EPISODE 69

[INTRODUCTION]

[0:00:10.8] SC: Hello and welcome to another episode of TWiML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

Thanks so much to those of you who participated in the TWiML online meetup last week, and to Kevin Tee from SigOpt for presenting. You can find the slides for Kevin's presentation in the meetup Slack channel, as well as in this week's show notes.

Our final meetup of the year will be held on Wednesday, December 13th. Bring your thoughts on the top machine learning and AI stories of 2017 for our discussion segment. For our main presentation, prior TWiML Talk guest, Bruno Goncalvez will be discussing the paper Understanding Deep Learning Requires Rethinking Generalization by Chiyuan Zhang from MIT and Google Brain and others.

You can find more details and register at twimlai.com/meetup. If you receive my newsletter you already know this, but TWiML is growing and we're looking for an energetic and passionate community manager to help expand our programs.

This fulltime position can be remote, but if you happen to live in St. Louis, all the better. If you're interested, please reach out to me for additional details. I should mention that if you don't already get my newsletter, you are really missing out and should visit twimlai.com/newsletter to sign up.

This week, we'll be featuring a series of shows recorded from Strange Loop, a great developer-focused conference that takes place every year right in my backyard. Not literally in my backyard, but here in St. Louis. The conference is a multi-disciplinary melting pot of developers and thinkers across a variety of fields, and we're happy to be able to bring a bit of it to those of you who couldn't make it in person.

Later this week, you'll hear from Soumith Chintala, a research engineer at Facebook AI Research Lab; Matt Taylor, Open Source Community Manager at Numenta; Allison Parrish, Professor in the Interactive Telecommunications Program at NYU and Sam Richie, Software Engineer at Stripe.

We like to send a huge shoutout to Nexosis who helped make this series possible. Nexosis is a company focused on making machine learning more easily accessible to enterprise developers. In fact, you'll learn a bunch more about the company and what they're up to in this show, which features my interview with Nexosis founders Ryan Sevey and Jason Montgomery.

Ryan, Jason and I discussed how they got their start by applying ML to identifying cheaters in video games, the application of machine learning for time series data analysis, and of course the Nexosis machine learning API.

If you like what you hear, they invite you to get your free Nexosis API key and discover what they can bring to your next project. You can do that nexosis.com/twiml. T hat's N-E-X-O-S-I-S.com/twiml.

Now, on to the show.

[INTERVIEW]

[0:03:21.6] SC: Hey, everyone. I am on the line with Ryan Sevey and Jason Montgomery. Ryan is the CEO and a co-founder of Nexosis. Jason is the CTO and also a co-founder. Welcome to This Week in Machine Learning and AI, guys.

[0:03:37.1] JM: Thanks for having us.

[0:03:38.5] RS: Yeah, awesome to be here. Thanks so much for hosting us.

[0:03:40.2] SC: Fantastic. Fantastic. As is the tradition here on the show, why don't we get started by having each of you introduce yourselves and tell us a little bit about your journey to machine learning and AI?

[0:03:54.0] RS: Yeah, sure. As you mentioned, I am the CEO and co-founder of Nexosis. Jason here is our CTO. Just going back to maybe the beginning of how all these came to be. Jason and I actually met each other at a company called American Electric Power, which is headquartered here in Columbus, Ohio.

Him and I were both on the cyber security engineering team. AP I believe is the largest generator of electricity in the United States. When you think about that, you can think about all the different assets that they have on the field, which directly correlates how much data is being collected. Basically, our task there was to ensure that all their assets were secure from both internal and external attackers.

We quickly realized that the amount of data that was being generated would be pretty much impossible for the analyst team to go through and identify any kind of numb lacing to that nature. Jason and I then started thinking and hearing more about machine learning and there were classes offered online, I believe through Stanford University.

Him and I both signed up to do these online courses, and we went and did all the homework and all the bonus material, if you will. We got to a point where the instructor said, “Okay, you now know more than everyone in the Valley.” We were like, “That’s a good stopping point.”

It was becoming very, very – just becoming very mathematical, and that’s necessarily a bad thing. But we are both very technical individuals, lots of development experience, things of that in our background, and just quite frankly like learning six layers of mathematics. Isn’t that appealing? That’s not like, “Cool. I guess it’s neat done or saying the math behind this one algorithm,” but quite frankly that wasn’t why we’re taking it. We were taking the class to really understand how we could use machine learning to solve a particular problem.

