EPISODE 1499

[INTRODUCTION]

[00:00:01] ANNOUNCER: In this episode, we interview Hossein Rahnama, Peter Kramaric, and Justin Lam from Flybitz.

[INTERVIEW]

[00:00:09] JM: Guys, welcome to the show.

[00:00:10] HR: Thanks, Jeff. Thanks for having us.

[00:00:12] JM: I want to start off by talking at a high level about banking and banking software. If I'm a typical bank, I mean, there's so many banks out there, and there's a wide variety of quality in terms of the apps that these banks have. How does the average bank get their software for managing mobile banking?

[00:00:37] HR: That's a great question, Jeff. And our observation today has been a lot of these banking channels are very transactional and also very passive. I mean, they're trying to mimic the classical role of a bank that is about giving you access to an account, or giving you money, or lending you money. But one thing that is changing and they are realizing after so many years is that, because of the advent of the big tech, because of data ownership, because of awareness of data ownership and privacy, they are realizing that they can play a much bigger role and they can become that trust hub for people to share data with them.

Most of us in North America, in Europe, we have a bank account, and we trust our bank account relatively. And these are also entities that have relationships with businesses that we work with. So, they have access to indirect repositories of data that may host our loyalty program data, credit bureau data.

But in the past, if you think about an architecture of a banking infrastructure, it's very centralized. It's very monolithic. And as a result, it's extremely expensive to maintain, scale and let alone change. With the advent of new systems, data-driven systems, decentralization that we are seeing in the architecture, we are also seeing an opportunity to change that as a company and as a group, but without following the old incumbent view that, "Oh, you need to go through a digital transformation? Gut out everything that you have and put new things in."

And this is where our approach and design patterns come in which we can go to a bank and say, "Keep all your existing and legacy systems; mainframes, CRMs, CDPs." Just build an abstraction layer that can orchestrate effectively between these siloed systems and create a unified view of these data assets. And while you do that, not just do that for the developers and engineers, but also build it as a service that creative people, designers, people in charge of compliance can access interfaces that can benefit these next generation of infrastructures that are coming up.

There is an old view of a banking environment which is relatively boring, very centralized and very monolithic. And there is a new view in which, well, it's very expensive to change all of that in such a short period of time. Most banks have a digital transformation project. But there is a way to do that by introducing decentralized abstraction and orchestration in which it's going to be a good discussion topic as part of our show.

[00:03:38] JM: Data integration, if you're trying to gather data from a wide variety of sources that a typical bank has, like a CRM, and a customer data platform like Segment, and be able to perform operations over that data, there has to be some kind of unification system for aggregating that data and allowing for transactions across it. Can you give me a sense of what the data aggregation process looks like?

[00:04:10] JL: Yeah, you said it right, Jeffrey. There needs to be some sort of unification layer. At Flybits, we have something called a context plugin that essentially creates that unified view of data attributes. And we purposely create this context plugin agnostic to how we bring the data in. Because we realize, depending on who we're working with, where we're getting the data, it can be in any form. It could be an API. It could be a gigantic flat file. And working with customers around the world, we kind of have seen everything under the sun. It was important to create a context plugin that defines kind of a data schema that we can map all different disparate data to.

Now, when we think about how to get to the data, sometimes data can be pushed to us, sometimes we have to go get data. And that's kind of where we saw the need to create kind of a framework around creating these integrations. And the framework defines a set of conventions and contracts that anybody building these integrations need to follow for the purposes of reuse.

We kind of saw working with a lot of different customers that every project required additional development. I got transformed this data format to another data format. I gotta connect to a proprietary authentication mechanism. We really looked at ways to reduce and potentially reuse this integration work. And that's kind of where the integration framework that we designed comes into play, that it allows for these what we call connectors to be reconfigured to be reused across different integration projects that we have.

Another benefit of this framework is that because it imposes some sort of contracts and conventions around these integrations that, now, Flybits can go opportunistically fetch data when we need it for decision making. It's kind of a both supporting push and pull kind of flows of data.

[00:06:01] HR: And one thing I want to add, Jeff, in at least what we learned, not just with banks, but also when we work with the large enterprises, that when data transformation or digital transformation strategies start, one thing that is missing is actually the context and the synthesis of context around these data.

A lot of the context building happens usually after some type of a use case is being created. And that creates a lot of DevOps challenges for the organizations that we have met.

