

Please note the following timestamps are approximate.

Elena Samuylova (Guest) (00:00):

We do believe in open source. You can eventually monetize it. You just monetize not over everything, so you just create more value, then you monetize. And it also seemed great in terms of community go-to-market approach, which I personally enjoy a lot. And that is something that we believe we can execute.

Peter Wang (Host) (00:19):

You're listening to *Numerically Speaking: The Anaconda Podcast*. On this podcast, we'll dive into a variety of topics around data, quantitative computing, and business and entrepreneurship. We'll speak to creators of cutting-edge open-source tools and look at their impact on research in every domain. We're excited to bring you insights about data, science, and the people that make it all happen.

(00:39):

Whether you want to learn about AI, or grow your data science career, or just better understand the numbers and the computers that shape our world, *Numerically Speaking* is the podcast for you. Make sure to subscribe. For more resources, please visit anaconda.com. I'm your host, Peter Wang.

Speaker 3 (00:55):

This episode is brought to you by Anaconda Notebooks. With nothing to install and nothing to configure, Anaconda Notebooks is a lightweight, ready-to-code, and fully-loaded data science environment entirely in your browser. Spin up new projects with the click of a button, with all the packages and files you need in one place—with fast and persistent cloud storage, no matter what, wherever you go, your code goes. And students? Listen up. You also get on-demand access to Anaconda's data science experts. No matter your experience level, learn through hands-on experimentation and you'll be predicting the future with machine learning models in no time! So what are you waiting for? Start coding with Anaconda entirely in the cloud on anaconda.cloud.

(01:15):

I'm so glad to welcome Elena Samuylova to join us today on our podcast. Elena is the CEO and co-founder of Evidently AI, and they do have an open-source product around machine learning monitoring, and we're going to have a great conversation around what correctness means in machine learning and how production deployments actually work today in the enterprise.

Elena Samuylova (Guest) (01:39):

Thank you very much for having me. I'm thrilled to be here.

Peter Wang (Host) (01:42):

Welcome. Thank you so much for joining us. Can you tell us a little bit about yourself and about your business?

Elena Samuylova (Guest) (01:47):

So as mentioned, I'm CEO and co-founder of a startup where we develop an open-source tool that helps you monitor machine learning models in production. So we basically focus on this stage in the model life

Please note the following timestamps are approximate.

cycle when you're really creating the model, you deployed it, and it's being put to good use and you actually need to take care of it and make sure that it delivers on the promise.

(02:06):

In the past, I have worked with different machine learning solutions with different industries: retail, manufacturing, you name it, finance, a variety of industries, for more than seven years. So I've seen how this whole ecosystem developed, how enterprises adopted machine learning technologies and this gave a lot of inspiration to the tool that we create because we saw that a lot of things are still missing. And we hope to contribute to it, particularly with the open-source approach.

Peter Wang (Host) (02:31):

Excellent. Yeah, that's great. And I definitely look forward to chatting a little bit about learnings from deploying these things in manufacturing, in environments that are not what we traditionally hear a lot of the conversation around ML being about, because a lot of it's focused on consumer retail analytics or online behavioral analytics, things like that. So that's great. That's great.

(02:50):

One question I have is, what motivated you to do this in the open source? Because there's many startups I know of who do all sorts of AI and ML products, but they use open source, but they don't make their underlying technology open source. So for you and the founding team, what was the decision-making around that?

Elena Samuylova (Guest) (03:07):

It was actually a pretty rational decision because when we went out to the market, we looked at what's there, and we were talking to a lot of potential users like business leaders, data scientists, about how they monitor their models. And first of all, we learned that many of them don't monitor their models; there are not yet proper tooling for that.

(03:26):

And when we asked how they adopt tools, how they choose what they look for, open source came up. It's so straightforward, so easy. If you're a data scientist or machine learning engineer, that's how you adopt the tools. You're used to working with open-source algorithm, tooling, and whatever you look up, you want to see an open-source tool out there and there was none that would take this leading position.

