EPISODE 60

[INTRODUCTION]

[0:00:10.5] 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.

The podcast you're about to hear is the first of a series of shows recorded at the Georgian Partners Portfolio Conference last week in Toronto. My guest for this show is Solmaz Shahalizadeh, Director of Merchant Services Algorithms at Shopify.

Solmaz gave a great talk at the GPPC focused on her team's experiences applying machine learning to fight fraud and improve merchant satisfaction. Solmaz and I dig into step-by-step the process they used to transition from a legacy rules-based fraud detection system to a more scalable flexible one based on machine learning models.

We discussed the importance of well-defined project scope, tips and traps when selecting features to train your models and the various models, transformations and pipelines the Shopify team selected as well as how they used PMML to make their Python models available to their Ruby on Rails web applications.

Georgian Partners is a venture capital firm whose investment thesis is that certain tech trends change every aspect of a software business overtime, including business goals, product plans, people and skills, technology platforms, pricing and packaging. Georgian invest in those companies best position to take advantage of these trends and then works closely with those companies to develop and execute the strategies necessary to make it happen.

Applied AI is one of the trends they're investing in as are conversational business and security first. Georgian sponsored this series and we thank them for their support. To learn more about Georgian, visit twimlai.com/georgian, where you'll also be able to download whitepapers no their principles of applied AI and conversational business.

Before we jump in, if you're in New York City on October 30th and 31st, we hope you'll join us at the NYU Future Labs AI Summit and Happy Hour. As you may remember, we attended the Inaugural Summit back in April. The fall event features more great speakers including Corinna Cortes, head of research at Google New York; David Venturelli, science operations manager at NASA Ames Quantum AI Lab; and Dennis Mortensen, CEO and founder of startupx.ai. For the event homepage visit aisummit2017.futurelabs.nyc, and for 25% off tickets use the code TWiML25. For details on the Happy Hour, visit our events page at twimlai.com/events.

Now, on to the show.

[INTERVIEW]

[0:03:02.2] SC: All right everyone. I am here at the Georgian Partners Conference and I have the pleasure of being with Solmaz Shahalizadeh who is director of the data team at Shopify. Solmaz, welcome to this week in machine learning and AI.

[0:03:16.9] SS: Thank you very much for having me, Sam.

[0:03:18.8] SC: It's great to have you, and I especially love when I get an opportunity to interview folks who have listened to the podcast before, so thank you so much for listening.

[0:03:27.7] SS: You're more than welcome. Yeah, I've subscribed I think like nine months ago or something like that and I've been a fan.

[0:03:31.1] SC: Awesome. Thank you so much. Why don't we get started by having you tell us a little bit about your background and how you got into data science and machine learning?

[0:03:40.6] SS: Sure. In my undergrad I studied computer science and towards the end of my undergrad one thing I realized is that I'm really interested and passionate about using computer science and a little bit of machine learning that I knew at that stage to solve problems in other domains. I figured that out. I did a master's in my informatics, and I went to Sweden and I studied there and as part of my thesis project I actually worked with Sloan Kettering Cancer

Center in New York trying to predict what happens in a cell after you give it sort of a multiple perturbations or different drug cocktails and how the structure of the cell might change.

At the time we used actually recurrent neural networks, which were not hot then, because that was 2006 and everyone was giving us grief about like, “Oh! This was 80s. This was cool in 80s. Why are you using it now?” It lent itself very well to our problem, so we used it there. After that, I went to McGill in Montreal and I did another master’s. That was like more focused on machine learning and computer science, and as part of my thesis project we’re I worked very closely with the department of oncology and molecular biology and we used the microarray data to —

[0:04:41.8] SC: Microarray?

[0:04:42.4] SS: Yeah. A very simplistic way to explain it is like you have a chip and on it you have these probes that you can measure the expression of different genes.

[0:04:51.1] SC: Like GDPCR?

[0:04:52.4] SS: Sort of yeah, but scale that to thousands of genes on it.

[0:04:55.9] SC: Okay.

[0:04:56.5] SS: We had on the data on that. When we’re trying to use that data, which we had captured at the moment before people have got any treatment to see if there’s any signal in that data that we can say if they’re likely to have a recurrence of the breast cancer. The usage of that is that by knowing it you can sort of like not give as much of a hard treatment to people who don’t need it and also plan for the recurrence for people that are likely to get it.

