ZETTA BYTES LIVE – Anthony Goldbloom of Kaggle – September 15, 2020

Ash Fontana, Zetta Venture Partners: I'm Ash, I'm at Zetta, and many of you may have met me. I've had the pleasure of knowing Anthony for a decade, and we got to know each other when he was first starting Kaggle. Then he went off to take that from objects that he was working on in Australia and over here and then grew it quite significantly. And we serendipitously were able to start working together in a more formal way about five years ago. So what we're going to go through today is angled towards something, which is, what do we really need for data scientists and machine learning engineers today? What do they have available to them that's great? And what would they like to have available to them? And the reason we're going with that angle today is because Anthony runs the biggest community of data scientists, machine learning engineers on the planet, by a long way.

So Anthony, I'll start by opening it up asking you to think back to when you were a data scientist yourself for the Reserve Bank of Australia. You still do data science today, but when you were starting as a data scientist, compared to today – and when I say data scientist, I mean someone working on an experiment or trying to learn something by themselves using more statistical methods rather than necessarily more complicated machine learning methods – what's changed in all those years, for that type of individual contributor work?

Anthony Goldbloom, Kaggle: First of all, I was using MATLAB initially; that's changed. Maybe to anchor with where I started my career: I studied econometrics and statistics. So that was my background, but through university or college I had a job as a programmer and I liked the definition that a data scientist is somebody who knows more statistics than the average programmer and more programming than the average statistician. So when my first job out of college was at the Australian treasury and then the Reserve Bank of Australia doing econometric modeling. It's data science-ish, is the way I would put it in that, in that you typically have some theoretically driven model based on theory, and then you fit parameters to that model. That's how econometric modeling is done as you can think of them as parametric models where you assume a functional form. And certainly the time, MATLAB was probably the main thing people were using. I actually remember, um, having a coworker in the cubicle next to me who used to read Slashdot and was trying to get our stuff running on Python at the Reserve Bank. and I thought, this guy is nutty. He ended up joining Kaggle and was a co-worker for a long time. But that was sort of the first I had seen of Python. When Kaggle first launched, I would say it was MATLAB lab and R, and maybe a little bit of Sass, were the dominant things. Python really started to rise. I mean, the random forest was dominating Kaggle early on and there was a nice implementation for Python. I'll give you an interesting fact, we launched our hosted notebook product -- you'll remember this Ash -- around when we started working together in 2015, and we supported R, that was the first language. It really

wasn't clear whether you should support R or Python. The other interesting fact is it wasn't actually a hosted notebook, but it was a text box with a Run button. So two things started really happening since 2015. One is, the switch from R to Python really became very clear. And the second piece is that Jupiter notebooks became by far the dominant ID or environment for doing data science. And that's kind of where we are. It started with MATLAB in their commercial ID, then went to R, a bit of R Studio, a bit of a Revolution R, and now, at least at Kaggle, Python and Jupiter notebooks are really dominant.

AF: And what about your workflow? So that's what you're using; what used to trip you up that hasn't changed, still trips you up, and what used to trip you up that has just been abstracted away and made so much easier?

AG: I think one challenge in answering that question is, over the years, I've used pandas more and more. I've been using the same libraries for a long time, and so I've become very familiar with them. So it may be hard to give you a really clean answer to that because partly the tools have changed but partly my skills have as well. One big thing that has changed from an infrastructure perspective is we went from mostly training on CPU's to GPU's, and I probably don't need to talk too much about what that has done, but that has been a big one. Maybe the change that is underway at the moment is, it went from CPU's to GPU's, the thing that is getting much easier to do now is going from training on a single GPU to training on much larger clusters. TF 2 has made a lot of progress; for instance, TensorFlow 2.0 has made it much, much easier to train across multiple chips. So that's probably something that has really changed of late.

AF: What we're going to do now is we're going to go into three different areas in this theme, and then we'll jump to bigger-picture things. So we'll talk about development, datasets, and deployment, and then we'll go into what's going on at Google, which is where you are and where you run a lot of products, and your experience running set up. To continue on from what we were saying, and to go into a little bit more detail into one element of it, I want to ask you about the core thing that the data scientists use, which is the notebook. For those on the line who are not necessarily familiar with this, think of it as just like a place where you edit text, where you write code, but is connected to other things, has other features. So it might help you auto-correct things that might have lots of little shortcuts, and then it might connect to something else, so it might connect to a version that another one of your colleagues has. That's just a little bit of a background, but think of it as the data scientist's version of Google Docs. So I was wondering what you think is going to happen to the notebook, because on the one hand we've got a very dominant, very good set of open-source options, and on the other hand we still see a lot of companies trying to build a better one. So on the one hand, you've got data scientists doing work by themselves and they'll just choose the best tool for the job. And the other hand, you've got a whole bunch of people that want to work together. So you've

got individuals and then you've got groups, you've got people doing stuff that doesn't require a lot of computing power, you've got people doing stuff that does, you've got people using simple methods and complicated methods. And then you've got open-source versions and other versions. So there are a lot of options for data scientists. Where do you think this is all going to go?

