EPISODE 1150

[INTRODUCTION]

[00:00:00] JM: Machine learning models require training data, and training data needs to be labeled. Raw images and texts can be labeled using a training data platform like Labelbox. Labelbox is a system of labeling tools that enables a human workforce to create data that is ready to be consumed by machine learning training algorithms. The Labelbox team joins the show today to discuss training data and how to label it.

[INTERVIEW]

[00:00:22] JM: Guys, welcome to the show.

[00:00:25] Ed: Thanks for having us.

[00:00:25] Team: Thank you for having us.

[00:00:27] JM: I want to start off with a simple question. Why is training data important?

[00:00:33] Team: Yeah. What we're seeing a big shift in computing including logic to solve really interesting problems in the real world to building these intelligent applications, but with data. And in this new paradigm of software 2.0 or data-centric programming, these models or algorithms are essentially neural nets and they do not have intrinsic understanding of the world or the systems. The only way that these models learn to emulate pattern recognition is by seeing many examples of decisions, human-like decisions. And these decisions are essentially what we call training data in digital form.

So for example, if you wanted to build an AI system that could detect a tumor in the images, we've got to tell a computer somehow what does tumor looks like and what all the varieties of tumor they can have. And the medium to communicate that to models is in a form of printing data.

That is really best way to tell computer systems about pattern recognition, and models are intrinsically kind of dumb. They don't really have any encoded understanding of the world. And so it all really comes down to the information that is fed to for its learning.

[00:02:12] JM: How do labels get applied to a piece of data?

[00:02:17] Team: There are quite a few ways to apply labels to data. Generally speaking, labels are applied by humans in some form or the other. For example, if you're using Facebook, you're tagging your friends, you're essentially telling the system that that person in a photo is your friend. In other cases, there are systems and teams set up that the only thing that they do is essentially look at information and apply some sort of observation to them. Then there are in real world, for example, in radiology for instance, when you go to a doctor, the doctor looks at the X-ray scans and sort of marks some of the areas for their own wreckage, like that's what the tumor is and so forth. So there is just a whole different ways of how we as human teams generally apply labels on the data.

Now, a lot of these activities are not necessarily done in spirit of building AI systems. They are done for just part of doing the job. But as these AI systems have become more prevalent, it has become more important for the AI creators to create labels in a systematic manner that is high-quality, unbiased. And in order to do so, most of these teams have to develop software as systems, software infrastructure and processes to create these labels.

Now, not all the labels are created by human teams. There are automatic ways to apply labels. So for example, if I have an image of – Again, I'll take an example of a tumor, I could probably write like a function like tumors are generally darker on a light kind of background. Maybe I can have like an OpenCV that detects like darker kind of pixels and automatically apply a label that that is likely to be a tumor.

So we kind of have a whole variety of ways that teams create labels on the raw information. And the important thing really is for the AI creators to subscribe into one or many of these approaches that are relevant and appropriate for their data and domain, the problems that they're solving, and essentially create an organized workflow so that their AI models are constantly being trained with a stream of high-quality training data.

[00:04:52] JM: On Labelbox, explain what Labelbox is.

[00:04:56] Team: Labelbox is a training data platform for AI teams. A training data platform is an emerging category of software, a critical piece in the new paradigm of software 2.0 data-centric programming world. Now, training data platform consists of a three core capabilities; annotate, manage, and iterate. So at the heart of all of this category of software is annotation tools. Annotation tools are essentially like labeling systems that human teams can use to create labels from scratch or review existing labels, like course labels and refine them. Then there is the AI teams that need the ability to manage the training data and understand as quality and bias, share and reuse and repurpose in their organization.

And then the third part of training data platform is iteration, where AI teams need a really easy way to diagnose errors in their AI models and use those insights to take actions such as source and select the right data to label to further increase the model performance. And that is what Labelbox does. Labelbox offers tools and organize workflow in these teams of manage, iterate and annotate.

[00:06:15] JM: So is Labelbox used by people who are working in the same company as the engineers who need the machine learning models or is it used by like outsourcing firms?