That went really well. One of the key learnings I had there was that if you are building a computational solution and you want people in other domains to use it, the first thing you have to learn is to actually speak their language and understand how they explain their problems. I remember as a rookie grad school, grad student, I had this presentation, all of my P-values and all the like statistical factors are why my predictory was awesome is on that presentation. I start and you could see after the second slide, the light in like the eyes of the molecular biologist is gone. They are not listening. There is nothing keeping them going.

Then through that I learned how to understand what's the actual real challenge for my collaborators and how to explain my solutions in relation to that. Then towards the end of my degree was like really engaging. I felt like I understand another domain on top of machine learning, and I think that was like a learning that I kept with myself. For a while I worked at Morgan Stanley, an investment bank, and then I joined Shopify four years ago.

Shopify is a leading cloud-based commerce platform. It allows you to sell on multiple channels from brick and mortar stores to huge, huge sales and enterprises, like Tesla motors or Budweiser, all of these big brands, but also very small merchants.

I joined Shopify. We were in the process of like changing our data warehouse and prepping for IPOs. It was really important to make sure we really understand the data we're capturing. We also have like clear definitions for the metrics that we were going to share and all of that. I was part of the team that worked on that.

Then after IPO we realized, well, we have all of these machine expertise in-house and we have also all of these data about different aspects of commerce. The beauty of commerce is that it's like messy. It's real-world. There's merchant trying to fulfill orders in their basement of their home and there are people having like thousands of orders coming to a second and they have to deal with that. That brings with itself so many opportunities to take repetitive tasks out of the daily life of a merchant and an entrepreneur and give them back either the gift of time or the give of money that they were spending on something else. That has been fascinating part of my job at Shopify.

Last two years I've focused mostly on machine learning teams, data science teams that build products that are powered data. One of this is our order fraud detection, that runs on real time on every single order. The other one is cash advance product where we basically give cash to our merchants, because we think that's the amount and that's the right time to give them the capital to help them grow their business and they return that money.

Now, we’re trying to like bring sort of this like basic level smartness to other products, such as our shipping and fulfillment and things like that. That’s where I am.

[0:07:56.7] SC: Nice. You spoke at length about the order fraud problem and your approach and solution to that earlier today at the conference. Why don’t you tell us a little bit about the problem there, the context that you’re trying to apply machine learning to?

[0:08:11.9] SS: Sure. As I was talking today, what happens is like all of our merchants, bit or small, have one thing in common. They are there to make sales to succeed. What happens is that they see an order come through their store. They go ahead and fulfill it. They look at the order. Nothing looks suspicious. They fulfill it. Six months goes by and then they receive a chargeback fine from the credit card company. They’re out of the item, because they’ve already fulfilled it and sent it to the customer. They are out of the money for the sale, because their credit card company refunds the amount and they also receive a chargeback fee. That really cuts into their cash flow. You don’t have to have too many chargebacks to feel the impact.

The other side of it is the emotional factor. We are saying like, “Okay. You know, you focus on building the best product that you can, putting it in front of the right audience.” Now, we somehow have to tell them to be okay that somebody across the universe from you is using stolen credit card information to buy from you. That, on an emotional level, is unsettling.

The other thing and the other reason we pick it as one of the first areas to go with machine learning is the fact that it’s really back office work. Becoming an expert in fraud detection is not going to make you a better product designer. It’s just not a core skillset of our merchants.

[0:09:23.9] SC: Something that the customer should have to think about.

[0:09:25.9] SS: Exactly. That’s where we thought, “Okay. We have the data. We have the knowledge and we can scale this solution,” so that not only the big merchants can benefit from it, but the merchants who start on platform and on their very first order they can get this analysis that backed by a decade worth of data, and that’s why we picked this problem as the first one to tackle with machine learning.

[0:09:46.8] SC: Okay. Was the problem previously being managed manually where there are some group of analysts that were doing this, or was there not a prior solution in place for order fraud detection?

[0:09:59.1] SS: We had actually a prior solution and we have a group of risk analysts in-house, so it's like a combination of the two. The prior solution was built like five years ago, and I think it was good for the time it was built, but it had very hard coded rules. It would things like if the order was placed using a web proxy, then probably it's fraud.

Right now we see many people use web proxies, even like you're here, you want to order something from U.S., you probably use a web proxy. Without going to the details of the rules that was there, the problem was like the rules were not learning on —

[0:10:30.2] SC: The static part.

