibility that we've memorized the test data sets or something.

00:34:52 Actually, a fun fact, I should mention this. The very first time we submitted TabPFN v1, it was rejected because the reviewer said, "The performance is far too good. You must be doing something wrong."

Jon Krohn: 00:35:06 Hmm.

Frank Hutter: 00:35:06 And what they thought we were doing wrong is you must be tuning on your test set because we actually had some complicated stuff in there that was actually doing some gradient-based updates, looking at some real data sets, and we just dropped all that. It meant an [inaudible] and we just never touch any real data during training, and that, yeah, made it just-

Jon Krohn: 00:35:30 That's-

Frank Hutter: 00:35:30 ... so much more easy to defend the next time. Went through with flying colors.

Jon Krohn: 00:35:35 And so, now, so you mentioned version one there. So, now, unless I'm jumping the gun too much, so it was a couple of years ago that version one came out, and that's what I was talking about at the onset of the episode. That is when I first noticed TabPFN, and it is still today the only tabular deep learning approach that is on my radar, but in January, you guys had a paper. So, you mentioned Noah Hollmann earlier. So, he's the first author on this Nature paper that you published. It's called Accurate Predictions on Small Data with a Tabular Foundation

Model, and I'm, of course, going to have a link to that in the show notes. We're going to spend a fair bit now talking about this version two release and the associated paper.

00:36:19 Something to answer, maybe kind of quickly at the offset, is when you're coming up with a venue to publish something like TabPFN in, how did you think of Nature, which is ... It's one of the world's most popular journals, and it's general. It's designed to kind of give a broad overview across all disciplines. And so, it's interesting because while ... I don't know. So, you can let me know why you chose Nature, but it's amazing, first of all, to get published in Nature at all. And so, it's amazing to think that ... to even have the audacity to submit to something like Nature. And then I'm guessing that the reason why you would pick something like Nature is because tabular data are so ubiquitous across so many different scientific fields. I mean, that's literally your opening sentence in the abstract is, "From biomedicine to particle physics to economics and climate science, tabular data, which are spreadsheets organized in rows and columns, are ubiquitous across scientific fields." So, I guess I understand. Well, I've spoken way too much. You can tell me. Tell me about this Nature paper and what led to it.

Frank Hutter: 00:37:26 Yeah, absolutely. So, yes, tabular data is super ubiquitous and so, we did want to reach a really broad audience. That's, of course, one of the nice things we do get with Nature, but it was actually the ... Already when we submitted the first version, TabPFN v1, we submitted it to iCLEAR, for which the papers are directly online, and also, if you retract them, they stay online. And literally the day after we submitted, I was like, "This is a real breakthrough. This changes everything. The fact that we can now actually have a deep learning model that does in-context learning and learn across tabular data sets, we ... There's so incredibly much potential in there." If I was

at DeepMind, we would've sent this to Nature because DeepMind actually, well, it does publish there a lot, and-

Jon Krohn: 00:38:24 Yeah, DeepMind is constantly publishing. Yeah.

Frank Hutter: 00:38:25 ... we read a lot of these papers, and we just ... Whenever they had a new paper, we're like, "Wow, this is so amazing," and we just ... Everybody talks about them, and everybody reads the papers. I read the papers, and there's ... They're really great. And I was like, "Hey, this is of the same caliber."

00:38:44 We had submitted to iCLEAR, so if you retract, it's still online. That would've been a problem. So, we're like, "Okay, fine. We can't publish in Nature, but let's go for a next version, and let's make this really, really strong," because the first version, all that that did is ... It was extremely limited. It only worked on numerical data. It didn't do missing values. It didn't handle outliers. It didn't have imbalanced data. Even categorical values were a problem, and, of course, tabular data is all categorical. It also didn't do regression. It only did classification. What it did do for me is it was an eye-opener in that this is possible, and we, quote ... or just need to scale up and just need to make this more general and so on. Of course, we had a bunch of extensions and improvements on architecture and so on, but at the core, going from TabPFN v1 to v2, it's very much the same in-context learning, just made to work really well.

00:39:58 And so, actually, it's a much better paper for Nature because the TabPFN v1 was like, "Yeah, great. You can do this on data sets up to a thousand data points with ... " What did we have? A hundred features, only numerical data, none of the stuff that you have in data science, none of the issues. So, not a whole lot of people used it because of that. And we have a repo with some applications, like 15 different papers that used it and

showed that it's awesome in different domains, but yeah, 15, not thousands.

