“A Conversation with Joe Hellerstein”

Full Podcast Transcript

Hi, I'm Craig Smith and this is Eye on AI.

Last year, I had Evan Sparks and Ameet Talwalkar on the podcast to talk about Determined AI, their startup, which aims to provide cutting edge deep learning software infrastructure for everyone. Since then, in line with their mission to democratize ML infrastructure, they have decided to open source their deep learning training platform in hopes that it will help machine learning engineers more easily build better deep learning models.

Ameet, a professor at Carnegie Mellon University and a leader in the emerging machine learning systems research community, is joining me in a series of five episodes to talk to some of his friends and colleagues about various aspects of the machine learning pipeline, from hardware and data preparation to model development, training and deployment.

As everyone listening knows deep learning feeds on data and the first step in feeding data to any algorithm is to make it machine consumable. This week, we talked to Joe Hellerstein, a computer science professor at the University of California, Berkeley, who created one of the most widely used tools for preparing data for machines. Joe talked about the challenges of data prep and how the field is evolving in the age of AI.

I hope you find the conversation as informative as I did.

Craig: Joe, why don't you introduce yourself? And then talk a little bit about how data prep is the first step in any machine learning project. And you can talk about some of the anecdotes that we've heard.

Joe: Sure. Well, happy to give a little background. I'm a professor at UC Berkeley in the computer science department where I've been since 1995.

Prior to that, I did my graduate work in database systems, and before that spent a year at IBM research. So, I've been in academia and industry and over the years been involved in startups and big companies as well; worked at Intel, been an advisor to EMC, but also startups. And in 2012, I was doing research with some colleagues at Stanford, who work in human computer interaction. And we were studying this problem that people who had data didn't seem to have the capacity to get it into a shape where they could work with it, to be able to plot it, or to be able to run analytic functions on it.

And we were thinking at the time about users, like journalists who were trying to work with data to support stories that were important to people. And we wanted to make it possible for these people to work with data in the way they would work with any other artistic medium, like video or audio. And, we were trying to wrestle with what was so hard about data. And in essence, what we discovered was that these are non-programmers, but fundamentally cleaning up data is a very time consuming and very boring programming task. And so, we asked ourselves, can we use a combination of visual interfaces and intelligent AI algorithms in the background to make it

possible for people to take their messy data, which may come from many sources, put it together, clean it up and get it into the rows and columns you need to build a chart or run a machine learning algorithm. And so that work was called Wrangler. And then we saw great uptake on the open source of that. So, we went ahead and built a company called Trifacta to commercialize the Wrangler technology. And that was in 2012 that we began, and today Trifacta's Wrangler product is offered by Google as Google cloud data prep, and it's also offered by ourselves directly as the Trifacta Wrangler product. And it's in use corporations all over the world and you can use it for free on the web as well.

Craig: And that's not only for machine learning. This is data prep for any system that ingests data, is that right?

Joe: Yeah. At the end of the day, anybody who's working with data has to deal with the messiness that's in that data. They have to make sure that things line up so that they can be analyzed. They have to make sure that things are coded uniformly. They have to make sure that outliers and strange readings are taken out of the data or rectified.

There's just all sorts of things they need to do for any purpose that you're working with data.

Craig: Ameet, can you talk about how data prep relates to machine learning systems in particular? We were talking earlier, Andrew Karpathy says that when he was a graduate student, he spent all his time worrying about algorithms. And now that he's at Tesla, he spends all his time worrying about data and data preparation.

Ameet: Sure, happy to do that. I think the, simplest, connection is just machine learning is powered by data. If you don't have access to data that you can reason about and learn from you can't perform any machine learning.

So clearly the underlying linchpin of machine learning is the data. And so, you need the data to be prepared and collected and labeled and so on before you can actually potentially use it. And kind of building on the story that you just told, you see this in academia as well.

And I'm thinking about this with my academic hat on as well, I'm also a faculty at CMU, and I'm teaching a course in the fall on end to end in machine learning. It's quite a departure from what you typically see in intro to machine learning courses, which is that in an intro to machine learning course, we'll often explain to you the 10 different supervised learning algorithms. What an SVM is, what a decision tree is, what naive Bayes is, what deep learning is, and so on and so forth. And those are all really exciting, really cool, really interesting, mathematically beautiful algorithms and methods to be thinking about. But, the general model development and training process is really large, but it's not the only part of the pipeline. I like to think about machine learning as very roughly being broken into three parts. Everything you do before developing and training models, so data prep, data cleaning, labeling, then model development and model training, and then model deployment. And even within the model development and model training part of world, the focus is really on different models, at least educationally, right?