[0:10:30.8] SS: Yeah, exactly. They were static. I think it served our product for the time it was working, but also we have had this crazy hockey stick growth. As we have had like these many more merchants and more visibility to sales, it became apparent that we can make a difference by using our own data to solve the problem.

[0:10:50.0] SC: What were the steps in kind of getting to a solution to this. What was the first thing that you had to figure out?

[0:10:56.6] SS: The first thing that you have to figure out, and it's common across any machine learning problem, is actually try to define what you're trying to solve. It sounds very basic, right?

[0:11:07.3] SC: We're trying to fight fraud.

[0:11:08.8] SS: Yeah, exactly. Then define like, "Okay. I want the cash fraudulent transactions," but they actually want to do it before the merchant has gone ahead and fulfilled the order. That brings some practicality requirements to the solution that we offer. Any machine learning algorithm and system I build, it has to be able to run on every single order as they go through without slowing down the platform, without having downstream processes have to wait for it. That with itself brought some hard requirements on the kind of solutions we can do.

The next step was, of course — Okay. We have a classification problem, so we want to classify this transaction to fraud and non-fraud, so let's see if we can actually clearly define what's a fraudulent order. I did some digging there to make sure our definition is correct, also can capture if there are anomalies or there are changes. For example, if we're relying on dispute codes from the payment gateway or a bank and that happens to change, like let's make sure we have automatic detection in place or some ways so we learn, "Okay. You know what? The fact that I haven't received any more fraud is not because fraud has gone down or has gone up. It's just that that code that I use to capture fraud has changed."

That making sure that basically the targets of your prediction are correct, and then we got to like investigating inputs. That's kind of an interesting story, because I talked about like the massive pools of data we have. It's like over the last year we saw — Just last year, hundred million customers place orders on Shopify stores.

For these customers, we know the path they took to go their product. How much time they spent? What's their color preference? Have all of these information. At the same time when you tackle a prediction problem, you also have to know, "Okay. Of all of these features that I have, which of them I'm actually going to have available at the time that they are making the decision?"

For example, if I'm going to use the number of orders this customer has placed in the past, how am I going to have that aggregate count of orders available at the time of production? That brings, again, another layer of practicality to features that you can put into your model. We went through that. Right now we can actually — We have built like internal services where we can get aggregate data, we can get real-time data aggregations. That, of course, gives a boost to the models.

[0:13:25.9] SC: Part of that is the architectural challenge of making sure that data is available, but then in your talk you also mentioned at one point you were building your model around some feature that actually wouldn't be available to the model for months later.

[0:13:42.1] SS: Yeah, that's true. Actually, that's one of the main challenges in fraud is that people can decide to place a chargeback up to a year. Credit card companies allow you to take a year and say, "Okay. I believe this was a fraudulent charge of my card."

They're looking at data and realize, "Okay. Most of the fraud actually comes back within the six months. We define our target as like, "Okay. Has this transaction resulted in chargeback in six months or not?" It also means for any transaction we actually know the full ground. We have to wait for the full ground truth for up to six months. That's a challenge. We have to make sure, "Okay. Are there any leading indicators that we can use?" Because if I add a new feature to a model and I want to see how that's going to work for new orders, I have to actually wait six months to see if the prediction is correct or not.

Some of it we deal with it by using historical data if we have the feature available in past. If not, it depends if the feature is like really out of nowhere, we don't know, we have to wait. If we can get a degree of confidence with leading indicators and we go with that, we'd say, "Okay. Has the ratio fraud we've seen within the two weeks gone up or down?" We try to do that to have like faster iterations.

[0:14:54.6] SC: Okay. Even in terms of the definition of fraud, I'm just thinking there's stolen credit cards, there's people that I was worried about this as like an eBay seller, like you sell something and you ship it to someone, they say they never got it, but they got it. That's another like — Did a lot have to go into actually defining the types of fraud? Did you model for every kind of fraud or just the specific subset of the possible fraudulent universe?

[0:15:26.2] SS: For the first step we actually decided to select a subset of fraudulent universe, as you say, because the features and the characteristics that we were studying and we're trying to understand were more around the financial fraud, so people using stolen credit card.

