EPISODE 620

[INTRODUCTION]

[0:00:00.3] JM: Every company has the idea of the nightly report. A business analyst comes into the office in the morning and sits down in front of their inbox and looks at yesterday's data. Did sales go up? Did the marketing campaigns bring in the expected number of customers? Was there an increase in help desk tickets? The statistics that these reports deliver to human analysts can change the direction of an entire business. Everyone within a company could use a regular report that documents how the business is changing over time.

Outlier.ai is a company that processes the datasets within a business and generates automated reports that are relevant to different people within the organization. If you're an email marketing analyst, your MailChimp campaigns will be analyzed. If you manage a customer success team, your Zendesk tickets will be analyzed. If you're technical support analyst, the crash reports and the error messages from your users will be analyzed. In all of these cases, the data gets processed automatically and a story is sent to you so that you can have the information in your inbox waiting for you instead of having to go ask a data scientist to generate it.

Mike Kim is the CTO of outlier.ai and in this show he described the engineering challenges of integrating all the different datasets of an organization and why there's so much value in the idea of the automated report or the story that will be received by an analyst. In past shows we've explored how data engineering has progressed over the last 20 years, from database administration, to Hadoop cluster management, to the emergence of the data bread lines where analysts wait for a data scientist to process the job that they asked for.

Outlier represents a step towards a world where the data science reports are delivered to us before we even ask, rather than us having to query the system. So it was a great historical venture into the past and a look at what the future might bring.

We are hiring for Software Engineering Daily. The jobs we're hiring include writers, researchers, a videographer, and you can find those positions along with some other jobs at softwareengineeringdaily.com/jobs. Some of these are part-time, some are full time, and if you're hiring, you can also post your own jobs on our job board. It's easy and it's free. Just go to softwareengineeringdaily.com/jobs and you can see how to post a job.

[SPONSOR MESSAGE]

[00:02:32] JM: At Software Engineering Daily, we’re always analyzing data to determine what our listeners care about, and we actually have a lot of data even though we’re just a podcast. So it always reminds me that organizations with much more engineering going on have an order of magnitude more data than a podcast like Software Engineering Daily, and that’s why the job of data scientist is such a good job to get.

Flatiron School is training the next generation of data scientists and helping them land jobs. Flatiron School is an outcomes-focused coding boot camp that offers transformative education in-person and online. Flatiron School’s data science program is a 15-week curriculum that mixes software engineering, statistical understanding and the ability to apply both skills in real-life scenarios. All of the career-changing courses include money back guarantees. If you don’t get a job in six months, Flatiron School will refund your tuition and you can visit their website for details.

As a Software Engineering Daily listener, you can start learning for free at flatironschool.com/sedaily. You can get $500 of your first month of Flatiron School’s online data science bootcamp and you can get started with transforming your career towards data science. Go to flatironschool.com/sedaily and get $500 off your first month of their online data science course.

Thanks to Flatiron School for being a new sponsor of Software Engineering Daily.

[INTERVIEW]

[00:04:24] JM: Mike Kim, you are the CTO at Outlier. Welcome to Software Engineering Daily.

[00:04:28] MK: Thanks for having me, Jeff.

[00:04:29] JM: I want to start by talking about statistical outliers, because that's the core of your business, that's the namesake. What is a statistical outlier and what are some examples when outliers are important in a business?

[00:04:43] MK: Actually, this is a really funny bit of a place to start, because while Outlier is the name of the business, and Outlier certainly is a cornerstone of what we do. A lot of our promise is actually not on the statistical finding of outliers, which is actually a very well worked out subjects and statistics which I'd be glad to get into. But I think these days outlier detection and anomaly detection is really table stakes. There are plenty of open source packages that even do this quite well for a lot of different cases.

But to get at your question, an outlier is really this idea of can we find exceptions in data? Can we try to identify something that doesn't fit a pattern? So the natural outgrowth of trying to find out what normal is or what expected might be is to find out when things are outside that range. And so this helps businesses, like our customers manage by exception. Like I was saying, a lot of times in our modern data environments, this is just simply not enough. If you don't mind, if I can elaborate on that point as well.

[00:05:33] JM: Yeah, please.

[00:05:34] MK: So modern businesses today, they're collecting metrics on so many different metrics along so many different facets and dimensions, that a typical business has millions of different ways you can slice, let's say, revenue. You could slice revenue by region, by demographic and then by every single combination of those metrics.

So when you end up with all these slices of your data, once you've done anomaly detection, you're still going to end up with tens of thousands of anomalies and that's still far too many for a human being to really wrap their heads around and process.

So that's really were Outlier is coming in, is after we process all your data, find out what's normal, find out what's not normal, and as you said, do some physical processing, find these outliers, how then do we reduce this tens of thousands of possible outliers into a human manageable set of things for people to understand and take action on?

[00:06:23] JM: I think the more general idea is that your company is trying to surface insights about data that may not be straightforward for a human to notice or a human doesn't necessarily have time to notice, a human doesn't have time to run an experiment to notice.

So the whole idea is that you have a system that is built to recognize outliers, which is an example of something that if there's an outlier in my daily revenue data, I probably want to know what caused that. If there's a dramatic drop in daily revenue or a dramatic gain in daily revenue, then that's something I would want to be aware of. But more generally, there're other trends that are going on, there are so many different datasets within an organization.

No organization can track every single dataset and see every single trend that is occurring. So it's nice to have an ability to surface things, and that's kind of what the motivation of your business is.

[00:07:18] MK: That's absolutely right, and I would only add that it's not even that human beings aren't able to do this, in a lot of cases human beings flat out can't do this. It's not feasible to ask a human being to sit through and sift through millions of different facets every single day and try to figure out which ones they should pay attention to.

I've built a lot of my career around let's let machines do what they're good at so that humans can do what they're good at. This is the task. This is a task for a machine. You don't want a human being to sift through thousands and thousands of examples of things and try to find which ones are irrelevant.

[00:07:50] JM: The idea of the product again is that every day I get a set of stories about the datasets within my company and how things are trending, outliers, anomalies. It's a condensed report that machines are able to surface about the data in my company.