So, the full pipeline and the full workflow is pretty complex. There's lots of steps. And it is the case that as ML people, we often just assume that the data is given to us. There's benchmarking data sets, there's ImageNet, there's CIFAR-10, whatever you have. And we

assume that it comes in beautiful tabular form preloaded, and we just can run our different learning algorithms to train our different learning algorithms on that pre-canned data. And obviously that's very different than the real world.

Craig: Joe: data prep for ML, is that different from data prep for other forms of machine ingestible data, or machine algorithms software systems?

Joe: I think the answer to that's yes and no. So probably 80% of the stuff is common, particularly when you're dealing with data that comes from multiple sources, which is what the real world is all about, is really getting data from many different places. A lot of the things you have to do for any application, whether it's machine learning, or building a chart, or running an analytic algorithm, or even just presenting the data in a meaningful table. A lot of the work you have to do there is common. You have to make sure that the codes for things are uniform across different representations of the data. You have to make sure things line up in rows and columns. You have to make sure that the units are the same. You know, there's all these sorts of things that are just basic hygiene, that if you don't do them, the data's going to be interpreted wrong.

And then for machine learning, there are some additional wrinkles, and I'll give you examples. One of them is dealing with categorical data, data that's got names instead of numbers. There's design decisions that you make in machine learning about how you want to represent categorical data. You have to encode it into some kind of numbers to give it to a machine learning algorithm. So, the challenge of dealing with categoricals is a particular challenge in machine learning and statistics.

Labeling data for supervised learning is its own set of challenges, right? If you use supervised learning, you need examples of data where the outcomes have already been provided. That's called labeling. And that's specific to supervised machine learning. That's an example where you have specialized algorithms for data cleaning and machine learning.

MUSIC: MUSIC

Craig: Trifacta deals primarily with, text-based data or numerical data as opposed to unstructured data, like images or video or audio, is that right?

Joe: Trifecta is mostly dealing with data that comes in textual form. It's not necessarily structured. So, when you think about a log file that comes out of a piece of software, it might have some structure, but it's very non-uniform semi-structured unstructured text. But we don't work with media files like video or audio.

Ameet: Can I follow up on that? I was curious, what are common examples of the raw input data that, you know, your customers typically start with that they want to wrangle?

Joe: Yeah, very common is data that comes out of software that interacts either with the real world or with people in forms. So, if you think about banks, banks are awash in data. Very little of that data is audio or video. Most of that data is swipes of credit cards. It's behaviors at ATM machines. Oftentimes it's customer interactions like web chats or transcriptions of audio chats.

So, there's a lot of natural language and quite a lot of semi-structured data that comes through software.

Craig: What are the steps of data prep, generally, and then specifically for machine learning?

Joe: So, you know, generally what you have to do when you're working with data is, if you have a single data set, then you need to get it structured properly for your usage. Very often that means getting it into a rectangle shape. Machine learning and charting packages, both like your data to be in tables or matrices, which are these, sort of, rectangular regular forms.

And that often takes quite a bit of work. So, structuring. Then encoding has to be uniform. So, you have to make sure that your units are right, that the names things are relatively uniform. If you have multiple references to the same object, you want those references to be the same. And then you need to deal with things like outliers, so if you have values that seem to be strange, you have to make decisions about what those values really represent in the real world. So oftentimes you're essentially trying to bridge the gap between what's been measured and what the reality really is for the purposes of your analytic application. So those are typical examples of things that you do.

And then again, in machine learning, as I said, there's some wrinkles. Like, dealing with null values turns out to be something machine learning algorithms are very sensitive to. When you have missing information, the statistics want to know a value. And if you say, I don't know the value, you have to come up with an encoding for that. So, machine learning is often brittle when it comes to missing data. So, you have to do what's called data imputation to guess at the appropriate values or at harmless values, you can put in. You have to deal with categorical attributes, attributes that are non-numerical. And then, with supervised learning, you have to do labeling as well.

Craig: And when you have a customer come to you and they have collected a large volume of data, how is that data generally structured at the outset? It's in some sort of a data storage unit or database? And does it have any structure at all or are you then searching and organizing, through fairly unstructured data?

Joe: Oh boy, Craig, the answer to that is it's everything. Data comes from all sorts of systems. So sometimes you get a nice file that you can read on a screen, that's not uncommon, but also very commonly the data's been locked up in some system, whether it's a database or an application like Salesforce.

So, you have to build a connector to get the data out. And then the data comes in like every format you could possibly imagine, some of which are outrageously maddening when you finally look at them. I'm reminded of data we were given, early days of the company, by Bloomberg as a challenge, and it was shipping manifests. Now these are national data that's required by the federal government, and holy moly, every line is formatted differently. You're like what is going on? After great study, you can discover that there's 32 different kinds of lines, and they tend to be in order, but sometimes lines are missing. You know, it's just remarkable how messy data can be. It's as if it's like there's an adversary out there making up ways to represent data to make your life difficult.