AG: I'd say we're in the process of seeing one fairly obvious change at the moment, which is going from everybody pretty much using notebooks on their local machine to a lot of hosted notebook products. And they're doing really well. One that I obviously I know about in great depth is Kaggle Notebooks, so we have a hosted notebook product. Also at Google there's another product doing exceptionally well called Colab. So that is a very obvious change. Hosted notebook have nice features from a collaboration perspective and versioning your work, such that others can pick it up and extend it. And there's a whole lot of advantages.

There are two shifts that I think are likely to come next. One is, if you look at the two hosted notebooks leading the market, Kaggle Notebooks and Colab, both of them are really not enterprise-grade notebooks. In Kaggle Notebooks' case it's really oriented towards public sharing. In Colab's case, they think of it as a scratch pad, a place where you play around. They have some more production-grade workloads as well, particularly inside Google. So one of the next big things is having something that is really an enterprise-grade notebook with granular permissions and LDAP integration things like that. That's sort of fairly obvious next step and a fairly active place for where a bunch of startups are playing. You've got companies like Hex, which I'm an investor in, going in that direction.

You may also have some folks re-imagining the notebook making it easy to build apps out of a notebook, and a nice example of this as a company called Streamlined which has taken some of the things that work well about the notebook, looking at some of the limitations of notebooks and building an alternative to the notebook, and that's another potentially exciting direction.

I think the third direction that's exciting is making it easier to run really scalable, large-scale jobs. One exciting thing that is certainly in the air is this idea where you run a notebook and underneath the notebook is compute that is a bit more elastic than what you get at the moment. That's another exciting direction there's a fair bit of activity around. So those are three variations. It's impossible to know exactly what's coming next, but these are three active areas that I think are all exciting for different reasons.

AF: You've made a very good strong statement there, which is that nothing available was really enterprise-grade. So two things: one, that is a really interesting, strong statement. And two, just to just to bring some other people back in who may not be working with these products

every day, the way I'm thinking about it as we have this conversation is, just like for most of us going from Word to Google Docs is one thing, going from Google Docs that plugs in through Zapier or through using a different tool like Airtable to having a little app built on a Google spreadsheet or having a a form that fills in a Google Sheet or whatever else, that's another step. So in the data science world, you've gone from working with something that's just the text file on your machine, to working with something that's hosted in the cloud and is backed up, and maybe you can share it with someone really easily, to having it pipe straight into a massive computer running it as an application. That sort of leads to what we're going to talk about next, which was deployment. So again, for those who don't know what I mean by that, deployment is: you built a model, what happens with it? How do you serve it to someone? You built something that makes a prediction about something, how do you get it out there? And there are so many options: you can just ask them what they want to predict and print a report, you can turn it into an Excel sheet, or you can put an application on top of it. And obviously it gets very complicated when you think about something like serving a prediction of the next ride in an Uber application. There's 30 steps that happen between the thing that the data scientists build running and the user getting the car. Of all those different options of how to get a model into the world, from printing a report all the way through to having a very complicated application built on top of it, of most of the work that data scientists and machine learning engineers today are doing, how is it being seen and used? I mean, of course a lot of it goes straight into the trash, being it's an experiment -- didn't work, all right, let's try another one. But how are models really being put out there for the most part in the real world?

AG: Historically, I think it's Josh Wells who breaks the world up into type one and type two data science where type one, the output is, as you say, a report or some sort of batch scoring, and the other is batch inference. The other is in real time, like serving up a recommendation for what movie you want to watch on Netflix or what Amazon product you want to buy. A data scientist working in prototype mode can easily generate their own report for the type one use case, but for the type two use case, getting a model into production is really tough. It's one of the seams. You have a bunch of difficult seems; the seam between UX design and front-end engineering is a difficult seam, and the handoff is not easy. Right now the handoff between a data scientist prototyping and getting something into a production system is hard. There are a few things that make it hard. Very often a data scientist experimenting doesn't really care too much about latency, but if a model is going into production there's a good chance it needs to have some low number of milliseconds in terms of latency. So somebody putting an algorithm into production needs to optimize it for performance. The second issue is, very often a model will take inputs known as features and we'll turn those features and we'll run those features through the algorithm and spit out a prediction. It's fairly easy; data scientists can go into a data warehouse, select the features they want to use. But then when it comes to productionizing the model, you have to have a productionized version of those features. In each of these cases there's quite a few different approaches that are being taken to solving

those problems. One pretty trendy way of making it easy to get features into a production system is using a technology called a feature store. There are some ex-Uber people that started a company called Tecton which has raised a large round from Andreessen Horowitz, focused on this idea of the feature store. And the idea is that these are already pre-built. You have features that are ready to be plucked off the shelf they're already quote-unquote productionized features. And so a data scientist who's prototyping, as long as they restrict themselves to that list of features, there's already a productionized version in the feature store.