What are the different types of datasets that people want to be looking at? What are some examples that we can think of throughout this conversation?

[00:08:19] MK: That's a really great question. I think the really interesting niche that we've kind of carved four our self is that because the end consumer is a human being, the kind of resolution at which we operate is really like the daily, weekly kind of resolution, right? So you can think of other machine learning problems and applications, like high-frequency trading, where the end user isn't a human being, right? The end-user is a machine that would go execute trades. So there, their resolution might be on the millisecond or a microsecond timescale.

But what we're really dealing with at Outlier is anything that you can use to help you make decisions on a day-to-day basis. So revenue is obviously a really good example of one where if you have revenue tracked by all the different types of sources or where it came from or your product lines, this is obviously one key source.

A lot of our users use this for product analytics, obviously A-B testing. You've had plenty of guests on the show talk about great A-B testing and why it's important. But a lot of times we don't think about all the unintended side effects.

So some of our customers use it in their product organizations to track what are the unintended side effects of the various experiments I'm launching. Because there, again, you have something that operates on a daily timescale where human beings can interpret and then kind of take action on them.

Customer support tickets is another kind of great example where you might not see the forest for the trees of just trying to tackle all these tickets that come at you. But once you have something kind of surfacing trends or trends and patterns within those trends, then that become something that you can act on at an organizational level.

Marketing is really another good example. You can imagine you have thousands of campaigns. A lot of these are now actually run automatically on your behalf and it's kind of hard to keep track of exactly how every single of these companies are doing and all of the various regions or all the various demographics that they're firing against. Or if there is brand-new trends that are coming up within your organic search data, let's say, like traffic coming to you straight from Google, and could you lean into those as new campaigns? These are all opportunities that Outlier is really well designed to surface.

[00:10:10] MK: Now, marketing is something that some of people in the audience might groan at and think this is not an interesting problem at all. In creating Software Engineering Daily, I've had to understand how marketing works at a lot of different companies, because my entire business is built around advertising. So I have had to get a sense for how people think about advertising, and there're a lot of hard problems.

So there're different flavors of advertising, different flavors of marketing. Marketing is core to almost every business and the thing about marketing is if you have a place where you can spend X-dollars and make X+1 dollars, you will always want to spend X-dollars until you can no longer make X+1, and its finding those X leads to X+1 dollars because you can make sales that equate to X+1, is completely a mathematical problem and it's a problem of running these small experiments. Because there's so many different channels and because they have different ways of converting with accuracy, it's a complex scientific problem.

You have these different things that are very easy to quantify, like Google Ads and Facebook Ads. These things are – Where you have a much tighter control of you run an experiment and then you can measure the entire funnel, as it's called, you can measure the entire like soup to nuts. I showed an ad to this person, they clicked on it and then they actually purchased something, and I spent $5 running the ads. I made $10. Let's run that experiment a little bit more. Very straightforward.

On the other hand, you have something like billboard ads, which is very hard to quantify. If you put up a billboard on the 101, who knows if that's driving traffic to your website? Maybe you put a promo URL, twilio.com/billboard and you get some promo from that and you can measure that. But the point I'm trying to make is that there are significant problems around attributing value to marketing.

Describe some of the problems that companies face around marketing data.

[00:12:19] MK: Yeah, multichannel attribution that you are describing is a super, super hard problem. I went back at my days at Google, I had coworkers who would describe this problem to me and I just remember scratching my head thinking, "That is just ridiculously hard."

I think that level of problem I think is almost strategic, like trying to figure out how do I optimize these various campaigns I'm running. I think Outlier's use case a lot of times in marketing data is actually quite tactical. We can tell you exactly which campaigns are no longer performing the way they used to.

Basically, the inflection point between when this thing used to make me X+1 dollars is now making me X-1 dollars, we should turn this thing off. A lot of times that's not obvious and is not obvious depending on the number of different kinds of campaigns you're running and all the places that they're running. So Outliers is kind of another safety net for humans who are operating these campaigns sometimes, or at the fly by wire steering wheel as you will of these massive campaigns to kind of help them co-pilot this.

I think actually just beyond the challenges, I think there are some really interesting opportunities that are hiding in data and Outlier can surface those as well. We had a fantastic case study with one of our customers where February of two years ago when we had that record-breaking heat February, they saw unseasonably early interest in a lot of their spring catalog.

Who's shopping for spring clothes and spring accessories in February? But we were able to surface for them, "Hey! You are actually getting a lot of organic interest in this category," and they were able to take that insight which they would've missed, because who routinely scans all the keywords that are being used at their websites? And they retargeted all their email in March around, "Hey, spring is here early. Come get your spring deals." And doing that they're able to make massive shifts in their sales in March.

So this is one of those really nice cases where it's like, "Oh, you can draw a very straight line from the data surface, something interesting to us that we would've missed, but was highlighted to us by Outlier, and we leaned into that and we had a huge impact year-over-year, March versus now."

We also know that they hadn't run that, that they probably would have missed that on all those sales.

[00:14:22] JM: The tooling around data science and data engineering and data-driven X, whether X is marketing, or sales, or whatever aspect of your organization you're trying to drive decision-making and change within the organization through data, the change in the tools seems to continually change how teams are structured, because if you have a tool that does the work of a certain type of business analyst and the business analyst gets either leveled up, or gets obviated, or can move on to doing something else in the organization, but you have this ever-changing structure of data-driven teams.

How are data-driven marketing teams organize today?

[00:15:10] MK: That’s a great question. One that I actually don't have terribly much insight into, but I can speak to the general question a little bit better, which is we've definitely noticed that exact same phenomena happening with Outlier. When teams deploy Outlier, they often can do so without having to hire another headcount.

One of our larger customers was about to hire a human being, an analyst, to go do exactly what Outlier now does for them, right?

I think that broader point of data teams shifting and changing composition, roles and responsibilities in the face of technology change is absolutely a really great observation and one that Outlier is also contributing to.

