EPISODE 1388

[INTRODUCTION]

[00:00:00] KP: Once a machine learning model is trained and validated, it often feels like a major milestone has been achieved. In reality, it's more like the first lap in a relay race. Deploying ML to production bears many similarities to a typical software release process, but bring several novel challenges like failing to generalize is expected or model drift. AI quality management might be the biggest challenge in AI today.

In this episode, I interview Anupam Datta, co-founder of Truera. Truera has a solution aimed at helping AI performance, monitoring, and model Explainability. We talk about some of the challenges of modern machine learning deployments in production and how companies are succeeding with ML ops.

[INTERVIEW]

[00:00:48] KP: Anupam, welcome to Software Engineering Daily.

[00:00:51] AD: Thank you, Kyle. I'm happy to be here.

[00:00:55] KP: Well, in addition to being a co-founder at Truera, the makers of AI Lens, you're also an academic. Tell me a little bit about the start of your career.

[00:01:03] AD: Yes, so I started out in grad school at Stanford in computer science. And after finishing up a PhD there working in topics around privacy and security, I joined the faculty at Carnegie Mellon back in 2007. Over time, my interests transitioned from privacy into the intersection of privacy and machine learning or AI. And gradually, a couple of themes bubbled up to the very top, one was around understanding the fairness challenges associated with machine learning, as machine learning and AI starts gets deployed at scale, in a wide range of areas.

© 2021 Software Engineering Daily

If you think about our starting point was actually a study of the Google advertising system back in 2014, where we discovered gender bias in the way employment opportunities were being targeted. And around the same time, similar kinds of discoveries happened as people looked at fairness in the context of credit decisioning systems, fairness in the context of recidivism and predictive policing, healthcare.

So, across a wide range of areas, it increasingly became clear that as machine learning starts get deployed at scale, questions of fairness are going to be extremely important to have guardrails around and deal with. The root causes of why machine learning models can sometimes produce unfair outcomes lead us to a deeper study of Explainability in the context of machine learning. So, at a high level, that means that we should be able to look at individual predictions coming out of models, or aggregate trends when a model might be, say approving women at a lower rate than men, and be able to understand what's causing the disparity. And if it is, in fact, forms of unfair bias, then how to mitigate them. That exploration of Explainability, and root cause analysis eventually led to a significant body of work around understanding machine learning models, debugging them, maintaining their quality over time, and that led to the formation of Truera.

[00:03:38] KP: I'd love to dive into some of those use cases. You'd mentioned being able to discover that there was gender bias in these employment ads on Google. I'm going to guess Google didn't just give you access to their database to figure all this out. Could you talk a little bit about how you study this scientifically and how you get to that result?

[00:03:56] AD: Absolutely. So, the starting point of the study in that, where we looked at Google's advertising system, we almost studied it, much like biologist's study mice where through a set of randomized controlled experiments. So, my wife is a biologist and when she studies whether certain genes are protecting against cancer, and she will have an experimental group, which has the regular mice, and then a control – or an experimental group where that gene is turned off and a control group which has the regular mice, and then she'll observe over time, whether turning off that gene results and the experimental group mice getting cancer.

We did something akin to that, as we studied the Google advertising system. We had an experimental group and a control group. These were simulated users that were largely identical,

except in the experimental group, we were able to set the gender to be female. And in the control group, we set the gender to be male and there was something called Google Ad settings that allowed us to control and set the gender in that way. And then, there were 500 simulated users, female users and 500 simulated male users and they went around and visited a lot of job related websites as a way to signal to the Google advertising system, that they are all interested in employment opportunities.

All of these simulated users went and visited other kinds of third-party websites, news websites, and so forth. And the system collected the ads that were targeted at these two groups, and systematically studied their differences. One of the things that came out of the study was that there was significant difference in the ads that were targeted to men versus women. That by itself may not be surprising. It could be that gender plays a role in the kinds of clothing ads and so forth that get targeted. What was more surprising, and somewhat concerning, was that some of the top ads, which were shown in significantly higher numbers to men, relative to women had titles like 200k plus exec shops. And that was where the alarm bells started ringing, that there was some amount of association between gender and the ads that were being targeted.

