EPISODE 1144

[INTRODUCTION]

[00:00:00] JM: Machine learning models are only as good as the datasets they're trained on. Aquarium is a system that helps machine learning teams make better models by improving their dataset quality. Model improvement is often made by curating high-quality datasets, and Aquarium helps make that a reality.

Peter Gao works on Aquarium, and he joins the show to talk through modern machine learning and the role of Aquarium.

Peter, welcome to the show.

[00:00:24] PG: Yeah, thanks for having me.

[00:00:26] JM: How has machine learning evolved over the last 5 years?

[00:00:30] PG: Ooh! The last 5 years has been such an exciting time. So I think the biggest thing I've seen so far is just the emergence of deep learning into all these really interesting useful tasks that have like a big effect on like the economy and these things that people are doing manually right now. So I think if you look at the history of deep learning, 2012 is sort of when it came on to the scene. And I would say by 2014, 2015, that's when a lot of the really useful applications started to get – It started to work, really, in a lot of things like image detection, and audio classification, and things like that.

But then even in 2015 when I was starting off at Cruise and we were doing deep learning, it was just super inaccessible. So the state of the tool chain was very poor. I worked on Café at Berkley, and it was essentially maintained by grad students. And so a lot of the sort of support and getting set up on your own model was very DIY.

But over the last four years, we've seen that like the state of the tool chain has become much better. There's much more access to pre-trained models. TensorFlow is pretty great. There's a

lot of stuff for deployment and things like that. And so it's gotten to the point where it's easier for someone to get started with a model on their own task or their own domain and apply in a useful way. And so the number of applications has exploded. I would say like the number of companies that are doing really interesting things in the space has exploded.

So even in just like four years ago to now, there was just so many really interesting things that people are doing with AI and ML that wasn't accessible even like 5 years ago even though the technology is fairly well-understood.

[00:02:13] JM: How has dataset management changed?

[00:02:17] PG: It really hasn't. So I think I've talked to so many folks who are working on ML, who are working on deep learning. And the status quo for dataset management is I have a bunch of images and a bunch of JSONs on my local hard drive, and I navigate it using Mac Preview, or a spreadsheet, or a Jupyter Notebook. And so I think there's just a huge gap in terms of what capability can be built there compared to what people have access to right now.

[00:02:51] JM: Explain a little bit about what you're building with Aquarium.

[00:02:54] PG: Yeah. So basically, if you think about an ML model, an ML model is a combination of code and data. And so the code is fairly well-understood. It's usually something that a researcher has made that you can pull off the shelf for something from like the TensorFlow's example repo. But then the data is really the thing that is specific to each client or each customer or each application, and that's where the magic really happens.

So it turns out when you look into a production application of ML, the vast majority of the improvement to a model's performance comes from the data, because that's something that you have a lot of control over. Whereas like the code is something that it takes a lot of effort to like move the needle on something that's very researchy. But then it's really hard right now to understand what is inside of your dataset. Where is it have good representation? Where is it missing representation? What is going on with your model? Where is it doing well and where is it doing badly? And where are the pieces of data that you need to collect next in order to best improve your model performance?

And so my experience at Cruise and I think pretty much every other applied ML practitioner we've talked to, the vast majority of their improvement comes from the data. So the vast majority of their time goes into understanding and improving the datasets. And if you have bad tooling, it just is a super, super painful thing where a lot of teams encounter these pain points where like, "Hey, my model is at 80% accuracy. I want to get it to 95% accuracy." But you have no idea what to do to get it there.

Or let's say that you've trained a model on your test set and it's doing pretty well and then you deploy it in production and it does terribly in this way that you didn't anticipate. Well, what happened, right? Another use case I see very commonly is like I've received a new batch of data from my labeling provider. I train my model on it. It's worse. I have no idea why.

So these three pain points collectively wastes, like in multiple cases I've heard, like multiple people for like two months at a time to try and debug what's going on. Understand why it's happening, and then take the proper corrective action to fixing it. And almost always it comes back to something where it's like a problem with the data quality or a problem in the way that things are labeled differently from like your previous batch of data to your new data. Or here is sort of like piece of the data distribution that you didn't anticipate that your model is now seeing in production and now it's like you're doing very badly on.

And so sort of like the basic tooling around understanding and improving your dataset is something that is super, super impactful for improving your model performance but people don't really have access to. So when I was at Cruise, we started off with essentially no tooling and nothing and we had to improve these models to make them handle the sort of full variation of what you see in the world. And we had to build stuff like this that made that process easier.