[00:06:30] Team: Labelbox is primarily used by AI teams across all different kinds of companies, whether it's small AI teams or one of the largest enterprises on the planet. The portion of our software is used by outsourcing teams for them to render their labeling services to our customers. But primarily, as I mentioned, manage and iterate part of Labelbox is primarily designed for AI teams, because ultimately AI teams need an organized workflow to systematically create training data, find errors and correct them and ultimately train AI models quickly.

[00:07:11] JM: Could you give an example, a prototypical example of a Labelbox use case?

[00:07:16] Team: So Labelbox is predominantly in computer vision. So over the last 2-1/2 years since we've existed, Labelbox actually is used in nearly all imaginable use cases in computer

vision, from medical imagery, to agriculture robotics, to scientific use cases. One of the recent examples I can think of is there are numerous companies in agricultural industry that are pursuing building robots that will essentially enable growers or farmers to very selectively kill the weeds in the early season.

So for example, in today's world, a farmer in Illinois would have to spray pesticides with airplanes or with a general purpose spraying gun, and you've got to apply that pesticides to nearly all of the plants. It's kind of like a shogun approach. And with the advent of computer vision systems, now the tractors will have these modules that will kind of with a camera look very precisely if the plant is of a corn type or if it's a weed. And if it's a weed, it would basically like spray like a small jet of pesticide to kill it right away. And that can reduce – We actually see in real world, that is reducing the application of pesticides by 90%, which is an insane amount. And if these technologies are pervasive, it will make really a big dent in the world.

So now how are these computer vision systems created? Well, it turns out for a robot to identify different kinds of weeds, it needs to first understand the kind of nuances of like whether if it's a plant or the ground. And then more importantly, it needs to identify if it's a weed or if it's actually the plant that we care about. And so we are actually leaders in this space where most of the biggest companies use Labelbox to create this very sophisticated computer vision systems where they have team of agronomists who have studied like these nomenclatures of plants and so forth for decades and they are essentially labeling different kind of weeds in different geographies, which is in turn used to train computer vision systems that can essentially do that and tell robots to spray at the right place. That's one example of many, many other examples that we see.

I think what's really cool is that Labelbox is a very general purpose kind of tool. So if anybody has a pattern recognition problem in visual domain, they can actually use Labelbox to build that computer vision system. And I think it's one of the most profound movements we've seen in this decade, to move from like kind of building these software systems with logic to basically domain experts teaching AI systems directly about the world that they know of.

[00:10:20] JM: So in that example, how work intensive is it to label those plants?

[00:10:27] Team: It depends on of course the particular use case and of course the maturity of the AI teams. Most of our customers actually have very less reliance on human teams, and that's because their AI models are participating in the labeling process using our software platform. So in that particular example, generally, like if a company, if a team is just starting from scratch, they've got to of course create initial – Like it's a cold start problem. You've got to create training data from scratch, and it's very labor intensive. But as soon as teams have just enough data so that they can train a model, even let's say to about 50% to 60% accuracy, they can start to leverage the models to guide them where they should be putting more effort into labeling such as if a model is weak in certain class of weeds, well maybe then the team needs to just focus on that particular kind of data and not label everything.

So most of our customers, if they're starting from scratch, they have to label data laboriously, manually. And over a couple of iterations, they can generally kind of automate or reduce the need for human labeling by as much as 70%. So it really is just a function of the problem that the teams are solving. If they problems are narrower, the faster they can achieve like that scale of automation. And if the problem is more broader, then they need to continue to create a lot of examples and data, because the model just inherently needs to have enough examples of all edge cases.

[00:12:08] JM: Can you describe how Labelbox fits into an overall machine learning workflow?

[00:12:14] Team: Yes, it's fairly simple. In organizations, Labelbox basically integrates with data storage or data lakes. So it could be like Databricks, S3, Google Cloud and so forth. And then the information is brought into Labelbox for labeling, reviewing and so forth. And then on the other side, Labelbox is integrated with the AI training and deployment pipelines and it's sort of is a cyclic process. So as more information is created in the organization, that is sent into Labelbox and things gets labeled. Then the model gets trained. Then the model gets deployed. The model is making decisions in the real world. It will make good decisions, bad decisions. Those bad decisions are brought back into Labelbox to be further reviewed and then the model is retrained again. So generally speaking, most of our customers obviously have their storage hosting problems solved. They're often using PyTorch or TensorFlow to train their models after they ETL labels from Labelbox.