(03:45):

So we thought, okay, we're going to create this one because that's great distribution. We do believe in open source. You can eventually monetize it; you just monetize not over everything, so you just create more value, then you monetize. And it also seemed great in terms of community go-to-market approach, which I personally enjoy a lot, and that is something that we believe we can execute well.

Peter Wang (Host) (04:06):

Yeah, I think in a sense what you're describing there is that the practitioners, there's like multiple pieces of this, right? Number one, the practitioners, they wanted to be able to pick up the thing, use it for themselves to validate that it works.

(04:19):

Please note the following timestamps are approximate.

So in a sense, it's a kind of transparency. You're saying, our technology is not locked down behind some closed thing you can't look at. You can look at it and see how it works. But we of course then, as a business provider, we have to add more value above and beyond just the tech.

(04:34):

And that's then where a lot of startups will now, will move to a hosted model, like a software as a service, something like that. I imagine in your case, the kind of deployments you're doing, that doesn't really work in all the cases. People are deploying on premises or in environments that are not amenable to a cloud-hosted SaaS solution like that.

(04:52):

Can you tell me a little bit about that? In terms of, from a business model perspective, from monetization, what percentage are you seeing people doing on-prem things? What percentage are you seeing people doing cloud things?

Elena Samuylova (Guest) (05:01):

Right now, we're still in the open-source phase, so most of the people we talk to, they eventually try to deploy it on their own. And we still see a lot of people that are working, a lot of companies that are deploying on premises.

(05:11):

I mean, cloud is great, but there are a lot of environments just like this manufacturing that you already mentioned, right, there are financial services, there are different limitations to what exactly you want to share, especially when it comes to data, data transfer because you are logging your model predictions. This might contain some personal data and whatnot.

(05:28):

So there are multiple considerations to why you want to host it on your own. So I cannot give you an exact estimate. It definitely varies on the industry, but we see still a lot of on-prem deployments, and I think it will continue like this for the foreseeable future.

Peter Wang (Host) (05:42):

Well, and the reason I ask is because 10 or 11 years ago, I feel like there was an inflection point that was passed and people felt safe moving to the cloud, and it was a lot of enthusiasm about doing this.

(05:54):

And then, three or four years after that, people started asking questions around, what is the actual total cost of doing base loads in the cloud? How much complexity does it really save us? Once we take the governance questions that we had on prem and we implement them in the cloud, it's kind of just as complicated, or there's other flavors of complications.

(06:12):

And then, once you go through this whole thing, I feel like now I'm starting to see a lot of trickles of conversations where people are over that now, and they're like, look, here's what the cloud is good for. Here's the things that we do, of course, in the cloud, for these things, but these other things, these workloads, we have to govern them in this particular way. And we don't feel bad that we're not moving them to the cloud.

(06:32):

Please note the following timestamps are approximate.

I guess that's the thing I'm trying to say. And that's the kind of sentiment I get. Do you see the same, when you talk to people?

Elena Samuylova (Guest) (06:37):

I think so. I think now it's more of a rational decision. So you are really choosing between alternatives, and one alternative is to deploy it on prem. And of course, in some cases, there is basically regulations or physical world limitations that guide you. If you have an [inaudible] device, you're not going to deploy stuff in the cloud.

Peter Wang (Host) (06:54):

There's something you said just a little bit earlier, where you said many of the people you talk to, they're not even doing monitoring.

Elena Samuylova (Guest) (07:04):

That is still fascinating to me, but I think that basically highlights where we are in the adoption curve. So people were creating machine learning models and not deploying them for a while. This was basically the first thing that happened. There was a lot of proof of concepts, pilots, testing it out, you never put this thing to work.

(07:20):

Then somehow we passed over this and finally started deploying the models. And the next barrier is maintenance, retraining, and monitoring. And we are right now facing and shaping best practices around it. But many people I talk to, they just say, "Hey, the first time the model broke, we didn't even notice. Then someone told us, the customer, the end user. Somehow it became obvious and we started to think about monitoring."

(07:41):

And I'm like, "Seriously? Okay." But this is something that you hear a lot, so I must say that this happens more often than you'd expect.

Peter Wang (Host) (07:49):

That kind of blows my mind a little bit, I guess. I could see how that happens. But were they using it for very serious things, then?