Ameet: Has that gotten better or worse over time? Are people adopting standards these days, or are things as bad as they've ever been?

Joe: Okay. So, I have a charming story of taking Trifacta to a startup company, which is now a very large payment processing company, and this was in their early days. And we went to them and I was in a room with a bunch of young engineers and one senior engineer who had been around in the industry for a while.

And I said, "You guys have lots of data and you're doing payment processing. You must have dirty data that we could help with." And they said, "Oh no, we have no dirty data because the data comes from our own systems. And whenever the data isn't the way we want, we tell the engineers who build the systems to change the way they represent the data so that we keep it uniform." And the older person in the room just kind of smiled and said, "Yeah, that, that works for now. But you know, later on, I assure you we will have these problems."

The story of that anecdote is basically that in any reasonably mature environment, you have a million formats of data and each one of them made sense to somebody. But by the time you get working on it, you've got a hodgepodge.

Craig: And then how does Trifacta handle that? Or how does anybody handle that? You don't lay it all out and have a bunch of guys looking at it and identifying anomalies. Does Trifacta scan the data and highlight anomalies? Or does it do an analysis, then, that you use to tell it how to massage the data further?

Joe: Yeah. So, we have a strong point of view on this idea that data preparation is a collaboration between human intelligence and AI. That basically humans can't do all the work themselves. It's too tedious. There's too many things to keep track of. At the same time, there's no automated algorithm that takes input data and produces good data. You can't say Siri clean my data and expect to get anything good out. And the reason for that is essentially that data is a medium. You use it like you use clay on a potter's wheel, you shape it for purpose. And so, depending on what you want to do with the data and depending what's in the data, you make decisions, design decisions, essentially, about how you're going to transform that data for use. And so, you need to be in dialogue, essentially, with these algorithms that are analyzing the data to try to figure out what the outcomes should be. And so that's a sort of lovely dance between analytic algorithms in the background and human insight in the foreground. At the end of the day, you want the people who know the data best making decisions about what the data should look like.

Ameet: I see. So, is that kind of where the HCI [human–computer interaction} comes in, to allow a human to interact as the algorithms behind the scenes are trying to process the data?

Joe: Absolutely. The biggest challenge, I think, that we were facing in the research and in the product with Trifacta is how do you get that collaboration between the human domain knowledge about what the data is being used for and their intent, and the algorithmic ability to look in the data and summarize and detect patterns?

You need to build a bridge there. The bridge is typically going to be visual. It's very hard to imagine doing this, say, with a voice recognition system. But with enough visualization, you can

build interfaces where the algorithms that are analyzing the data and the people who want the outcomes can begin to be in a feedback loop with each other.

Ameet: Is that sort of like visual unit tests, or visual testing of some sort to eventually convince the user that the algorithms are doing the right thing and they can trust the data that's coming out?

Joe: Yeah. What's interesting about it is it's a bit like software engineering, but it's a bit like media editing, like film editing or audio editing in the sense that you're looking at this thing. You really want to kind of poke it and shape it in ways that are pretty visual. But at the end of the day, you're building a program, a script to transform the data using algorithms. And so that back and forth between kind of the visual intuition and the programmatic expression of a recipe for cleaning the data is where some of the magic lies.

Craig: And the visualizations, are they curves on a graph? What kinds of visualizations are they and how does the user manipulate them? Do they literally manipulate the shape of the visualization and that's transferred to the data? Or are you setting parameters and watching how that affects the visualization that you're working with?

Joe: So, you actually asked a question about visualization, and then you asked the question about interaction and those are actually two different things that are both hard. For visualization, I have a colleague, Maneesh Agarwala, who's a professor at Stanford in human computer interaction. And Maneesh made a very good point, which is that a table, like a spreadsheet or a table in a book, is a form of data visualization. And it's a very powerful one. So very often you want to see some of the data, like explicitly a sample of the data. So that's one kind of visualization that's traditional, extremely powerful, and useful.

Another thing you'd like to see is, say if your data lines up in some columns, you might want to see summary statistics of those columns. And bar charts are surprisingly effective; histograms for numeric data, or sort of most frequent value bar charts for categorical data. And then sometimes you want to see multi- attribute or multi-variate as they call it, visualizations of how columns might correlate or rows might correlate.

So, we use a mixture of visualizations in Trifacta, but it becomes very familiar and seamless after a while because of the HCI work that's been put in. I think the key is simple visualizations are often much more powerful than exotic ones.

Craig: And then how do you adjust the underlying data? You see the visualization, you see that something's out of whack, it isn't where you want it to be. Do you then go into the data, and find the correlating data and adjust it, or is there some automatic link between the visualization and the underlying data?