And now when I left Cruise, we look outside and see there's all these other mature ML teams, like Tesla and Waymo who've built very similar things in terms of dataset understanding and improvement. But then the vast majority of people out there who are doing machine learning, doing deep learning, don't have access to this type of tooling. So what we're trying to do with Aquarium is make ML easier for everyone by giving them the same sort of tooling that these top-tier teams have for understanding and improving their datasets, understanding and improving

their model performance.

[00:06:24] JM: Is the idea of the name Aquarium, like you're looking into this big ocean of data points and you're kind of seeing what's going on inside of it? Or is it just a random name that was chosen?

[00:06:38] PG: I wish it was as principled as that. Yeah, I need to come up with a better answer for this, but it's kind of like a mix of reasons. I like fishes. I like the see. I love going to the Monterey Bay Aquarium as a kid. I do think there is a sort of like vastness in dataset that you can think of in terms of like the variation of marine life and how datasets are also very diverse in the types of scenarios you want to handle in them.

We also had some systems at Cruise. One of them was called Terrarium. One of them was called Vivarium. And so I thought this is kind of like the nice spiritual successor was to call it Aquarium, but yeah.

[00:07:15] JM: Okay. So I load a dataset into Aquarium. Give me an example of what like a dataset would be and what happens when it gets loaded into Aquarium.

[00:07:26] PG: Yeah. So really what we take in as input is here is your dataset and then here is the results of your model on that dataset. And so what this lets us do is a few things. So first of all, like a lot of users can go into there and just understand what is their dataset look like? They can see here is like a bunch of different examples in their dataset. They can see like, "Hey, if I want to zoom in to one example in particular and see metadata about it, like what is the label? What is the model's result on it? What is like some metadata of like when it was captured or where it came from?" But they can also zoom out and ask questions like, "Hey, what is the distribution of certain parameters on my dataset? What is like the class distribution on my dataset? Where do I have class imbalance? Or where is the distribution across time of day or something like that? Or a device ID or something that you've used that you have some sort of idea that you want to be relatively balanced across the dataset." This helps you just do basic understanding and management of these imbalances that you may want to correct with like data collection.

But then on top of that, we go a step deeper and say, “Let’s not just understand your dataset, but also understand your model’s performance upon this dataset and what you might see in production.” So we can, for example, plot here are the places where you model has the most disagreement with your dataset. So like the highest loss examples. And these typically surface a lot of places where the model is confident that it is correct. But then it disagrees with the label and it actually turns out that’s because the label is wrong. And so you can just look at these examples, and it’s really easy to find all of these places where the labels are wrong. And pretty much every customer dataset we’ve looked, we’ve been able to find issues with this just from doing this very simple check that is really hard to like do for a lot of teams either because they didn’t know that this is a question they should be asking. Or because the state of the tooling meant that it takes like half an hour to an hour to write a script to answer this question, and they kind of forget about it. Whereas, we make it easy to access all these stuff in like three clicks, right?

Another thing we help do is use these neural network embeddings, which is sort of a special property of deep learning models that gives you really good visibility into what a neural network thinks about your dataset and essentially lets you index this unstructured data. So for example with like images and point clouds and audio and natural language, it’s really hard to index in the way that you do with like regular tabular data instead of like a SQL database. So what you can do is you could run a neural network on these individual data points, extract out a layer in this neural network that represents this descriptor. It's an embedding that you can use to compare individual data points to each other and find things like outliers, or find things like clusters. And then once you go and see, “Okay. Well, here'slike the distribution of like these clusters or outliers and how my model does on these clusters, outliers,” you can start to find places where you have places where the model consistently fails. It has a pattern of failures that surfaces that from this like very large dataset in this very easy visual way.

And so you can very easily find, “Okay. Well, here are a lot of places where, again, like and the labels are bad, because these are high-loss examples. But also, here are places where I have clusters of failures where my model is doing really badly.” And so this helps find these problems on specific edge cases so you know about it before you deploy this type of model into production.

And then once you do have the sort of like idea of like, here, where the failures are in terms of bad data or bad model performance, Aquarium helps you take the right corrective action to solve this with the proper application of data. So if it's a bad data problem, then we integrate into your labeling providers so you can send this data back to your labeling provider. They can fix labels. You retrain your model. It gets better. And the flipside is that if it's a problem where your model is doing really badly, then what we can do is we can actually get more data of this hard case that your model is not doing well on so that it can do better the next time you retrain it.

And so, in this case, we can leverage these neural network embeddings to search within your unlabeled dataset to surface here are the things that you should label next and retrain your model on in order for it to do better the next time you deploy it.

[00:11:47] JM: So in some ways, it's kind of like a universal QA tool for machine learning models.