Out of that, we – just long story short, Jason ended up leaving American Electric Power. He went to Veracode. He was a principal .NET researcher for them. Veracode is a information security company that does – well, why don’t you tell over?

[0:06:09.0] JM: Yeah. We do binary static analysis. You submit binaries and we find flaws in the code and give you a report that says, “You need to fix code in these places or whatnot.” I was a researcher there for about three years.

[0:06:23.0] RS: Reason we bring that up is when he left AEP to go to Veracode, I shortly thereafter left AEP to go to Hewlett Packard, but Jason and I kept doing joint research projects together just mostly out of fun in our spare free time. That means that we lost access to the AEP data, which was that's fine and we were still in broad with this notion of what machine learning could do, and we started thinking about other use cases.

I forget exactly how it happened, but I asked Jason one day why he didn't enjoy online gaming. His response was basically, "There's a lot of cheaters in it. I want to know if I'm getting owned or dominated by somewhat a video game that is because they're actually better than me, not because they're using cheats." I thought, "Well, that's an interesting feedback."

I said, "I wonder if we could use machine learning to identify patterns in public data sets available via the Steam API that would identify a cheater." I think the whole notion back then was if you're cheating, you're trying to do something that a normal human wouldn't do, so it show up in the stats, right? Like if you see this ridiculously high headshot percentage, you're probably cheating unless you're a professional.

Or if the stat show like for the last two years you have an accuracy rating of 15%, then suddenly now your accuracy jumps at 50%, well that's suspicious, right? Steam makes it pretty easy to get all that data that we needed via their API. We spent about six months and we created a perfect concept that basically had a pretty good accuracy rating. It was like 88% accurate on identifying whether or not a player should be banned by Valve if I cheat.

[0:08:11.5] SC: If I can interrupt and ask a question, or a couple of questions. I'm assuming then that you are using supervised learning for this task. If that's the case, what did you use for labels? Did you manually label some number of users that you thought were cheaters? Then how did you validate this to determine your 88% accuracy rate? Did Steam also published what they thought from sort of cheating, or did you just look and see if they were eventually banned or something like that?

[0:08:42.4] RS: Yeah, that's a good question. What we did was we took the professional gaming league websites. We took data from there. Our theory behind using them to train not cheaters was that people tend to be watching the professional gamers more. There's more eyes on them. Cheating would be more obvious.

Then we went to the – there is a website called VACBanned. Its whole purpose was to take Steam IDs and categorize those that got banned based on certain dates. We were able to correlate those with the Steam API and pull down their game stats. We did some cleaning of the data. We looked at when they purchase the ESCO, if they had the game, because the data set wasn't perfect like all data has lots of crap in it. We spent a lot of time analyzing it and making sure we had a pretty confident set of cheaters and a pretty confident set of those we felt weren't cheating. That was the process we went.

[0:09:35.6] SC: Awesome. What did you do with those proof of concept?

[0:09:39.7] RS: Yeah. At this point in time, we released a research to the public at a security conference called DerbyCon in I think 2014. There were some Reddit post and it got a lot of traction. Actually Valve reached out to me about the research and was asking how we did it and whatnot. I told them, because of time that was like, "Oh. Fixed the issue that would be boarding it up for us."

That naturally led to the formation of Nexosis, because as we were going through the model building, we tried using different services like Microsoft's AML Studio. We tried using Amazon's API for machine learning. We tried using Google's predict API. At the time, Dato was still a company which became Turi, which then sold to Apple.

We tried all these different things on the market. The inclusion that we reached from all this was nothing is really out there for the developer. Everything is really catered and aimed at the data scientist. You still have to understand how to train and build a model, then you just get into this question of how do I make the model better?

Fundamentally you have to know the algorithm you want to use and you have to know why you would want to use algorithm A over algorithm B. Just lots of issues and from a local point of view IE, we want to just get something out there that's doing a good job a very, very painful experience.

We then set out with that knowledge. In fact, and we said, "Look. Why don't we look at machine under the lens of a developer? How would a developer want to consume machine learning?" This has been done in other industries for a while, right? The one that I can point to just off the top of my head is Twilio. You think about what Twilio did, kind of very similar when we looked at the data science, machine learnings landscape.

Part of Twilio, if you wanted to curate a communications application, you would have to really have someone on staff that understood the telco infrastructure, understood voice over IP protocols and how to configure it. We actually have a slide here internally that compares, here is what they look like before Twilio, here is what it looked like after.