We are looking at adding on our models and capabilities to pick things like item not as described and also with better shipping information integration we can also see if the item was delivered or not. Before the first version, it was really important for us to have a very clean cut definition. We went with the charges that we thought were fraudulent due to financial reasons. This included things that the merchant had gone in the admin and cancelled due to fraud either because of a call they got from the credit card company or the bank, and we called those ones as fraud as well. We were very sure.

Of course, we have this internal platform built on top of Spark and Pi Spark and we made our definitions into jobs, but you need to run on schedule. As a byproduct of this, regardless of which part of the team you work on and what day of the week you queried you database, you're always going to get the same orders as being fraud and same orders as being non-fraud. That helps a lot in being able to sort of validate results and models as we go ahead.

[0:16:39.0] SC: Okay. You have your definitions set, like what's next.

[0:16:42.8] SS: Okay. We have a definition. Now, we basically have labeled data. What are the inputs? What are the features that we're going to put into models? We started with very basic things, like things around payment gateways and credit cards and those were like the easiest things to think about. We had for also for that do some checks, like make sure — We had also another features where we started looking at it, it look predictive, but then we realized, "Yeah, but this feature has not been fired for the last six months, so actually I can't use it, because for whatever reason we're not producing it anymore."

[0:17:11.5] SC: Meaning it just got pulled out of the platform somewhere underneath you?

[0:17:14.1] SS: Yeah.

[0:17:14.8] SC: Okay.

[0:17:15.1] SS: That's the thing, like there are lots of data. There are massive amounts of data, but you also have to understand who's producing it, what's the expectation level on availability. Sometimes, like for example, orders that come from like a point of sale are not going to have all the features of the orders on the web. The features you pick, if they're going to run across different gateways and across different channels, have to have a representative value for all of these scenarios, otherwise the model is going to be biased towards one versus the other.

We realized the feature frequency mattered a lot. How often are we going to see this feature? Then the distribution of values and how often it's going to be null or not present. Once we figured out this list, one of the things we are always focused on in Shopify is like, "How can we scale this?" I don't want every single data scientist in the team to every day have to code how to figure these things and have this checklist in their mind, because I think there are like way more interesting things that they can do. We made templates. We made Python template codes that they can run a new feature true and it would produce this sort of descriptive statistic. None of it is super complicated, but it's just a fact that we have thought through this step gives us a boost in the speed for our delivery.

Now, if you're part of the team and you want to add a new feature, before you even put it in a model, you can run it through this sort of scripts or a Python notebook or job and you would get a report that says, "Okay. Over the last 12 months, this is when this feature has been null. This is when it has been missing. These are the distribution of values you see for it as how it looks across different segmentation." That's something that's very simple, but in action it helps a lot.

[0:18:56.2] SC: Those tools, are they primarily used for kind of exploratory analysis or do you have like a whole framework for back testing and things like that?

[0:19:08.7] SS: This specific one we use for exploration, because we're trying to add features into new models and see how they work. We do exploration by pulling these features that have passed the checks and the target and then trying in different learning models. We started with the simplest ones that are really easy to explain and easy to sort of debug as well, because we knew we had to scale it for 500,000 merchants.

For the first one we said, "Okay. We're going to go with a random forest." Then we have a pipeline where you define your features, you define the transforms you want to apply on those features and the models.

For example, I want to say — As an input data, I get where the order is placed, but I transform it to a feature that says, "Is the order placed on a tablet or not?" It's like a level of change you do on top of a feature, but it's really important that that transformation is well-defined and also well-understood by the downstream parts of the pipeline.

After the transformation we also have the model training and then we do our testing. Then we optimize for different metrics and we really tie those metrics based on what's accepted in the fraud detection industry rather than like optimizing for true positives or true negatives. You really want the balance. You want the merchants to be able to take as much sale as they can with peace of mind. That's the optimization.

[0:20:28.0] SC: Elaborate a little bit on this difference between the kind of this generally accepted metrics versus the ones you might otherwise track. Is the idea that you can't report AUC to a merchant, because that doesn't mean anything to them or are there industry-specific terminologies or metrics that you need to kind of map things to?

[0:20:55.3] SS: Sure. There are industry metrics within frauds. For example, like in varies a little bit, but above 95% of your order should be accepted. There're 95% of your transactions we actually don't have to worried about. They're not going to be fraud. Within the next 5%, how much of it do you ask a person to investigate, call the customer, try to verify a little bit of extra steps versus the ones that say, "No. This is fraud and you have to go and cancel it."