[00:11:54] PG: Yeah, kind of. I think like the way that I like to describe it is it's like in interactive learning tool. And so what I mean by that is if we look at sort of our end goal, like our end goal is to make machine learning easier, right? So if you look at any ML model right now, like any ML project, it takes more time, more engineering resources. It has more unpredictability than any equivalent software engineering product.

And so the reason why people use ML right now is because it's actually like allowing you to do things that you can't accomplish with regular code. But let's not forget, I think the purpose of machine learning in the first place is to make this easier for people to do certain tasks without having to hand write a bunch of rules. They could just like show examples to a model, train on those examples, and then you achieve their desired behavior or result.

And so with Aquarium, what we're trying to do is make it so that you don't need to have a very technical person to get some sort of like code functionality to work out for you. You shouldn't need to necessarily write a lot of code. You shouldn't need someone who's an expert in ML. What you should be able to do at the end is have someone who is a domain expert in the task that you're trying to accomplish who can say, "This is good, and this is bad." And then they can work with the model to build something that accomplishes that without needing to have a lot of

like very technical or slow interaction so that you can just consistently say, “Okay, this is good. This is bad. Train a model on it. Look of the model’s results. Correct in the places where it's wrong. Correct the data in the places where it's wrong, and continue to iterate on this model.” And just through clicking a lot of buttons, be able to make something that does really well for your use case and will continue to improve into production.

[00:13:48] JM: Okay. So you said there might be a couple problems that could emerge from a bad situation. You could have problems in the data. You could have problems in the model. Could we go through each of those examples in a little bit deeper? So like let’s say there're problems with the data, problems with the data labeling. Tell me more about how to triage that kind of situation.

[00:14:12] PG: Yeah. So, really, really common problems we see are with label quality. This is like, for example, if we go back to the self-driving domain. Let’s say like the task is to detect objects in an image. So you tell your labelers to do to draw bounding boxes around all the cars, or the pedestrians, or cyclists that you see in an image.

And so a common label quality issue will be, say, “Hey, they didn't draw a box around this pedestrian,” and so it's missing a label, or maybe the label is off or incorrect in some way apart from that. And so when you have a lot of these errors in your dataset and your model trains on this, it gets confused, right? Because this is something that should be labeled and the system that – And that should be looking for. But it doesn't do well on it and it's being penalized on it, because like there's no human label for it. So that can hurt the model's performance. It can also hurt your evaluation of the model because you're penalizing it for something that’s doing correctly on in certain cases.

Another sort of very common thing we see is like labeling training standards. So let's say like you have two classes that are very similar to each other. Sometimes like your labelers will get confused and make systemic errors. So one of them I've seen is having, say, like different types of signs. May be like one of them, like a slow sign versus a stop sign or something. Sometimes labelers don't have the proper cultural context to understand the difference. Or sometimes they will be instructed in a way that is inconsistent with your expectations as the person who wants to build the ML model. And so these inconsistencies can cause your model to get confused

between very similar classes. And of course then you don't achieve your desired behavior.

So these sort of like bad data issues, like these are sort of things in like your labeled dataset. And so what you can do is you can find these, again, with your sort of model's divergence from the dataset, these sort of high-loss examples, and then that can surface to you, "Okay, here are like the points within this like massive dataset that I need to pay attention to and decide is the model right here, or is the data right here?" Who's wrong?

And then in that case, if it turns out that the data is wrong and then you can say, "Hey, I'm going to change the way that I label. I'm going to adjust my guideline so it's clear in terms of like the instructions to labelers what they need to be doing in these sort of hard or difficult cases."

Or another case is that it's actually like labeling quality standard, in which case like I can resubmit it to my labeling provider and hope that they can fix this example, because the model has already pointed out the place where they are wrong. Or that's also a signal sometimes that maybe you should like be more strict on QA-ing your labeling provider or that you may need to switch labeling providers entirely.

Another sort of like data issue we've seen is not having proper representation of the type of environments that you may see in productions inside of your label training set. So one example I like to give is like let's say like at Cruise, like we're training a cone detector. So our cone detector, what we do is we take, again, our off-the-shelf code and then we collect a dataset of cones for that, our cars had captured. And so we would label these cones and train the model on it. And then hopefully now we have something that detects cone in production. And then we go out in production and then our drivers come back and complain to us that, "Hey, it's not detecting green cones."

And so now you have to go into this sort of debug cycle, which is like, "Okay, number one, why didn't we catch this before it went out into production?" Number two, you go and see what's in your existing dataset, and you realize that all the cones you labeled were orange. So your model has never seen green cones before. So it doesn't know how to handle it. And then now you have to go and take the proper corrective action of saying, "Okay. Well, let's go collect green cones to retrain our model on so that we can handle them in production the next time we

deploy it."