One of the things that we created is actually a semantic layer on top of our data layer, which really allows you to quickly map that data into these templated semantic-driven experiences. For example, before you start your data integration, we ask you questions. For example, in the context of a bank, are you building a fraud use case? Are you building a wealth use case? Are you building a use case which is supposed to upsell and cross-sell products?

Just by knowing that semantics and context in advance of data aggregation, data synthesis, data modeling, you can really make that data integration more effectively during the build process, during the aggregation process, and especially during the exposure process for people who actually want to use the data as an asset for predictability, for experience design. So, that context layer is also something that we have invested heavily making the job of IT professionals, software engineers and data scientists more effective.

[00:07:48] JM: Can you describe in some detail what your infrastructure looks like? If I'm a bank and I come to you, and I want data integration and higher quality data science across my platform, what does the integration process look like in more detail? And what are some of the tools that you're using? The databases and runtimes that you're using?

[00:08:12] HR: I'll comment on kind of our view and our deployment model. And Justin and Peter will comment on the stack that we have. The first thing we have done is that we built the whole platform as a service as a set of microservices. Because as a scaling organization, we could not expect to go to a large complex enterprise and say, "Okay, procure my monolith versus another monolith." We had to expect that any organization that we go to, there are different systems, different data contracts, different authentication models. And that end-to-end microservice-based design really allows us to integrate with these complex organizations effectively.

The other thing that we do is that we allow a very effective containerization of our context models. Because there are business units in these banks that they want to do their business using risk assessment. There are ones that may want to really increase subscriptions or upsell and cross-sell.

As you may imagine, there are different AI and machine learning techniques for these types of business units. You may need Bayesian models when it comes to risk. You may need a neural network when it comes to cross-sell and upsell. Building that containerization and orchestration, there's actually a capability we have called the inference router. Really allows the CTO's office or the organization to create a graph, a database graph, that includes all of these models contained, orchestrated. And based on what the bank wants to do, we will go and pick, let's say, the risk model for a credit card. Or we will go and pick a neural network model for upselling a product. Those capabilities allows us to integrate with banks very quickly and shrink their time to market significantly when it comes to their digital transformation strategies.

In terms of the stack, maybe Peter and Justin can comment on it.

[00:10:16] PK: Yeah, from the infrastructure perspective, as a SaaS platform, we move more towards an AWS cloud that we orchestrate all of our microservices from. Everything's kind of hosted there. We are looking at ways to move to other clouds to be as agnostic as possible. But we use a lot of the proprietary services that are used on those clouds as I mentioned.

I think, from a database perspective, it really depends on the use cases, right? Sometimes structured data makes the most sense, whether it's a postgres database. Sometimes it's unstructured, whether it's MongoDB. For long term analytical purposes, we use Redshift from AWS directly. It really depends on the use cases that we are working with our customers.

As justin talked about the integration framework and the connectors, the nice part about those is they're all developed as microservices and can be hosted anywhere. They can be hosted inside Flybits if a customer doesn't feel comfortable operating it themselves. Or if they build it and want to host it internally inside their systems, they absolutely can as well.

We try to limit our customers in terms of what they can do. Because working with banks, it's very hard to go in and say, "I need you to use this database. To use this service." They're just going to come back and be like, "Ah, this is what we use. You guys have to deal with it." That's part of the reason we built our own framework, the integration framework, to be able to be flexible and be able to connect to any type of system and be hosted anywhere in the world that our customer is.

[00:11:49] JM: Do you have a one framework per customer deployment? Or maybe you can talk a little bit more about like if you have like one single deployment, what are the AWS resources that it consumes? And like, do you have a one framework per customer deployment model?

[00:12:05] PK: We're a fully multi-tenant solution depending on the type of customers that we have. If it's an enterprise customer, they will have a full deployment that includes our entire stack, from the databases, to Kubernetes, to Sagemaker, to all the different aws tools that we use. If it's uh not an enterprise customer, then there's a lot of shared resources inside one of our clouds that they would use. It depends on the customer.

Most banks tend to want their own deployments to their enterprise customers. But our system is used by other people outside of the financial institution space. We have corporate campuses that use our technology. We have device management systems that use our technology. We've done things with fashion weeks and soccer clubs and others as well. And those tend to be the non-enterprise customers that get deployed into our shared resource pool.