Elena Samuylova (Guest) (07:58):

Of course not. You may be deploying a model that does some term prediction for your marketing folks to send out a newsletter. And there are different scenarios when this can still harm your business. You might send something to the wrong person, you might churn a customer instead of retaining them, and so on. But this is not a life-and-death decision or some mission-critical scenario.

(08:17):

When you talk about manufacturing and use cases, which has different cost of error, of course, people would think about this from the very beginning. But still, it takes them longer to actually deploy it because they're starting to think about all this stuff.

Peter Wang (Host) (08:30):

Please note the following timestamps are approximate.

Well, there's no free lunch, if you want to use this stuff for serious things. When you're talking to folks about monitoring, and I like this term you used, the cost of error, and as they're thinking about this for the first time, tell me what that's like.

(08:42):

Because I have this theory that one of the big things we're trying to figure out with doing machine learning and production and employing it more successfully within businesses is that people in general have to think about correctness differently than we used to think about it for just civil transactional software systems or data integrity for just managing data systems.

(09:03):

But there's this combination, this fusion of concerns about value-sensitive software and what correctness looks like for that. And building on that, I think that it forces businesses to have a hard conversation at a business level about the business objectives. Do you feel like there's some of that when you're having that conversation with people? What is that conversation like? How does it generally tend to go?

Elena Samuylova (Guest) (09:24):

Well, absolutely. You touch so many important points. One of them is connection to these business KPIs. And this is the hard discussion that you should have before you create a model, to be honest, because sometimes the expectations, they're not matching the reality at all. So you just want some magical solution that will solve all your business problems.

(09:41):

Whereas with machine learning, you need to have really well-identified KPI, what you want to optimize, and choose it from the very beginning because that's the objective goal for your model. Are you increasing revenue? Are you optimizing cost? Are you increasing conversions? Are you saving time here and there? So there should be some real business KPI.

(09:58):

And some companies, they sort of figure it out, like if they're doing deployments for the first time, after they deploy. So they learn that, actually, the intended purpose of the model was different from what they initially envisioned. And that is a hard conversation that builds them to the question of how we trade on the model faster, how we build and put them to deployment faster.

(10:17):

This is something that I've seen a lot, that is probably not what we discuss now when we discuss monitoring because if you start monitoring the model, it is probably doing something right already. And what you need is you need to ensure that it continues to do so. And here we touch on this question of correctness, or maybe model relevance, that you just produced, because like you said, these systems, they're not deterministic.

(10:39):

There is not a set of right answers that they might necessarily supply. It's very hard to test them because in reality, you might get completely new data inputs. The world might change completely. I mean, just imagine all the models in the pandemic. So the behaviors changed, all the patterns changed. In machine learning models, they will still give you the response. So they don't know what they don't know.

(11:00):

Please note the following timestamps are approximate.

They were just like, "Hey, that's my prediction. You asked; I answered." And the question is, how do you monitor this? So the model relevance, the business KPI, the data that is incoming, and still the software system, because machine learning system, that is still software. It can be real-time API that you need to monitor precisely. It can be a batch job, which is somewhat simpler, but still.

(11:21):

And there are these four layers. I see it as a pyramid, like a software system, data, model and business KPI. And you need to monitor it all, and this is a hard task.

Peter Wang (Host) (11:31):

Yeah, there's a hierarchy of needs. If the software isn't reproducible, if the data is not well-conditioned, then you have no hope of ensuring correctness at the higher level. But even if all those kinds of things are met and the hardware's the right kind of hardware, and it's giving you the right kind of responses within certain constraints, SLA, route constraints, all that is set and then you get to actually monitor the model itself.

(11:52):

And so I'm curious, in your open-source tool, how does a user express the concept of the guards or maybe the equivalent of, in software we have the concept of doing an assert, an assertion about what you expect before going into a particular phase of computation.

(12:07):

So do you have something similar like that? Or what is the analog of that in machine learning models where, as you say, things can go wrong quickly, they can go wrong slowly. You don't quite know what to expect in all cases.