And so I think that example is so illustrative because it captures like the most common problems that a lot of ML practitioners will see, which is that if you just understand the world, like the world is not like sort of uniform. Your model doesn't make uniform errors on the type of things it sees in productions. Like the types of errors it sees are distributed into certain edge cases that you need to go and discover and then address. And it's often not trivial to do this if you're just working with like a spreadsheet or a Jupyter Notebook, right?

And then beyond that, it's like, "Okay. Well, once you have discovered these problems, like how do we actually go address it? How do we find this very rare class within like the sort of CF and labeled data that you might be collecting?" And then, "Okay, how do we detect this before we deploy next time so that we don't have to like scramble in this fire drill when someone, like in our customer or something, starts complaining to us about this?" You see the exact same thing in so many different domains where like, let's say, for example, a company making license plate reading cameras. They train on license plate reading, license plates that they've collected from like Massachusetts, and then they deploy it to a customer site in Utah. And all of a sudden like their cameras aren't working and then they have to do the same sort of debug understanding cycle and then fixing that problem. And it turns out at the end of the day it's because, "Oh yeah, these Utah license plates look nothing like the Massachusetts license plates that we trained on."

So you see in like all these different ML domains the exact same problem of like once you have something that like works at like a basic level, you need to like understand the failure points and understand why they are happening and then go and collect the right data to solve these. And this is the vast majority of the work that an ML engineer does in production. And so what we want to do with Aquarium is to either speed up or completely automate a lot of these tasks so that it's just easier for you to go through the daily flow of making your model better.

[00:20:04] JM: Are there any other user flows that you can describe for the listeners? Why is it people use Aquarium?

[00:20:11] PG: Yeah. I think Aquarium is actually kind of like a Figma for ML. And this is actually a very surprising insight that we only sort of realized a lot more when we started working with

customers. So if you think about Figma, it's something where it has like a somewhat niche user who is like the designer and the design teams. But then a lot of the magic about Figma is enabling collaboration with all the stakeholders and the people around the design team who want to understand what is going on with the product and the product development cycle.

And in a similar case, now we see that with Aquarium, the core user is the ML engineer and possibly like the ML product manager or product operations manager who are tasked with improving a model that is key to the company that has like business value for the company.

And so of course, like we want to make their workflows much easier and more automated and smoother and all that to make their model permit better. But then actually a lot of like the interesting value that we've seen from our customers is the ability to now include these non-technical stakeholders in the ML development lifecycle. So that is to say like all of the people who want to understand how this ML system that is so key to their company is doing. So like let's say that you have a customer that you're selling this ML model to. Like they want to understand the sort of performance characteristics so they understand where they're buying, what they're buying, and like where that model is doing. Where is it doing badly? Or your manager who wants to be able to plot, “Here is like the model performance over time,” and understand what is the ROI on their engineering time that they’ve put into this pipeline, or to the product operations manager and to the product manager who want to understand where things are going well and going badly so that they can apportion their time appropriately.

And so there's a lot of like people now at these companies that are doing some innovative stuff with AI who want to understand what is going on with this ML system. And so I think a lot of the magic about Aquarium right now is facilitating collaboration not only within the ML teams, but within the wider company. So that's really where we want to be. Again, going back to this idea of making ML easier for everyone.

[00:22:38] JM: So let's say I’ve got a model. It’s been running in production for a while. I'm using Aquarium with it, and I want to improve the performance of that model. Take me through how I would use Aquarium to improve the performance of an existing model.

[00:22:53] PG: Yes. So let's say that you've got this model and you've got some dataset that it's

trained on, right? So the first thing you would do is you would get it into Aquarium using our API. We have the Python client library where we ingest sort of the metadata about your dataset and then also the performance results and embeddings from your model. And then now you can go take a look and see, “Okay. Well, here are the places where my model is doing badly. What do they look like? So let's chat out, “Okay. Well, here are like the high-loss examples in my dataset. Are these places where the model is doing badly or are these places where the data is wrong? Okay. I found that. Here are some places where the data is wrong. So I'm going to collect them into Aquarium's issue tracker and then click a button to send it back to my labeling provider so they can fix it. Okay, now let's look for places where the model is wrong.” So we can analyze the embedding space of like essentially the results of our model and look for places where there are clusters of errors or patterns of failures with the model. “Okay, let's go look at those and examine. Okay, these are actually issues with the model. So let's go collect more of this type of data in order to fix that problem the next time we retrain it again.” These are like the green cones within like the larger dataset of orange cones or something like that.