[00:13:04] JM: And can you go into a little more detail? We're about using a neural network potentially to identify upsell opportunities. A bank obviously has lots of upsell opportunities. That seems like a good prototypical use case to maybe describe an end-to-end example in more detail.

[00:13:22] HR: Sure. The view we have from a deployment perspective is that we call it install once, deploy everywhere. The way we do it is that we have this ability to really replicate the structure of a tenant not just within a geography, but across all our deployments globally. This comes with all the templated experiences, the data points that you need, the data assets that you need for building experiences. Then these are all interconnected with a policy engine. You may imagine that, in parts of the world, there are different privacy laws, there are different regulations. In run time, we will be able to use the same tenant from anywhere in the world. But when you launch that experience on someone's digital channel, automatically, it will adhere to that particular privacy policy or regulation during the runtime.

And the way we do this is that, as Justin mentioned, there are connectors and plugins for not just public data assets that we offer as part of our tenants. But we also enable any organization to turn their data assets into these context plugins allowing them to interconnect and synthesize with other data elements that are available.

Now, some of these data points that are being used to generate experiences are atomic data. Think about like properties and attributes. Some of them are outputs from a machine learning model. That model is already containerized as a connector and is being exposed to the experience designer. This model may use a neural net or it may use any other network. It could be a causation. It could be a Bayesian network. Or it could be some other classification tools.

Specifically, when it comes to neural nets, the way we use them is that we gain the ability to interconnect classifiers during the runtime. We have classifiers, let's say, for movement data. We have classifiers for transactional data. And we have classifiers that, let's say, can look at a group or an aggregated basis demographies.

Because of the way neural networks are created, it's very difficult to build a composite classifiers at the beginning to do all of these at the same time. We have gone and containerized each classifier with a particular context and purpose, as I mentioned.

When the output comes in, then what Justin talked about allows you to really chain or interconnect the output of these classifiers together to really do that synthesized inference at the end. You can use this for understanding patterns, like people who tend to do these types of traveling, tend to have this type of a purchase when they come to the shopping mall. You don't have to go and build this as essentially a compiled element and just output it. The experience designer can actually use the output of every single classifier separately and turn it into an experience for their customer. That's one example that we have seen on how our machine learning tools are being used.

[00:16:43] JM: I'd like to know a little bit more about how you design the right abstractions to plug into since banks have a pretty wide variety of data they could be capturing that they might want to plug into for building a model. Describe in more detail, like, how do you expose the right abstractions to plug into models and generate those models when some banks are going to have some pieces of data, some banks are going to have other pieces of data?

[00:17:15] JL: I think it case by case basis, depending on the bank. We kind of establish what use cases they want to solve. I think a lot of our customers sometimes come to us and say, "Where's the AI?" right? But it really depends on what they're trying to solve. And then we kind of drill down from there to understand what data is available.

Take into account data that they might have, proprietary data that might change in real time. Or we might consider data from the device or even external data that might provide additional segmentation to feed into the models. But it really kind of depends on what we're working with.

[00:17:49] JM: I'd like to zoom out for a second. If you're working with a bank, like depending on the bank, they're going to have a variety of kind of available engineering resources that they might be able to devote to a project. And they're going to need – Kind of depending on how engineering intensive the organization is, they might need a smaller or a bigger amount of assistance when integrating with a data platform. I'd like to get some sense of to what extent you try to avail yourselves to the customer as a kind of consultancy or customer support versus having a self-serve platform?

[00:18:32] HR: I can share some of our observations with you on that regards. If you think about almost any major financial institutions, or as a matter of fact, any large enterprise organization, they all have a project or an initiative called digital transformation or data transformation strategy, in which in these types of big, expensive initiatives, they want to really leverage their data assets to either create automation and productivity for their organization itself or to engage with their customers on a more effective predictive basis.

Now, if you look at most of these projects even into six months of their start, you realize that about 90% of the budget allocated is being spent on managing IT and data complexities. And although, at the beginning, they came up with some amazing use cases and creative use cases, they realize that they will not be able to deliver on them at scale. Number of reasons. Some of them naturally are technological that we talked about. Their view that, "Oh, we need to centralize everything in a data lake. And making sure that all the data sets are normalized before we can expose it as a service layer or an API for others to use." Lots of things that we can discuss there.

In terms of – With capabilities, like trusted execution environments, with algorithmic capabilities that allows us to ask questions from the data rather than co-locating the data. And building the abstraction models and layers that we talked about, you can really address some of those challenges in an organization.