Joe: So, we took a fairly principled, but also fairly opinionated stance on what we thought a good interaction model was for working with data. And this followed years of experimenting and research and user studies. We use a technique we call predictive interaction, and the idea is that someone should be able to gesture on the screen at something that looks interesting to them. "That looks wrong. That looks good." And as they point at things on the screen, the AI algorithms can come in and take that interest of the user as well as the data that the algorithm is

looking at and suggest ways to transform the data based on that interest. So, if you point at an empty cell, you're telling the algorithm something about your interest in empty cells, right? And then, rather than the user sort of trying to manipulate the data, the way you'd manipulate a video file, you know, by cutting and trimming and so on, it automatically suggests outcomes. So, it'll show you previews of transformations of the data if you were to take one or more steps. And then for each of those previews, there's an associated sort of English description of what it's going to do to the data. We call this the guide-decide loop. So, the user guides the AI algorithm to interesting features in the data, the AI algorithm presents a ranked list of suggestions for what to do next. And then based on visual representations of those suggestions, the user can decide what their next steps should be.

Craig: And if you're dealing with, let's say it's fairly structured, but a million lines of tabular data with a hundred columns, you're not going to look through it to find things that stand out. So, I was asking that before, does the system spot anomalies, spot potential problems and bring them to the user's attention?

Joe: This is one of the places where having a combination of visualizations is very effective. So obviously a table with a million times a hundred cells is hard to spot things in, but if every row or every column has a chart with the outliers and the mean and the median and so on, you know, the core tiles, then you can quickly eyeball for each column what might be interesting, what might be wrong. Similarly, and this is all too common, you have a column, it's mostly numbers, but some of it's not numbers. And so, you could scroll around looking for that. Or you could have a chart at the top of the column that basically says, this is 90% numbers and 10% other stuff. And then you want to interact with these charts as you interact with the data. So, if I click on the indicator, the red bar that says this is bad data in this column, then it'll make suggestions. "What would you like to do with this bad data?" And again, like it zooms into context then on that data and makes intelligent suggestions.

So, a combination of visualizations is very powerful, and then the ability to interact with different kinds of visualizations in a uniform manner that just says I'm interested, become something that users are very able to do very quickly.

Craig: I'm fascinated by this idea of data as a medium that you manipulate. People tend to think of data as ground truth. But for example, in these suggested ways that Trifacta offers up to deal with different anomalies or different potential problems in the data. Each one will have a different effect on the outcome. At some point, it really is an art on the part of the data wrangler or the data scientist preparing the data. Is that right? Because you could have very different outcomes depending on how you treat these different situations.

Joe: Yeah, I think it is quite a bit of judgment that goes into data preparation and it absolutely has effect on outcomes down the line. I don't want to over emphasize it in the sense that some algorithms are more robust to dirty data than others. Some tasks you're trying to do are more robust to dirty data than others. So sometimes you want the quick and dirty and you take the data and you just kind of get it to be in an okay shape and you move on. Sometimes you really need to do creative things to the data to be able to see what you want to see with your algorithms.

So, it varies. But It's a big word, art. I don't think it's always required. Sometimes it is.

Ameet: I have two follow-up questions there. One is, does the process that the wrangler is going through using Trifacta end up ever leading the wrangler to realize that they need to collect more data or ingest different sources of data? That there's something fundamentally missing to get the data to look like the way they want it to be?

We have something similar in machine learning called active learning, where instead of labeling all your data at once, you label some of your data, and then you realize, okay, this subset of the space is not well understood. So that's the area we should focus on and gather more labeled data. I'm wondering if there's an analogy in this sort of setting.

Joe: Interesting. It's certainly the case that you never know your data as well as when you're wrangling it into the shape you want it to be, for using it in your next step. Your head's completely in the game at that point, with the data. And so, it's actually a very powerful time. It's sort of like when a software engineer is deep in their code, you really have a context on what's going on and you lose that context very fast the next day, the week after you can't remember anymore. And so, yes, while you're in there, you notice all kinds of things about the data, what's wrong with it, what you're missing, and so on.

I think the idea of active learning that you bring up is an interesting one, because active learning is a statistical method, right? It says, "Here are places where we have high entropy that we want to resolve." And that's something that we would treat as more of an analytic task and less as data preparation. So, what this leads to though, which you alluded to before, is you may have a pipeline -- data prep, training, deployment -- but that pipeline is a loop. You do your data prep, you do some training, you realize your model is not very good. Why? Well you go back? And you see there's really high entropy in some of your data and you might use active learning to try to augment it. And then you start the pipeline again. So that looping between all these stages is super valuable in the machine learning life cycle.