And so in that case, what we can do is you click another button, which is collect more examples of this data. And then what we can do there is look inside of your unlabeled dataset, your production database, or your production dataset rather, and we can see, “Okay. Well, here is the embeddings of everything inside of your production dataset.” And so we can phrase this problem of collecting the right data to improve your model as a search problem. And so we can search with these neural network embeddings to find here are the pieces of data that are most similar to this failure case that you've previously seen. And so if you want to do better on this failure case, let's grab these similar examples from your production unlabeled dataset and send them to labeling for them to label so that the next time you're like training your model on this new dataset with these new labels, it's doing better on the green cones, right?

And then past that, once you have pretty much solidified these two things in terms of what you can see with your existing labeled dataset, now you can start to do comparisons on this is the distribution of my labeled dataset as compared to my production dataset. So you can spot places where, for example, like your production dataset is seeing things that isn't represented inside of your labeled dataset. So you can catch those issues of the green cone even before you go into production, or even before you have any representation of that in your labeled dataset.

So one example that we saw with one of our customers is we plotted, “Here's the distribution in embedding space of your labeled dataset. And then here's the distribution of your production environment.” And we saw that there is a section of their production environment that didn't have any representation in their labeled datasets. So their model had never seen this type of data before.

And then you go look at it and it is just completely like black imagery. It’s just like someone had turned off the lights in that room. And so the model had never seen this sort of data before. And so it was just doing really weirdly and inconsistently on it. So we were able to identify like, “Okay. Well, here's a place that your model has never seen before and it's not doing fairly well on. You should probably collect some of these data and label it so that the next time your model gets deployed in production, it knows what to do in the scenario that it's never trained on before.

So I like to say that like this is sort of like back to this analogy of like if you think about like a human studying for tests, what Aquarium helps you do is find the places where you need to study more, or may be like the practice questions that you're looking to use for your studying are incorrect. Or here's the places that you need to go get certain test questions to study on so that you do better when you are in the real thing. So yeah.

[00:27:18] JM: When you say Aquarium is a Figma for ML. When I look at Figma, I see such immense opportunity for expanding into other verticals, other features. There's almost unlimited potential for where Figma could go. Tell me about some of the potential opportunities you see for Aquarium.

[00:27:37] PG: Yeah. I think the core thing is that AI and ML is going to be this super transformative thing for the way that people, all these complex physical tasks in the economy are going to be done in the future. A lot of things that people do right now or cannot do just because of like their physical limitations are things that AI and ML can either completely automate or augment.

And so I think in the next like 10, 20 years, there’s going to be a huge shift in terms of the way that people work just because AI is going to like really disrupt a lot of these complex processes

that are done right now by people. And so if you start with like the previous generation of machine learning, like pricing algorithms, recommendations, ads targeting. These are the things where it's very clear that algorithms and data can do better than people. And there is just innumerable things that we've seen out there in our customer base that are really interesting that can be done better, again, by algorithms and data. And the big blocker is just like the availability of tooling to make that process easier and more comprehensible and accessible to people who may want to apply those techniques to their domains.

And so I think the number one biggest thing that I see us expanding into right now is just like the myriad types of tasks that people can do that have useful economic value. Our customer base extends from people who are working on agriculture, to logistics, to drones for manufacturing inspection, to people doing food waste inspection, e-commerce, surveillance cameras. There're so many different things that are really interesting applications of AI and ML up there that are very economically viable.

So what we've started off with with Aquarium so far is focusing on visual and perception data. So things like imagery from cameras, or from LiDAR point clouds, because this is a domain that we know very well and it's also the most mature in terms of the industry use cases for computer vision and perception.

But then there are a lot of really interesting stuff that we can do with audio, with NLP, and even for like structured data tasks, tabular data tasks, where I think deep learning is really taking over the world in terms of the types of things that you can do with it and what the future is going to look like for machine learning.

And so the sort of core thing about Aquarium, the core technical thing that makes this work is, "Okay, you have these very large datasets, and you have these very complex models. Let's use a deep learning model to sort of surface. Like here are the places in your dataset in your model performance that you as the human domain expert need to pay attention to."

And what's beautiful about that is that the neural network is really doing all of the hard work for you. And so that means it's very easy for us to trivially like extend into places that you can also apply neural networks to, which again starts with imagery and point clouds. But also the audio,

the NLP, structured tasks that I think are going to make up the vast majority of how people do software in the next like 5 to 10 years.

We focus very hard right now on the sort of like analytics and understanding piece of like the machine learning process, right? Understanding what is going on with your datasets. Understanding what is going on with your model performance. And the reason why is because, like we've had 40 years to come up with really, really good ways to debug your code in regular software engineering. And so that just hasn't happened for data yet. And so this has just been the biggest pain point for the people that we have talked to and the customers that we serve.