But also, another interesting element is the organizational element of how such projects is being run. Many of these IT organizations that you talked about are very hierarchical and have layers and layers and layers of governance and bureaucracy around them really slowing down the integration process, the build process and the launch process.

What we have seen working well is that when such teams are broken down into interdisciplinary agile units that are essentially acting as product teams. And having that abstraction layer, having that semantic layer that has broken down data based on a taxonomy against the need of a business or a use case will really provide tools to these smaller teams to essentially act more autonomous while there is a governance layer built around them allowing them to scale and launch more effectively. Those are some of the things that we have seen. So, the technological element of that initiative should go hand-in-hand with the organizational. And then, of course, the support should come from the very top of the organization. Allowing such teams to have more autonomy while the governance is understood.

Those are interesting things that we have seen. And by doing that, basically, you will go from a very slow bureaucratic IT organization that will take almost months, if not years, to launch use cases, to an essentially a well-orchestrated group of product teams that are following and adhering to the same principles. And you will see that use cases and capabilities are launched more effectively on the digital channel.

You can contrast that with what's happening in a lot of organizations now that basically goes like this, "Okay, let's go buy a multi-million dollar monolith." Okay? Let's go find a system integrator that can put it together for us. Okay? And then after that, let's go work with consultants who can help us to understand use cases that we can build on top of that.

Well, our view is that, for that to change, you need to really understand technological elements and data elements as we talked about. Because data is a key asset and an enabler for these projects. Really, break the organization down into more autonomous units and then really change your creativity practices and design practices at the same time.

[00:23:02] PK: And if I can just add on to your original question as well, Jeff. I think, as a startup, there is a level of expertise that we are bringing to these organizations as well. And we try to help them as much as we can through the – Especially the integration process. We want to be as self-serve as possible. But these are large financial institutions, and sometimes they can get in their own ways. And so, we try to help them as much as we can. Most of our customers have some sort of input coming from us and helping them. And it's something that we we do pride ourselves in, is that we want to help you through this process as well. It's not a hands-off process from our perspective. Obviously, the bigger the organization, the bigger the challenges. That's the same kind of mentioned and then some of the internal things that they have to handle. But we do kind of have boots on the floor and trying to help everybody to reach that integration strategy.

And one example that we always come across is, let's say, there's five use cases the customer wants to do this year. And maybe next year they want to do 20. We want to try to avoid looking at those use cases specifically to get the data, right? Because you can pigeon yourself into those use cases and not think about the 20, 30, 40, 100 use cases that you want to do in the future. When you take a look at the data, that's the most important thing actually. Not the use cases. Because from one piece of data, you might be able to create 50 use cases in the future. Whereas, if you're just focused on the five use cases I have to build this year, the data that you expose and the data that you abstract tends to be focused on just those five. Aand then next year, when you want to change to another ten, you're like, "Well, but the data was structured in a way to meet those five. Now we have to go in and change the work."

What we really try to help our customers understand is, "Hey, holistically, in the future, don't think about these five. Think about what happens when you want to do 500 use cases in a year? How can we get that data to be available to you?" And it's not specific to one use case. You don't necessarily build APIs just to do the single use case. Build generic ones that can be reused from multiple use cases.

It seems trivial, but it tends to go down the route where someone's like, "I just want to solve these five use cases. We'll think about the use cases later." And that's where things tend to be you become a bottleneck. And we've seen that with multiple customers in the past.

[00:25:34] HR: And Jeff, one thing that I really want to highlight is that, for sure, the physics of data is changing, portability of data is changing, the magnitude and the volume data is changing. But one thing that is missing in a lot of these large data transformation projects is our relationship with the data. In the old days, even if you think about a business to a consumer-facing company, the thinking has always been, "Okay, as long as the data is there, I can put it in a repository, I can put it in a lake, and I can put all sort of mining capabilities and inference capabilities around that. And then whatever I synthesize from it, I can use it to push content to the user or give offers and all that."

In the large enterprise, especially in banks, that is fast changing. Regulation is pushing for that. Consumer awareness is pushing for that. The way we should treat that the relationship with data is that that data does not belong to the bank anymore. That data belongs to the end customer. And how do you design systems and architecture that the permission of the user, consent of the users, policies that we need to adhere to are all being considered in real time when we are processing, accessing or porting the data?