I think the rule of today are similar on a data science point of view, right? You would have to know all the different – just how do you do VTL, how do you deal with missing variables. There's just so much to it. Then again, how do you know what algorithm you want to use and when, blah, blah, blah, right? It becomes very complicated.

The whole notion with Nexosis is saying, "Look, developers you use the same language that you're used to using. If you're a .NET developer, come to us and use .NET, C#. If you're a Java developer then go use Java. You don't have to learn a new language to incorporate machine learning to your project." That's really the whole basis of Nexosis.

[0:12:42.6] SC: Okay. When I think about what – let's take Azure ML for instance. When I think about what they are trying to do, it sounds exactly like what you're describing, right? They're trying to create an API that lets a developer deliver machine learning types of applications. They've got the studio. You can even do it drag and drop if you're so interested, if you're interested in doing so.

Maybe we can dig a little deeper and you can – when you're faced with skeptics that say, "How can you enable a developer to do machine learning without knowing anything about data science? How do you address that?"

[0:13:21.9] RS: It's a good question. When we looked at ML Studio for instance, a lot of the work as you know is data preparation, data cleaning, ETL. It's like 80% of the work. Then they're scaling. How do you do imputation strategies? How do you aggregate data? All these sort of questions.

Developers are very used to using – working with data and data types. You can typically take a data type identify and you know what step you need to do to impute or aggregate at that point.

Giving the developer the tools in their hands to define maybe the data types with metadata, but not really having to worry about what needs to happen before it can go into the algorithm, before we convert it all into numbers, whether we need to one hot encode categorical data, things like that.

How do we pick features? All of that stuff is – there's a lot of automation that can be done to simplify that. Now you still have to know your data, you have to know a good question you want to ask and you have to validate that you are submitting features that make sense. We're not going to know your data for you, but in a lot of ways we can automate some of that heavy-lifting at scale and then we can build lots and lots of models looking at different combinations of those things, and then finding what falls out of that.

[0:14:34.4] JM: Just to add to it, the other thing with all the different platforms is that you would still have to know how to go and immediately tune the models to make them better. If you're in Azure Studio and I don't know, you picked an algorithm which I think is the first hurdle, right? You're a developer, you're in Azure and now you have 15 different algorithms underneath your direction. Which one are you going to pick?

You're a developer. You're sitting there. You're seeing 15 algorithms. How do you know which one to use? How do you then go about making that model better via tuning, or maybe you have new features. With Nexosis, we do all that for you.

We have probably close to 150 different algorithms at this point that are in the platform. Basically, just high-level how it works is we hold these tournaments and the algorithm that's performing the best based on scoring metrics is the one that's used, so you as a developer don't even have to come to us and say, "Hey, I'm going to use decision trees." You don't even have to know that. You just have to say, "Hey, here is the problem that I have. Here is what I want to solve for. You guys build all the models and then you guys tell me the one that's doing the best."

[0:15:47.1] RS: Right. It's like a universal pipeline we use depending on the type of that problem you want to solve and it makes a lot of those decisions based on metadata that they provide as well.

[0:15:56.8] SC: Are there specific types of problems that this works for? It sounds a little bit too good to be true to be applicable to everything or anything, like I can give any kind of data and they could do any kind of algorithm.

[0:16:10.6] RS: Yeah. Right now we're not going super deep into what is traditionally viewed as the deep learning space. We think that's pretty well solved. If you want to go predict this is a hotdog or not, like there is funnier things after that. We'll do that for you already.

[0:16:28.7] SC: Like in Valley reference.

[0:16:31.4] RS: Right. That relates super focused on deep learning. We are starting to get incorporate more things around voice and speed, so NLP and not exactly sure how deep we'll end up going in that vein. But we're really more focused on the true machine learning and the layer that's above deep learning.

We're not going to probably release anytime soon a image recognition element into our API. Again, I think that's been solved and that's been solved pretty well. But we don't see a lot of stuff as underneath, like regression and classification, clustering, and again all the real machine learning types of element.

Today we launched the API doing time series, which I think is also a little bit unique. You don't see many platforms out there that have true time series capabilities in it, and time series is just naturally a very hard thing to create models in and you really apply machine learning through them. We launched with any kind of time series problem set today, if you have a time series like question, like demand forecasting, API, we could do that.