Elena Samuylova (Guest) (12:19):

We actually try to learn a little bit from software and from data monitoring. I mean, it's obvious, right, you want to inherit the best practices. So one thing we introduced to how you can approach it is through testing. So you apply some, like you said, assertions. So you would validate maybe your inputs, maybe your outputs, maybe if you already got the labels, you can evaluate the model performance as in accuracy and error and so on.

(12:41):

And you do it at the right points of time by introducing certain constraints that come from your reference dataset. So you get a new batch of data and you say, "Hey, is it approximately similar to what I've seen before? Or, there are some red flags that just tell me that I should not use this data at all?" Maybe you generated the predictions and then you can evaluate, are your model predictions looking weird?

(13:01):

And maybe you should not act on them, but instead use some fallback or some alternative way of taking a decision. And this is very interesting because it's not just about monitoring but also about the system design. That's probably not the part that we cover, but still very important in how you introduce the certain guardrails, alternative fallback systems...

(13:19):

Please note the following timestamps are approximate.

A lot to learn from manufacturing, probably, and from other mission-critical systems that you deploy in the real world of how you make sure that when the model fails—and it will fail, you know this a hundred percent, you just don't know when—how you react and how the system reacts.

Peter Wang (Host) (13:34):

You know, I'm a huge fan of airplanes and I've always loved airplanes, even as a kid.

(13:39):

And when you start looking at airplane design, especially commercial-facing stuff, not like fighter jets and things like that, but for the stuff where you really expect to have thousands of people, or hundreds of thousands of people, using this particular device, or flying through this vehicle and having an expectation of a safe landing. So much of what the engineering is about is all about redundancy and safety systems.

(14:01):

The actual thing of bolting an engine onto an airframe with an airfoil that basically works, that's actually pretty straightforward. It's all of the stuff, what happens if this fuel pump dies? What happens if that pneumatic or hydraulic pump dies? What happens if there's low tire pressure in one of these tires? What happens if a bird hits an engine?

(14:21):

So much of the engineering is actually around the safety systems and the redundancy systems. Not that building an engine or building an airplane is in itself easy, but that's still pretty straightforward. And what really takes you from five nines of uptime to six nines or seven nines of reliability in uptime, that's a whole different kind of engineering.

(14:40):

In the software world, as software developers, we know this. We don't engineer for that level of quality at all. We talk about space shuttle flight software and avionics, things like that is an entirely different class of software than what most people are getting paid to write in the world today.

(14:57):

So I think when you talk about the machine learning models and what goes into it, you're testing even the training data coming in to say, "Does this conform to the assumptions we made about the training data for the previous cohort or for the previous version of the model?" And then once it's in production, we're going to constantly be injecting, hitting it with test things, both fuzz testing it with things that we know should give us bad results, and then also good things.

(15:22):

Just that whole mentality of engineering this online information system, it's very different than the fast and loose, move fast and break things kind of mentality of a lot of web development, I feel like, or development in the last generation of web applications, where you expect things to break, you just reboot it. You expect the box to go down, well, you build a cluster of 20 of them so it's resilient to the failure.

(15:42):

And that's a very different kind of mentality when you're trying to build for resilience. In the manufacturing scenarios, can you think of real-world examples where this worked really well or this worked really poorly or something that was deeply counterintuitive to you as you were trying to deploy a model?

Please note the following timestamps are approximate.

Elena Samuylova (Guest) (15:57):

Yeah. Manufacturing was really interesting because there are a lot of automated systems already and machine learning is no different from them. You might have some process control systems that use physical statistical methods already and you just upgrade them and add one more way of making decisions.

(16:13):

So this world is actually pretty well prepared for this sort of automation, and they already have some existing practices that I would actually be very interested to see some exploration, what we can learn from them and just transfer it to the machine learning world. But that said, one thing that I noticed, and I think it's applicable to many automated decisions, is that you still don't want to really automate everything.

(16:34):

And you introduce some sort of human in the loop who would approve these recommendations or whatever the model is giving you as an output before you act on them. And this is the first obvious failsafe that you might add. The model is still helpful. The person might be approving most of the decisions, but they are the ones who help you catch the one wrong decision that you did not foresee.