[00:13:19] JM: Let's start to switch into talking about a little bit of engineering of Labelbox itself. Could you give an overview of what the application does? What it looks like?

[00:13:28] Team: A Labelbox application does essentially three things. At the heart of things are world-class annotation tools. These annotation tools are generally kind of very complex because ultimately they support images, videos and to be able to label those formats, these systems have to be highly performant and efficient. So we deal with a lot of computer graphics like problems upfront on the annotation tool side.

Then there are management of training data and iteration workflows, and those problems are essentially around work primarily around data, big data, slicing and dicing, searching across all these complex information. And then workflows of how to understand quality of training data, improve the quality overtime, create labels with different kind of configurations of teams and so forth. So generally speaking, those are the three components of our application, is purely software. It comes as a cloud SaaS as well as on-prem. Ed, do you have any more to share there?

[00:14:32] Ed: Yeah. Maybe the other thing to add is just the – Maybe you already touched on it, but the workflow and like the queue management. How we assign tasks out to the labelers.

[00:14:44] JM: Do you want to describe that in more detail?

[00:14:48] Team: In the annotation part of our application, one of the big problems typically is collaboration with any number of people. And generally creating lots of training data requires of course a lot of people, and these people can be rather inside the organization. They could be of course part of external organizations or outsourcing companies. And so the system has to be able to kind of enable that collaboration and enable these people to work at the same time. So one of the things that we – One of the early innovations of Labelbox was the queuing system. So Labelbox sort of automatically can take the dataset that needs to be labeled and share it across the different people in real-time dynamically.

So for people who are mostly were used to using desktop tools where, literally, in order to create like lots of training data, they would be people who would divide their information into small

chunks and take those USP6 and put them in the computer, load that for person A, person B, person C. And that's kind of like – That was the state-of-the-art two years ago. So Labelbox kind of brought that into the cloud and made it very collaborative. It's almost like coming from like Word, like desktop Word to Google Docs. So that's one of actually very hard problems to solve that can kind of scale for lots of people. And so that's an example of the thing that Lablebox provides among many other things in our platform.

[00:16:23] JM: Is Labelbox and electron app?

[00:16:26] Team: It is not. Our technology stack is essentially React. So most of our UIs are in React and Redux systems, and then of course we use Typescript as a language throughout the frontend world. I'll let Ed describe the backend system.

[00:16:43] Ed: Yeah. There's just a lot of Java and some Go running on Kubernetes in GCP. And of course we have on-prem solution as well.

[00:16:54] JM: What do you put in Java? What do you put in Go?

[00:16:57] Ed: So Java is mostly the services. The Go stuff is just really the two – Things that were running on Kubernetes. I actually don't know the specifics of what we're running in Go right now. Apologies for that.

[00:17:11] JM: That's all right. So you have a bunch of –

[00:17:12] Team: We use Go for asynchronous systems. So webhooks delivery is an example that is implemented in Go.

[00:17:22] JM: Got it. It's still surprising that you wouldn't just use Node. What do you need Go for? Why does it need to be so fast?

[00:17:29] Team: That's a good question. Actually, a lot of our services are node currently. And when we looked at that particular implementation for webhooks, at that time it seemed to us that Go was most prudent choice, because our teams had knowledge building that exact application

in previous roles. So a lot of our current decisions, you have to make them at endpoint at that time of a company's phase. And a lot of it is driven by teams' capabilities and what we are most comfortable with. And so that is an example where kind of team members had a lot of comfort around building Go application for webhooks. And we had seen all the kind of nuances of it, and it was least riskiest decision for us to take to go build that.

[00:18:21] JM: And so let's go through the usage of Labelbox a little bit more. So the plant labeling example. So I've got a bunch of plant images. I open up Labelbox and I do what?