Now, that study was raised or helped in bringing to attention, the observation that large scale systems, not intentionally, but because of the way historical data is used to train machine learning models could be picking up on and reinforcing some of the disparities in targeting employment opportunities. But it did not answer, and that's where the next body of work that we worked on started getting into was, what is the reason for this disparity? What is driving differences? And this case, ads that are targeted, but more generally, other kinds of high stakes use cases around credit decisioning, healthcare, many other areas where these disparities might unintentionally get picked up and reinforced by machine learning models. What are the reasons for those? And how do you mitigate those problems?

[00:07:28] KP: So, there's a certain point of view that might say it's garbage in garbage out, that we have a systematically sexist, systematically racist society, and the machine learning model gets the training data from the real world. What can we do?

[00:07:44] AD: That's a very good question. I think one way to think about the problem is, it is true that machine learning models will pick up and reinforce historical bias in training data, the

data on which they're being trained. That said, I think what we do need to acknowledge and work towards is there is a lot potential here for large scale societal harm. That's the first observation, whether it's advertising systems, credit decisioning, recidivism prediction, and other uses in the criminal justice system. Our facial recognition systems, which have also been shown to have significantly lower accuracy, sometimes for minority groups, which are not that well represented in the training data.

I think the question that you're asking is extremely relevant and that, once we have recognized that there is a problem here, how should we address it? In my mind, the solution to the problem is to understand the root causes of what's causing the disparity in the performance are fairness of the machine learning models. Unpack that and then have targeted ways of mitigating the problem. So, just to give you a concrete example, if you ground our conversation on some concrete examples, let's talk about facial recognition systems. There has been a set of studies showing that with facial recognition systems, they sometimes end up performing really well with high accuracy for white males. And the accuracy tends to go down considerably for other groups. For example, for African-American women.

Now, the root cause of that is that in the training data, white males are represented at a disproportionately high level, whereas some of these other groups have very little representation. And just the way machine learning works, it's going to try to do well on average. So, if one group has very high representation, it'll end up being very accurate for that group, while ignoring the accuracy requirements for the other group. So, one way to address this problem is to ensure that the training data is diverse, and representative, so that accuracy levels are balanced across different groups.

That's one concrete example of where the root causes and how to address it. There can be other kinds of root causes. This is not the only way in which bias and unfairness creeps into models. Let me give you a different example. Let's think about an example where the model is accepting, we observe that the model is accepting men at a higher rate. This is a credit model, and it is approving men at a higher rate than women.

Now, typically, these models would not be explicitly given gender. Gender is not an input feature into the model. And at the same time, when you see this kind of disparity between the two

groups, men being approved at a significantly higher rate than women, then you may want to drill down into the reasons for the disparity. So, that requires technology capabilities for root cause analysis and that might surface that one of the big reasons that men are being improved at a higher rate than women is that they have a stronger record of continuous employment. Let's say continuous employment is one of the features that is fed into the model.

Now, if that happens, at that point, there is a judgment call to be made on whether continuous employment is something that you would want to use effect in making decisions about credit. It could be that if the credit data is coming from a long time ago, and historically, women tend to take more time off during childbirth, and also to take care of aging relatives, and which in turn impacts the continuous employment feature, and if that data is driving the models, predictions, and let's say, currently, that trend is changing, then you might decide to not use that feature, which ends up being a proxy for gender in this case. Certain kinds of proxies are already disallowed from use in the US like geolocation, zip codes and information of that sort, which are known to have historical associations with race, for example.

So, that's another way in which you could address the problem diving into the root causes and making sure that certain kinds of data attributes or features are not used as part of the decision making process, if they contribute too much towards accelerating or reinforcing historical disparity, without serving a very good purpose of meeting the business goals around separating, in this case, good credit risks from bad risks. If on the other hand, disparity is being driven by differences in income, or differences in net worth, which are known to be strongly associated with credit worthiness, then even though there's a disparity, it's justifiable. It's not being caused by the machine learning model. There are other structural inequities in society that need to be addressed through addressing differences and gender pay, for example. So, the flow there, there is perhaps a problem, but the problem is not with the machine learning model, it's at a different stage of the pipeline in the world.