For the deploying models, obviously I'm most familiar with what we're doing at Google Cloud. We have these deep learning virtual machines, and the idea is that, so let's say you use a notebook for instance, using one of the already set-up virtual machines, they're set up to play well with Google Cloud's prediction service. S the idea there is that you prototype, you get your model working, as long as you don't change the libraries on that virtual machine it should productionize decently well. Obviously that depends on exactly what was done and what the model involves, but, the idea is to make it quite easy to go from prototyping to production.

And then similarly, Google Cloud has a bunch of different ways to train a model: you can use the raw VM, you can use the managed training service, you can use auto ML, and in each of those cases there's a nice path into an inference engine that should give a pretty good performing inference. The feature store is something that's becoming popular, and with these smooth paths to prediction end points, the products are still being developed and still maturing. But you can definitely see the direction of travel here. Pulling back, if you look at where software engineering was 15 years ago, it was a lot less fun to be a software engineer 15 years ago than it is today. I feel like data scientists in 15 years' time are going to look back at what we have to do today and think, oh, that looks awful. So a lot of the companies that are being built now, and the technology that is being built now, will make this a much more pleasant profession going forward.

AF: I was going to dig in on just one thing you said, which is really, as we see it right on the cutting edge, as in with just seeing some companies build some really cool stuff here, you mentioned Tecton, and there are a few others, and how it parallels with what's happened in software engineering. When I'm referring to this idea of a feature store, to of provide a bit more background for those on the call, you know, a feature of a model and machine learning is something that these predictive of something it's like, one of the things that sort of generates the prediction, just like a feature of the product is one of the things that, you know, perform some function, you click a button to calculate something, and it spits it out to you.

Speaker 1: (20:27)
You click a button, it runs many, many, many calculations, and you're able to check out, um, a card of stuff that you've bought on, on a clothing website, and in the software engineering

world, you know, the big thing over the last 10 years now, even more 10-15 years, is this idea of having Lego blocks, whether it's having lots of reusable code internally, but more so having APIs that you can access. So the example I just gave of checking out of a store, previously you had to just write all of that software, yourself, all the little steps to check various things and make sure the credit card was correct and run it and talk to the bank. You have to do it yourself. Now, you just call a Stripe API or Magento or another API. And this is sort of just beginning to happening in data science and machine learning. It seems where you can start picking and choosing features that people in your own organization have written, um, for a different model. They've built, they've wrote, they wrote them to put into one model to make one sort of prediction, and you can pull them off the shelf, so to speak and put it in to your model for the next thing you want to work on predicting. Um, and then you say that the big companies with, you know, these pre-built or pre-trained models, and that's a whole model, another feature that you can just call and just say, throw it an image, and it throws back some sort of structured output. So, um, I, I don't know. I draw that parallel. Does that, does that make sense to you? Do you think this is how, um, it will help data scientists and machine learning engineers be more productive as in they can just pick and choose things off the shelf like you can with APIs? Or is it just not that simple?

Speaker 2: (22:17)
No, I think, I think it's a reasonable, um, I think that's a reasonable, um, um, mental model for how data science becomes more fun to do. As you said, you pluck your features, um, you grab your pre-trained models. Um, um, you've, you've got things like TF hub, and I think there's a pie torch equivalent, uh, and I'm forgetting off the top of my head, what maybe it's called pie chart. Um, um, um, uh, you've got all the hugging face transformers, like already, you're starting to say this, uh, starting to happen. Um, and it's actually interesting. I was reflecting as, as you were talking about, you know, what software engineering is modularized, uh, this is the direction data science is going. It's interesting to think about the other, you know, one of the other big legs of the technology store or which is, um, designed, which has also going in this way with, um, getting capital has done a big migration to material design.

Speaker 2: (23:04)
And so we have this big, uh, uh, library of components, uh, so that we don't have 20 different ways that a button looks, but we have a button it's already coded up as a material components. We can just, it's been designed and coded so we can just stick the button in. Um, and it makes, so it's, it, it just seems to be the direction, um, you know, the direction, um, that a lot of pieces of the technology stack are going. And so, yeah, it seems like a logical direction for data science, machine learning as well.

Speaker 1: (23:35)

Yeah. It's really amazing how in software, you know, you're able to, and obviously I'm over simplifying things. They can choose things off the shelf and in design, as you said, you can pick and choose different buttons and

Speaker 3: (23:46)
Different banners and colors and everything else off the shelf with Figma Canberra or whatever else. And then in data science, we're trying to get those as well. We're trying to get to that point where people can just like quickly assemble things and, you know, a word that doesn't exist that we could use here is modularize things. Um, so going back to deployment for a sec, and then we'll get into some stuff about datasets. Um, you know, you get to talk to a lot of companies, uh, big companies and small companies you did when you're a Kaggle, because you know, they're in the community and you interacted with them whether they wanted to run competitions or just to get to know the community. And you do a Google to, um, when you think about going back again to the beginning, when you think about the individual data scientist, what is the biggest bottleneck for them into getting a model onto a, off their machine and onto a bigger computer so that someone else can use it access at 24 seven?