And so when we think about what we want to do next, like we first want to make sure that we are doing what we focus on right now. This sort of analytics understanding piece very, very well, and that we are solving this problem for our customers in a way that is like world-class. But the end goal is to make ML easier.

And so I think if you look at the different aspects of what it takes to get an ML pipeline up and running, it spans everything from data labeling, to model training infrastructure, to experiment tracking, to deployment and monitoring. And I think my view of the world in terms of the ML tool chain is that it's going to be a lot of great players in each category who are going to be really, really good at what they do. And we can integrate into those different providers of labeling or different like providers of training infrastructure to provide a great end-user experience to the person who ultimately has to come to Aquarium to understand what is going on in order to like build and improve their ML models.

So I think there is a potential for us to say like, “Hey, maybe we want to get into this space of building some of these things ourselves.” But I think at the end of the day, what we do is going to be dictated by what provides the best end-user experience for the ML practitioner.

[00:33:00] JM: So we’ve talked a little bit about this, but let's say I have a model and I want to improve that model, and I want to use Aquarium, and I want to do so by improving the quality of my datasets. Could you go deeper into the process for improving the quality of datasets?

[00:33:21] PG: Yeah, sure. So in terms of a dataset, like you can only really edit what is inside

of the dataset right now or you can add data to that dataset, right? And so if you think about like editing your dataset, like you usually would want to do that if you have bad data. Like you've discovered like, “Okay. Well, here are some mistakes in the label in your dataset, or that there is like malformed data or things like that.”

And so with that, like what Aquarium helps you do is surface, “Here, are the places where you have labeling errors or like malformed data,” with those like high-loss examples I mentioned before. But then the other thing is, “Okay. Well, we want to add more data to this dataset. What is the right data to add to our dataset?” And that's another thing where Aquarium helps you out by understanding, “Okay. Well, you determine what to add to your dataset by seeing the places where your model is not doing well. Where does your model mess up? Where does your model have cases that it hasn't done well on in production that haven’t been represented in the training dataset?” Or like finding the gaps with your current system, right? It's like debugging code. Like you want to improve the places where you have bugs in your web server or in your web app, or you want to find the places where you need to add new functionality that your web app doesn't have right now.

And so Aquarium's analytics help you find those places where you have these gaps. And then once you know what these gaps are, we help you find that right data in like the larger sea of data that you may have available to you that might be larger than your labeled dataset.

So if you have, for example, like in the standard sort of computer vision tasks, you have these large datasets that you’ve collected from your products that are unlabeled that your model has run on in production. And then you have the smaller dataset of like things that you can dedicate labeling time to. So this labeled dataset, this training data set that you train your models on. So what we can help you do is once you know the places where you have gaps in your model performance, well, we can help you find here are the pieces of data in your unlabeled dataset in your production environment that you need to label next and add to your training dataset so that your model does better the next time you deploy it. Does that make sense?

[00:35:46] JM: It does all make sense. So pulling back a little bit, what kinds of biases could exist in datasets and how would you potentially iron out those biases?

[00:36:00] PG: Right. Yeah. So I think one of the examples that I gave before applies again here. So like the sort of orange cone versus green cone example, right? So you have a dataset that you have collected and labeled and trained your model on. And the bias there is that all of the cones inside of it are orange. And then you go and you deploy this model and start to see that you didn't incorporate green cones. And so now you need to go and label green cones and retrain your model on these green cones that can handle it in production. That's a classic example of a bias.

Another classic example of the bias is the stuff that I talked about with the license plate examples. But even like a lot of the controversy right now of like, “Hey, there is that GAN recently that does facial reconstruction from low-quality imagery, low-resolution imagery. And they found that the vast majority of the faces that it reconstructs look white. And it's not the model’s fault. Like nothing about the model’s code is at problem. The problem is that the dataset that this model was trained on is biased, and it's biased basically with whatever was accessible to that researcher at the time, which is I’m guessing stuff that they've pulled from the Internet.

And so, essentially, there is this bias in a data that reflects bias in society. And it's an undesirable bias and the model has just been taught to reflect what it's found in this data that was pulled from society. And therefore it's producing these results that are obviously not great. And so the first step in fixing these types of biases is to understand that they are happening in the first place. And then once you understand that this is a problem, then you can start to take the right corrective action. In the case of cone detection, it's like, “Hey, we should collect more green cones and train our model on green cones.” And then in the case of like that, like facial reconstruction thing, maybe the answer is like let's collect faces from more ethnically diverse group of people and retrain our model on that so that you it handle them better once you deploy it at your production, right?

So a lot of these problems with ML algorithms and production can be traced back to inadequacies or biases in the data, and Aquarium is basically about helping you find and fix these problems either easier or before you know that they’re happening.