[00:13:53] KP: Well, machine learning has been exploding and how much it's being deployed in use places. Everyone's taking an interest and seeing where they can get it working, and it's doing some surprisingly good things in a lot of applications. So, there's a big market here and I think people are becoming aware of these problems, but may not know how to exactly address them. Can you talk about how Truera helps companies solve this type of problem?

[00:14:17] AD: Absolutely. So, Truera is a platform that helps with testing, evaluation, and monitoring of machine learning models to ensure that they are of a high quality before they move into production. And that that quality is maintained over time through monitoring and updates to models. So, in some ways, I know a lot of your audience here comes from a software engineering background. If you think about the state of the world, we are in right now, with machine learning and AI, it is quite analog as to how software develop worked in the early ‘80s. When there weren't good frameworks and platforms and tools to support systematic testing, evaluation and monitoring, we did not have CI/CD pipelines at that time, we did not have application performance monitoring. Those are some of the big whitespaces. That's the area where there is a gap now in the tooling for machine learning development. We have seen a lot of advances in tools and frameworks and platforms to support training and deployment of machine learning.

The next big hump to cross is really having very good and systematic tools for testing, debugging, and monitoring of machine learning models. That's where the Truera platform helps users and practitioners of machine learning. We have two products on the platform. One is a diagnostics product that helps with testing, evaluation, and Explainability. For machine learning models, during the development process, you can think of it as a form of quality assurance before you move models into production. And then once models are in production, we have a second product, a monitoring product, which helps with tracking various kinds of model metrics related to model performance, fairness metrics, as well as model drift, or stability related metrics. There is a nice flow between these two products.

‘So in the monitoring world, if the model is starting to degrade along a certain attribute or dimension, be it performance or fairness or drift and stability, alerts can get sent automatically to the data science teams, and then they can quickly come in and debug and understand the reasons for why these metrics are out of whack, and how to readjust the models to make them high quality again.

[00:17:16] KP: So, when I sit down to build my own machine learning model, new greenfield project, I've got a lot of choices that are going to be affected by my experience in the languages

I use. Maybe I build it in SK Learn or Spark ML or TensorFlow, I've got a lot of options. What is your platform support?

[00:17:33] AD: That's a great question. So, one of our driving principles and developing the platform was to make it universal and agnostic to the platform in which the models are being built. So, there were serious architectural and engineering decisions that we made to ensure that no matter what platform you build your models in, you can still use that we're a platform to evaluate and monitor it. So, in particular, concretely, to go back to your examples, we have native support for Python models, so people who are building their models in SK Learn, extra boost is used very widely, like GBM, a whole slew of Python model building libraries and capabilities. We support natively as part of the platform.

We also support all the popular neural network building frameworks like TensorFlow, Pytorch, and Kairos. We also have support for proprietary platforms from which you can export models into standardized format. So, for example, people who are building their models in DataRobot or H2O, they can export models into these Java model formats, which can also be then evaluated using our platform, people who are building models in R, or SaaS, there are ways of exporting those models into standard formats like PMML. So, we are evaluation and monitoring platform is truly, truly universal.

That's actually a very good point that you're bringing up, Kyle. Because one of the things we're noticing, as we are working with data scientists and enterprises, is that often within the same enterprise, different teams are building their models in different frameworks, and having a single pane of glass to explain those models to test and evaluate them, to monitor them, is emerging as a very strong need.

[00:19:35] KP: So, once I've gotten to the point where I have some binary asset, I think I've got my model finished. I want to run it through your diagnostic tool. Do I upload that or do I use a library? What's the actual connection to get the insights?

[00:19:48] AD: Great question. We have several different form factors. We have a version of the product that we can deploy in a containerized form on the enterprise's, on VPC, meaning that their data and models, they get ingested into our platform, but our platform stays within their

computational infrastructure, which can be either in a cloud VPC. So, if you're working with a bank, then their data and models are not leaving their network boundary, they're not coming to our cloud hosted solution, they are being installed within the premises of their own on-prem computation infrastructure, or in their own cloud VPC. So, from a governance standpoint, some customers find that to be very, very attractive, because they're very wary of their data and models leaving their boundary. We are also working now on a SaaS offering, where we will have a hosted solution, where customers will also be able to host their data and models on our platform.