Speaker 2: (24:44)
Um, you're talking about productionizing a model, or are you talking about somebody else being able to edit? Uh, yeah, I think it's what I said.

Speaker 3: (24:51)
So someone else can putting it into the cloud.

Speaker 2: (24:55)
Yeah. I th I think it's the two things I mentioned strike me as the largest time. Actually, there's a third as well, but the two biggest pain points, um, or two of the big pain points are what I said that they're getting the product, the features into a production grade setting, and then having the model, um, um, be deployed as something that is low latency enough, uh, that it's a good end user experience, the final paste by the wire, which is a really important one. There's also model monitoring. Right. It's all good and well, too. Um, it's kind of scary with machine learning, um, because models drift over time, right. You know, you might have a composite to change in the composition of your customer base, for instance, or you have a global pandemic or something, very drastic changes. Um, and, uh, and your model, you know, it's, um, software engineering.

Speaker 2: (25:41)
You can, you can write tests and it's a deterministic system. And, uh, you can test that when, um, uh, you started this into a function. This is the output, right. Um, machine learning is somewhat scary to have in production because it's, um, not quite a black box, but it's also not,

not a, not a deterministic system, right. Where, um, um, and they're often quite complex systems. And so, uh, having really good monitor monitoring in place to see when a, you know, when the distribution of features going into, uh, a model that's been deployed, you know, when the features start to drift or, um, um, for whatever reason when the Mo you know, if you're doing regular retraining, when the, when the, the, uh, the retraining is giving you, um, you know, causing big changes in the model or the whole lot of things that could be. So that's a, another big gotcha. With respect to getting models into production.

Speaker 3: (26:35)
And is this different for small companies and big companies, like once you get to the big company level, you know, is there another totally, totally different quality of

Speaker 1: (26:44)
Problem that they're facing?

Speaker 2: (26:45)
I think, um, I, I mean, obviously the high volume, um, particularly historically very high volume, very low latency models, um, had to be, you just had to be very, um, you have to be very conscious of how heavy the models were. Um, um, I think that's, there's a lot of good technology being developed now by places like Google, um, being able to try and large, uh, models and have them, uh, uh, execute pretty low latency. Um, the other issue with, um, depending on the industry, that's another way to cut this. Some industries are heavily regulated. So I know to the extent that cattle's worked in insurance or, or, uh, financial services, um, uh, financial services has the fair credit reporting act. And so if you're building a credit scoring model, um, um, not only can the model not discriminate use features like, um, gender or race or things like that.

Speaker 2: (27:36)
Um, but the effect of the model, um, uh, you know, the, the, the model, uh, it's not just what you've put in, but you can't have variables that are proxies to those things. Um, and, uh, and you know, his zip code, a proxy for, um, rice, for instance, um, um, in some cases it actually might be. And so that's often a big challenge. Um, um, and then, uh, for insurance, uh, regulation is on a state by state basis. And so you have to get your, your, um, uh, a new algorithm, um, approved. It gets approved as a pricing matrix. And, uh, uh, and so making changes to models is really hard because you have to go around to, you know, 50 plus state regulators. So there's a lot of challenges on an industry by industry basis, uh, um, uh, with respect to deploying models that limit how often they're refreshed and what sort of, you know, what sort of things about what, you know, what, what is, how, how, yeah. Uh, how the model is trained and how it's, um, how it's used.

Speaker 1: (28:44)
So let's move over to datasets. What's the state of datasets today on Kendall. Um, how many is using them? What are some sort of headline numbers that give us a flavor for what's available there?

Speaker 2: (28:56)
Yeah, I mean, for those who aren't familiar with Cara, we have three main components. We have machine learning competitions, which we're most famous for. Um, we have the hosted notebook products, um, which I've spoken about already. Um, and then a third component, which is actually the fastest growing part of Kaggle is, um, we have a data sets platform. And so, uh, the way to think about this is it's almost like what YouTube is for videos. We have the datasets, anyone can upload any dataset, uh, to Kaggle. Um, we have over 50,000, um, uh, public, we share datasets, um, and it's really nice, you know, one nice trend we're starting to see is people are using us, um, to keep very high quality, um, always updating, um, datasets available to community. So we have really nice set of COVID-19 datasets, um, um, ranging from, you know, the obvious John Hopkins university dataset on, um, cases and, and, uh, recoveries and fatalities by city.