And when you do that, of course, that will give you a new transactional capabilities as part of the abstraction layer that we talked about, which, again, it's consulting the user on what data attributes can and cannot be used. We are working on capabilities that is building data alliances around banks. For example, are you going to to trust your bank to share some of your data attributes with a telecom carrier, with a utility company, with an airline, in exchange of value? Those are all algorithmic capabilities that we have built and we are using.

But one thing that we are also realizing as part of this relationship change is the role of design, user experience and UI. Most of the channels that we currently use don't consider the fact that you need to embed trust into that design so the customer feels comfortable sharing the data with you. In this case, it's a bank. But it could be any other organization.

When you understand that in the data strategy that my team talked about, which is, "Okay, use the data, but really consult a policy and consent while you're accessing it in a decentralized fashion." And also, how do you use that to create interfaces and UIs that the customer is feeling more comfortable sharing data with you? And whenever they don't feel comfortable, they have this ability to quickly de-link their data or essentially have this right to be forgotten as part of the experience.

When all of these come together, we can see that the uptake of the digital channels or apps go significantly higher. The data sharing and the comfort of data sharing will increase. And naturally, the impact of the experiences that you can create on the channel will grow.

[00:28:45] JM: Okay. I think we've talked a lot at a high level. And I'd like to get a little bit more into engineering. If you think about the variety of use cases you can have for a data platform, specifically for banks, and you've got home buying, lending, credit cards, I'm wondering how you architect the code base to be reusable across different use cases. And to what extent it is reusable? Maybe you could talk some about how you have architected to be extensible. And talk through some of the programming language and architecture decisions you've made.

[00:29:30] JL: Some of the philosophies around Flybits to begin with wasn't necessarily predicated around the banking industry. I think our goal as a platform was always to build personalization, whether it was in finance, or zoos, or whatever.

Underneath the hood, it's an event-driven decision-making platform that has interactions with mobile, or web, or any other digital channel on the delivery layer. I think once we realized that that's where we're going, we embarked on this microservice-driven approach and kind of rewrote the platform from the early days when we had a monolith, because we knew that it was not going to scale when we started to encounter more and more data sets and more and more decisions happening every second.

That also led to our choice in language as well. All our microservices are written in Go lang, which at the time, three, four, five years ago, was a little progressive in the financial space for sure. But we kind of saw the merits behind the choice. And we haven't looked back ever since.

In terms of kind of having that scalability across use cases from the data layer, it's kind of what we talked about in terms of building that context plug-in data structure. It creates an abstraction between how the data is coming in, where the data is coming from. Basically, unifies our understanding within the platform so that we know that, "Hey, this is a float. It expires every 10 minutes. And we might have to go get it again." And it allows us to provide that kind of comparison and decision making around it agnostic to what it actually is, whether it's your account balance, or the temperature outside that's kind of our approach.

[00:31:24] JM: Let's dive a little bit deeper into a specific model. Lending, for example. If I want to build – Like, let's say I'm a bank. I come to you with five different pieces of data for each of my customers. I've got their transaction history, their credit history, like whatever public credit history I can acquire. I've got where they live. I've got – Or let's just say those three. Where they live, credit history and transaction history.

I'm a bank. I'm trying to rank order them in terms of leads for extending them loan offers. Can you describe what the integration process would look like? Maybe talk through some of the services that would be used in architecting a machine learning model for continually being able to rank order them?

[00:32:22] HR: Before we get to the machine learning model, the first thing we do is to really figure out that data ecology that is needed to build this use case. But some of the data assets that you mentioned, they are inside the bank across different silos. And there are some data points that may need to come from outside the bank.

For example, you may need a data point coming from a credit bureau. You may need some data assets coming in from someone's liquid assets, savings assets, risk models. These may all come from different repositories from inside and outside the bank. That's the first thing we do.

The second thing is that some of these data assets may not be available as part of a service layer. An API may not be available. So, we may need to do an FTP dump. Or we may need to read the flat file. The one coming from outside the bank, it could be a Kafka stream. It could be an API that we need to call and authenticate.

The first thing we do is to really figure out how effectively we can unify the access to these data assets and provide them as connectors. So, whenever a business unit exactly following what your use case said, they can actually build it yourself. All of these data points that you mentioned first will become a connector as part of our integration framework capability. Aand within that, you also have capabilities to say, ", I want this to come as a file. I want the other one to come as a Kafka stream. Should I pull the data? Should I push and trigger from the data?