[00:21:01] KP: So, when I'm in that situation, I've got my model ready. I've got your software deployed, and I want to evaluate my model I've built, do I have to do some configuration or is it out of box? What's the time to insight? I guess, is the question.

[00:21:15] AD: Yeah, the time to insight can be really fast. There are a few different form factors. In one setting, the first step is to ingest the models and the relevant train test data into our product that can be done straight out of Jupyter Notebooks. A lot of data scientists like to work out of notebooks. And in the notebook flow that they're usually working with, they need to add just a few lines of code, and then they're up and running with the product.

At that point, for individual predictions coming out of the model, they can get explanations for let's take a concrete example, let's say they're building a model for credit decisioning, or fraud, and they want to understand why did the model deny Jane credit, then from within the notebook, with a few lines of code, they have connected to our service through some simple API calls, and then they can within the notebook itself, see, for this particular individual, the model said, "We should deny Jane credit. And here are the most important reasons for that denial." So, they can get a notebook experience up and running very quickly, and get insights in the context of their notebook, supplemented with user interface, which is a web based React front end, and start getting insights. The onboarding process can be really fast, they can be up and running very quickly under an hour, and start getting new insights within a day.

[00:22:56] KP: So, we gave sort of a passing glance to the fact that there's a lot of research here during the early part of the interview. We could probably spend a whole episode exploring lime and SHAP values and all this kind of stuff. Maybe we'll get into it. But to start with, I'm

curious if I'm going to use your product, how much background knowledge do I need to know about model interpretability?

[00:23:16] AD: Yeah, that's a great question. So usually, to get up and running with the product, the product offers a very simple interface. So, you can start understanding the different kinds of analysis fairly quickly. For example, the abstraction of feature importance is something that most data scientists are very familiar with already. And so, as they're engaging with the product, I return to my running example, Jane was denied credit, and the feature important scores are going to say how much each feature is contributing to the model's assessment of risk. At that point, there are different ways of generating feature importance, that product offers some choices, and it offers also documentation on what to those different feature importances scores mean, and what the origin of those methods are. So, Shapley values as is one of the, as you mentioned, one of the standard ways of generating feature importances.

And some of the work that my research group did at Carnegie Mellon, back in 2016, was one of the early works that introduced the use of Shapley values in the context of explaining machine learning models. That's something that at a high level is the product documentation itself summarizes for the users. For folks who want to go deeper there are good resources that we pointed to we have also written some accessible articles on this topic as others. So, there's some amount – you're right, there's some amount of education training that's part of getting up to speed in this area.

In fact, one of the things that I have started doing myself is to have a more broader education effort. So I have been teaching a course also at Stanford, on trustworthy Machine Learning where we talk about their relevant background in this area on topics like Explainability is a Shapley values and lime and neural network explanations which make use of gradient based explanations, topics around fairness of models around drift and stability of models, and their robustness. And so, I think you are absolutely right that, with the use of products like this, there has to be almost a parallel track, where data scientists are going through an education process. And there is starting to emerge a good set of resources on this topic. What I'm also finding, and this will perhaps not surprise you, Kyle, or the audience is that data scientists are very passionate about going deep into these topics, as they're building up their expertise.

So, the engagement with educational topics is actually very high and there's an eagerness to go deeper, understand the foundations, understand when to use which method, where the limits are various methods are so that the matching, if you will, between the wide range of methods that are now available, and the problems to which they're applying those methods start getting clearer in their minds.

[00:26:45] KP: When I think about the broadest category of diagnostic tools, I see this gradient in my mind at one extreme, there are things like the smog test, I have to submit my car to here in California, where I show up, they measure it. Unless I'm a criminal who circumvented it, it's all on them to take the measurement and I pass or I fail. I don't worry about it after that. Maybe at another extreme, there's a home COVID test where if I don't administer it correctly, I'm going to get the wrong results. Where along that gradient is your product? Am I just going to kind of throw my model over the fence? Or do I have to really be an operator of the system?

[00:27:22] AD: That's a great question. I think this product is somewhere in between the two ends. And maybe as we talk about it, the other dimension which surfaces in my mind, because you mentioned the smoke test is that of a third-party independent audit model. So, in certain cases, over time, but I think we'll – let me tell you how various users are using our product right now, which is somewhere between the smog test and the home COVID test. So, they'll typically install the product and its data scientists at the enterprise that's using the product. We help them get started and running. But they are the users of the product, not us. Right?