[00:18:35] Team: So typically AI teams, what they do is they need an organized workflow for training data. And an AI team would essentially first integrate Labelbox with their data sources so that the information can come into Labelbox. And then they're going to configure a few things. Number one is the anthology. What is the taxonomy or anthology, the words, needs to be applied so that I can train a model that will be able to detect those items in the anthology.

And the second step is to essentially configure the project where how am I going to create this training data, whether I'm working with internal team of domain experts, whether I'm outsourcing to any number of kinds of these service providers. And maybe the configuration is hybrid. So essentially the team set up those parameters and then they kickoff he labeling project. And as soon as the projects are kicked-off, the labels are created. They're streamed. AI teams in real-time can see labels being created. They can provide feedback to those labels. They can instantly use them for training their models. And in most examples, when our teams have AI models, they essentially integrate their AI models with Labelbox. So these human teams are not labeling from scratch. They are essentially reviewing AI predictions. So the cost of – And that's really important, because cost of correction is a far lower in most cases than cost of creation of labels.

So these are all the things that a machine learning team would kind of do as a part of set up, and that's it. And this is just one project in an organization typically in the enterprise. What we see is they are numerous AI initiatives. They are solving tens of different problems with AI, whether it's computer vision or NLP. And so Labelbox offers these AI teams sort of standardized tools and workflows to be able to simultaneously create all kinds of training data in any kind of configuration while keeping control of the training data in a single place.

[00:20:44] JM: When I click on an image and I label it in Labelbox, what happens on the backend?

[00:20:52] Team: A few things. Well, firstly, before you're able to click an image, the chances are Labelbox system has preprocessed that image to aid in labeling. So there is probably some metadata. There might be some easy ways to kind of like create segments and parts of the image. So we can preprocess super pixels. And then we've got to also ensure that the image is delivered wherever you are around the world as quickly as possible. So there's a lot of preprocessing step that happens before the image even shows up to the client computer.

And then once you create the labels in Labelbox, all of those are essentially annotations and each individual annotations are saved in a database. And so in real-time we are essentially saving them behind-the-scenes so they can be resumed anytime. They could be loaded anytime by other person and so forth, and that's it. So once you submitted all the information and a lot of the attributes, like time and any other metadata that might be generated during that process, it's saving that database. And then our system determines what is the next image for that person to label.

[00:22:04] JM: So we're talking earlier about the work queue for different labelers. Can you tell me more about how the labelers interact with Labelbox and how a team of labelers, their labeling work can be synchronized and parallelized?

[00:22:22] Team: Yeah, for sure. So let's say you've got 100 people that are of label to label data for a particular project, and you've got 10,000 images that needs to be labeled. So in Labelbox it's literally just three steps. You set up a project, upload those 10,000 data rows for images and add 100 people. And in many cases, people simply click just a button to access a very large number of people, services. In that particular project now, Labelbox actually determines like how is that 10,000 rows going to be distributed by 100 people? And the system is essentially a pull system. So these 100 people might not be available online at the same time. Some of them will be in the morning. Some of them would be in the evening. So they are essentially kind of eating away from this queue as the work gets done.

But what gets really complicated is let's say if you wanted to have consensus or if you wanted to have benchmarking system. So these are kind of essentially the tools for quality control. So for example, it turns out in highly specialized industries like medical for example, doctors disagree about a particular kind of diagnosis as much as by 40%. So it turns out like if AI team is trying to build a computer vision system that detects tumors and if doctors disagreement is 40%, that is going to translate to the model and it's going to be really difficult to kind of pursue FDA approval and things like that with that kind of performance of the model.

So teams want to know, teams want to supervise a system of consensus where each image needs to be labeled at least by three or four people or any number of people, different individuals. So our work queuing system gets very complex when it's parameterized like that. So we automatically – Like if the settings are set up where three people needs to label the same image, we are determining that. We are in real-time understanding like what the asset has been labeled by how many times and show that the next asset should be labeled by person B instead.