Speaker 2: (29:55)
Um, then people do cool things like they'll take that dataset and they'll join. There's been a question around what is the impact on weather on transmission, for instance? So, um, uh, when weather is warmer, does, does it slow transmission, uh, humidity is set to set to, I don't have an impact on transmission as well. And so it's really cool. You can, uh, somebody has taken, we had a weather, very detailed weather data, global weather data set. We have the, the John Hopkins university COVID data set, and somebody joined the two together, matching every city in the John Hopkins university dataset to the nearest weather station. So it's not just that we have these individual data sets where people were starting to do really cool things. And so if you're a researcher and you want to write a paper on the link between, um, uh, you know, um, cases and fatalities and weather like sitting on Kaggle right now is a beautiful dataset for you to grab, um, that's continually the code is there that continually updates it.

Speaker 2: (30:54)
So there's some really nice stuff, um, um, that you see starting to happen with data sets in our community, um, pulling back to machine learning more generally, I think it's, uh, um, data sets is, uh, is still one of the challenging pieces. Um, um, you know, I like the joke that 80% of data science is cleaning the data and 20% complaining about cleaning the data and it sort of rings right to me. Um, um, you have, um, uh, it's, it's just like a very, um, yeah, it's probably to the extent that this toil involved in data science and there is, um, it's probably the place where there is still the most toil, um, also the place where it's least obvious what sort of tooling would be helpful and generalizable.

Speaker 1: (31:41)
Yeah. I mean, w so I'm sort of wondering as you say this with so many datasets on Kaggle, where else do people go? Like, what are some other places people can go to get data just to play around with, if they're doing a data science course at university or whatever else just to practice these methods, these, um, these methods.

Speaker 2: (32:03)
I mean, the idea that there's a single place to go for datasets where it's, it's something that I believe should exist in the world and doesn't end to the extent that it exists. Kaggle's probably further ahead than anywhere else. And we're not very far towards that goal. Uh, just to, um, um, there is a, Google has a dataset search, um, which, uh, is a nice resource as well. And so that draws on CA you know, when you search on Google it's dataset search, you search category, but you also search, um, basically anybody who, um, who is, uh, using the right schema.org, um, um, uh, uh, um, some semantics. Um, so that's, uh, that's a place you can go, there's obviously general Google search, um, um, uh, but there is, you know, if you want government datasets, you've got at a state or federal government that you're there, isn't there, it's, it is, uh, yeah, it's, you know, catalyst running after this problem, trying to be the one place you go, uh, uh, to find data sets. And, uh, so if you were going to search a single place, actually, I'd probably do a Google status set search ahead of us. Um, but, uh, um, good chance you'll end up on our website. If that, if you do make that search,

Speaker 3: (33:18)
Uh, many governments, local, municipal, uh, federal sort of getting involved in posting data sets here, or people just reposting what

Speaker 2: (33:28)
People are already posting. Um, um, we have not, um, there's a company, uh, called Socrata that, um, and then there's an open source project called [inaudible]. Um, and those really have been the dominant places where, um, government, uh, in Europe, uh, I think it's mostly see can, um, um, and in the U S a lot of local governments using Socrata. Um, uh, and so that's the thing with Socrata is every it's, uh, every state government has a white listed, you know, Socrata instance, right. Um, um, so it's, it's not like you can go to this CRADA website. In fact, let's say I'm 80% sure this is right. And someone can correct me if I could have this wrong, but I'm fairly certain, you can't go to the Socrata website and search all the state. Like everyone has their own instance. So it's more of a software as a service

Speaker 3: (34:21)
Corrections, they start throwing questions, comments, et cetera, into the chat. We're going to go through them pretty soon, actually in a couple of minutes, about five minutes. Um, so, and

as people prepare datasets for competition, so we see a big product catalog is competitions, or wasn't still, it's about a pot of Kaggle. And that is a big company, has a problem. They think can be solved using some, some cool stuff that the community can come up with. So they put a data set out there. So, you know, in helping those companies put these data sets out there for competition, you probably come across a lot of data prep issues in general. And, you know, some of those data prep issues, especially for competitions that might involve supervised learning is a solution there's a lot of labeling of data to do. Um, so that is cleaning it up, tagging it, et cetera. So people can start using the tags to predict certain things. And, um, what the big companies sort of do when they need to label a huge amount of data today. Um, you know, are they doing this internally? They're contracting that, who are they contact? What's the store to say, to play there for, you know, a large corporate that needs to re um, label a lot of data? Yeah.

Speaker 2: (35:44)
I mean, there's a mix it's companies like Google that have built their own infrastructure for it. There are, um, and, and Tesla is doing that as well. And just as a mild aside, one of the talks that I think is most interesting on, um, on data labeling is, uh, by Andre Carpathia who runs AI at Tesla. And, uh, um, the talk is called software 2.0. Um, and he, uh, I think he paints a very compelling partial vision around what machine learning tooling should look like. Um, he makes the case that, whereas with software 1.0, I mean, conventional software engineering, it's IDs and debug as in profile as a, uh, kind of the key pieces that help, uh, um, help, uh, software engineer do good or good work. He admit the argument he makes is that for software 2.0, actually it's all the things around managing label diet or in the labeling tools, um, that matter most for the performance of a model.