So, imagine a data scientist at a major bank or insurance company that built models for credit, built models for fraud, and now they are using our product to assess the quality of their models. How it is doing in terms of performance, in terms of fairness, model drift and stability, robustness of models, and so forth. And through that process, they might identify weaknesses in the model, and then get guidance on how to address those weaknesses, and they had an improved model, which that moved into production. And once it's in production, the monitoring product would then help to alert them whenever the models are starting to behave out of whack and need to be adjusted.

I guess, if I think of that use on your spectrum between the small test and home COVID test, the reason I say it's somewhere in between is that like the home COVID test, the data science team

is doing the testing. But like the smog test, sometimes, what we see in big enterprises like in banks, they have separate teams, separate from the model development teams, there are sometimes called model risk or model risk validation teams or model risk management teams, who do an independent testing and validation of the models, separately from the folks that actually built the models.

So, in the US, in Manx, they have a separate team called the model validation team that does that for certain kinds of models like credit models and fraud models. And then a third team which does this kind of evaluation for fairness, which are fair lending teams. So, in that sense, you're starting to get closer to the smog test model because these are separate from the model development teams that are building the models.

Now, to complete the thought here, I think what might emerge over time, and we're starting to see some of that discussion now in the broader industry is that in certain areas, we might get a higher and stronger requirement for third party independent audits, much like the smog tests. So, if you think about financial accounting, the big audit firms do that for enterprises, and that trend is starting to happen a little bit in the context of machine learning in high stakes use cases where third party firms might come in to do these kinds of audits. And in that context, they might use platforms and products like the Truera platform.

[00:30:38] KP: When it comes to models in production, where your modeling solution is available to help people, a lot of companies and organizations have a concern about latency. They want a really fast decision on their model for whatever their use case is. How does your monitoring solution fit in for someone who's concerned about response time?

[00:30:56] AD: Yeah, that's a great question. So, one of the architectural choices that we made early was to initially focus on an offline part for monitoring. So, meaning that the monitoring product, model monitoring product that we have, does not hit the production inference engine at all. Let's imagine a model that is making decisions about credit, then the credit decisioning engine is on one part, and it's making decisions on an ongoing pace with some amount of latency. We don't impact that at all. The monitoring product sits on the side, it has a copy of that model, a shadow copy of that model, it's getting a feed in some kind of mode batches or micro

batches of the production data and it's doing its analysis in an offline manner, to identify whether the model is degrading or not.

So, in that context, the response time of the production part is not impacted at all. This is valuable, because if you think about one of the big problems that monitoring is trying to solve is trying to identify when the model performance is degrading, or when the model scores or data distributions are drifting in a way that it requires a careful look, that maybe the models are no longer fit for purpose, and need to be readjusted. That doesn't need to be done in real time. That needs to be done on some kind of cadence, it could be a daily update, or it could be weekly, or in some cases on a slower cadence than that.

[00:32:46] AD: And is there any alerting involved? Will I get, some kind of heads up that drift has occurred? Or is this more of a tool where I keep track of things and use my own judgment?

[00:32:56] KP: Yeah, there's absolutely alerting involved. So, when you configure the monitoring product, you'll set thresholds for alert on a wide range of metrics. For example, you could say that when model accuracy drops below a certain threshold, alerts can be configured to be sent to the appropriate teams, or when model drift, that drift, meaning the model score distribution now looks very different from the original model score distribution during training. When that happens, then alerts can be configured to be sent as well on pre-selected thresholds. That alerting and interrupting mechanism is extremely important in this case, because otherwise, there's just a lot of data to be tracking manually, a lot of information to be tracking manually. So, having configurable thresholds on model performance metrics, on model accuracy, on data quality related monitoring, on data drift or data distribution shifts, those are some of the things on which alerts can be sent. And then beyond the alerts is also guidance on – once someone has received the alerts as a data science team, they come to the dashboards, and then they have guidance through the diagnostics product on the root causes of those alerts, which in turn, inform how they might adjust their models.