Peter Wang (Host) (16:55):

Yeah. Well, and deploying these things in manufacturing or in physical systems in general, there is a nice back pressure coming from the physical world.

(17:03):

At some point, there is an information system virtual model. The map hits the territory in a very direct way at some point. You control certain machine things, but the machine can't go faster than this, or it can't move to this boundary or to this position. And so you can reflect those kinds of constraints back into the computational system.

(17:25):

And in online things, online banking, automated feeds, recommendation systems, a lot of these kinds of things, there's not a physical limit. It's all...you're dealing with a virtual system in any case, that you're actually going and manipulating. And so I wonder if maybe we'll see more rigorous engineering and thought around productionizing ML coming from when it intersects with real-world engineering, like proper engineering, not software engineering.

(17:55):

And I wonder if there's things you've seen there or anything like that, that might be relevant from the manufacturing world, where you had a hard physical constraint that changed the way you thought about what you could do with the ML models.

Elena Samuylova (Guest) (18:08):

So it is of course a spectrum. So there are these more mission-critical systems when we have to care about every single failure a lot. And then, even in manufacturing, there may be very simple use cases when you don't really put so much on the line, so the cost of error is moderate. Maybe you're just suggesting in which order you should sort your spare parts, so some small optimizations, and of course, [inaudible] for your machine learning systems.

(18:32):

Please note the following timestamps are approximate.

Still, there are a lot to learn. The monitoring and this level of rigorousness should match the use case complexity. So you don't always need to have all these airplane-level [inaudible] from manufacturing in terms of how they introduce these guardrails.

(18:45):

So one of the things that we've seen in practice is actually how you can mix traditional physical models and machine learning systems. Because you might [inaudible] system, but the physical models use, for example, a machine learning model to basically add one more layer of being more precise on top of your physical model; learning from the real physical model, the machine learning model gives the real-world constraints.

(19:10):

So your model will not predict something entirely unreasonable. It will follow, not just the model, your system, because the system will be comprised of the business logic and guardrails that you put on top. But this is a real product. So this is a real solution.

(19:22):

Manufacturing is this rigorous approach, this combination with physical-world models and systems and being very pragmatic about where you want to deploy automation. [Inaudible] you should develop having in mind the goal that it should be justified in terms of complexity and cost. So it's not just sprinkle machine learning and suddenly things become better because you've started to collect the data.

(19:43):

So I think what you could learn from manufacturing is this rigorous approach, this combination with physical-world models and systems and being very pragmatic about where you want to deploy automation. Maybe you don't need machine learning in every single use case that you can come up with.

Peter Wang (Host) (19:59):

Yeah, I guess if we're coming from the perspective of data science, we're looking for...we [inaudible] walk around with a hammer looking for nails to hit. But what you're describing is in manufacturing, but in engineering in general, they use computers all the time to do engineering simulation. We have closed-form solutions for many kinds of engineering simulation and that's used every day to build all sorts of things.

(20:21):

And so machine learning is then just an additional fancy kind of thing that you can bring in to help improve some of the simulation or make certain kinds of predictions of things that you might not have thought about when doing certain other kinds of discrete numerical modeling. But it's then, sort of, some icing and sprinkles on top, versus it's like the main thing.

(20:38):

So that's really interesting about that. There's a lot of startups that famously pivoted to an unlikely use of their thing. Or companies or whatever, they found, on their path of doing A, they got to B.

(20:50):

SimCity, one of the most popular games in history, started because a guy was trying to build a helicopter flight combat game and they had to design target cities to go and attack. And then he found that designing cities was way more fun than blowing them up from an airplane or helicopter and then SimCity was born.

Please note the following timestamps are approximate.

(21:06):

And I think Slack was born the same way, they had an internal chat tool, as they were trying to build something or the other and it turned out that that internal chat tool was really popular and really good.

(21:15):

And with Anaconda, same thing. We were trying to do all this other stuff around scaling Python computation and PyData and all this other stuff. And we were like, shoot, we just got to get people packages so they can install them and use all this open-source software. And Anaconda and conda were born from that and that became the thing that we became known for. So maybe you pivot from monitoring to model visualization and who knows? There's all sorts of things that happen.