Ameet: My second question, as you mentioned, this isn't just an art, it’s somewhat, it's maybe more reproducible than that. But my question is related to reproducibility: given that some decisions that the wrangler needs to make, in terms of how to manipulate the data and fill in or resolve ambiguity, is there any sense of reproducibility? Like if the same or a different data wrangler took the same set of data and went through the process on different days or, you know, different people on the same day or the same person on different days. Do they typically end up with the same outputs?

Joe: So, there's, I think, two aspects to this that you're alluding to. The first is simply, is the work that I did yesterday actually reproducible, like deterministically? So, if you edit data in a spreadsheet, which happens all the time, you'd be surprised how often that's how people clean their data, you lose that. That context of what you did. You started with data, you changed it, and now you have a different spreadsheet. The end.

So, it's very important that whatever tools you're using or whatever processes you're using documents your data transformations in a programmatic way that you can rerun. And so, absolutely in Trifacta, it's essentially generating code out the bottom, and that code is a data

wrangling, we call it a recipe. It's a program, for manipulating the data, and folks who are software engineers do this by nature. Right? They write data wrangling programs. So that's really important. And it's kind of table stakes, I think that's basic. If you're going to do a governable, reproducible data analytics or machine learning.

A second question you asked, which is, I think, deeper is: suppose you took two different people, you gave them the same data set and the same task, and you locked them in two separate rooms. Would they prep the data in the same way? I think the answer to that's almost certainly no.

Craig: And, depending on how robust or how sensitive the software algorithms are, you would get different outcomes.

Joe: Yeah, that's correct.

MUSIC: MUSIC

Craig: Can you talk about the role of labeling for supervised learning as a subset of data prep? And does Trifacta handle that at all? Or is that just one of the kinds of data that Trifacta handles?

Joe: So, you know, supervised learning, traditionally, you have a data set that's been augmented with outcomes, with labels, right? Of what the predictions should be. And in the traditional setting, those came organically in some way, hopefully. So, the labels came because, for example, they were part of a process and you're trying to predict how that process will go in the future, but you have the past data. And so certainly we see lots of data that people are using for supervised learning that has labels in it already. The separate issue of, "I have unlabeled data and I would like to extract labels from it," is not something that Trifacta focuses on. But there are two general approaches to doing this. One is to get humans to label the data. The second is to write what are called weak supervision scripts, which is like little scripts that you might write that are probably pretty bad at labeling the data, but they make some sort of sensible guess. They're like rules, essentially.

Ameet: So, we're talking to Alex Ratner in a future podcast.

Joe: Alex did his PhD in this area of weak supervision. And it's really interesting work. So, I'll leave it to Alex to explain the technical parts, but basically any piece of software can host these little rules for labeling the data. The interesting bit is mixing them together to get a statistical signal. And that's the kind of work Alex was doing in his PhD and is doing at his company.

Craig: Do you have any experience with synthetic data? I had a fascinating conversation yesterday with a guy that's working on building synthetic data sets, mostly visual, but also still images. I guess in that Trifacta is only dealing with non-visual data, that wouldn't matter. But then again, I would presume there's other forms of synthetic data that are not visual.

Joe: There's a long tradition actually of test data generation being a product feature in the data preparation ETL [extract, transform, load] space. Everybody needs test data to see if their scripts are working and their systems work end to end. If you want to show demos to people and you don't want to use private data, you need made up data. You can go to a website called

Mockaroo.com and build yourself a surprisingly rich table, full of names and emails and all sorts of stuff, and it looks very plausible. That's like a fairly traditional space. Here's the thing. People are very bad at recognizing whether a table of data is realistic or reasonable. People are uncannily good at recognizing whether a video is from the real world and is reasonable.

On the flip side, what that means is that people are very bad at cleaning up data. But they're actually reasonably good at recognizing imagery and audio. And so, the generation versus recognition tradeoff here is quite different.

Craig: That's interesting. How does the prepared data then connect to a model? Whether or not it's a machine learning model. Is this through an API to the cloud? Is there some tighter connection where Trifacta actually launches the data training or the next step in the pipeline?

Joe: Yeah, there's as many end-to-end engineering processes as we have customers, essentially, for taking data from raw to refined and then giving it to downstream systems to execute on. Some people use workflow systems. Some people use our built-in connectors, some people write hand code to do this. Some people like literally, you know, put a file into a well-known space and hope that somebody later on will get that file and use it for something else. It's just a wide variety of automation techniques for doing this end to end.

We offer some of that in Trifacta. You can self-service schedule your data to be cleaned as it drops in, and you can hand it off to downstream applications, publish it using Trifacta. But I would say, a large fraction of people have their own processes for doing that. And I'm sure Ameet can talk about that as well.