00:40:36 And so, that was the rightful criticism of the community in TabPFN v1. I said, "Hey, this is so amazing," and then they're like, "Well, why is nobody using it on Kaggle? This is not really breakthrough in terms of the impact yet," but that is really what changed with TabPFN v2 because it's just so darn generic now. It can just do whatever. It can tackle any type of tabular machine learning problem, just like XGBoost can, with still some limitations. I definitely need to be very clear here. It still has a size limitation, so small, and that is in the title of the Nature paper, small tabular data sets. So, in particular what we evaluated was up to 10,000 data points and up to 500 features.

00:41:29 And so, we already scaled up a fair bit from the thousand from before and yeah. I'm pretty confident that, based on a combination of different approaches, we can also scale up to a hundred thousand or a million or something like that.

00:41:43 Once you have billions of data points and you don't really need to be Bayesian about your data, then you have enough data that you just let the data speak, but when you have a hundred data points and you fit a neural network or you fit an XGBoost or something, you will typically overfit the data a lot. But if you have a strong prior that has ... and emphasizes smoothness and so on, then you overfit a lot less. It's learned to, using cross-entropy loss on the test portions of the sample data, that it has learned not to overfit, and so, it just doesn't overfit as much as a standard method.

00:42:22 And yeah, so it was a breakthrough, but not really in terms of methodological improvement. Yes, we have a new architecture that is nice. That could have been a paper by itself. We could have written individual papers on, "Hey,

let's do this for missing valuables or missing variables." We can do a paper for imbalanced. We can do a paper on just a regression, et cetera, et cetera. So, we could have papers on all of these, but we didn't go for that because that would have ... We would have had to have, yeah, ablations comparing against all kinds of different approaches particularly for that. We just wanted an all-encompassing framework that just works for all kinds of data, and Nature is a great venue for these types of papers where just the end result counts. It's not the individual contributions in terms of methodology to get there, but what do you have now [inaudible] for example? Yeah. There were also some methodological contributions there, but they weren't mind-boggling. It's just that this whole thing put together really worked. And so, we're also of that category, and that's why, yeah, we did have the audacity to try for Nature, and it did work.

Jon Krohn: 00:43:46 Yeah, yeah, it's very cool. So, this version two, relative to version one, to kind of summarize some of the key attributes, you can now handle ... Well, it's well tested on up to 10,000 data points, 500 features, which is quite a few features, and it can handle different kinds of data, not just numeric data. It can handle text data, even, correct?

Frank Hutter: 00:44:09 It now can in the API, but actually not in the paper.

Jon Krohn: 00:44:11 Did you know that the number one thing hiring managers look at are the projects you've completed? That's why building a strong portfolio in machine learning and AI is crucial to your success. At Super Data Science, you'll learn how to start your portfolio on platforms like Hugging Face and GitHub, filling it with diverse projects. In expert-led live labs, you'll complete an exciting new project every week. Plus, through community-driven projects, you'll tackle real-world, multi-week assignments while working in a team. Get hands-on experience with

projects like retail demand forecasting, building an AI model from scratch, deploying your own LLM in the cloud and many more. Start your 14 day free trial today and build your portfolio with superdatascience.com.

00:44:52 And yeah, it handles missing values. It handles outliers. This is very cool. I think I already said, "Very cool." I don't mind repeating it because this is something that is going to be a game changer, particularly, as you say, in situations where we have tabular data, where we don't have huge amounts, where we don't have billions of rows, where we have hundreds, thousands, tens of thousands, maybe hundreds of thousands of data points. Having these kinds of Bayesian approaches allows the priors to be able to fit the data much better than other kinds of approaches out there.

00:45:28 Before we get on to kind of specific, real world examples of TabPFN, I understand that in addition to working with tabular data, you also recently had a breakthrough with time series data.

Frank Hutter: 00:45:40 Yeah. It's really mind-boggling. So, it is the same model that we have in the Nature paper. We also tried it for time series data. You can think of a univariate time series, just a signal over time, such as, maybe, a trend. And so, basically, you have a time signal and then a size of the signal. And all we do is taking the time index, basically saying, "Well, this is the time of day. This is the day of the week. This is the day of the month. Do some sine and cosine features of that, and cast it as a tabular problem." So, basically, each timestamp gets these six features, including the timestamps in the future, and then you have, for each of the known timestamps, you have your X train, and for the future timestamps, you have the X test. And so, this works for the next timestamp or for 17,000 timestamps in the future. You can encode each of these just as one new data point. And so, you can predict as far

ahead as you want, just in ... not auto regressively, but just directly in one forward path.