We just released regression. Any type of regression problem can be solved with the API. Later this quarter we'll be releasing the classification endpoint. Then we'll just going down that vein of machine learning, if that makes sense.

[0:17:53.8] SC: Now it sounds like when I think about the tournaments that you described as conceptually happening on the backend, it strikes me that for a given problem, let's say I've got a bunch of time series data and I am trying to do a predictive maintenance type of application. Is that the kind of thing that you might do?

[0:18:19.8] JM: Yeah.

[0:18:20.0] RS: Yeah, that's one of it. Most just because of where we were and we did the tech [inaudible 0:18:24.8] tel accelerator, a lot of the early use case with the API are more about, "Hey, I have this store. It's in Columbus, Ohio. I need to know of the 100,000 products that I have how many I need to have on the shelf for next week. All right, so when I place my reorder how many new products I need to have in my warehouse?" More that the main forecast and type of element, but yeah predictive maintenance and work as well.

[0:18:50.7] SC: Okay. The time series in that latter case is and the retail case is transaction history?

[0:19:00.8] RS: Right. Yup.

[0:19:02.1] SC: Okay. Either of these cases, like I'm thinking about the – you've got some set of models then that you are training the – that you're training against this data. So there's a fan out there, and then for each of these models you've also got a – do the hyper parameter tuning and all that, so there is a fan out there that strikes me that I guess I'm trying to put my hands on the scale aspects of the problem.

It seems like for any given individual problem you end up doing a ton of different training runs. I'm wondering if there's some way you can characterize or help me understand the way that that looks from an underlying resource perspective.

[0:19:51.8] RS: Yeah. We have a very dynamic workflow engine that it's very – it's queue-driven, so we can scale in and out of number of CPUs we want to use. It scales up and down automatically based on demand. We spent a lot of time building automation to handle that, so we're not – we have the cloud to work with, so we can do scale sets, we can do a lot of different things in parallel.

We could build a 100 or a 1,000 all simultaneously and compare different results and try different hyper parameters, try different feature combinations and things like that. Then once we get down to a solution that works well, that they're happy with, they can tune more parameters around that, or go with that model and use it to predict.

[0:20:37.4] SC: Okay. Is there any particular method that you're using to do the hyper parameter optimization?

[0:20:43.6] RS: I would have to ask our data science team to get in the details of that. We do not have PhDs and 10 years of experience in the industry. While we do have some understanding of machine learning, we thought it important to hire research team to solve some of the more complicated issues that we're not qualified to solve.

[0:21:02.3] JM: Just as we – he needs him and I, but we have be issues at Nexosis.

[0:21:07.3] RS: Yes, we do. Yeah.

[0:21:09.8] SC: Got it. Got it. When we talk about the time series, I guess it makes sense that you – that that was a initial place to start and that that was a little bit of the kind of data problem that you ran into it at the power company. Is that right?

[0:21:28.4] RS: Yup.

[0:21:28.9] SC: Can you talk it all to the unique – any of the unique challenges, or things that you do with regards to time series?

[0:21:37.2] RS: One thing that I don't think people think about is using the product skew example that we were discussing earlier. If you have a 100,000 skews, that means ideally you have a 100,000 models so every skew has its own model. The real power of what we're doing is that each individual skew could have a completely different type of algorithm winning.

If you're selling snow shovels as one skew item, a different type of fundamental algorithm might be winning based on location too. If you have a store in Columbus, Ohio then you have a store in, I don't know, like Atlanta, Georgia. Very, very different results, even though that's theoretically the exact same item, right? That's really the other power is that we're able to take into account, okay you're selling a snow shovel in Columbus, Ohio and it's probably going to be snowier here, so you're going to sell a lot more.

Again, just a different algorithm might be winning here in Columbus. You know it's the same exact item. Then just thinking that even further out when you think about things like bottled water as an example, an algorithm that's going to predict how much water you're going to sell might look a lot different than an algorithm that's going to predict how many white t-shirts you're going to sell. The feature importance and both of those might be dramatically different in [inaudible 0:22:56.1]. They are different, right? That's the other thing when you start thinking about scale, one model or one algorithm doesn't fit all.

[0:23:09.4] SC: You're able to – if as a developer, and this – maybe dig into this retail case. As a developer, how am I giving you my data? I'm assuming that's the starting place.