I like the fact that you mentioned that sometimes you get up leveled, because a lot of these analysts are now no longer having to be at the mercy of these fire drills, right? Where Jeff might come to you and say, “Hey, all our campaigns are going whacky. What's going on, Mike?”

And suddenly my day is ruined because I’ve got to go dig through wherever the data lives and try to figure out what's going on. Instead, we can now try to get ahead of problems by saying, “Hey, look! Outlier’s surface these five potential issues or these three issues and two potential opportunities. What do we think of them?”

So suddenly the analyst is not caught always on the back foot, but now is able to be proactive and think through, “Hey, what does this opportunity mean for us?Hey, what does this organic interest in our spring line mean to us? What can we do about it?”

And the creative act of how do we intervene for the better or for the worse to either remediate a situation or take advantage of an opportunity, I think that's that creative work that humans are so great at that we would love to empower.

I love the phrase you used, up-level analyst to that, instead of being at the whim of all these tools and trying to go find the needles in the haystacks with an ever more powerful and more interesting tools, right?

[SPONSOR MESSAGE]

[00:17:00] JM: You listen to this podcast to raise your skills. You’re getting exposure to new technologies and becoming a better engineer because of it. Your job should reward you for being a constant learner, and Hired helps you find your dream job.

Hired makes finding a new job easy. On Hired, companies request interviews from software engineers with upfront offers of salary and equity so that you don't waste your time with a company that is not going to value your time. Hired makes finding a job efficient and they work with more than 6,000 companies, from start-ups to large public companies.

Go to hired.com/sedaily and get $600 free if you find a job through Hired. Normally, you get $300 for finding a job through Hired, but if you use our link, hired.com/sedaily, you get $600 plus you’re supporting SE Daily. To get that $600 signing bonus upon finding a job, go to hired.com/sedaily.

Hired saves you time and it helps you find the job of your dreams. It's completely free, and also if you're not looking for a job but you know someone who is, you can refer them to Hired and get a $1,337 bonus. You can go to hired.com/sedaily and click “Refer a Friend”.

Thanks to hired for sponsoring Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:18:43] JM: We do think about the relationship between the analyst and the data engineering team, where the team that owns the Hadoop cluster, there is a story arc to how that has changed. The relationship maybe 6 or 10 years ago, or even still today in many places, the relationship is, I'm an analyst, I need a new query against my Hadoop cluster. I walk over to the data engineer and I say, “Hey, can you get me a report on this thing?” And they say, “Yeah. I'll run it tonight and you'll get it tomorrow,” and that's fantastic compared to what we had 20 years ago. It's not great by today's standards.

So you have that relationship, the relationship where people wait in line to get their big Hadoop cluster queries answered to the tools like Looker, or I think Periscope Data, that kind of get – it's like a flexible empowering tool that a business analyst can potentially use to access a Red Shift cluster in a way that's a little more friendly. Then they get data given to them - it becomes more of a self-served data thing.

But then the Outlier model is a step further, where you don't even have the requirement for an analyst to know what query to ask. I mean, obviously I think we can humbly say that Outlier probably doesn't answer all the queries that you would potentially want. It's not a mind reading machine, but it certainly will surface some of the more obvious things that people might be running.

You've been in the data engineering industry for a pretty long time. Can you describe how you've seen this arc of multiple people having to answer a query, to one person answering a query, to potentially the machine insights just being delivered to you?

[00:20:36] MK: Yeah, I'd love to speak on this. I can actually speak all the way back to – I remember my very first job out of grad school. I get there. I'm a plucky – before things were even called data scientists. So whatever the term was that we used before that, and my very first thing I do on the job was go instrument everything and go dump it into a database so that we could eventually go query it.

Very much - not even being able to go to the data engineering team, like I had to go be the data engineering team and then go answer that question, right? Then eventually, I think there's a great book by [inaudible 00:21:07] and who is his co-author on that.

[00:21:10] JM: *The Looker Guy*.

[00:21:10] MK: *The Looker Guys*, yeah. They described the data bread lines, right? Where you have to get in line for your data.

[00:21:15] JM: That’s actually exactly what I was thinking when I was reciting that question to you.

[00:21:19] MK: Yeah, and it’s a vivid image of exactly these cues of people. Quite frankly, these data engineers have probably more important things to do than answer your specific query, right?

So I love the democratization of data, the stepping forward, helping everybody answer questions. I think the continuation of that arc really is what brought us to Outlier, which is all the tooling we have today is really about – pre-outlier, is really about better answers, right?

Where better can be defined as faster, or deeper, or more faceted, or more openly accessible to everyone in your organization. These are all better answers, right? You can get answers in all of these fantastic better ways.

The real twist that we kind of put on this is, that's great, but what if I don't know what to ask? The real insight there actually came from my co-founder, Sean. Sean was the cofounder Flurry, and he’d go on all the sales calls, like a good salesperson. At the end of your integration, you deliver the product and the last question you ask on the way out the door is, “Jeff, do you have any questions? Any last questions before we go?” Because that's how you wrap up all the sessions?

To the last customer, the question was always, “This is great. What do I look for?” At the time, that seemed like, “Why are you asking? I just gave you the keys to the kingdom.” You can go ask anything you want. What do you mean what should you go look for?”

It wasn't until a little bit of reflection and some time off that Sean realized just how profound and deep of a question that is and how profound of a need that is, that it’s never going to go away by giving people the ability to ask better questions. Ask questions better. It doesn't matter if I've got a Looker instance or a Tableau dashboard or Periscope, I'm still going to want to know what should I go use this amazing query answering thing for. That's really the that kernel of Outlier, is can we help people ask better questions.

A lot of what I think about in the terms of going back to your original question about the evolution between - where we've been 10, 15 years ago today, a lot of like when I think about my own personal journey, it's developing the experience to ask better questions. When something bad happens, like, “Oh, wait. I bet this is in this part of the organization. I bet this problem is in this part of the code base,” or, “This seems like a monitoring problem. Maybe I should go see if the monitoring is still up.”