Craig: This data, once it's prepared, it's valuable. Are there markets where you can trade this prepared data so that people don't have to go through it over and over again? At least sort of generic data.

Ameet: I think that's a super interesting question. I'm sure there's more than one answer to this, but to me, the first thing that comes into mind is kind of the cost of data, and privacy considerations, and also things related to kind of meta learning. So, it makes a lot of sense to share data and to use data across multiple organizations, multiple individuals, or what have you.

But there's a question of how to incentivize organizations to actually share their data. Mike Jordan, a professor at Berkeley, gave an example of this a few years ago at a conference talk he was giving about, imagine you have three big credit card companies and all of them are collecting a lot of data on individuals. If they pool their data together, you can imagine all of their risk models could potentially be better. But there's privacy considerations, but there's also incentives that really have to be considered in terms of why would Amex share their data with MasterCard?

What happens if the Amex's data is better than MasterCard's and Visa's is better than both of them? There's actually some interesting work by some folks at Berkeley, Dawn Song and her group, trying to quantify the value of data, which might lead to help people incentivize how they should share and how they should be monetized, or incentivized monetarily to share their data.

But think the privacy concerns are kind of big there.

Joe: I have actually a number of real-world examples of this, but they are constrained. So, the typical example is financial markets. In the US, Bloomberg, and in Europe, Deutche Boerse, who's a Trifacta customer, they sell data sets all the time. In fact, basically, for Bloomberg it's most of their business. And then another example, another Trifacta customer, is a company called Nation Builder. They went to the trouble of going to all the county courthouses around the United States, getting the voter rolls. All of those are completely different formats, cleaning that all up and then selling it to political campaigns, including all the presidential candidates in the last election, except for the democratic nominee. And that was their entire business.

Ameet: And so is the idea there that you're starting with data that is publicly available, but it's so painful to scrape that nobody's able to do it. So, if you do that, that then becomes very valuable.

Joe: Yeah. So, the key example of Nation Builder is exactly that, they're selling the benefit of them having done a ton of work. They are absolutely selling the fruits of their labor. It's public data, and they're just working super hard to bring it into a uniform format.

Craig: And who does this work? The data prep work. Is it done by the data science team at a company that has collected the data? Is it outsourced to people like Trifacta? And what are the skill sets? How do they fit into organizations, and how do they relate to folks doing machine learning or deep learning?

Joe: One of the things we really wanted to do in our research back on campus and then when we started Trifacta, was empower the people who actually were going to get value out of the data to do their own data preparation. And to not rely on programmers or IT professionals to get the data into shape.

And I think to a large degree, we've succeeded at that. So, we see customers of Trifacta who are not programmers, who are not necessarily in the IT department. They're in a part of the business that generates value, doing their own data preparation. The category actually Trifacta is in, the data preparation category, was originally called self-service data preparation to emphasize this point. So, we see a lot of that, but inevitably in lots of organizations there are teams of experts who are really good at working with data who are tasked with this. And that can work well, but oftentimes the social distance between the person who needs the data and the person whose job is to wrangle data leads to inefficiencies, right? I say, "Please clean this data up for me." You give it back to me a week later. It's not the way I want it. My questions changed in the interim. I go back to you, I say, "No, clean it up differently. I want something a bit different." And we iterate on that for weeks. if I'm doing that myself, it could be a daily task that I could even change my mind about on a regular basis.

Ameet: Thinking about this as a business, coming from Berkeley, there's a huge tradition of open sourcing everything. And for our startup, we recently opened essentially our entire product. And it sort of felt like, in the machine learning field right now, if you're targeting machine learning or deep learning engineers, they pretty much demand things to be open source. That's just the way the entire community is working. But kind of a key difference is that these folks are all engineers. And so, it'd be interesting to hear your thoughts on how you were thinking about open sourcing or not, when you started Trifacta and what the implications were.

Joe: Yeah. Before I get into this, I should say that, some people come at open source with a sort of fan kind of outlook or sometimes even a religious outlook. And I want to make clear that I cut my teeth on open source. My PhD was part of the Postgres project. So, I was one of the Postgres kids when we were building Postgres back in the nineties. I'm super proud of the adoption of that software over time. Warms my heart. Having said that, mostly, what I've seen over time is two things. One is that open source today is typically a business model as opposed to an inherent good. A lot of the successful open source projects you've seen now are supported by large well capitalized companies.

Secondarily, open source works best when it's in a community where people are contributing back to the software. That means that the software has to be of service to that community. In essence, they're scratching their own itches. So, you've got software engineers who are building software for software engineers.

That's when open source really works. There are a lot of people whose job is not software engineering who need software. Right? And so, part of the sort of hypothesis of the work we were doing back on campus was there's all these people who work with data who are not programmers. How can we serve them?