Speaker 2: (36:38)
So, as an example, um, if you're a model you seeing where your model is systematically underperforming, um, um, if you are piping more of that kind of data, you know, if you're trying to do, I don't know, recognize stop signs, for instance, and model is consistently missing stop signs, getting more data that with stop signs in it is, uh, is one thing you can do to improve the quality of your model. Um, um, uh, you know, that's, it's not as active learning. Um, so th th that sort of stuff I think is, um, that sort of stuff I think is really, um, um, um, or what Andre talks about is very, very exciting from a data labeling, you know, the vision for data labeling perspective, um, um, pulling back, I think, um, that market is fairly fragmented. I think a lot of the autonomous vehicle companies use scale. Um, um, but then you've got figure eight and you've got, um, label box and you've got, um, actually this cool new technology, um, out of Stanford called snorkel, which is an open source. It takes a bit of a different approach where you use heuristics to inform your labels. So that's a promising new angle. Um, so there's, there's, uh, yeah, there's, the data labeling is, is crucial. Um, uh, there's a lot of options out there and, um, yeah, I think it's, uh, it hasn't settled yet. Yeah.

Speaker 3: (38:05)
Okay. Um, so let's talk quickly about one more thing before we get to questions. Um, you've mentioned throughout, so whether it's about notebooks, whether it's about development monitoring, and then, uh, whether it's about labeling, you just mentioned, you know, Google's built their own internal thing for labeling. Obviously you can't talk about what Google's got available to its own developers, but what are some just, just really quickly, what are some Google cloud products that came recently

Speaker 4: (38:36)
That, um, people should know about? Um, you know, they might've missed the announcement or whatever else, but what's some really cool stuff that's come out recently.

Speaker 2: (38:46)
Yeah. I mean, one of the, um, perhaps the thing I'm most excited about that's come out of Google recently is, um, um, uh, uh, to be, sorry, not recently, but, um, is recently maturing as TPU. So, um, uh, we had, um, the jump from CPU's to GPU's has completely transformed machine learning and, um, tip use have, uh, um, GPS, um, uh, have real strengths on certain workloads and TPS have real strengths on certain workloads. And, um, I think that, um, some of the challenges with TPU is historically is they have been quite hard to use, um, and usability has really improved. Um, um, there was the launch of, uh, TensorFlow 2.1, um, which had really nice, um, uses the [inaudible] to train [inaudible], which train on T is rather, and it makes it trivial to switch between two years and GPU's, um, solves a lot of the challenges people had from a usability perspective and now PI torch.

Speaker 2: (39:44)
Um, the PI torch SOI, I think, has now GI. So this is a flavor of PI torch that runs, uh, on, uh, Tiffy who's in the, so I think, um, some of the areas where tip you shine in areas where, um, historically we ha where I think we, it is possible to see, um, um, re really strong model for performance improvements and probably the biggest are, um, TV use of very, very good for very large gal models. And they're also good, particularly good for very large, uh, recommend, uh, systems and ranking systems. So, um, these are th that's an area I'm, um, excited about. Um, um, and then the other one is, um, this is not as a single product, um, but the idea that there are, um, probably applies to all the clouds, um, the idea that, uh, that a lot of the toil around doing data science is starting to get easier and easier. You know, some of the things I mentioned earlier, like the, the various investments in feature stores and, and, uh, and, uh, um, you know, easy, easy, easy ways to push models into production, uh, make data science a lot less, um, uh, I guess, tedious to do and makes data scientists much more productive. I'm sorry. It's just like a lot of little, little pipette guts that are getting cleaned up, um, by some of the cloud providers, as well as a lot of startups running it this as well.

Speaker 4: (41:12)
Okay. Um, let's go to a question. Uh, I think we've got a question wedding from Nathan, um, and when he jumps on, uh, we'll let him ask that question. Um, there we go. Um, yes, I'm, I'm Nathan joining in from, uh, from Portugal actually. Um, so I've been, uh, a big fan of Kaggle over the years. And, um, one of the things that I was curious to get your take on was, um, especially why I guess, like cargo, um, was held back and its like potential that I saw to basically be like a problem solving as a service provider. Um, cause like you were on sourcing all these really neat problems from enterprises, but I sort of feel like you could have probably gone a bit further and maybe systematized that offering to many more problems that were out there and like you've seen kind of mini Kaggle's as it were and finance like a numerous, if you could cast that across lots of different industries, like why didn't you do that?