[00:38:33] JM: Are there other platforms out there like Aquarium that are doing something similar?

[00:38:38] PG: No. Not really. That's an interesting thing. I think ML is so early right now, that when we go and talk to people, the biggest competition for us is actually in-house tooling. So people who are working on this ML problem encounter these pain points with understanding and dealing with their datasets, and then they just have to fix it themselves. And so the most basic thing they do is like they us Mac Preview or they use Jupyter Notebooks or spreadsheets, or in one case, iPhoto, actually to manage and understand their datasets. And then sometimes the more advanced teams will go and build like a web app, that's fairly simple to do some of the functionality of dataset querying and understanding a model analysis. So that's actually the biggest competition we see.

But the people who've built really good tools are in this category of very high-budget, high-sophistication groups, like the Waymos, the Teslas, Googles, the Facebooks of the world who have huge internal tooling teams to build stuff to make this ML workflow easier and also have like an army of data scientists who experience this pain point on the day-to-day. So what that means is that when you go talk to these startups who are doing really interesting stuff with ML and they have like maybe like 1 to 5 ML engineers or people working on the pipeline and they can't afford to hire their own team of 10 people to exclusively work on ML tooling, then it's very easy for us to show the value proposition of like Aquarium and their workflows and give it to them at a much higher quality and a lower cost than if they were to try and hack something together in-house. So that's actually been our biggest competition is from in-house teams where they have like maybe like sort of enough people that they have the prospect of building something like this in-house and they're sort of is just a buy versus build decision that has to happen.

I think in terms of like the broader landscape, I'm surprised that there hasn't been more people doing this sort of stuff. I think Scale and Label Box have made noises about getting into the space more seriously. Scale Nucleus came out very recently, which is trying to tackle the exact same problem as us. And I think at the end of the day, like this is still one of the biggest problems in machine learning. And I think the reason why there isn't more people in this field yet is because – Or in building the exact same thing as what we're doing is just because, number one, they haven't had a lot of production deep learning applications to experience these pain points until quite recently when there is this massive influx of like new companies doing deep

learning and like bigger companies, like the FedExs the world, the Honeywells of the world, whatever, investing into deep learning, because they see the disruptive potential on their businesses. That influx of interest is what it took to start having people think about these problems. Whereas, with the self-driving field, we've been encountering them for like quite a few years now.

And the other thing is that to build the right tooling, you need this very particular skillset. You need to understand the ML workflows the ML products, the ML techniques and pain points. But they need to marry that expertise with the ability to build really, really good high-quality web applications and be able to use that to visualize and analyze massive datasets. And it's very rare that you find people who have both. And so the benefit of like me and my cofounder Quinn is that we were the rare people within Cruise that had that crosscutting expertise. And I think a lot of the people out there who are either working on ML or maybe contemplating getting into this space of building what we're building only have one of them. And since they only have one of them, they will fail. So, yeah.

[00:42:38] JM: Wwhat are some other general trends around machine learning that you think will emerge over the next year or the next five years?

[00:42:45] PG: Yeah. I think over the next year, with machine learning, what we're you're going to see is that there is going to be a lot of I, would say, sort of bare-bones applications. I've seen a lot of companies now that are doing things with very little to zero hardware components. So there're a lot of companies that are just taking imagery either from like a basic camera or from an existing stream of imagery or video and doing analytics on it with deep learning and then aggregating those analytics for a customer and giving it to them. And that's generating incredible amounts of value, and it's being applied across spaces from construction, to e-commerce, to industrial kitchens and all these other places where just having analytics on things that come from a video stream is actually super valuable and it's super relatively easy from an ML standpoint and is low on the cost requirements standpoint. I think a recent company that was really interesting was Aquabyte, where they put cameras into fish farms in Norway. And as a result, they can monitor and optimize like the environment in which the fish are being farmed. And they just couldn't do this before, because they couldn't get humans to like dive underwater and examine each of the fish one by one. But now with these cameras underwater with deep

learning running on them, like they can do the sort of like flake individual plus aggregate analysis across like large periods of time. So I think in terms of ML, that's a really exciting field and I think there's going to be a lot of folks working within that.

I think within the five-year mark, that's going to be a point when you're going to have this sort of question of what is the way that the ML ecosystem evolves into. Because like as more people get into doing these sort of ML applications, they're going to want to seek the best tool chain to help them solve their problems. And I think like there's sort of this this diametrically-opposed view of the world that I see on the one hand where people want to make these end-to-end platforms, end-to-end pipelines, where you can just sign up and everything is there for you from like the labeling, to the model training, to the deployment. All of that is just like packaged into like AWS SageMaker, or it's packaged into Metroid or whatever. And maybe that's one way that the ML field is going to go.