What authentication models?" All of them will be embedded into the data structure of the connector that we talked about.

Now, when the data assets in form of connectors are available, then you will choose the type of a machine learning model that you need. When we will take you through some sort of a wizard, some people don't know which machine learning model may be useful for this. We'll take the user through our interface that, "Okay, what do you want to do? Do you want to do a recommendation? Do you want to do a comparison? Do you want to do an association? Do you want to do a prediction?" By answering that question, then we will take them to the second step, which is what is your intent? And those are some of the business objectives that are mapped based on the industry taxonomies that I talked about.

Then the third step will be, okay, certain models will be recommended to the user based on what they want to do. In this case, it could be a recommendation. Some of these models could be provided by Flybits. Some of them could be provided by other vendors that are all orchestrated and available. When you choose the model, then you feed the data streams through the connectors that we talked about, and then the orchestration will happen.

Use cases that we have seen are – One of my favorite one is the one that different units in the bank are using this capability to create a synthesized experience for the user. And in this particular case, they get a trigger based on a fact that someone just got a mortgage. They also realize that the person has a credit card with a certain loyalty program. And they want to nudge the user that if they go to a particular hardware store or a home related store and purchase items that are related to their new home, they are going to get additional points.

Now, this is an interesting use case. But in this particular case, one of our customers just thought about this use case and could go and build all of that without writing a line of code by interconnecting the experiences and the models that we talked about. That's more from the usage perspective, that we differentiate between first the data and then the model. Then we have a governance models on our data, which we basically do not allow the person who is building the model to basically build an experience with it. There are two different interfaces just to follow the governance on model building. And then there will be a dashboard that is providing that explainability auditability to the user. So, if they have a question that, "Hey, why did you give me that offer? Or why did you recommend that?" We can provide that explainability through a visual dashboard.

From a technical standpoint in terms of what is needed to turn those disjointed data assets that I talked about, file, APIs, JSON objects, Kafka, and how we unify them is something that Justin can talk a little bit about.

[00:37:14] JL: Yeah. It depends on the data source. We've built connectors that are microservice-driven to facilitate API integrations. We've built kind of the – We've taken the serverless compute approach to kind of break down flat files. Some connectors could live in AWS as just a bunch of Lambdas that listen to S3 buckets and file changes there that kind of split and chunk up data. It really depends on the use case that we can build different connectors for. And I think that's kind of the flexibility that we're trying to bring with the framework.

[00:37:45] JM: Do you use any kind of systems for machine learning? Like, any frameworks? Like, Sagemaker or Tensorflow? Can you talk through some of the machine learning tool choices you've made?

[00:38:01] HR: Yeah, we use Sagemaker. We use Tensorflow. We have number of our build built between R and Python. But what I always believe is that you should give a lot of the autonomy to the model builder, right? And you should gain an ability to, while you give them autonomy, you also give them a container that when the code is built, irrespective of the syntax and how it's been structured, you can containerize it, you can vet it to make sure that it's secure. But then you can orchestrate it as part of a graph database allowing people to pick and choose models.

Although some of our data scientists and ML experts use things like R, Tensorflow, especially for our neural network approach, Sagemaker, but we also work with a lot of other entities, whether they are startups or research labs that they're like, "Hey." We tell them, "I really don't care what tool you use. As long as you can put it in one of my containers and I can vet it, I will orchestrate it as part of my inference graph."

And I use context to tell the user, "Hey, if you want to do –" Let's say you want to give a credit card to a first year university student, well, risk is a context. Automatically, our inference router will pick a risk model for the person who is building the experience.

From a tooling perspective, we really don't have much of a preference for how the machine learning model is being built. We have certain guidelines in terms of how they are built so that we can provide that explainability. But we do not impose restrictions on the model builder. We want them to use any tool or any syntax that they like for building and verifying their model.

[00:39:51] JM: Have you experienced any feedback from the machine learning engineers on what they have more success with Sagemaker versus Tensorflow?

[00:40:02] HR: Tensorflow, I primarily work with research engineers that primarily want to use neural networks. At least that has been my experience. And they use a combination of Python and Tensorflow at the same time.

Related to Sagemaker, at least the group that we work with, they are more on the infrastructure side of things. They use more raw level data. And they use the infrastructure that Sagemaker provides to do some of their work.