And the answer is not going to be, we'll build an open source committee and they'll help build the software. That's not how it works. And so, we set out to do it, and we ended up not doing open source because as a business model, it made no sense. Now having said that, if you look at open source, you'll also see very little of it is user interface oriented because coders are going to code, which means that most people writing open source tend to write text interfaces to it that you type at. And so, if you expect to see excellent user experience in open source, you'll often be disappointed.

Ameet: That very much resonates with me. I feel like the reason to a large extent that ML folks want to be using open source technology is because they want to be able to, if not at least contribute to it, be able to influence it in some way and feel like they're part of that community. If they fundamentally didn't feel like they were part of a software development community in any way, I think things will be quite different.

Joe: So, I'm curious about that. Let me ask a question back to you, which is when I think of the sort of larger data science community, particularly in the field, a lot of those people were not trained in computer science and they're amateur programmers at best. They're data people, they're statisticians. How does that relate to this hypothesis about the machine learning community?

Ameet: I think that the data science community is kind of somewhere in between. It is certainly the case that, for instance, what we're building is somewhere at the intersection of machine learning and just building a fundamental distributed system for machine learning and our target user is not going to be a distributed system engineer or a systems engineer in any sense. Said another way, a deep learning engineer, a data scientist can have a very varied amount of software engineering skills.

Some of them are quite strong, and some of the early adopters who were doing this stuff today kind of need to be quite strong because there's a fair bit of heavy lifting to actually get these things to work in production. Over time, I would imagine that that's going to move more and more away from folks who are stronger stuff for engineers. These things need to be more abstract, move away from being traditional software and more towards higher level APIs that someone with the background you were describing is actually able to access.

I'm just curious, Joe, with your research hat on, what do you see as kind of the big open questions with data prep these days?

Joe: I mean data prep is sort of the gift that keeps on giving. There's always more hard problems in it. So, another way to say it is it's miserable. But as a technologist or a researcher, there's always more to do. I think the idea of appealing in a uniform way to people who do want to code and people who don't want to code in a single platform is really interesting to me right now. So, I'm spending a lot of time actually circling back on the Trifacta experience and asking, "How would coders want to approach the benefits of these AI algorithms in the background that do the analytics in their coding?" So, I'm personally excited about that.

Ameet: And is that developer then more like a machine learning engineer? Or is there a different persona that you're thinking about?

Joe: Right now, I'm thinking about that machine learning engineer. If you spent any time doing data wrangling with a library, like Pandas or R, you know, it's just, it's boring and it's super time consuming and you write really brittle code that tends to break.

And yet people are doggedly doing it that way. They're not using tools. And there's a reason for that. So, what is that reason? Trying to understand that better. Try to meet the programmers where they live, and make their lives better.

Ameet: That makes sense.

Joe: I'm kind of curious, Ameet, about the longer-term iteration in the machine learning life cycle from where you sit. So, this idea that like data is not a one and done, right? Data comes in over time, formats change, models topic drift, models stopped working, et cetera. This is this whole longer loop between the kinds of things that you spend a lot of time thinking about, the kinds of things that I spend time thinking about, maybe we can tee that up somehow.

Ameet: I think just generally the life cycle or that loop is exactly what we see. You don't go through the step kind of linearly. You have your data, you train your model, you deploy it, and you're done. I know less about the loops during the data prep process that are internal just to the data prep part, but it is certainly the case that while you're developing models, you don't take your data as fixed and then call it a day.

Right? You typically see gaps and holes in your data, or you're training on a subset of data, knowing that new data is going to come in later. Or, for some of the customers we're working with, for instance, they have sensors that are collecting massive amounts of data every day. And so, they're kind of retraining their models every day. One way the cycle works is that you're just constantly getting influx of new data and you constantly want to be checking your models.

Another way the cycle works is that, at deployment time, you're monitoring in some way. And as you're monitoring, if some alarm bell goes off because all of a sudden you notice drift in the form of, say, your model's accuracy has gone down or something. You have to go back and figure out what went wrong and retrain from that. So that's conceptually how things work. To be honest, this process is very far from being smooth, right?

I'm describing how the world would work in an ideal setting where all of these things were automated and there were natural alarms that went off. In an automated fashion, you could just perform this process. Right now, it's completely manual.

There are fledgling or emerging bigger and smaller technologies to help with model deployment and modeling. They're not necessarily as well connected as they should be with the training aspect or with the data prep aspect. So, these things are more siloed than they need to be.

And we think, for instance, in the context of Determined, that model development and model training is really important, and timely in part because data prep, while it's the problem that keeps on giving, it has improved a lot or especially in the domains where model training is a really hard problem. But making that full cycle much easier and more automated is a work in progress. We're not there yet.