We have built in our pipeline, we can actually pass it the product metrics and say, "Okay. This is bucket size of each recommendation that I want you to have and optimize for metrics within this bucket sizes." That's one way to go, because then we actually know what the practical impact of this metric is going to be on the merchant. We also look at the losses that they would have. The value of accepting this order and receiving a chargeback versus like the loss of not having — Not actually letting the order go through, and we try to optimize for that.

Yeah, I would say like if you want to remember one thing, it's like the metric that you use for tuning your model has to be something with very understandable user impact, because these models, these products go in the wild and power real live people. We have to have an understanding of what happens if I actually push this too much one way or the other and what's the impact is going to be on the user.

[0:22:10.9] SC: You've got your model now trained up. What's next?

[0:22:15.2] SS: What's next is what we call production back test. In that, what we do is that we say, "Okay. For the most recent six months of the data," which is I said we might not always have all of the prediction results ready, we're going to still run the model and do the prediction and look at the distribution of the predictions across the six months and across different merchants. We're going to say, "Okay. We trained this model to say put X% of the orders in cancel bucket." Is it doing the same in the most recent orders? Has something changed in the patterns that's not allowing for that?" Then we even go find our grain and we say, "Let's look at the model predictions within the segments of our user-base. Let's look at people in a specific geography or in a specific channel to see if there are any problems there." Then we go one step deeper.

If we have merchants whose individual recommendations have changed significantly. That means like that person is going to have a very different experience when they log in that day. If that's the case, we reach out to them ahead of time and would let them know that this change is happening and this is why this model is better and this is why your experience is likely to change. They're actually really positive because they know we have put our focus on making their business better.

[0:23:23.3] SC: Okay. What are the things that trigger that last scenario? Is that when for whatever reason that merchant or that category becomes a target of fraud or is it something more like statistical drift in a distribution or is it stuff that you've done in terms of just tweaking models?

[0:23:45.1] SS: Most of the ones we've seen so far is by additional features that are totally different. For example, if I started looking at historical data on the merchants and I see like, "Okay. This merchant is more likely to have fraud, or less likely to have fraud." That is what's changing. Most of the time, introducing not a single new feature, but a class of new features, like historical values or when we started looking at the features about the browsing behavior, then that brings an extra level of detail.

That being said, even when I say the change is drastic, it needs some checks, but it's just like for one person, maybe one day they get like 10 more orders than they are used to to investigate, or 10 less orders to investigate. You'd be surprised how much people build their own workflows and their own understanding around them. You want them to feel safe and protected, because that's the whole goal of this product, is for the merchant to feel save and protected. That's the goal. Yeah.

[0:24:38.5] SC: Okay. You've done your back testing. Are we there yet?

[0:24:44.2] SS: No. We got all of these things, but then we've done all of it in local data LAN. It's like everything in data — Everything in data in our company is done with Python, Pi Spark, but Shopify is a very big Ruby on Rails application, and even the services we've built inside it, they're all Ruby on Rails. We have to find a way to transform this model that we have made in Python to run in a Ruby application just a risk application.

For that, there are many different ways for us, because we know there are many other applications in Shopify that also use Ruby on Rails. We wanted to find a way to see if we can actually run our machine learning models in Ruby applications. What we did is we went with this model serialization definition, which is called PMML. It's predictive modeling markup language. It's been around for decades. It was mostly used in academia, but now it's having a comeback and different languages and packages have PMML transformation. So in [inaudible 0:25:38.0], you can save your model in PMML or you can save your model in PMML, and it's very similar to XML. It has a very well-defined spec, so you define your inputs, you define your transformations, the model, and then decision making at the end.

It kind of ties into our pipeline, and we got ideas from that for the levels of abstraction that we've put in our pipeline. What we do is like once we get the model, we serialize it to PMML, and then we have built a gem that we are planning to open source next year. Basically, it's a Ruby interpreter for PMML. Once that gem is included in the Ruby application, it's able to sort of like understand this PMML model and evaluate orders using it as they come through.

Then we have the model —

[0:26:19.9] SC: Now, before we go past that, I've been waiting for us to get to this part, because I first came across PMML probably like four or five years ago or so and couldn't find anyone who is really doing anything with it, and there's another one — I forgot the name of it, but it's another kind of model serialization. I'm sure there are a bunch of model serialization things.