00:47:00 And the mind-boggling thing is that this model that we had in Nature, that is trained only on synthetic data and has never seen a time series and has never seen a real data set in the first place, actually is the best on the public benchmarks on time series, is better than all the foundation models that are trained specifically for time series, that are trained on, yeah, synthetically generated time series, real time series, et cetera. And with this model, we didn't even try, and it just works out of the box.

00:47:40 So, as of today, there's this benchmark, GIFT-Eval, which was in Europe's DBT paper just a couple of months ago and yeah. So, this is the standard benchmark for time series, and it's number one on there, outperforming Chronos.

Jon Krohn: 00:47:54 Whoa. Whoa.

Frank Hutter: 00:47:57 And Chronos is from Amazon. It's a really cool paper, and this just goes to show how much there is to gain here. Once we fine tune for time series and we iterate on this or we have a time series prior, the sky's a limit. So, I'm super excited about this and, yeah, really looking forward to building more there.

Jon Krohn: 00:48:21 State of the art, out of the box, that is a nice outcome. Wow. Great. Yeah, so very exciting, all of these big updates from version one to version two. With version one, as you mentioned, there was relatively limited applicability of TabPFN, but nevertheless, there were still some great use cases that came out of it. One of them was a science paper. So, in addition to Nature, the paper that you published in, there's one other big kind of

general, broad science paper out there, and it's called Science.

00:48:52 And so, there's this paper. I'm not even going to try to get into the biology of what this means, but we'll include the paper in the show notes. It's called Large-Scale Chemoproteomics Expedites Ligand Discovery and Predicts Ligand Behavior in Cells. And so, I can't really explain what this is all about. It's something to do with determining protein structure, but the key thing is that TabPFN was used as a part of the inferences that they made in that paper. And I'll have a link also in the show notes to repo, a GitHub repo called Awesome-TabPFN that lists about a dozen existing applications of TabPFN across health insurance, factory, fault classification. There's financial applications. There's a wildfire propagation paper, a number of biological papers in there.

00:49:50 So, yeah, clearly lots of different applications out there, even for v1. I don't know if you want to talk about them in any specific detail, Frank, but I know that you are, of course, looking for more people trying out TabPFN, especially now that version two can handle so many more kinds of data types, can handle missing data, can handle outliers, and can handle larger data sets. So, listeners, if you've got tabular data out there, you can head to the TabPFN GitHub repo that we also have a link to in the show notes, and you can get started right away.

Frank Hutter: 00:50:24 Yeah, awesome. Thank you so much for mentioning this, the Awesome-TabPFN repo. I literally, actually created this today, so I hope by the time that the show actually goes out, there is a lot more than a dozen applications there, and please, yeah, whenever you have an application, a use case, just either send us a note. Or, actually, this is one of these repos where you can just do a pull request with your own application, put your own paper, and yeah. We'll basically advertise it. Also, if there

is cool applications, we'd love to have blog posts or, yeah, just retweet your content and so on. I think we really want to build this community of people who love TabPFN and build on top.

00:51:15 And the open source community has already picked this up, and within a couple of days of the Nature paper, there's this repo on CHAP IQ that's all about interpretability. Directly put TabPFN in there, and so, yeah. It's really amazing to see the speed at which the open source community works, and I'm really looking forward to what else people will build with this.

00:51:41 One cool thing about the Science paper I wanted to mention is, yeah, I also know nothing about chemical proteomics, but that's kind of the neat thing. I can still work on this because, well, we have this really generic method, and if there is data chemical proteomics out there, then we can fine tune on that and get something that's even better for this use case. And so, those are the types of things that I'm really excited about doing for all kinds of use cases. There's also already something out there on predicting ...

Jon Krohn: 00:52:20 Algal blooms.

Frank Hutter: 00:52:21 Algal blooms, yeah.

Jon Krohn: 00:52:22 Yeah, algae.

Frank Hutter: 00:52:23 So, yeah.

Jon Krohn: 00:52:23 Green-

Frank Hutter: 00:52:23 Algae, I know, and algal blooms are the sort of ... but yeah, sort of-

Jon Krohn: 00:52:28 Yeah. I suspect-

Frank Hutter: 00:52:29 Things that are good for the environment and so on, I think I'm really excited about those types of applications. There's lots and lots of applications in medicine. There's not that many published papers on applications in finance and so on because, well, typically, people don't publish-

Jon Krohn: 00:52:44 Finance companies, yeah, exactly.