MUSIC: MUSIC

Craig: With the advances in deep learning, will there come a day when you throw a bunch of data at a deep learning model and tell it what you want that database to be prepared to do. And it finds all the outliers fixes them and basically you press a button and you get clean data. Is that science fiction?

Joe: I don't think that's science fiction. I just think that it's maybe not the right attack on the problem. So, if we wanted more automation, and I believe that we'll get it, I think you still want it in this context of reproducibility and programmability. So, I think the idea is you give data and a description to a model and it generates a data cleaning script for you.

Generates a recipe. And then you can walk through that recipe visually or programmatically. And you can say, yes, this looks right. This looks right. I like the output and it'll work the next time I load more data. So, this is somewhere where the explainability of the model doesn't have to be as good if it's output is a process instead of its output being just data.

So, I'm very optimistic about that. And in essence, it's just an extension of what we're doing today to longer steps. Rather than one step at a time, we're asking the AI to generate a whole series of steps to clean up the data. And I do think that's very tractable and also it can be attacked piece by piece and get better over time.

Ameet: Do you still envision that humans are in the loop just maybe less interactively in that sort of setup or is it fully automated?

Joe: I think humans are always in the loop if only to examine the outputs and say good or bad. Right? Given that this is probably a process that's a many step outcome, humans can intervene at different points in the process.

I'll point out, there's a name for this. This is called software synthesis or program synthesis, and it is what Trifecta is doing, and we're just doing it on a finer grain right now. Sort of a line at a time. But the program synthesis community is doing very well. It's growing. And I think over time, we'll see it generating much more rich data cleaning outputs from much more interesting specifications of input.

Ameet: We're seeing something similar with program synthesis in the field of auto ML. There was work by folks at Google recently trying to automatically generate programs to kind of learn gradient descent or to learn learning algorithms. And it required a massive amount of compute and the outputs are still pretty modest these days, but it's pretty impressive that you can do anything in that sort of area.

Can you say a little bit about the type of ML or AI that you're using in Trifacta to help do the wrangling?

Joe: We use like many dozens of models inside Trifacta of different sorts to do different things. I'll give you an example of the tasks we're trying to solve. So, one of them is type induction. You have a column that's got a mixture of different kinds of data in it, and you have to guess at a data type or a set of data types for the column. That's one example. Another example is this predictive interaction stuff. So, you click on a value in a column. Maybe it's a high value. And I have to infer a ranked list of program steps to recommend to you. Maybe you want to delete it. Maybe you want to set it to the mean maybe you want to set it to null. So, the choice of task. Another example is you highlight a bunch of different text snippets in a column, and I need to learn, say, a pattern, a regular expression based on your examples.

So, there's lots of different little models in there. Some of them there's been research papers written on them. So, the programming by example with regular expressions, that's a topic of many papers. So, you know, it's not like we have one magic algorithm that cleans your data by any means. There's a lot of tasks involved in cleaning data and they're somewhat different right now.

Ameet: it sounds like the main kind of persona that you guys have been dealing with is not necessarily somebody who then wants to use their data for machine learning. And in particular, right, labeling is sort of a complementary thing to what you're doing. But we're seeing more and more in ML that unsupervised models are really powerful, maybe not as end models, but as starting points that you can then specialize or adapt for your particular setup.

I don't know if you've been following Twitter, you must have been, about GPT-3 and the incredible texts that comes out of it. I'm wondering if you see, with unsupervised data becoming more important for machine learning, if that changes things for target users for Trifecta.

Joe: Well, I think in the decision tree of persona differences here, I'd say the most important thing is, are they a serious programmer or not? And if the answer is they're not a serious programmer, they still might be involved in a machine learning life cycle using AutoML products or tools. And there's some that are reasonably successful now. DataRobot's a company that does pretty popular AutoML product. H2O does these things. And so that space is where we've

been engaging more because it's a similar user persona. And whether the model is a supervised or unsupervised model is an orthogonal cut.

Now GPT-3, I mean, we got to talk about it, right? It's very exciting. very impressive. In fact, including doing pretty interesting software program synthesis, right?

So, I do think that labeling is very important to us today because we have lots of supervised learning algorithms. Whether labeling will continue to be important and to whom is an interesting market question that I don't have an answer to.

Craig: That's it for this week's podcast. Ameet and I want to thank Joe for his time. If you want to learn more about what we talked about today, you can find a transcript of this episode on our website, www.eye-on.ai. We provide a scrolling transcript so you can listen and read at the same time. We love to hear from listeners, so if you have comments or suggestions, feel free to reach out.

And remember the singularity may not be near, but AI is about to change your world. So, pay attention.