In terms of feedback, again, our view is that it's their preference. It's their tooling. The way Flybits looks at it is that, "Okay, as long as you have the right precision, as long as you have the right context in your models and I can validate it and verify that is explainable, I can audit it, I can basically review in a transparent matter," we just use their models. But I don't have a particular comparative view at least from the people that we work with in terms of which one is better than the other. Usually, they are being used by different types of people with different expertise and depending on the type of machine learning model that they want to use.

A lot of them, however, what I say, which I think is a missing thing in the industry, is all of them expect the data to be available to them. And in many cases, what we have seen is that data is missing. Data is noisy. And a lot of companies and even data scientists spend a lot of their time kind of cleaning that data or having an impediment to start their job because data is not available.

Some of the work we have done to help them is to help them with techniques like classification and fuzzy clusters to go and tell them, ", even if you don't have all the data and you have some attributes, you can basically cluster the data with probabilities. And as long as you give me that, I will be able to use it."

A combination of these techniques coming together with an effective orchestration. What I have seen is that, through the lens of our customers, we'll give them the scale and the precision they need to use models built with a heterogeneous set of tools like the two that you mentioned.

[00:42:19] JM: As we begin to wind down, I'd like to get a bigger picture view of how you see financial institutions changing particularly with regard to machine learning and data. We've been in kind of a big data revolution, so to speak, for, I don't know, 10 years at this point? Maybe longer. And I'd like to know your perspective for what happens over the next five to ten years.

[00:42:50] HR: Sure. My view is that most banks and financial institutions have a huge opportunity ahead of them to transform themselves to become this trust hub of a much broader ecosystem. First of all, they are all very worried about what big tech is coming and doing to them in terms of, "Oh, many of them are getting into the financial sector. Many of them actually want to have access to the transactional records of the consumers. And open banking is helping with that."

Now, banks are really feeling that urgency, that, "Okay, my old business model of just charging service fees and just providing account instruments may not be enough to stay relevant in the next 10 years or so." However, they can actually become an exchange hub or a data hub allowing their existing customer base, which is large, to share data with the broader ecosystem.

A lot of work that we are doing now is to really build these data alliances around banks. You can go to your bank and ask your bank to share data with your utility company because you want to contribute better to your ESG metrics. Or you want to have a better travel experience with your airline. This is where I can see banks are really shining, that in addition to money and financial elements to be their assets. I think data is becoming one of their biggest assets that, with the consent and permission of the users, they can monetize it to build a much stronger reciprocal ecosystem. That's one thing.

The other thing that I observed with banks that has changed over the past two or three years, you may or may not remember this, there was this AI craze in banks over the past five years. Board members went to the CEO saying, "Oh, we need some AI. And Ceo went to the CIO saying, "We need some AI." But no one asked, "Well, for what? To do what?" I mean, even under the umbrella of AI, machine learning, there are 50 different ways that you can do machine learning. Which one the bank needed to solve what type of a pain point?

A lot of that literacy in banks happened over the past five years that, "Okay, there's nothing – Very few things is very new in AI in the past five years." A lot of it is statistical modeling, that now because of the volume of data, because of the compute power we have, because of the storage capabilities we have, we can actually demonstrate its value. That literacy has changed and is actually bringing value to the bank.

The second thing that has changed is that, although most banks talked about AI, very few of them understood the relationship between data and AI. Because without data, AI is meaningless, right? And how many fintechs or how many entrepreneurs or even banks are frustrated because their mindset is still the mindset of a graduate student in a research lab, which says, "Hey, professor, give me the data and I'll do my work." It doesn't work like that. Because that data is sensitive. That data should be privacy preserved. That data, in many cases, should be computed and transported in an encrypted state. So, now, understanding all of those and then combining it with AI and understand the importance of a data strategy, data portability strategy, against AI is something that the banks are realizing more and more and more. That is not just about building models. It's also understanding that how do we use data, in which in many cases is now not their data, it's their customers data, is becoming a big transformational factor in financial institutions.

With that, I think there are two things that I can see are opportunities for banks. First, positioning themselves as a data trust hub for a broader industry. And then play a much bigger role than just being a transactional enabler. To become a trust enabler to help their customer to live a better life. And all of that will happen based on the correct usage of data and the ability to turn data into experiences that are valuable to the customer.

[00:47:18] JM: Well, Hossein, that's a great conclusion. Thanks for coming on the show.

[00:47:20] HR: Thanks for having us, Jeff.

[END]