You serialize this model and you have your inputs and your outputs. What's the level of abstraction of the serialized model? Are you telling it a model type, like this is a random forest, and then it's up to the implementation that is interpreting it to actually know how to implement the random forest or is there some other mechanism for actually implementing the model?

[0:27:07.2] SS: PMML spec, for example, has something that says, "Okay. This is a logistic regression. These are the inputs I need for a logistic regression and this is how I'm going to give you the output." The actual implementation of the code that does it, it can be done in any language.

What it is literally just a transform mechanism for one language to the other one. It's an XML document, you can open then you can see then it has like the inputs and the model definition and the sort of weights and things that go into the model and how it's going to make the decision at the end. It doesn't do anything. It doesn't have any execution engine tied to it.

[0:27:42.7] SC: Right. I guess my question is do you run into — Do you ever run into issues where, I guess, if you're deployment environment is all Ruby, it would have to be something like you've got some weird dependency thing where you've got one version of one rails app that's pinned to some Ruby machine learning library version and you have another that's been to another version, and because you're serializing this model at like a level of abstraction higher, you get different results based on where you run it?

[0:28:19.3] SS: Right now we have one gem that we use across Shopify, and right now we are using it in a single application. Like any gem, if we don't pay attention to which version of the gem is running, we can run into problems, but we have checks for that. What we do is that the model is ready, but it's still not ready to meet the user. What we do is we do what we call life model back test.

We deploy the model in the application, so as the orders come true, it evaluate them, it makes a score and a recommendation, but instead of powering the users with it, it only writes it to a

Kafka topic. What we do is that we're also observing the same data in data land and we're constantly training. What we do is we compare what in data we predict using these inputs and what did we predict in production using these inputs? Match those. It's only when we match like 100% match between the two systems that we take it out of the shadow mode and we power users with it.

[0:29:13.6] SC: This process, is this an automated process or data scientists kind of manually overseeing this transition into production and making sure that monitoring the performance overtime and then hitting a button to deploy it?

[0:29:29.8] SS: Parts of it are automated. The jobs that reconcile the two are automated. The reports that show you what the reconciliation looks like is automated. The last step of just like making sure everything reconciled and then pushing the deployment, that's not automated right now. I do want us to get a bit more experience before we like fully automate that, or at least learn how to catch things if something goes wrong.

The other benefit of live model back test is, also, it gives us some metrics and real understanding around performance of this new model. As we make more sophisticated models, we're reaching out to more internal services to get data with different SLAs and models that are getting more complicated. What's the runtime of it going to be. I remember, I said we'd really want to have the fraud detection and risk analysis ready as soon as the order is placed. It allows us to say, "Okay. What's the real performance of this model?" Once all of that are good, then the model is ready to meet the user and hopefully the user would be delighted by the results.

[0:30:25.0] SC: Do you currently deploy new models out to all users or do you do like A-B testing or something like that? Do you feel like all the steps you've taken up to now mean that you don't have to do like partial deployments and A-B testing?

[0:30:40.9] SS: Right now we deploy to all users, but I foresee there are other things that we want to do. We want to be able to look at the voting scheme between different models. We want to be able to have more specialized models for specific segments of the users. When we get to that, we're going to do more A-B testing. It's something that I foresee in the future for sure. Then we would do holdout sets so that we're not impacting the actual real prediction. Yeah, the fun with that.

[0:31:06.7] SC: How far do you see that going? The way you described that, I can envision like different models for kind of arbitrarily small customer segments.

[0:31:16.9] SS: Yeah, you can do that. It's actually with the pipeline, it's very easy. Then it becomes like — Just because we can do it, should we do it or not, because then it brings also maintenance.

[0:31:25.9] SC: There's an overhead associated with it.

[0:31:27.1] SS: Yeah, overhead associated with it. Yeah, in terms of like platform and scale, yes, we can do it.

One of the things that's interesting is like by us going through this problem, first thing we focused on fraud. We actually built this pipeline that now can be used in other parts of the platform as well. If I want to deploy an algorithm that tells their shipping service, like what are the default dimensions of something that the user has added. I can use the same pipeline, like the same encoding of model and then deploying it and running it in another Ruby app. Those are the steps that we wanted to make easy, because I think companies in general, they use AI, so they encounter AI into different ways. Sometimes the company is built as an AI company and the challenge they have is finding the data. Sometimes the company is build, the business is working perfectly, they have ton of data and now they want to introduce machine learning to parts of existing company, and I think we are in the later one right now.