Frank Hutter: 00:52:44 ... these types of applications as much, but medical and so on, there's a lot, and, yeah, really hoping for a lot of people to use it to do good things for the world.

Jon Krohn: 00:52:55 Yeah, fantastic. Very cool. So, yeah, we've got the TabPFN repo available to you to access this Python library right away. It's been downloaded almost a million times at the time of recording, which is pretty cool, and yeah. And then we'll, of course, also have a link to this Awesome-TabPFN repo that has all of the applications.

00:53:20 And so, speaking of applications, you are spinning out a startup to help spread the good word and presumably applications of TabPFN and associated technologies, and appropriate given how much we've talked about Bayesian inference and priors and posteriors. Your new company that you're co-founder and CEO of is called Prior Labs. So, tell us a bit about Prior Labs and how it complements or how it's different from the research that you're doing at Tübingen and Freiburg.

Frank Hutter: 00:53:55 Yeah. I mean, I'm super excited about this startup. I've been wanting to build something for many years now, but, really, for the last 10, 12 years, I've been ... I co-started and have been co-leading the AutoML community, so this community on automated machine learning that's all about democratizing machine learning, making it easy for everyone to get state-of-the-art machine learning by not having to worry about picking

the right hyperparameters, picking the right method, et cetera. And we've had a lot of great research and many, many really nice papers.

00:54:43 We've also had some tools. In particular, Auto-SKLearn was our most widely used and widely known tool that wraps around scikit-learn and allows you to figure out the right method in scikit-learn, the right pre-processing, the right algorithm, the right classifier, the right hyperparameters, et cetera, and made that much easier, but it sort of always ... Coming from the university and being at the university, having just a few research engineers who happen to, yeah, want to work in a university setting, we were never really in a position to build something for the masses. We've always built something that's sort of good for our research friends and good for ourselves to do our research with. And, yeah, if you want to reach a broader set of people, of course, we need a commercial entity for that.

00:55:39 And also, with TabPFN really being this breakthrough that will allow so much cool new stuff, yeah, we just need more workforce. We need really strong engineers to build amazing products. And so, that's what we will be doing in the startup.

00:55:57 In the university, I will keep an academic co-affiliation, and in the university, I will focus very much on tabular data as well then and research about the tabular data, things like interpretability. What does this network do? It is the best learned algorithm, but how precisely does this algorithm work? How precisely does it change when you change the priors? What are the failure modes? Where is it particularly good? How can we improve it further? There is so many avenues to do research on, and, of course, with the startup, we also want to push the boundaries of what's possible in terms of capabilities, but with a university hat on, we'll be able to focus more on,

yeah, maybe some moonshots, things that might turn out or might not work. It's good to have, yeah, this open-endedness of research, and that you can really only have in an academic setting.

00:56:58 So, I'm really excited about combining the two and also provide the PhD students an opportunity to have amazing engineers to actually build products out of what the PhD is published. And so, I'm, yeah, really excited about these energies and the future of Prior Labs.

Jon Krohn: 00:57:15 Fantastic, and I know that you are doing hiring at least of PhD students because you post about it. You posted about this recently on LinkedIn, so I'll include a link to that in the show notes, and I wonder also ... I mean, it sounds like you're also hiring engineers at Prior Labs.

Frank Hutter: 00:57:33 Yeah. We're hiring a lot of people, actually, at Prior Labs. I haven't posted about that on LinkedIn yet because we'll have our funding announcement two days after we tape the show, but by the time it goes out, it will have long happened. And, yes, we are hiring sort of full-blast AI scientists, ML engineers, backend engineers, community people, at some point in the future, also sales, but we're actually really not focusing on that now. We're focusing on building the community and building amazing tech.

Jon Krohn: 00:58:09 Nice. I usually have my last question be how to follow you, but I'm actually just going to jump to that right now because we were kind of just talking about how you're going to have this big funding announcement, which will be live by the time that this episode is published, and you'll be announcing more hiring on LinkedIn and that kind of thing. So, how should people follow you to get the latest on TabPFN, but also maybe opportunities to be involved in the open source community or even as a paid employee?

Frank Hutter: 00:58:37 Yeah. So, I'm on Twitter/X and LinkedIn, evermore on LinkedIn. Also, want to, at some point, start Bluesky, if I ever have time, but yeah. So, we have this, the GitHub repo you mentioned, so there's a TabPFN repo. There's a TabPFN API repo, and there's a TabPFN extensions repo, and it's particular, these TabPFN extensions. That's a repo where we strongly encourage the community to push extensions, push cool things people have done with TabPFN, such as this work on interpretability, work on, yeah, doing better hyperparameter optimization, post hoc ensembling, and stuff like that. Auto TabPFN is in there already, so we strongly encourage, yeah, interactions there.