[0:23:21.7] RS: Yeah. What we do is we have a couple different ways you can import data through the API right now and it's through JSON or CSP. Then we have some S3 endpoints where you can put larger data sets, then we'll adjust them from there.

[0:23:35.2] SC: Okay. I'm giving you this data via the API, and how am I – am I doing anything then to describe this data, or are you figuring it all out somehow?

[0:23:50.9]RS: We try to have what we call same default. If you didn't give us any metadata and you uploaded a data set, we would do some basic analysis of that data set and try to create an appropriate data types for each one. Now is that going to be – is that going to work great? Probably not. But will it work well enough to get some results? That's our hope.

Then as people learn more about what they need to do with the data, we have metadata that you can use to describe this is a string, this is categorical data, or I need imputation strategy on this numeric field that does mean mode, or that sort of thing.

You can start to describe and we'll try to make some of those decisions for you. But it's better, certainly once the developer gets in and gets their hands around their own data and understand what they need to do with it.

[0:24:40.5] SC: Okay. You've mentioned imputation strategy a couple of times. Tell us what that means.

[0:24:44.7] RS: the idea there is if you want to look at aggregating data over time, the question is around do you add it up, do you sum it, or do you take an average? If you're looking at the weather heading up, temperatures doesn't make any sense. You want to average that over time.

It's that general idea there that you can indicate what type of data it is, then we'll take that step of how you want to handle that aggregation or imputation around that. Fields are missing. We might put in something else. Or if you want to roll up your daily or hourly for pass up to a monthly forecast, we're going to use different strategies to roll that data up, as well as filling empty values.

[0:25:24.3] SC: I gave you this time series of transactional data for this retail use case that has date time, skew, purchase price and maybe some other stuff. How do I then – how do I tell you what problem I'm trying to solve?

[0:25:44.4] JM: You're basically going to tell us the column that you're wanting to predict off of, right?

[0:25:49.6] RS: Yup. In that same metadata you just say, "These are features and this is a target." You can turn those on and off for each one as well. If you want to turn off a feature, you could then predict on a com. You could turn it back on. You could do one of scenarios that way too to sort of say, "What if we did this or that with those features too?"

[0:26:08.1] SC: Then are you able to do anything with around like artificial features? Features that are in in the data and the predicting home price example. You might want to look at the number of rooms, times and number of bathrooms and that particular – artificial feature might have some predictive value, that either those features by themselves doesn't have.

[0:26:31.5] RS: I mean, it depends. I would say in those cases, no. We're not going to just brute force to earlier columns and try and multiply. Then see if something happens or dividing by some – you know what I mean? That's a hard problem. I think in that sense, you need to know a little bit of your data and what the indicators are.

In other cases, we do some interesting things with maybe holiday calendars and we can automatically overlay with your time series data. We've done some work with launch locations, we incorporate whether automatically certain weather features that might help. It depends on what it is. We have some interesting ideas and plans in the future for that as well. I mean, at some point you have to understand what the indicators are that may help you predict, right? We're not a crystal ball.

[0:27:18.8] SC: I tell you what columns I want to predict on. You mentioned earlier that the platform wall, build individual models for each of the skews. That's something that I need to tell it to do, or is it always doing that?

[0:27:39.1] RS: It always does that with caveat. Once you have a model build, you can reuse it. The problem with time series data is of course is often yesterday is a better predictor of today or tomorrow than a week ago. There is the notion of how long my algorithm is good for.

On the time series, you end up rebuilding models a lot more frequently than you might with a regression model. That's just not as concerned about those sort of yesterday as the future, or two days ago, three days ago, last week.

[0:28:09.8] SC: Okay. Maybe tell me a little bit about some of the technical challenges that you had to overcome to put all these together.

[0:28:19.3] RS: Yeah. That's a big question. Where to start?

[0:28:22.1] JM: That could be a second podcast.

[0:28:25.8] RS: Yeah. I think really when you try to build a generalized platform, there's so many challenges around handling all sorts of data. We can't boil the oceans. We have to make a lot of tradeoffs and choices around what is the MVP going to look like? What is the next version going to look like? What can wait? What do we have to have now? How do we get something to market?

From the product side it's been a big challenge to make those tradeoffs, then get the work done in a way that we feel good about the results. Yeah, I think building that ETL pipeline in a general sense, sort of that pipeline around do you scale the data before you run the algorithms, when do you have to do that. Building and sort of all that core capability around how do we get any sort of data into a common matrix to do ML on? There's lots of different pass to get there.