That's all intuition and experience, and quite frankly bias, that as humans we built in and learned. By the way, not all bias is bad. Bias sometimes help us quickly narrow down where a solution space might be. But sometimes our biases lead us to flat out miss solutions. So one of our customers really had the situation where they spent six weeks looking for the answer and is right under their noses the entire time, but it was just in this ignored corner of the organization that nobody thought of. Nobody thought could possibly be related to a problem as big as this. But sure enough, that was exactly what their problem was.

When you turn those decisions over to the machine, and this is one of the great parts of introducing machines into your decision-making processes is you can hopefully eliminate a lot of those hidden blind spots, because the machine doesn't know that that was an important. It looks statistically important. I will surface it to you, and human being, you can quickly triage it and decide whether or not this is important.

[00:24:30] JM: Again, this Outlier product, it's creating stories by looking at my data and detecting outliers, or detecting anomalies, or detecting trends, or detecting whatever.

So let's say I'm selling t-shirts. I've got some ads that run across different channels. I've got some email marketing campaigns. I've got a lot of different marketing things that are going on. I wake up in the morning and there's an email from Outlier that has a story in it. What kind of story would Outlier give to me?

[00:25:05] MK: So we actually tell something upwards of 13 or 14 different kinds of stories. One of my favorite one, just because you keep bringing up the Outlier thing, is we'll even tell you a story of when your top line metric is not anomalous, but should be. So let's say let's use that example for this. Let's say you get an email this morning in your t-shirt business. You might get the story that says, "Jeff, your t-shirt sales are growing at the rate that is always been growing at, let's say, 4% growth, which is fantastic for your mature t-shirt business, but did you know underneath the hood the following segmentation of customers are actually growing at 6% and this other group is actually falling off a cliff. The only reason you're growing it 4% is because of the mixture of these two groups."

So suddenly your dashboard on the wall might look totally normal. In fact, it's going up into right on the exact same slope that is always been going on, and Outlier is now showing you that it could be growing faster if you addressed what's going on in these subpopulations, where one is growing faster and this other one seems to be having problems.

A very similar thing happened with one of our actual customers where I believe in this case it was device type. They had shipped a version of their app to the App Store, which broke their app on older versions of iPhones.

So the rest of the users actually increased their engagement, but their older legacy iPhone users were actually falling off a cliff, like literal cliff, because their app was bricked. They bricked their own app. And it would've taken them a long time to eventually surface these support tickets and eventually prioritize them to realize, "Oh, yeah. This small but previously active portion of our user base has gone zero," because the uptick in engagement from the rest their user base actually hid the fact that they broke this small fraction.

What Outlier will do is immediately identify that the small fraction had gone off the rails even though, again, the top-level metric did not change. We have another story like that where an e-commerce company had done an A-B test. They thought it looked great. They launched it, but they broke that all-important people will buy coffee also by creamer and sugar widget. The extra widget that accounts for 30% of their sales, and they only broke it on this specific page that they’re doing their A-B test on. So it worked out to about 4% of their revenue that they were missing.

Because of revenue’s highly stochastic nature and it’s got a lot of ups and downs, it would've probably taken them weeks to notice that they’d been slipping and losing and bleeding 4% of revenue every single day. Of course, even on their dashboards, nothing looked unusual immediately, but literally the day after they pushed the change, Outlier told them a story saying, “Hey, you previously used to make this amount of revenue off this widget off these pages, and now you're making zero.”

They were able to spin into action and quickly fix that and they recovered all that lost revenue before it turned into this fire drill where six weeks later someone’s going to run to my desk and ask, “Why is revenue down 5%?” Then I've got to go figure out why. Hopefully, I'll have the intuition to be able to figure out quickly why that's the case.

[00:27:56] JM: We should dissect how this happens. So I think there’re a couple different areas of this conversation we can have. Like, first of all, if a customer wants to start using the Outlier product, they have to be able to feed their datasets into Outlier.

Then second of all, you have to run jobs, machine learning jobs or just data science jobs against those datasets that the customers are plugging in. I imagine there's some phase in between where you either have an account manager talk to the customer and say, “Hey, what are the things that are actually important to you?” Or maybe there's a configuration that the customer does themselves.

But let's just start with the integration. So there's all these different things that emit data. So we've got MailChimp data around who's clicking on emails that we’re sending out from our T-shirt company. We've got Google Ads that were running that are emitting some kind of analytic data back to us and we want to plug that in. We've got Facebook Ad data. We’ve got all these different kinds of data. Maybe potentially log data if we’re talking about surfacing insights for engineers.

How do you standardize that integration process where you have all these different pieces of software that you need on the input side that people need to integrate into the Outlier platform? What's the process of writing an integration, of presenting the integration to the end customer that's trying to plug in to Outlier?

[00:29:32] MK: Yeah, this is something we gave a lot of thought to, and actually I'm going to reference an earlier question that you had about taking things to a data engineer. We noticed that a lot of traditional BI SaaS products actually had this really horrible step between you convince somebody to buy it. Do they actually get value out of it? That stretch is usually sometimes weeks, sometimes months long where the integration team has to come on-site or work with your immigration team and data teams.

So you imagine you're a poor data engineer named Mike and your CIO made this decision probably not even consulting you, and the next thing you know you have this quarter long project to get up and running on a new fancy tool, right? Whatever that tool new fancy tool is. We recognized this as kind of an anti-pattern and like the BI SaaS kind of industry.

So Sean and I thought long and hard about what can we do to make that integration process as painless as humanly possible? So we made a lot of thought and put a lot of design into this. So for a lot of our cloud services, integration is simply as easy as go find your API keys and your secret tokens or your passwords. Plug them in and press go. That's literally all you do. There's nothing else to do in terms of integration.

I guess a little more complicated with SQL databases, because obviously we need to know which tables you want to integrate, or which columns and which tables you want to integrate. This is where in our onboarding process we actually work very closely with customer teams to figure out what are the data sources that you want to plug in and what are you hoping to answer from these things?