Speaker 2: (42:16)
Yeah. I, I, um, I guess a bunch of different directions, casual experimented over the years with a lot of different directions. Um, our initial phases was that there would be three ways of getting data science problem solved. One is you have an internal team. The second you have a consultant, the third is catalog. It's like a crowdsourcing style offering Hodson are exactly what the broke up would be between the three. But you know, there'd be some share would be done by internal teams, some share by consultants, John's share by a crowdsourcing mechanism. Um, a few things, um, um, uh, kind of made that challenging. Um, uh, um, one is like we would work with, um, an insurance company for instance, and we'd work on an, uh, claims prediction problem. And uh, then that would have the recipe for how to solve the claims prediction problem and that, so take the recipe and roll it out across all their models.

Speaker 2: (43:09)
Right. Um, and uh, and then we're sort of useless, uh, on that problem, but you know, we started on get more, uh, more renewal business. Um, the other thing is that taking models, how to Kaggle was always, it's kind of what I was talking about before. It's hard to productionize them, right. And so there's a tremendous amount of work if you think of an end to end machine learning problem, um, where some fraction of the solution, but there's a heck of a lot of work before a cackle competition in terms of framing the problem. And then, uh, after it, um, so let's say we're 30% of the work. And I think what really, what we are most useful for in practice was, um, you know, companies are trying to problem internally. They're done as well as they could. They wanted to see like one it's to say how well it was able to do what, how well the outside world could do on the problem and what the methods were.

Speaker 2: (44:01)
Um, and then, so that really was the strongest use case. And there's just like a limit to how far you can scale that use case. Um, then the other thing we toyed a little bit with was this idea of like a consulting marketplace, this idea where we could match companies with individuals in

our community. And I think there's a lot of challenges there as well. Um, um, particularly for larger companies like giving access to the, to, uh, people from our community to datasets was, was certainly big blocker. Um, also I think very often for more open-ended problems having really good domain context is helpful. Um, um, and then a lot of those in, um, to do well as a consultant. I think you need a combination of very strong project management skills as well as a strong machine learning and doing well on Kaggle. Um, definitely demonstrate strong machine learning skills, but not necessarily project management skills. So, um, to the extent that we tried different permutations of, um, scaling up the idea of companies, posing problems and our community solving them, those that some of the blockers that we experienced and some of the reasons that rather than, you know, digging in data there and trying to really scale that we went broader to doing things like Kaggle notebooks and, uh, datasets.

Speaker 4: (45:22)
That answers the question. Thanks. Uh, thanks for, for that. Um, I think you could probably like look at all the topics that you guys discussed over the last, like 45 minutes and actually you would probably try this whole adventure again, it would make your life a lot easier if like things like synthetic data, like easier pipelines, stuff like that. So maybe we'll find it somewhere on the line. Please go ahead and ask the question.

Speaker 2: (45:56)
I'm pretty happy with Kaggle 1.0, checking, sorry, Mac,

Speaker 4: (46:02)
Um, uh, uh, work at, uh, New York MTA, uh, et cetera. You talked about active learning, uh, and transit. We don't do a whole lot of active learning. We pretty much provide service and with COVID is kind of, uh, expose a lot of the vulnerabilities that we have, um, and our inability to react to those vulnerabilities. Um, I was intrigued by, you know, whether or not there's opportunity. Like, look, we don't have any data scientists in our, in our agency guys, 77,000 people working for us. Um, how do we leverage some of the communities that are out there in the work that you're doing to try to figure out a way to be more expensive COVID, uh, challenges differently, but more importantly, how do we, you know, the data sets and the information, it's not that we don't have smart people in our agency. It's just that we're not programmed that way. And if you had any suggestions on how it's not just us, by the way, it's not just empty. Uh, and, uh, and we, we are, uh, you know, we, we have a very strong vulnerability right now. Right. And how do we use either your approach on data or machine learning? You'd get us where we can be more reactive.

Speaker 2: (47:20)
Yeah. I don't think Kaggle is a good fit, to be honest. And I think the, um, I think we are a better fit for, you know, you have a data science team already and we have a bunch of things that are

useful as a way to make them more productive. Um, I think that, um, and I think it's kind of hard to tap the community while they're outside the walls of, uh, the S the, the MTA. Um, my, um, uh, my guidance on this always is, um, uh, you know, start with a very small team, um, and start with small wins, right? Like, um, uh, and, uh, and have the team, you know, small win, um, bigger, win, bigger win, like, you know, start to, um, just like build up the muscle. Right. Um, um, and it's, it's the sort of thing that it's not a quick fix, but, um, you know, maybe you want to start off with, um, I don't know, a better demand forecasting or something, and there's a way to, I don't know, I'm just getting into a demand, uh, mentality.

Speaker 2: (48:25)
Okay. Fair enough. Look, the problem, as you know, with public sector is you get the muscle memory in the muscle. Yeah, sure. Um, although if you know that the, um, P people, people turn out, I think, you know, public sector, private sector, you look at the Bay area and there's always a shiny, shiny thing, but, um, if you ha, if you can get a bit of a critical mass of a team, you know, three or four, so you'll lose one, but then you have two who have learned from the one and then one of those lakes and he sorta, yeah, exactly. Uh, uh, I think the term they used as boss factor, you know, make sure there's not a, um, a bus factor of, uh, of one being, you know, that person lays all the knowledge lays. Um, yeah.