I think one of the big challenges was trying to define the best use cases for us to get the broadest hit. Now, we have to tradeoff some accuracy for that and that's okay. As we go on, I think we feel like we can get better and go deeper in a lot of areas.

[0:29:35.9] SC: Yeah. That was a question that I had earlier and I didn't ask it. Often when I've talked to folks working on generalized machine learning platforms, the idea tends to be – you're trying to get the developer, their organization from zero to 80%. Then maybe once they're at 80%, have some maturity, they may find that they need to invest further to get to 90X percent. Is that the general way you think about the problem as well?

[0:30:14.0] RS: Yeah. Up to a certain degree. I just think where we are today, getting people familiar with what machine learning can do is the first hurdle. Then as you were saying, how do we get better performance after we actually have something in the environment, or some kind of application deployed that's using machine learning. I think there's this natural evolution, and that's one of the things that we've built into our own platform is the [inaudible 0:30:39.8] like a regression problem.

You were talking about how is prices, I think earlier. Let's say your initial data set has 15 features and is like room size and how many bedrooms and whatever, right? You build your model and it's pretty good. But then as time goes on, you might think, "Well, hey you know what? Maybe this new thing that I just thought about is going to have a big impact. Maybe that new thing and this scenario is the school rating in your areas and how you're going to bring in the whole new feature of school ratings for the house that you're looking at."

Then with us it's just naturally going to build a new model. So you don't only have to think about, "All right, well now I have this new feature. Do I need a new algorithm or not?" If you do need a new algorithm, our platform is just going to figure that out and it's going to say, "All right, well maybe you're using like classical regression or something before and now you're going to use, I don't know, whatever else, because you added all the same features.

I think that's the idea of the big power of this. As people start looking at the accuracy and the results, I think the natural question is always going to be, "Well, how can we do better?" We are trying to spend a lot of time and energy on that educational component, which is answering that question of, "Okay, we have this today. How do we get up better?" Is it going to get better if we add in school ratings? Maybe, probably. But let's figure out what happens and we actually do it. Then they'll figure out, "Okay, well this had no impact or had a big impact."

Then that should lead to the next question of, “All right. Well what else could we add maybe to make it even better?” We could just keep going down that whole trend and naturally just our platform is going to figure out, “Okay, as you mature and add new features, we’re going to pick maybe a new algorithm and it’s going to perform even better.”

[0:32:27.2] SC: Yeah. It’s funny. A lot of those things that those activities that you’re describing on a spectrum of improving your algorithms or things that I think of data science, like if you separate out the knowledge of the underlying math from the process and the way of thinking about data and features that have some predictive value and using those to create predictions, that to me – a lot of that is what a data science is really about.

I’m wondering if you – you know is that that you end up teaching developers those parts of data science in order to get them productive on this platform, or are you finding that there are maybe folks with different roles that understand that stuff, but don’t understand the math. How do you see the audience and for what you’re doing and is it evolving at all?

[0:33:29.5] RS: I think the audience is absolutely evolving. I think as we look at the future, we’re releasing some additional features that should really marry this notion of developers teaming up with data scientist. We want to enable more collaboration in that space.

We think one of the main things as we look at more mature organizations is shorting the time from R&D to production. If you’re a data scientist and now you’re collaborating real time this maybe beta application that a developer has made, that really speeds up things up. Then as the data scientist is looking at what the developer maybe made as an initial proof of concept, yes the data scientist and might really be honed in on, “Okay, how do we make this better?” That’s on approach.

Then I think again, we just have so many – from an innovation perspective enabling developers equip prototype things has a tremendous value. Then just being able to show if you’re a developer and maybe you have a data science friend or someone at your company is a data scientist, being able to say, “Hey, look. I made this prototype qualification and it’s doing X. What do you think? What else maybe could I do to it?” Just bolstering that collaboration is obviously really important to us as we look at structure 28 key and you just see more and more things start to be released.

[0:34:49.5] JM: Just to add on to that, there is really not enough of data scientists to go around unfortunately. Also enabling I think people to get some capability up is better than just having to go, I guess go hungry, so to speak.

[0:35:05.6] SC: Yeah. Are you doing anything where it's industry, or vertical focus where you're able to – you've got data scientist that are thinking about the problems in those verticals and what the data needs to look like and what the features are, so that you can guide or offer special features for a specific vertical, or is that not a focus for you right now?