I think this process is something that obviously we've learned and grown into, but we discovered is really valuable to making sure that this Outlier experience is something that's going to be really valuable for them.

As you mentioned, there's obviously personalization steps that we can also take. The standardization format or what the crux, the funnel or the lens that we shove everything through, is we transform everything that we can into a multidimensional time series, and that's kind of the lingua franca that we kind of make everything inside of Outlier speak.

So once you've integrated data from whether it's a cloud data source or a SQL database or wherever it's coming from, once it's on our side of the firewall, it's turned into a Outlier standardized time series. Which from that time series object, we know all the data about it. We know its relationship to other time series inside your dataset, and then that's the starting point of all of our algorithms that we run.

[00:31:49] JM: Is that your own custom time series database?

[00:31:51] MK: Actually, we just built this on top of Postgres. There wasn't a lot to really kind of think through a new one. We did evaluate several of the other open source time series databases, but at our size and scale, we decided that that was probably not worth it and we went with something that was easily understandable and could be extensible and probably scale for us for the foreseeable future.

[00:32:10] JM: You're storing all of this data in Postgres database instances. Are you flushing it on a regular basis or do you just like store all the data?

[00:32:21] MK: Yeah. We actually do store all of the aggregate data, and this is actually a really key difference. So you can imagine, let's say, you pointed me at your t-shirt revenue database. Let's say you have a Postgres table on your side, or let's say it's a MySQL database or whatever flavor of SQL you happen to be speaking.

You have this row for every single customer that came to your online e-commerce portal and bought all the various t-shirts. You can imagine this might have like the date of the transaction, the amount of the transaction, any promotions that it might have, the sources they came from. You can imagine that each one of these rows might have a lot of different columns and a lot of different information about every single sale you've ever made.

Part of our ingestion process does, basically, aggregations on that table and what we output or what we input into Outlier outputs from your data systems and into ours is just the aggregate. So I won't actually ever know that Jeff bought a men's t-shirt from the new spring line. All I'll know is that men – there were, let's say, 322 men from California who bought this t-shirt from this line.

In this way, we sidestep a lot of the PII ingestion problem. There's also another kind and nice quick side benefit that we get out of this, which is since we're not actually storing raw event level data, our data storage cost actually are roughly logarithmic to the size of data that we're ingesting.

Let's say you're a massive, massive e-commerce company. You might be doing millions and millions of new rows a day, but every single day I just add a single new data point per dimension that I've sliced, right?

So I don't care if you do 10 transactions a day or 10 million transactions a day, because on my side I'm just going to store one new value. Let's just take, for example, sales in California. For sales in California, I literally don't care how many rows of sales you did in California, because on my end that's just yet another single row for, "Oh, Jeff's t-shirt store did 30 sales in California, whereas name brand giant e-commerce company did 30 million in California."

To me, that's just still one value in a single row on our side.

[00:34:24] JM: So you copy the entire database or the database for a set of days or a set of months or something and on a periodic basis you are aggregating that data into something like revenue by day, or revenue by month, or revenue by sector, or whatever, and then you store that aggregation in your Postgres database. Is that right?

[00:34:46] MK: The last half of that is right. We actually never make copies or do anything of their database. Their data stays inside their cluster. Never leaves their firewalled protective place. We're very, very adamant about this. All row level data stays on their side. The only thing that leaves their cluster is the aggregate result.

So for a SQL database example, we'll connect to them. Typically, most customers ask us the middle of night. So we don’t have a problem doing this. So middle of night their time. We’ll connect to the database. We’ll run a aggregation queries and we’ll only pull back the aggregate result and store that on to our side.

[00:35:21] JM: Okay. I see. That’s cool. So it's basically the same if you're hitting a MailChimp API or hitting a –

[00:35:30] MK: Bingo!

[00:35:29] JM: You just don't have to –

[00:35:31] MK: Our storage cost are fractional compared to the storage costs of our customers. Yeah.

[00:35:37] JM: Fascinating.

[00:35:37] MK: On a per customer basis, obviously. Once you aggregate all the customers, we also have a lot of data. But on a customer to us basis, yeah, we store a minuscule fraction of what they actually have.

[00:35:48] JM: So just to make sure I understand this right, you basically offload the bandwidth and the resource consumption of querying or data engineering or I guess queries aren’t even that – probably not even that data intensive.

[00:36:02] MK: No. These are really simple, like sum of this by this. These aren’t like really crazy analytics queries. All the crazy analytics happens once we’ve got those aggregates and on those aggregates, and [inaudible 00:36:13].

[00:36:15] JM: Okay. I see. Okay. So let’s say you get all these aggregations, on a nightly basis, you’re running a job, you're finding my sales within California or something like that, and now you've got a hundred – let's say after I've run Outlier for 100 nights, I've got 100 records of sales in California.

[00:36:34] MK: That's the beauty of it. You have records that go back years. So when you plug us in, we'll immediately populate records going back years, right? Because I can ask what were your sales yesterday? The day before that? The day before that? Last year? And I can just populate all of that.

Out of the gate, you don't have to wait for Outlier to catch up to you. Out of the gate, Outlier will already have a sense of what kind of seasonality does Jeff's t-shirt store have? What kind of cyclicality does it represent? What kind of things are normal? What are not normal?

So long as you have the data and we can access it in a reasonably efficient way that's not going to take down your database cluster, we can populate things back years. A lot of our larger customers, in fact, that's required for us to really better model their seasonal effects of the year on year effects.

[00:37:16] JM: Okay. So I get that you would want to run the nightly queries, for example, like the previous day's sales. If you want to surface an aggregation for a business analyst tomorrow morning, then you want to run the query on yesterday's data or maybe even yesterday's data and like the early morning hours of today, for example.

But these aggregations that are over the - analysis over the aggregations that you've collected over the last – from all the customers data over the last hundred years or whatever, these could be more complicated queries and it's less clear when you would want to schedule these jobs.

I'm guessing there's some time constraint, but you have a lot of flexibility. For those, that's a query that's running on your infrastructure. Can you help me understand how those jobs get spun up and spun down? Give me the lifecycle of a job that is getting executed on your infrastructure.