Speaker 1: (49:11)
Okay. Um, we do have one more question in the Q and a, but I don't actually think that that's a good question. Um, because it might involve talking about too much stuff that's internal to Google. So, um, do appreciate it, but I don't think it's a question that we can probably answer in a relatively public forum like this. Um, if there are any other questions, please shoot them through. Um, all, just ask one more question. Um, while we wait to see if we get any more, um, and you built a community, um, from the ground up, um, and I guess as you sort of went through that process of building a community, um, it's something that you, at least when we were working together, like you always knew that this was a very valuable, important community, but it was sometimes hard for investors to see that ahead of the more obvious things that tell you a business's working on, which is we've got money coming in the door.

Speaker 1: (50:19)
And so for those founders out there building sort of community type businesses, community led businesses like developer communities and whatnot, what do you think is some important metrics, um, that will tell them they're building something really valuable. And this just sort of takes you back to when did you know that this community you were putting together, um, was a really valuable community that was doing great things and would go on to do important work. And if you sort of were running that community, you could be part of that, but yeah. What, what was it, what, what did you see early on and what do you think are important metrics you building community?

Speaker 2: (51:00)
Um, I don't know. Um, I guess what I, um, one of the reasons, you know, catego had lots of ups and downs and one reason that I always felt very motivated to push through the downs is that, um, I thought it felt like we had built something special. I don't know that I, um, I don't know that, um, commercially valuable was the Mo was the motor was like the thing that, um, um, was my driving force through a lot of cattle, but, um, I've gotten to know a lot of the community capital is a big part of their lives, had made a big difference. Um, um, and I think probably I had the somewhat naive assumption. I no longer believe this. Um, you know, I guess as you, as you see more is you see more? I, um, I used to, you know, I think when I first started catego, I used to think, Oh, you create value in the world and you can capture some of it.

Speaker 2: (51:46)
It's actually, um, um, there are a lot of open source software is a great example of this. Um, a lot of cases or consultant projects create a tremendous amount of value. And, uh, in some cases you might, they monetize nicely and turn into good companies and other cases I don't, um, I think now it is easier for certain types of businesses, particularly open source businesses. There's a lot of templates to follow. Um, um, um, you know, you've got open core, you've got, um, um, and, um, support models, like, yeah, yeah. And so I think there's probably more prior art now, um, uh, to give an idea for how to monetize, uh, depending on what the type of community is. Um, but Carol Calgary's community, uh, is somewhat it's, it's different to a lot of other communities. I look at it as more analogous to get hub in stack overflow than, uh, than an source community.

Speaker 2: (52:41)
Um, and I think if you look at communities like Kaggle, like you'd have like stack overflow, um, monetizing them is, um, is not like a single, um, I don't know that there's like a single template, um, uh, that each of us have followed in order to monetize effectively. And I think also the other thing is we've monetized, um, uh, you know, each of us have monetized more or less with more or less success. So I think our type of community is a little anomalous. Um, as far as community health is concerned, um, um, uh, one dynamic we certainly see with Kaggle, which is a dynamic where I'm really trying to work hard on is I think a lot of people pass Kaggle as a result of, you know, through their machine learning, um, journeys. Um, but one thing that stack overflow and GitHub do have now really nicely is there's probably not a day or a week that goes by that a software engineer is not on GitHub and or stack overflow.

Speaker 2: (53:40)
Um, and I think that, um, with Kaggle, um, there's a lot of reasons to come back on a daily and weekly basis. Like if you're starting a project, we have an example notebook that will show you how others have tackled that project. If you're looking for a data set, we have a lodge dataset

repository. Um, but, um, we ha we are less of a daily habit for data scientists and I think stack overflow and get hub. And so to give you maybe that rather than answering, you know, what metrics tell you about health, the metric that we are we care most about, um, is, uh, you know, the ratio of monthly active users to, uh, um, to the size of our user base. That's when we really focused on at the moment.

Speaker 1: (54:21)
Okay. Yeah. But in the early days it was about knowing them. Um, that's a good, that's a good one to end on. Well, thank you very much. And, um, and thank you very much everyone for your time. Um, we will be having another one of these, um, in about a month or so. Um, we'll announce that in the next newsletter we send out, uh, there's this survey link and the sign up link so that you don't miss that, but yes, a survey link in there to tell us what you'd like to see more or less over the next one. Um, and thank you very much again for your time and thank you AMS, um, for the work that you do and for sharing some of it with us today.

Speaker 2: (55:01)
Thanks for having me nice to meet everyone.

Speaker 1: (55:03)
So you guys.