[0:35:28.3] JM: Yes. Shortly, we should be releasing this notion of data templates. The idea behind that is at a very minimum, here are the features that you would need in order to be successful with this type of problem. Skewed level forecasting as an example, that the bare minimum features would be like time, so ideally you want a year of data. It can work with lesson app, but best performances a year.

Then you're going to want probably daily sales activity. Then beyond that – like those are just the bare minimum. You can create a forecast off just of those two things. Because the platform automatically extracts data out based on the timestamp if it's Monday, Tuesday, Wednesday and will find it weekly and daily trends, things of that nature.

Then you could start adding into that template additional things that maybe aren't necessarily required, but they're nice to have. A nice to have would be, "Do you have a promotion going on for that particular item?" That could a binary 1 or 0. You could extrapolate what kind of promotion it was, or it is like a buy one, get one, is it just a percent off, things of that nature. But yeah, we have plans to put there here is a bare minimum that you need and here is the nice to haves and go from there.

[0:36:46.7] SC: Interesting. In that example where you've got a developer, they've gone out and collected some sales and marketing, historical data and they're doing a forecast, how do you articulate to this developer that may not be statistically sophisticated, the extent to which they should rely on this result that your platform has put up for them.

[0:37:11.2] RS: Yeah. Currently we put out some metrics on that. What we've determined is we need to get a little more friendlier on those metrics. We have some education around what the metrics mean and some of our learning on our documentation site. But we want to really go to the next level, I feel like and really hold their hand a little more on what the metric is saying about the model base on what the data they have.

[0:37:35.3] JM: Currently, we're returning MAPE scores, that is a time series at forecast. Yeah, we plan to make it even more friendly than that. Because drop, we might not understand what made this.

[0:37:44.6] SC: What's that score?

[0:37:45.9] JM: MAPE, Mean Absolute Percentage Error.

[0:37:49.7]SC: It sounds like this is – this interface is between what you might expect the data scientist to know and what the developer might not know, or some of the key challenges that you face as you evolve this and bring more people onto the platform.

[0:38:07.7] RS: Honestly, I think that's a – just quite frankly, I think that's a minor challenge. I think the biggest challenge that we have and I think just the industry has in general is access to data. If I want to build – honestly, I was thinking about the server the weekend, I want to build a couple different applications. The hurdler is getting the data sets, right?

If I wanted to build an application off of predicting cancer, there is cancer data sets already out there that's very famous. The breast cancer data set that has the size of tumor and things of that nature. But maybe a big important feature there is where the people live. If you live next to, I don't know, a waste site, that's a little bit extreme, but let's say that you live there. Maybe that's why you have cancer, right?

Maybe where people live might actually be a huge indicator whether or not you're going to develop cancer, or that lump that you discovered is benign or malignant. But it's very hard to get that data set, right? Number one, you have the HIPAA issue. There is that little hurdle to get over. But then more than that, it's just our people are going to want to share the data.

The other idea I had was predicting, more than at a startup would be successful. One of the things that I think you would need to do that would be the financial information on a startup, or how much money they're spending where in the return. But again, trying to get reliable data that's going to help me build that model is what I think right now is the biggest obstacle in the field.

[0:39:42.0] SC: Any other thoughts on that particular point? Any other obstacles that you see?

[0:39:46.8] RS: Yeah. I mean, there is plenty of them. I think really walking them through what their data might be telling them is going to be helpful. One of the big challenges is you submit a data set, and I think there needs to be more of an indicator of you really can't do anything with this potentially in some situations.

Others, it's like, it's okay. Then this is really good data. I think defining and helping them along without just this raw metrics, like give them a little more I guess safety feeling about their models is still a challenge, and then we have small ideas around that.

[0:40:22.8] JM: Yeah. I think just also to add to that, one of the challenges is how machine learning, AI, whatever you want to call it today has been talked about in the media. It's put into the minds of a lot of people that machine learning is this magic bullet and they give kind of a real example. We have some people – the data that's sign off with the API that want to use it for things like Bitcoin price.

The data set is just the price point on a day. They think that machine learning is just going to magically figure out what the price is going to be. Sometimes we have to talk to the people who sign up on that use case, or and like, "Well, if that's all the features you have, do you really think that Tuesday is the reason why Bitcoin jumped up? Or do you think there is this other feature out there that is really impacting and influencing it, right?"