(21:38):

That's the beauty, I think, of open source, is that when you start talking to those users, and if you have an open mind and an open heart to what it is they are trying to do, you can find yourself moving to adjacent spaces very easily. Versus if you go out there with a kind of already shrink-wrapped, closed proprietary tool, it is what it is. And there's a much harder boundary or barrier to getting that feedback from users.

(22:01):

So what else in terms of contributions from the community, have you seen some good stuff coming in? What does some of that look like?

Elena Samuylova (Guest) (22:06):

I'm always fascinated when I find content about us. So people are just taking the tool and they're writing a review in Japanese, or they have a meetup in Singapore or in Mexico, and I'm like, "Wow, I didn't expect that, but that's really fascinating to see."

(22:21):

And of course, yes, contributions. So people come up and they just suggest their own way of how you want to monitor the model, their own metrics, their own integrations. And that is really the best feedback loop that you can get, just like you said.

(22:34):

In a way that with enterprise software...and still open source, we are dealing with enterprise...your feedback cycle can be like a year until you deploy the system and understand no one needs it. So this is the joy of it and really great for product development as well.

Peter Wang (Host) (22:49):

And it requires the whole organization, I think, to be aligned. I mean, your business, right? Everyone in the business up and down has to understand that that's one of the reasons you're doing the open source is to get that kind of product feedback very quickly.

(23:02):

As you were talking, I was thinking about the fact that...you mentioned integrations. That's the other thing that we always see with open-source tools and frameworks. People can take them and they will...I don't know quite how to phrase this, but it seems to me like what we've seen is that situations, or use cases, where there's a lot of things to be integrated together are places where open source does well. And a well-designed open-source framework lets the user build their own integration.

Please note the following timestamps are approximate.

(23:30):

And if you provide a way for them, like a community hub or something for them to contribute that back, then you build this user-to-user innovation loop that is very powerful. Now, do you guys have something like that or are you thinking of doing something like that?

Elena Samuylova (Guest) (23:43):

MLOps is a fascinating field. So we belong to these machine learning operations, which is, it's like a Wild West of a lot of tools they don't understand really how to stitch together. So there are hundreds of them. And maybe people are trying to solve this by building internal machine learning platform, and they would take Evidently, or any other tool, and try to combine them all together to solve this problem end to end.

(24:05):

So yes, we see that a lot. Probably the challenge here, is a little bit that even the boundaries are not yet defined. So I have my vision of how many tools you need in the workflow and how they should connect, but there are other ways of approaching it. And we are still in a world where this is not yet solved. You don't understand where deployment stops and monitoring starts, experiment managements, data prep, how they all interface, so I think we are collaboratively figuring this out.

(24:30):

And this is where open source is great, because you see what people are doing with it. Where they expect these boundaries to exist. What should integrate with what, on the database level, on the deployment level and so on. We are just now basically in the observation phase, in a way. So we do propose some of the patterns ourselves. Right here, this is the integration idea that we have. We suggest you to use workflow manager, for example, with Evidently and integrate it as a step in your pipeline.

(24:56):

But there are other ways, and I'm looking to learn how people will start using our tool to update this vision.

Peter Wang (Host) (25:02):

Let's get a little geeky. So you're saying you have your own point of view on how this stuff should come together. What would you say are maybe the most unconventional or differentiated aspects of your perspective? Like things that other people in the MLOps space might have the most disagreement with?

(25:17):

And what are the things that you think are essentially settled arguments, things that pretty much everyone agrees is how you should do it? So I'd love to hear that. Again, bias and variance, right? What is the highest baseline? And then what is your highest-variance set of opinions about the architecture?

Elena Samuylova (Guest) (25:32):

One thing that many people imagine when they think about monitoring is literally this dashboard that sends you alerts when something goes wrong and you have a lot of different visualizations. And our sort of contrarian opinion is that you often don't need it. I mean, it's a kind of nice-to-have thing, you want to know that it exists somewhere. But to really integrate it in your workflow, it's sometimes better to do pipeline tests.

(25:53):

Please note the following timestamps are approximate.