[00:38:11] MK: Sure, and I'll backup one step, which is after we – Let's say you're a new customers of our's. After we do all the onboarding, you don't actually immediately – especially if you're a large customer with years and years and years of data. That's going to take a little bit of time to ingest. We're not going to want to take down your production systems while we ingest all your data, because you can imagine that that could slow things down for mission-critical apps on your side.

In that case, what might happen is we might schedule a call. You might give us your credentials. You might talk about what's important to you. Then our team will go and figure out how to best, within the resources that you've given us, issue all the queries we need to populate years and years and years of data. So that's one clarification, right?

Then that's done in a similar way to - you can imagine that we just rewind the clock back three years and just start running nightly collections from three days ago all the way to today. Naïvely, you can assume that that's what's going on.

We do some clever batching to make it more efficient, but in a nutshell, you can think of that as the process, right? So however long it takes us to play forward the number of years of data that we agreed upon, that's going to be happening over however many resources you give us.

One of our customers, for example, said, "You can only run queries between 1 AM and 3 AM our time." So, great. We can run, say, 90 days at a time in that window. If you want us to go back two years, that's going to take us however many times divided by ninety, right? So it'll take us, I'll say, eight days to get all the way back, right? So we'll say, "All right. We'll work on it and we'll talk to you in a week," and that's what we might agree upon, right?

So to get to your question, once everything is on our side, so now everything is in the Outlier standardized time series format. So what this means to us is that, like I said, we have a metadata about the time series. So we know, "Oh, this is a time series about revenue, or it's a time series about this, or about that."

Then we can build a graph of how this time series is related to all the other time series in our database about you. On a level of each individual time series, that's time series modeling. This is again a very well worked out field that we've borrowed heavily from existing knowledge that we have. Whether this is ARIMA models or a Bayesian structured time series models.

There's a fantastic paper from actually a number of groups, but the one I'm thinking of is from Google Research on how they do time series modeling and combine various models that they do time series modeling on. We can draw inspiration from a number of those things. The trick that we have though is we have to do this in a fully automated fashion, because a lot of traditional time series modeling actually involves a human to kind of like tweak parameters, or even if you use a machine to tweak parameters, you still have a human kind of evaluate per time series, “Oh, is this a good model? Is this not so good of a model?” and then tweak things, or maybe apply combination of models and do very clever things to really get good models.

So the one disadvantage we’re at is because of the scale that we’re operating at, all of our time series modeling has to happen fully automated. So that happens. One of the actually really cool algorithms that we have that I like to brag about is we have the ability to do the all by all comparison across a sliding window of time and figure out which of these time series that you have are correlated together or used to be correlated together and stop being correlated together, or previously uncorrelated that started correlating.

One of the examples I'd like to get is we have these examples of, let's say, you have checked out data and you have warehouse inventory data, and you can imagine that your sale of in-stock and your sales is probably strongly correlated, right? Every time you sell something, something ships in the warehouse, right? So you'd hope that those things were very correlated.

So let's say one day your warehouse inventory system goes off-line. Well then suddenly Outlier will tell you this story that, “Jeff, these things used to be totally perfectly correlated. Now they’re not correlated anymore. There's something wrong. You should probably go take a look.”

This is an example of things that’s not even an anomaly, right? You can imagine that, let’s say, you had a number warehouses and only one kind of went off-line because it was flaky. So that might actually be within any statistical models, norms and lanes that it might've drawn for you, but there's actually something wrong because the relationship it has with other metrics is broken.

So that's a different kind of outlier. It’s not an outlier of the time series value in and of itself, but it's an outlier in the sense that the relationship it has with others is broken and that's something we can also highlight for you automatically.

Or in another example, let’s say two things didn't used to be correlated at all, but suddenly are correlated, right? What happened here? What kind of behavior shift happened that these two groups of people who previously were unrelated at all are suddenly acting in sync? Actually, in that case, it might be fraud or that might be an indication of bot traffic, or you can imagine any number of other things that could be causing things that were previously uncorrelated that suddenly become correlated.

In any case that’s a good start for an investigation. It’s like, “Why, why did that happen? Let's go look into it. Let’s maybe fire this up in Looker and try to dive deeper into this.”

[00:42:52] JM: Yeah. So that's what's cool, is if you can surface – so if you can surface that there is some kind of link between dataset X and an increase in bot traffic or an increase in fraud detection, something like that, in one organization, then you would probably want to surface that for similar organizations.

If I understand you correctly, you have to do some sort of manual labeling for these kinds of datasets in order to make them correlated in order to surface useful correlations, because I don't think we’re at the point where we can just sort of let loose the machines and let them find the right insights.

There is still a great deal of sort of saying like, “Hey, these - warehouse inventory tends to be correlated with sales in California,” or something like that. You still have to do to some degree some labeling and maybe you have to prove that it works in one organization, then you can apply that insight to another organization.

I’m having trouble articulating a question here, but it's just an observation, that if you just sort of let loose the machines on the datasets in a naïve way, they would probably surface insights that would frequently be less useful and then you would just be getting the story on a daily basis and you're like, “Okay. The machine learning system is detecting that there may be an increase in fraud detection because our warehouse inventory is low.”

Well, that seems like kind of a spurious correlation. That’s probably not useful, and then you start discounting what Outlier is feeding you every day. I guess the question is how do you make sure that these are useful insights?

[00:44:32] MK: That’s a great question. I actually this the piracy and global warning problem, right. It's like, “Oh! Did you know that there was an increase in piracy along with global temperature spikes?”

[00:44:39] JM: Exactly.

[00:44:39] MK: Oh, great. Clearly, it’s pirates causing global warming. Whereas actually it’s probably the other way around, right? It’s global warming taking away these people's livelihood and forcing them into piracy.

But regardless, there’s this unmistakable correlation between an increase in piracy and global temperatures. It’s like, “Well, that's true, but what really is going on here?”