00:59:30 If you're interested in, yeah, applying TabPFN to your particular domain, like fine-tuning, et cetera, do reach out to us, actually, also, particularly on our Discord channel. So, we have a Discord channel that is particularly for ... particular for TabPFN. We already have over 200 people in there starting to build this community, so I'm super excited that that is working. I already did an AMA there last week and yeah, great questions, and yeah. It looks like it's going to be a really cool community.

Jon Krohn: 01:00:05 Nice. Yeah, no doubt. It's interesting. I hadn't noticed this before, but I can see on the GitHub repo for TabPFN how new people are online in the Discord channel right now. There's 55 online-

Frank Hutter: 01:00:17 Wow. That's so cool.

Jon Krohn: 01:00:17 ... which is an interesting little widget included in there. Nice. Yeah, so fantastic. I'm sure you'll get a lot of interest from this podcast episode and just how amazing this project is in general. It really is transformative. It's been so exciting for me to have you on the show because of my longstanding interest in TabPFN. Before I let you go, I need a book recommendation from you.

Frank Hutter:	01:00:41	Book recommendation, let's see. I really like Asimov, the Robot series or Foundation series. I think, yeah, if you haven't read them, I strongly recommend.
Jon Krohn:	01:00:53	That's a great recommendation, especially at this time.
Frank Hutter:	01:00:56	Yeah.
Jon Krohn:	01:00:58	Thank you so much, Frank, for taking the time, busy between getting a startup going and your university responsibilities. It's amazing that you can take the time to be on a show like this, so we really appreciate it and, yeah, wish you all the best.
Frank Hutter:	01:01:13	Yeah. This was super exciting. I love your show and yeah. Yeah. I'm really honored to be here, actually, so I'm super excited. Thank you.
Jon Krohn:	01:01:22	Yeah. It's mutual. Thank you as well. All right. Yeah. Maybe we can check in again in a few years and see how the TabPFN journey and the Prior Labs journey is coming along.
Frank Hutter:	01:01:31	Absolutely. Love it.
Jon Krohn:	01:01:39	What a fascinating and practical episode with Professor Frank Hutter today. In it, he covered how TabPFN is a deep learning model specifically designed for tabular data, that uses a transformer architecture combined with Bayesian principles to make accurate predictions, even with limited data. He talked about how version two of TabPFN significantly expanded its capabilities, now handling up to 10,000 data points, up to 500 features, missing values and outliers, numerical and categorical data, and, through their API only at this time, text data as well.

01:02:07 The model was trained entirely on synthetic data, over a hundred million generated data sets, eliminating any potential data leakage while ensuring robust performance. We talked about how TabPFN version two unexpectedly achieved state-of-the-art performance on time series prediction without any specific time series training, outperforming Amazon's Chronos and other specialized time series models. And he talked about how Prior Labs, his new startup, has been created to commercialize TabPFN technology and build products that make the TabPFN breakthrough accessible to a broader audience while academic research continues at the University of Tübingen.

01:02:41 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 Frank's social media profiles, as well as my own, at superdatascience.com/863. And if you'd like to meet in person as opposed to online, I'll be giving the opening keynote at the RVA Tech Data and AI Summit in Richmond, Virginia on March 19th. Tickets are a bargain, frankly, so if you're in the Richmond area especially, come on down and see me on March 19th. It'd be awesome to meet you there.

01:03:10 Thanks, of course, to everyone on the Super Data Science podcast team, our podcast manager Sonja Brajovic, our media editor Mario Pombo, partnerships manager Natalie Ziajski, our researcher Serg Masís, writers Dr. Zara Karschay and Sylvia Ogweng, and our founder Kirill Eremenko. Thanks to all of them for producing another horizon-expanding episode for us today.

01:03:27 For enabling that super team to create this free podcast for you, we are, of course, super grateful to our sponsors. You can support the show by checking out our sponsors' links, which are in the show notes, and if you are

interested in sponsoring an episode yourself, you can find out how to do that by going to jonkrohn.com/podcast.

01:03:45 Otherwise, share this episode with people that would love to be applying deep learning to tabular data. Review this episode on your favorite podcasting app, or on YouTube, subscribe, obviously, if you're not a subscriber. Feel free to edit our videos into Shorts to your heart's content, but most importantly, just keep on tuning in.

01:04:00 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 SuperDataScience podcast with you very soon.