[00:34:26] KP: I really like the idea that I could be alerted on a drop in accuracy, but I'm not sure every use case is going to have that. I'm thinking of maybe a loan application where I'm trying to predict will this applicant defaults in the next 12 months. That's a heck of a lag. I

probably need to look at something different. What are the options available for a situation where there's no ground truth like that?

[00:34:46] AD: There are two different ways that's a very big problem. You're absolutely right. This is sometimes called the delayed feedback problem, and it applies to credit models, to fraud models, a whole slew of models where there's often a delay between when the model makes a decision, and when you get to know whether it made the right one.

So, for that setting, we have at least two different kinds of approaches to address the problem. One is that we have created metrics for estimated accuracy, which is a way for the system to automatically estimate the accuracy of the model on the new data and the new production data, even when labels are not available. For your credit model example, the model has made, let's say 20,000 credit decisions in the last few months for which labels are not available, and the estimated accuracy metric would provide an early warning system indicating whether the model is in fact degrading and performance. The idea there is to, I won't get too much into the details, but at a high level, the idea is we have the new production data, we don't have labels for it, but we can find some segment of the older data for which we do have labels, which is the most similar to the production data, and then estimate the accuracy on that segment.

Now, this is a hard problem, because you have to first figure out what similar even means. And second, identify quickly what the relevant similar segment of the old data is. But that's one way, this early warning system and you can set thresholds and that to issue alerts and then have the data scientists dig in deeper into why the estimated accuracy is degrading and what they need to do to address the problem.

The second capability here is to track model drift, score drift, and feature drift. There, some of the things we are noticing is that when model score distributions shift a lot or data distributions shift a lot, having alerts on those are useful. And one of the unique things that we have created is not just to alert on when the model score distribution is shifting, but also to have automated support for root cause analysis that can help the data scientists understand the reasons for the drift.

So, just to give you one example, in the pandemic times, there were big shifts in models called distributions for a bunch of different reasons. And when there is a big shift in the model score distribution, and let's say we are talking about a credit risk model, and suddenly the risk scores are going up a lot, then the root cause analysis might surface that there are folks who have started applying for higher interest rate products. And if that's the reason why the risk is going up, then although the models for distributions are drifting on estimated accuracies, indicating that the model is still fit for purpose.

In other cases, you might find that the risk scores are going up a lot, and the reason for it is because of some factor, that may not be a strong indicator of race. For example, if travel spending has gone down, and that's the reason why risk scores are going up. So historically, it could be that travel spending was a good indicator of risk. But post pandemic, at least right after the pandemic started, a lot of people stopped traveling because of safety concerns, not because they had lost their jobs or things like that.

So, this also really helps in understanding whether the model has degraded in its performance, once you get into understanding the reasons for the drift. This kind of root cause analysis is really powerful and one of the things that has been missing when coupled with monitoring for machine learning models.

[00:38:59] KP: Well, the pandemic certainly is an immediate change point in a lot of different datasets. Lots of things changed overnight, relatively speaking. I guess, you were maybe firsthand to how some companies had to react to that. Are there any stories from the front lines you can share about the overnight changes people may or may not have experienced?

[00:39:18] AD: Yeah, so I can't speak to specific customer stories. But at a macro level, what has happened quite a bit are the kinds of things that I mentioned where model score distribution shifted, read scores may have gone up for various segments because of things related to the pandemic in marketing models. There's actually a very good article in the MIT Technology Review that talks about several interesting use cases in the E commerce site where invent machine learning models that are being used in supply chain or logistics, they went really out of whack because certain kinds of products the demand shot through the roof, and for other things, the demand went down a lot. So, think about hand sanitizers, for example. Post pandemic

suddenly, the forecasting and inventory supply chain logistics models were completely out of whack on those kinds of products because of the unexpected and unusual surge in demand.

[00:40:22] KP: Well, earlier, you'd alluded to the changes in sophistication in software engineering in general, the fact that we didn't have CI and CD pipelines in the ‘80s. And in many ways, it's like we had leaky rowboats and now we have, you know, massive Titanic, more ships available. There's been quite a lot of advancements in how we develop, deploy and monitor software, and maybe it's time for machine learning to have its day. What are some of the lessons we can learn from software that will apply to machine learning?