You do highlight on another really interesting kind of aspect of what we're doing here at Outlier, which is we do have a feedback mechanism. Once we present these stories in a feed, we rely on human feedback. We rely on people to either say, “Hey, this is super interesting. I want to start with Jeff.” Oh, and then Jeff shares it with everyone else in his organization. People are commenting on it. People are engaging with it, interacting with it, issuing sub-queries off of it.”

Well, this is clearly something that's hit a nerve, right? So then the system learns that things like this, stories like this might be worth surfacing again. We have obviously the opposite kind of feedback, the equivalent of the thumbs down, right? We have the hide button, which is like never show me this again. Then we also have the inexplicit negative feedback actions of like, “You know what? Actually, I don't care about sales in New Mexico. I’m a California brand and I only sell in California. So just don't ever show me things about New Mexico.”

So in all of these different kinds of feedback, the system is layering in ways to learn about businesses. I think one of the other things that we've done is now that we've been around and building models of this for a number of years, we actually have a good sense for what stories out of the box tend to be interesting versus not.

Generally, even out of the box without a whole lot of personalization, we tend to be closer to the - like what people find interesting rather than farther, and then over time the models obviously learn and improve on your feedback and the feedback from your organization.

[SPONSOR MESSAGE]

[INTERVIEW CONTINUED]

[00:47:52] JM: This reminds me of another company I've have done some interviews with, which is Dremio, and they have this thing where – so Dremio is this data engineering platform and one of the features of it is that they figure out what queries to cace with - your entire data platform and they speed up queries by caching certain inquiry results and they materialize those views more aggressively so that the query time is faster.

So the way that the queries get cached is often times you have different teams in their organization that are running very similar queries and so you can cache materialized views based on who is up-voting certain queries. It’s literally the process of up-voting. A certain team says, “I want this query faster, so I up-vote it.”

I'm hearing the same thing in Outlier where you basically have so much data across an organization and so many things that could be generated from that data that you require people in the organization to do labeling. It's almost like the – it’s like an internal mechanical Turk labeling system.

[00:49:03] MK: Of what’s interesting. Because there isn't a mathematical definition of interesting, and even if we had one, it wouldn't be the same for every organization and even within an organization. So we do need the human input. So we can get close, but if you really want this to sing, we really need that human partnership.

That goes back to my earlier point about. This is about helping people be better at their jobs by helping them building a machine that will help them do that and give them superpowers.

[00:49:30] JM: The only piece of infrastructure we’ve really talked about is the fact that you use Postgres. Can you give me more of an overview of the infrastructure?

[00:49:38] MK: We’re actually pretty vanilla shop. We’re running on AWS, EC2 and we use RDS, and we built a lot of the things on top of those pieces of infrastructure, but there's nothing really glamorous underneath the hood. Most of the secret sauce is really just in the algorithms and the design of how we put things together.

[00:49:56] JM: Do you have any queuing systems or a workflow scheduler? Anything like that?

[00:50:01] MK: We use Kafka for queuing and messaging back and forth, but, again, no groundbreaking, earth shattering secrets to share unfortunately on that front.

[00:50:10] JM: Any workflow scheduler? I think Airflow is a common workflow scheduler in this kind of thing.

[00:50:17] MK: Yeah. I think in that case we did actually roll our own and that was more of a expediency thing and didn't know better thing than it was like a well-thought out decision.

[00:50:26] JM: People use Airflow. People use a workflow scheduler when there are, I think the number of – well, actually I shouldn't talk on this too much, but I think it has something to do with the fact that if you're in a resource-constrained environment and the number of jobs that want to be run can sometimes outstrip the number of resources you have, or like if you're Netflix, you have all these jobs that want to be run.

So you let the jobs specify their priority and so that the ones with a higher priority or the lower priority depending on your view of priority, get run more aggressively and then the other ones have to wait. But I guess do you just not have that level of workload?

[00:51:05] JM: That would be a fantastic problem to have for us.

[00:51:05] JM: A good problem to have. Yeah.

[00:51:08] MK: I look forward to having problems that I need Airflow to solve.

[00:51:10] JM: Yeah. Yeah, it makes sense.

So no reasons to go multi-cloud, no reasons to start using Big Query, anything like that?

[00:51:18] MK: Like I said, a lot of the secret sauce really is just in the core idea of like what happens if we give people questions instead of answers? Then the algorithms and the software design into making that experience as seamless and as useful as possible.

[00:51:34] JM: What are the hardest engineering problems you have today?

[00:51:36] MK: We have several. The ingestion of data, while we kind of glossed over it and made it seem like it's magical, is always a bear. There are all kinds of issues dealing with cloud providers.

One of our favorites is a lot of cloud providers will have sample data that is returned to you. This is fine if you're doing one-off craze that you kind of need approximate answers for, but if you're trying to surface analytics and insights and you have an insight that's based on a sampling error, this obviously becomes quite a headache where like, “Wait. How come your numbers don't match our numbers?”

And you can imagine that depending on this exact way that we issue our queries versus the way that someone issues their query on their end, you can end up with different results. That, of course like you're saying, like kind of erodes confidence, right? Dealing with how to work with various cloud providers is one trickiness.

Anytime you’re dealing with somebody else’s schema – I made this joke at a different talk I gave. I think every data architect are like drivers. If you ask a bunch of drivers, “Are you above average?” They all are going to answer they’re above average, and that mathematically can't be true.

You ask all database architects, “Do you write clean schemas or better than average schemas?” Everyone is going to answer, “Of course. My scheme is better-than-average.” That can’t be true either. So we’ve obviously come across, for lack of a better word, let’s just say interesting schema choices, and then having to integrate and work across those has been an interesting challenge.

We’re trying to build a tool that's as simple as possible to use and generalizable as possible. Of course, that flies in the face of this exact – every unique butterfly of a snowflake of a schema.

[00:53:04] JM: Right.

[00:53:04] MK: So that's definitely one challenge. And on top of that, we have all the classic time zone problems and all – because we have customers all over the world now, and so we’re dealing with time zones and dealing with all those things is super fun.