What matters for me right now an my team is to be able to sort of unlock that capability across many services so that we can give the benefit at scale to different parts of the platform.

[0:32:29.6] SC: Right. Interesting. You've got this pipeline going and you said a number of times, Spark and Python, and is the data HTFS? Is that why you're using Spark primarily or —

[0:32:44.1] SS: Yeah. Our internal platform, that was initially built as an ETL extract transform load system, uses Spark, Pi Spark, and it's really good because the volume of our data is really high and Python is really easy to adopt, so Pi Spark was like a sweet spot for us as a company to adopt. We've built this like platform so it abstracts so many things from the user. Now, the Spark data frame is actually like really easy for anyone who has worked with tabular data and SQL to like write a Spark job.

We use that, because we also have like an amazing team of data engineers that have built this platform, and by free we get so many things. I was actually just talking to someone upstairs. We get things like metadata on every input data that we've got. For every model, I have a UUID, and by that I can go and say, "What was the git SHA of the code that's ran this model?"

[0:33:34.0] SC: What was the what?

[0:33:35.0] SS: Git SHA, like the —

[0:33:36.0] SC: Oh, the git SHA. Got it.

[0:33:36.4] SS: Yeah. Then the path —

[0:33:38.5] SC: The hash of the git commit of the model.

[0:33:41.1] SS: Exactly. I can exactly go and reproduce what the version of the code was run, and I also have the information on the sort of snapshot of the data that was run as input. All of these capabilities to reproduce to be able to audit, to be able to like go back and check or reproduce. Those come for free. My team didn't have to build those. Those are — Yeah, exactly. Those are amazing things that were built for us and we just use them.

[0:34:06.3] SC: Okay. Nice. Is all that infrastructure and Spark, is that all deploy time or not deploy time, but like inference time, or is that involved in training as well?

[0:34:18.7] SS: It depends. Spark has machine learning library, so we use it for training as well depending on the models that we are using, but it's well-known that like scikit-learn has just like a broader set of machine learning capabilities that implement it. I'm hoping that the community, including our company, gives back to Spark by adding these different learning algorithms.

Yeah, we sort of go between for loading of the data, for doing all the transforms. Most of the things are at Spark level, but then for sort of training, we use scikit-learn heavily. It's a mix of two. Yeah.

It also makes onboarding people really easy. Many people in the data field are picking data or they've already worked in data and Python, so it's like a familiar interface and just breaks the barrier very easily.

[0:35:03.3] SC: And it's a lot more accessible than Spark.

[0:35:05.6] SS: Exactly. Yes, it is.

[0:35:07.3] SC: Okay. Awesome. Anything else that you would want to leave folks with or, left folks with in your presentation?

[0:35:16.3] SS: Right now I focus a lot on the mechanics of making this model be alive. Part of that is because I had the luxury of being in the same company, being in the same domain for the like two and a half years before we even started tackling this problem. That means like the understanding of what the features mean or where do I have to poke to find out what the features means. That came with me and my team.

I think one thing that I want to emphasize to people is that try really hard to understand the domain and the problem you're trying to solve, because at the end of the day, so many times, machine learning is sort of builds up on top of heuristics that already humans are in that field. Spending that time that does not feel like you're working on a fancy learning algorithm is actually one of the most important parts of having a successful data product.

[0:36:03.2] SC: Yeah. Interesting. Thank you so much for taking the time, Solmaz, to sit down with me. I enjoyed your presentation and I enjoyed talking to you about your presentation and I appreciate it.

[0:36:16.3] SS: Oh, thanks a lot and thanks for having me, and keep producing this awesome podcast.

[0:36:20.7] SC: Thanks so much.

[END OF INTERVIEW]

[0:36:26.5] 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 Solmaz or any of the topics we covered in this episode, head on over to twimlai.com/talk/60. To follow along with the Georgian Partner Series, visit twimlai.com/gppc2017.

Of course, you can send along feedback or questions via Twitter @twimlai or @samcharrington, or leave a comment on the show notes page. Thanks once again to Georgian Partners for their sponsorship of the show. Be sure to check out their whitepapers, which you can find by visiting twimlai.com/georgian.

Thanks again for listening, and catch you next time.

[END]