Because usually you still need to do batch checks. You don't really need to [inaudible] the change in data input statistically every 10 objects. You would rather do it after every new batch of data arrives, maybe every hour, maybe every day. You don't really monitor it on this level because it's not software monitoring, per say. And having this as a pipeline test introduced as part of your machine learning workflow also gives you an opportunity to formulate these assertions.

(26:19):

It is hard, but it is important because otherwise, what are you going to do with these metrics? You can come up with some statistical test, but then you will not be even able to interpret its results. So we are really proponents of this software-informed testing approach that you can move to the machine learning world. That said, it's not the only one, and we actually have different integration patterns, but this is something that we are pretty bullish on.

Peter Wang (Host) (26:42):

I agree with you. I mean, going into the thing like this, not to get too meta, but it's like human learning about machine learning. You should form an a priori prediction as to how the pipeline behaves, or what the condition, the data quality looks like. If you are doing your sensemaking post hoc, after it all comes down, and you're like, "Well, that doesn't quite look right," or, "Well, maybe that's okay,"

(27:06):

You'll be doing a lot of retrospective narrative formation, as opposed to, from the outset, saying, "Okay, we expect we put this into the system, these things happen, we expect this amount of variance and it shouldn't wiggle more than this," or, "We should almost never see this at a frequency of more than whatever."

(27:23):

And if you don't do that intellectual exercise, you're always going to be playing defense. And once you're on defense, that means an anomaly has happened. You're already in a bad state and you're trying to figure out what happened. And now you're going to go do this trying to drive with the rear-view mirror, as opposed to having a picture in your head as to what the road should look like ahead.

(27:41):

So I completely agree with you about that. It forces, however, people to have a certain level of intellectual rigor about business processes, which may not exist in the organization. And data scientists may not have the clout to force those conversations, either.

Elena Samuylova (Guest) (27:54):

Maybe that's a good moment to introduce it. So yes, you need to have this thinking before you deploy a model or very early in the life cycle, but maybe it's in the shadow mode. You're just testing it out alongside the model or existing decision-making approach. You should already think about what can go wrong, and how will I catch what is wrong? What is the acceptable boundaries and expectations and what is definitely a thing that is a red flag and I should be alerted immediately?

Peter Wang (Host) (28:21):

Right. And the thing that I've observed is that a lot of technical...this kind of gets to this thing about a lot of technology, big technology transformations in particular, within businesses, requires more of an organizational shift. It's not the technology that's the gating or the bottleneck aspect of the

Please note the following timestamps are approximate.

transformation; it's changing people's mentalities or changing the way that groups interact with other groups.

(28:43):

So have you observed that to be true? Do you find anything surprising about this in the course of building your business?

Elena Samuylova (Guest) (28:51):

Yeah, it is spot on. I think all problems in the end are people problems, organizational problems, communication problems, not about technology at all. And specifically with machine learning, I like to just think that you're deploying, I don't know, linear regression. You don't care about algorithms. Really, this part is irrelevant. You're just putting some sort of...it can be even a rule-based system, so just like a decision tree.

(29:13):

And then just forget about all this technological complexity. Imagine how you're going to deploy a decision tree in your business process and all the things that are coming to organizational change management, these guardrails, this communication, how you're going to update about changing priorities that might affect the way your model is used, how you're going to communicate that there will be an actually new data source appearing that might improve your model quality.

(29:36):

So this has nothing to do with the technology or defenses, deep learning algorithms out there that will be state of the art.

Peter Wang (Host) (29:43):

Yeah. Well, it's entirely about bringing a certain level of rationalism and empiricism into business, is the way I like to think about it. But speaking of business, it's very interesting, you're obviously very technical. So in business programs in university, are they talking about this at all? Or what is the level of education and pedagogy in schools relative to what the reality of ML-infused business looks like in the real world?

Elena Samuylova (Guest) (30:09):

Well, probably when I started, no one spoke about machine learning, but they did build some regression models in the past. So if you study economics, so you understand this basic principle. So that gives you a good enough understanding of some stuff.