[00:40:50] AD: I think the two big things are having the analog of systematic testing, debugging, CI/CD style pipelines, and that key role of monitoring. Those are the two big areas where if you look at the software industry, we have now a market that has over 100 billion in market cap from various companies, the data dogs of the world and new relic and on the monitoring side. And also, if you think about GitLab, and things like that, which are supporting the development process in CI/CD pipelines, Jenkins, and you have some of the modern cloud native ones like Harness. So, the analog of that, I think, that's where the big opportunity and whitespace and machine learning is to arise playing in that space now. But overall, I think that segment of the industry is a very strong analogy with software engineering and how the market has evolved there. In the next five years or so, I would expect this to be a huge market that will make sure and meet the strong needs that we are starting to see.

[00:42:05] KP: Are there things that are distinctively different, maybe unique lessons that machine learning is going to have to figure out that we can't get inspiration from software engineering techniques on?

[00:42:14] AD: Yeah, absolutely. The data dependence of machine learning is very, very different from traditional software engineering, which has focused quite a bit on code level analysis. So, that has implications across the board in these pipelines and tools for testing, debugging, and monitoring. What a bug looks like in machine learning is quite a bit different from a software bug. Software bugs get triggered, and can be discovered through static and dynamic analysis, which look at art through programs. If you go to a certain part, then you

trigger that bug, and to trigger that bug you need to go through that path. And so, you might be doing fuzzing and other kinds of static analysis, software analysis techniques.

In machine learning, a bug is more of a statistical trend. So, for example, a fairness bug because there is a significant gap and their rate of approval of women relative to men, or model drift related bug. It has to do with distributional differences in how the model score distributions look like, at one point in time post endemic relative to what it looked like pre pandemic. So, these more aggregates statistical properties of machine learning models and data distributions. That's very unique and very different from traditional software engineering.

[00:43:50] KP: Are you seeing any patterns in either use cases or industries that are early adopters who's really on the forefront of leveraging tools like yours? I

[00:43:59] AD: I think the demand is really, really starting to come together in some of the higher stakes use cases and industries. So, financial services, banking, insurance, fintechs, that's one area. We are seeing that emerge more in healthcare and companies that are using machine learning for employment. And then that's where some of the regulatory pressure is also driving the urgency of the problem. But what we're also starting to see and what initial evidence off, and that I firmly believe will emerge is that much like in software engineering and software equality, it is a very universal need. Every team that's building and deploying machine learning models will need this kind of product. So, we are starting to see wider adoption outside of the high stakes use cases as well with companies in the retail and manufacturing sector going in that way, tech first firms starting to get more into adopting, monitoring, and debugging solutions as well. So, those are the early trends over time. I really envisioned that this will be an integral part of every machine learning pipeline.

[00:45:23] KP: So, it sounds like there's a really strong case for enterprise adoption. Do you think it's appropriate for a scrappy startup to be considering solutions like yours?

[00:45:32] AD: Absolutely, and they already are. I think it's a very interesting dynamic where for a scrappy startup, it's a small team that's working on a focused problem and that's where they want to be usually deploying their resources. But if they're building machine learning models and fielding them, then having automated support for testing, debugging, and monitoring those

models is almost a no brainer. So, for example, a scrappy startup will typically have CI/CD pipelines. They don't want to build those things themselves. And if they don't use existing products, we are a scrappy startup, and we have CI/CD pipelines from a pretty early stage within the first few months, right? So, this is very much the analog of that. For scrappy startups working in the machine learning AI space, this is absolutely something for them to use,

[00:46:33] KP: To wind up, what's the onboarding story for anyone who wants to test out your solution? How do they get going?

[00:46:40] AD: To get going, we have set up a very easy to use Cloud sandbox. And so, they can pretty much get credentials and get started very quickly to test out the product. When they've done their initial testing, and examine the features, if they want to start using the product, we have a fairly streamlined process to help with installing the product on their behalf and then getting them up and running.

[00:47:07] KP: And where's the best URL they can go to to learn more?

[00:47:10] AD: That’s truera.com.

[00:47:13] KP: Well, Anupam, thank you so much for coming on Software Engineering Daily.

[00:47:16] AD: Thank you, Kyle. It's been a pleasure. Thank you for the probing questions. I really enjoyed the conversation and I look forward to engaging more with you and your audience on these topics.

[END]