Now adding to more complexity when customers have AI models participating in the labeling process, well it turns out AI models inherently has knowledge about the data. Meaning that the AI model seems generally know that their AI models are performing weak in identifying objects of type B or type C. And that they might want to prioritize certain data rows up in the queue so that they can create those labels early on in the process and retrain their model to boost the model performance as quickly as possible across those classes. So now our queue system has to be able to also allow these teams to be able to do that. Not only prioritize which asset should be labeled when, but also on top of that have system for consensus or quizzing labelers randomly behind-the-scenes and understand their quality, their performance on that particular task. So it's generally a very hard computer science problem to solve in a scalable manner that meets like 99.99% SLA, and that's an example of kind of a work queuing system in Labelbox.

[00:25:49] JM: Tell me more about quality management, quality management of the labels that are being applied to given images.

[00:25:57] Team: Yeah, quality is a very subjective thing. I think we hear a lot in the world that the data, it's all about high-quality and unbiased data. Actually, what we find is actually not true.

So for example, we have customers in insurance industry and they are labeling data in a super biased manner about how like a claim instance should be adjusted, because what they're trying to really do is they're trying to encode their institutional knowledge, which is essentially proprietary to that organization into training data so that their AI models tries to mimic the behaviors of their own claim adjusters who are trained in that organization to make decisions in a particular way.

So Labelbox as a software enables organizations to create training data and administer quality. Do whatever they see fit. And the way kind of teams assess quality is essentially three things. Number one is the consensus. So in some places, you might want to administer consensus system to collect multiple words for a same kind of image or a video. Second is benchmarking, which is randomly teams might want to administer some sort of a quizzing mechanism to ensure that the person is attentive and making the right decisions.

And then third is essentially having kind of like a review system. And what we see is actually review system to be very important. So taking ideas from like classical manufacturing processes, what you want to do is the biggest innovation of a manufacturing period was to take process and compartmentalizing to small steps and to do them serially and have the people who are trained to do those small tasks at different stages.

So what we see is companies also have, let's say they would label data with any number of kind of like outsourcing teams, internal teams, but they would have a different team that is essentially reviewing all of the labels and ensuring that they are correct or wrong. So there is essentially three kind of ways people use our tools in Labelbox to administer and monitor quality. On top of that, then sometimes our customers also use their AI models to make predictions on the human labels and understand how much overlap there are in those instances. And that also is yet another way for teams to discover gaps or things related to quality.

So I think the answer really there is that there is no silver bullet for quality, because quality in itself is a super subjective term. But what companies, what AI teams wand to be able to do is have the tools and workflows to administer the policy of the quality that they see fit in this workflow and consistently create training data according to that policy. So Labelbox provides basically a lot of collection of tools and techniques to be able to do so.

[00:29:04] JM: There are several different types of labels we might want to apply; images, video, text and audio. One example of a particular difficult labeling task to set up, video seems challenging. Like how do you build the video labeling processes? Do you have to label every frame? Or do you have to label something and then you have some computer vision model that figures out what entities are moving in the video? How do you do that?

[00:29:32] Team: It's a collection of things. Yeah, certainly, video is hard. Harder than just a single image. I think generally speaking as compute gets more pervasive and models get better and better, what we are seeing from Labelbox perspective is that the labeling complexity is increasing. So people are generally more interested in capturing higher order perception and more quickly. And video is actually a great example, because generally models do not have a capacity to understand the time domain. And there's a lot of innovation that is happening that enables these models to kind of remember the last few frames and so forth.

And the hard part in video is essentially being able to tell what is happening over the time period. And we have seen techniques of frame-by-frame labeling and so forth, but that essentially loses the sense of the content. The fact that there is a time domain and that it should be used as an information to decide what's happening on the video.

Particular challenges in video are of course around just the bandwidth. It's sort of a high data, high volume of data. It needs to be able to work in all kind of computers. The tools have to be able to be precise enough to tell what is happening in a video. Like how the pixel is moving from time, like one minute to two minutes and to be able to do that precisely.

Challenges like that are generally hard problems. And by the way, all of these has to work in browser. So they have to be just very performant and snappy and then becomes like computer graphics problems. Like kind of the problems that video game industry continues to solve for video games in the browser.

[00:31:24] JM: Could you tell me more about how you solve for the problem of bad labels or erroneous labels? How do you get those figured out in future iterations of the training process?