(30:20):

But what I see now, is that there are quite a few programs that actually try to bridge this tool, so the technological aspect of it and the business or pragmatic, managerial aspect of it. There is even this role that people try to introduce called analytics translator, which I really like the whole idea. It's not like a job role; it's more of a hat to wear. It might be a data scientist, it might be a product manager, project manager who is wearing this hat.

(30:45):

But basically, it's trying to translate the business needs and business concepts into technological solutions, particularly when it comes to data and then feed back the insights and understanding from the data to the business on what can be done. So bridging these two is incredibly, immensely hard and I think this is something that this should be invested more, even on the organizational level. So you don't

Please note the following timestamps are approximate.

just hire data scientists; you also build the culture of making this decision of automating these processes and of learning from it.

Peter Wang (Host) (31:14):

I agree with you. It's the sort of thing where, I think for too long, computers were seen as geeky stuff that computer science majors deal with. And at the end of the day, the business will kick over some specs for them to build some business application and it does something...

(31:27):

And it's really amazing, for me, actually quite encouraging to see how, in just a few short years, we've gone from, is data science a thing? To, can we find data scientists? To business managers and product managers and everyone being like, "Okay, we need to understand data. We need to understand how to think computationally and algorithmically in a world where machine learning infuses every aspect of our products and of our businesses."

(31:54):

I'm personally quite excited to see that happening because I think that's the right thing to have happen. And I certainly think that's a lucrative development for anyone who's got a career in any of these things right now. If you can find a way to get yourself into that translator position, you're actually in the middle of a lot of really important conversations. And it's an incredibly, I think, incredibly good career move, if people can swing it.

(32:18):

Now, looking back, with all the experience you have as CEO and as founder and everything, what are some tips or some advice you would give to a younger version of yourself or to others who are listening who are maybe just getting started in their careers? Whether it's making the career change from being a business major to going and getting deeply involved in this technology stuff, or really any of it. What are some takeaways, top three lessons you would give yourself?

Elena Samuylova (Guest) (32:43):

My top one is to follow the curiosity. I once took a very boring job, and understood that I was so, so bored and this was not the adult life I envisioned. So I was always later guided, hey, I don't understand how it works, but it sounds exciting. So between the sure path and the "I have no clue how it works, but I'm going to go there," I always chose the second one. But it's a personal choice, still. So I like building things from scratch.

(33:08):

There is a lot of uncertainty, and of course, probably not everyone would choose this and that's fine. So it's all sorts of things to be done. But, yeah, I think definitely that was a defining understanding I got about myself and I later pursued. So if you're like me, probably follow what you find curious and you'll figure it out.

(33:25):

So new jobs, completely new technologies. I came into AI and machine learning before it was used, I mean, even the term, before it was used. There's probably some new technologies that are appearing right now that I have no clue about, but maybe someone else can pick them up.

Peter Wang (Host) (33:39):

Please note the following timestamps are approximate.

Yeah, I love that. Curiosity and courage are two fundamental character traits that everyone has, everyone's born with them, but unfortunately, as we get older and older, there's more and more things in the world that wear those down, that make us numb to those things. But in all cases, one should be courageously pursuing curiosity to the maximum extent possible.

(34:02):

Great. Well, Elena, we're at the end of our time, but I really appreciated you coming on the podcast and having this conversation. I encourage any of our listeners who are interested in doing machine learning monitoring or just getting nice visuals and nice dashboards for some of their models to check out Evidently AI.

(34:19):

Yeah, this has been absolutely fantastic. I wish you the absolute best of luck, and I will hope to see you in conferences and whatnot in the months and years ahead. So thank you, Elena, very much for coming on the podcast.

Elena Samuylova (Guest) (34:29):

Thank you for the great conversation.

Peter Wang (Host) (34:32):

Thank you for listening, and we hope you found this episode valuable. If you enjoyed the show, please leave us a five-star review. You can find more information and resources at anaconda.com.

(34:43):

This episode is brought to you by Anaconda, the world's most popular data science platform. We are committed to increasing data literacy and providing data science technology for a better world. Anaconda is the best way to get started with, deploy, and secure Python and data science software on prem or in the cloud. Visit anaconda.com for more information.