Again, it's just people really have to I think get beyond that machine learning isn't a magical bullet. It's only going to find the patterns and the correlation in your data set. If it's not there in the data, it's never going to find the pattern.

[0:41:25.5] SC: Yeah. That's awesome. Anything else that you'd like to share with the audience as we come to a close?

[0:41:33.2] RS: This is the other – only thing that I was thinking about is that we're talking about our overall goal. I think really most of the 28 key is going to be this thing of how do we enhance collaboration between developers and data scientist.

One of the newer features that we'll be releasing in 2018 is something called 'Bring Your Own Algorithm.' What this is going to do is it will allow a data scientist to create maybe a very specific algorithm that is really good at solving for a particular thing. Let me give example.

Let's say that you are a data scientist at Best Buy, or some big box retailer. Let's say that retailer really cares about the price, or the sales, or I don't know, LADTDs. Well, the notion is that they might create a very specific algorithm that's really good at figuring out, "Hey, how many flat screen LED, 55 inch TVs that we're going to sell?"

All that they would have to do is take that hour, and if they build in-house they can just plop it into our API and their algorithm will live just in their instance. It will live by itself. Well, it will live just in their instance by itself. But it will live side by side our 150 plus algorithms. They won't be able to see real-time, "Hey, here is what our algorithm is wanting and here is where it's not wanting."

[0:42:53.6] SC: I guess, a related question that I had from earlier, as you were starting to get into collaboration and data scientists working with developers and developers exploring their data and creating new features and things like that, it opens up this whole set of issues around model management and model governance and model providence and stuff like that. Is the company thinking about any of those things? Or do you offer support for those kinds of challenges?

[0:43:24.9] RS: Yeah. That certainly come up in the past and we've had it come up primarily in the insurance industry. The regulations around the model building, you really have to understand how the decision is coming about, because you can't discriminate and things of that nature.

Yeah, for the more enterprise customers will allow them to actually download that model and algorithm and have that supporting documentation they need to have to prove that we're not using, for lack of better word, illegal types of data sets to create the prediction result.

I think as time goes on, we're going to see more and more of that. Again, right now one way that we can solve it is just by giving a dockerized container that contains the algorithm and explain how it works and things of that nature. Yeah, I mean as time goes on I think we're going to see more and more of that.

[0:44:18.6] SC: Okay. All right, great. Then did you have an offer for free access to the platform, or free API key or something like that available to listeners?

[0:44:31.1] RS: Yeah. Anyone can sign up for the API for free. That's part of our philosophy as a company and being developer-first is that we want people to use it, so we do have a community edition. It's a 100% for free. It will always be free, that community tier. Yeah, anyone can go to nexosis.com, sign up for an API key and get started in under five minutes.

[0:44:54.1] SC: Okay. Awesome. Well, guys thank you so much for taking the time to chat with us. I think what you're doing is really interesting and I will certainly be looking forward to keeping up with you and I'd love for you to keep in touch and let me know about, you know as the platform evolves, your continued success, etc.

[0:45:15.1] RS: Definitely. It was a pleasure being on here. Thanks so much for having us.

[0:45:19.3] SC: Before we go, why don't you take a second and tell me a little bit about what's the traction been to date? How many users do you have, or how many companies are using it?

[0:45:30.5] RS: Yeah. Currently, we released the API to the public on July 11, 2017. To date, we have over 4,000 developers that have signed up and we have hosted 500 applications that have been deployed.

[0:45:44.5] SC: Wow. All right, well once again, thank you so much for taking the time to chat with me. I appreciate it.

[0:45:51.2] RS: Yeah, likewise. Pleasure. It was fun.

[0:45:53.1] JM: Yeah. Thank you.

[0:45:54.1] SC: Awesome. Thanks, guys.

[END OF INTERVIEW]

[0:46:00.3] SC: All right everyone, that's our show for today. Thanks so much for listening and for your continued feedback and support. For more information on Ryan, Jason, or any of the topics covered in this episode, head on over to twimlai.com/talk/69. This interview kicks off our Strange Loop 2017 series. To follow along with the series, visit twimlai.com/stloop.

Of course, you can send along your feedback and question via Twitter to @twimlai, or @samcharrington, or leave a comment right on the show notes page.

Thanks once again to Nexosis for their sponsorship of the show and this series. For more info on them and to get your free API key, visit nexosis.com/twiml.

Of course, thank you once again for listening and catch you next time.

[END]