[00:31:37] Team: So there are quite a few ways, again, to find bad labels. Beyond the things that I mentioned to you before, like those are actually all prudent ways to identity bad labels. The other way that we often see are most advance customers do is essentially they would do diagnosis of errors in the models. Essentially, typically when you train a model, you're training a model on a validation set. You're testing the efficacy of the model on a validation or test set. Those validation and test sets are also labeled by humans, generally speaking.

And so what often teams do is they essentially overlay model decisions on those subset of data that already is labeled as well with a separate method. And they try to find out like where are the discrepancies that are occurring in the model? And based on those insights, they find like, "Okay, in this example, this weed plant that I predicted is completely of a different type. But it looks like it can detect the edges of the weed plant perfectly." So then the problem is around classifying that particular weed plant. So maybe I need to collect more examples that I should find more examples in my data that could contain this kind of particular classification. And maybe I want to have a human team review that again and make corrections trough them. Or maybe I need to just source, collect more unlabeled data that might contain those examples and label that again and train a model.

So generally speaking, like the errors in the labels are found during the process of labeling and as well as after the fact, once you've trained a model and you kind of compare the model performance with the training data that it was used to test against. So those are the two areas where people often find anomalies in the training data.

[00:33:34] JM: How is Labelbox used for collaboration in ways that we haven't discussed yet?

[00:33:39] Team: Yeah, great question. Just like coding. Coding is a very collaborative process, because it essentially requires people, it requires a personal logic and that to be wedded with different teams in pursuit of solving a systematic problem or creating a system that once it runs, it solves those problems. In the world of AI, training data is the new code .This is how these intelligent systems learn. And it turns out training data is, in other words, essentially are human decisions. And human decisions tend to be erroneous or diverse and so forth. And the way to solve for collecting decisions that are uniform, that are representative of the problem is often done by collaboration, where a team of people who are experts in that domains – Let's just take

an example of the weeds. Agronomists of different backgrounds, they are coming together to label that data. But then there are team of experts who are ensuring that the examples that are labeled are correct or wrong. And if there are discrepancies, they are providing feedback to the different team members that here's a reason why this decision was wrong, or here's a decision, here's a reason why this decision was right.

So in many ways, like you're kind of creating a system where human teams needs to collaborate, do knowledge sharing in spirit of ensuring that things that they label are uniform and correct to the purpose of building the performant AI model. And that is how Labelbox is really powerful for our customers, because it gives this collaboration with any number of people, whether it's in-house team of domain experts. Whether it's outsourcing teams, whether it's machine learning engineers and labelers. It kind of brings all of these three different kind of personas, like labelers, domain experts, machine learning engineers, program managers or product managers together in a same environment. And then they can have this sort of deep conversations and feedback and ensure that the things are being labeled the right way inside the platform.

Now there's another layer of collaboration that happens in our more sophisticated AI teams. Well it turns out that machine learning, they make software development in a sense that the team has high propensity of succeeding in AI initiatives if they are able to iterate faster. So basically like the faster iteration cycle remains to be the cornerstone for the success of machine learning.

And typically when teams are solving a kind of a problem or a multitude of problems, what we see is that they – Let's say they were labeled data with certain anthology and train a model. But as their business connections have changed, market has evolved, they've identified that, "You know what? They need to be able to identify more higher order things in that data or perhaps identify a lot more classes of objects in that data. So they need to be able to go back to the existing training data and simply just make edits to them. In Labelbox, it's super easy to do so. So we see a lot of people just do this iterations with their data and keep massaging it and keep kind of – Yeah, massaging it to the right form to solve the business problem at that time.

We also see in large organizations that the different departments when they create training data, they're often siloed in their kind of like desktop tools and so forth. So when the company is using Labelbox, all of the training data is in cloud, is in the same environment. So it turns out that if a team has a new project to identify like roofs, for example. Instead of labeling everything from scratch, it's very likely that a data person can just search and they will box like does my organization have any data that contains roofs? Okay, I'm going to use that as a starting example and add more things to it to quickly make a training set for my model.

And this is a completely different class of collaboration that we had never thought in the early days that we would see, but it sort of makes sense. That's kind of like how GitHub and GitLab platforms enable collaboration. So we see that kind of aspect of collaboration as well.