But then the other way is going to be I think more like how you see web tooling right now, where like there's very different places where you can build tooling for web development from like the debuggers, to like the underlying infrastructure in which like web applications are deployed, to backend databases, to like monitoring stuff like Sentry. And I think that my view, it's going to be a similar endgame with ML where there's going to be best in class tooling companies or tooling, I guess like open source tooling available for each of the individual aspects of the ML pipeline. And then people would want to be able to stitch those together into whatever works best for their use case. But I could be wrong. Maybe it ends up being that everyone uses one provider or another for absolutely everything and that just works out well for them. But I think that's can it be a really interesting trend in the next five years to see where that goes.

[00:46:21] JM: For people who are out there and they're considering using Aquarium, what would your process be for convincing them to use Aquarium? What's the kind of day one onboarding experience?

[00:46:32] PG: Yeah. So typically what we'll do for new customers, new clients, is we'll just say, "Hey, give us a subset of your dataset. Maybe it's like a thousand examples, 2000 examples, and give us your model's results on that, and we'll whip up a demo for you. And in this demo, we're going to like get your data and your model results into Aquarium and show you here are

the problems in your dataset that you probably didn't know about before. And also, look, it took us only three, four clicks to find them instead of you having to spend like hours writing like Jupyter Notebooks or like compiling reports into slide decks or something like that."

And once we show them, "Hey, here are all the problems that we found on just a subset of your data. Think about all the ones that we can find on your entire dataset. And not only that, we can help you fix them and make your model even better." So that's the typical sort of like sale that we are able to do just from showing a value from day one. And we've made our API for ingesting datasets, ingesting model performances like really easy. So what people give to us is not their raw datasets. So let's say, for example, it's not the raw audio files. IT's not the raw imagery that they may have collected. It's actually you give us like a URL to your image or a raw data and then you give us some JSON fields as like here's the time a day or device or timestamp or whatever, and here are some JSON fields of your labels and your inferences and then your model's embeddings upon that.

And then we ingest that through our REST API, and that we can give you a really great analytical view into what's going on into your dataset and your model performance. And you can use our Python client library to integrate into whatever part of your pipeline or your ML workflow is most convenient for you. So we make it really easy to see value from day one in terms of here are the problems and here are ways to fix them.

[00:48:31] JM: Tell me little bit about what went into building Aquarium and what your software stack looks like.

[00:48:38] PG: Yeah. So our software stack, it's a fairly normal web application that does really interesting stuff in terms of data visualization and in terms of, I guess like taking advantage of the sort of deep inner workings of neural networks. So on the frontend, the sort of basic stuff is like, "Oh, yeah. We use React. But on top of that, we also use WebGL for doing data visualization of these like hundreds and thousands or millions wide like datasets."

And then on the backend, we're built on Python stack on top of GCP. So we heavily utilize things like BigQuery, like Dataflow, Firestore all of that. But then also we do a lot of really interesting stuff in terms of running pre-trained neural networks on large corpuses of data and

extracting out embeddings and doing embedding analysis and dimensionality reduction in order to surface here are the places that users should spend their attention on in order to improve their datasets and their model performance. And then ,yeah, we have a Python client library that we give to folks to make it easier to onboard on to our platform. And I guess those are really the highlights.

[00:49:53] JM: Anything else you'd like to add about Aquarium as we begin to wind down?

[00:49:57] PG: Yeah. I think ML is something that is just so exciting and interesting and also valuable not only in terms of just like the economic effects that it has in terms of augmenting or replacing human labor and all that. But I think it also is interesting from a practitioner's perspective of just thinking about how do these algorithms learn and how you can apply that to how people learn or think about how is it that people understand things and like drawing these sort of serve analogies with human cognition with what you're working with with your machine. And so I think machine learning is something where working in this field doesn't just give you like practical improvements to like important systems that people use for business use cases. But it's also something that explores the nature of cognition and the nature of what it means to be human. And that excites me every day that I work on it.

And so I think, of course, the pitches, it's great. If you are doing something with ML that's really exciting and you want to improve your model performance, definitely reach out. We'd love to work with you either as a customer or as like an employee. But then I think beyond that, I would just encourage people to get into machine learning, because I think it has the potential to unlock not only all these very interesting problems in like, say, climate change, to like the economic optimization, to agriculture, all these different like interesting applied use cases, but also because it helps us understand what it means to build something intelligent. And I think that's a really, really interesting goal for humanity.

[00:51:47] JM: Well, thanks for coming on the show. It's been a pleasure talking to you and thanks for your vision of the future, and Aquarium looks great.

[00:51:53] PG: Yeah, thank you so much. Thanks for having me.

[END]