[00:38:05] JM: What's the hardest engineering problem you've had to solve in building Labelbox?

[00:38:09] Team: There's quite a few, and I think I'll share some of the problems that I have, and maybe Ed had more examples of problems coming up ahead of us. Well, there are two classes of problems. Number one is computer graphics. To be able to make the labeling systems that are highly performant that can work in the web browsers. These labeling systems are essentially human computer interface. They're not just like labeling – Like a human is sort of like doing paintbrush and so forth. The AI models are also making predictions. And so you want to be able to interact with these AI results in a dynamic fashion, in a performant fashion. Go generally that is a pretty kind of at the apex of problems in frontend world that we continue to address.

The second class of problems is around the backend systems. To give you a scale, one customer can be creating hundreds of millions of annotations for one single model. And Labelbox is used in all industries, and we have hundreds of customers in different industries and many of them are like at that scale. And to be able to kind of manage all of that training data in a single place and slice and dice and search for the right information, almost like a Google search experience, like, "Hey, find me all the training data that contains the cars and it has all these nested JSON attributes," and you want to be able to slice through these different levels. It's a

fundamentally hard problem to solve in computer science in a distributed kind of backend systems and so forth.

So those are two classes of problems that we have solved, we continue to solve that our customers don't have to reinvent and solve them again. But moving ahead, we have a different class of problems, and Ed will share that.

[00:39:56] ED: Yeah. I guess for me moving ahead some of the challenges that I see is really around how customers use our platform in so many different ways, and really are coming up with new use cases that maybe we haven't even thought of really stretching the platform in different ways. And so building a platform that is robust and can handle this huge amount of load that these customers are putting on us, like we're dealing with a lot of data. And being an enterprise software company, our customers are really depending on us and using us in very critical business processes for them. And so building services that are five 9s available, like when we're down, it has a huge impact for our customers, right? Like time is money. And so the challenges ahead for us is really building that enterprise-grad five 9s available system that can scale with the customer demand. That's probably the biggest challenge I see ahead of us.

[00:40:57] JM: Tell me more about those challenges of scalability.

[00:41:01] Ed: Yeah. I mean, a lot of it has to do with technology choices. Some of it is also just the fact that we have to support both a cloud version as well as an on-prem version. So certain technology decisions have to be made that kind of align to both product lines that can raise very interesting engineering discussions and decisions. Yeah, in general, just being able to manage all that data and load and designing your system so that it can be tolerant to any number of failures.

I think one of the things that we're learning now is like expect anything can fail, right? We had some recent outages that were due to some cloud providers, partners. You basically just have to assume everything and everything is going to fail at some point. And designing a system that is resilient to those kind of failures is very challenging.

[00:42:04] JM: Cool. Well, anything else you guys want to add about Labelbox?

[00:42:08] Team: Yeah. The parting parts is essentially that Labelbox's vision is to essentially build a standard data infrastructure – Training data infrastructure for AI teams so that they are able to easily supervise their AI systems and maintain those AI models in production. And why you should work, come work at Labelbox – Well, there are a few things. One, it's really intellectually stimulating problems to solve. We're essentially building products for AI teams in nearly all industries. So instead of choosing one AI kind of problem to solve, at Labelbox, you basically get to see nearly all of the AI problems in the world. We kind of joke internally that Labelbox has this like sort of a house of nearly every AI initiative that is out there in the world and we kind of get to kind of talk about it and learn about it.

As I mentioned, we have incredibly difficult problems to solve in computer graphics and dealing with insane amount of visual data, processing them, creating data ready products from them. We have customers in scale. So teams who are really – People who are just interested in solving these really interesting problems but also want to deal with scale. We've got that. We've got some of the most important customers across different industries. So from agriculture, to healthcare, to defense and intelligence and so forth. And lastly, we really have fun working together. So what we are looking for is similarly-minded people who are excited about AI systems and the problems that we kind of share today. That's it.

[00:43:44] JM: Okay, guys. Well, thanks for coming on the show. It's been a real pleasure talking to you.

[00:43:47] Team: Thank you for having us.

[00:43:47] Ed: Thanks.

[END]