[00:53:16] JM: Yeah. My scheduled 6 AM job is not the same as somebody in Uzbekistan’s scheduled 6 AM job.

[00:53:24] MK: Exactly.

[00:53:25] JM: And the schema thing, my definition of warehouse inventory is not the same definition as the Uzbekistan t-shirt company's definition of warehouse inventory.

[00:53:37] MK: Exactly. So one of the ones that bid us recently was surely 255 characters is plenty for naming a column, and we'd be plenty safe to prepend this outlier-specific tags.

Because who could possibly use that many characters in a column name? Of course, someone found a way to use all 255 characters, and then we would break – our queries would break. We can create longer column names than that, right?

There are all these little gotchas where you make assumptions based on what you think is a reasonable view of the world, and it turns out that there are perfectly reasonable, other ways to do things that you have to accommodate.

[00:54:11] JM: So there seem like a lot of opportunities for growth in the business, and it seems like you could have a pretty straightforward – well, there's a lot of different onboarding strategies you could use.

You could have freemium model, but freemium would be tough because you still need some human in the loop, some account management in the loop in terms of getting people started.

How are you thinking about like customer onboarding? Pricing? Those kinds of questions, the sort of go-to-market customer by customer question.

[00:54:45] MK: Yeah, we did a lot of early experiments on this and we discovered kind of like what our ideal customer profile would look like. At least in our early phase that we're in now, and it's is really funny, because discover things like, "Hey, despite the fact that any business might have this need, it turned out that you had to be a business of a certain size to not only have enough data and make this really worth your time, but have the resources to do something about it once I told you.

Because let’s say I told you, “Jeff, your California sales are slipping 4%.” If you're a one-man t-shirt operation and everything else is on fire, that's nice Outlier, but I can't do anything about that, right?

On the other hand, if I told a very large e-commerce realtor that a state in the United States were slipping 4%, you could be sure an analyst was going to get assigned to that immediately and go fix that, right?

So there is a matter of scale where the organization that's receiving these insights actually has to be able to act on them. That was in interesting lesson that we learned in the early days of Outlier when we’re trying to just give this to anybody who would take it.

In terms of pricing, something we settled on the very early is we wanted to make sure our pricing aligned with the value that our customers are getting out of it. So we do not price by seat. We do not price by volume. We do not price by any number of these other metrics. What we price by is by integration, because what we have found is that as businesses integrate multiple different types of data, the value they get actually compounds.

You can imagine that if all I had was my web analytics data, okay that's nice, Google. That's nice, Outlier. I've got Google Analytics in here.

But the moment I’ve got that and MySQL revenue database in there and I can start seeing how relationships are forming between my unique user traffic and this other table. Now, suddenly Outlier has become way more useful than having either of the sources alone.

Then once I layer on all my customers for tickets on top of that, then a totally new insights surface between the combination of these things. So what we discovered is that when we price by integration, our incentives and our customers value are aligned really well. So that's kind of the model that we’ve stuck to.

[00:56:45] JM: Yeah, that does make a lot of sense, because it sound like there's just like there is still this work that’s associated with each integration that you have to do particular to these just custom SQL databases and you've got to contort yourself to the customer’s schema. So pricing by integration, I don't think I’ve heard of that one before. I'm sure it's out there though.

Okay. I think we've done a pretty good job. I want to wrap up, but we’ve done all these shows recently about different data engineering, data analytics, data science, business analyst views of the world and the ways that things are changing. What’s your perspective there?

How are the roles of business analysts and data scientists and data engineers changing, and how is it affecting the teams that you're seeing? Do you have any forecasts? Can you even wrap your mind around the process of data engineering and the different teams that are being built around that or is that too hard to generalize?

[00:57:45] MK: I actually have a slightly different take on this. I think the – yes, this is by way a massive philosophical question that you’ve lobbed my way in the last few minutes here. I look at it as there's this promise that we've all been sold that if we’re data driven, then surely we’re all just be living these better lives and running better businesses.

So based on this promise, we've invested millions upon millions of dollars, trained tons of people and built great fantastic tools and infrastructure to do this. One of my questions I wondered is if we don't start getting real return on this, when is this house of cards eventually going to come crumbling down? Because, if it weren’t for the ever falling price of storage, at some point some CIO is going to be like, “Hey, I'm literally – My Red Shift bill is what? Wait, why am I storing all of these?”

At some point somebody is going to wake up and do this sacrilegious thing and be like, “The emperor has no clothes. We are spending millions and millions of dollars on this and we’re not getting anything out of it. If it weren’t for, like I said, the ever falling price of data storage and these ever new, shinier, better tools that keep pulling out more promise, then we might already be there.

So Outlier, we look at it as a synergistic piece to the rest of the ecosystem. Where it’s just like if you had something that could surface insight from all of the mounds and mounds and mounds of data you’re storing, you wouldn't mind storing it anymore.

If you had something that you knew what to point your fancy visualizations at, you would love having those visualizations tools. If you had something that could give every analyst in your organization – by the way, that's the other one big piece of the future. Everybody is now an analyst, right? I don't care if you're in marketing or if you're in product or you're in sales. Everyone is now an analyst. Everyone is required to be able to speak to the data or speak from the data, and that makes us all analysts, right?

So suddenly the investment I made in making sure that everybody on my team can be an analyst, which is really lofty wonderful vision. Now they are empowered to go ask really insightful deep questions that they previously couldn't.

And I always like pitching my hero, Paul Meehl, and I don't know if you're familiar with him. He's the clinical psychologist who really pitched the idea of using mathematical models over human decision-makers. He had this landmark paper at the end of his career where you can hear the exasperation in his writing where he says like, "In no other field of human endeavor has a result been so repeatedly and unrefutably found than this results," and his result was that if you use an algorithm to make a decision or prediction, you're going to either match or outperform humans 90+ percent of the time, and it is like an overwhelming thing.

If I said, "Jeff, you're about to go make a prediction, and I can give you a tool that with 92+ percent of the time you will either be better than or match human experts." Wouldn't you take that tool every single time, right? Why wouldn't you?

So the exasperation of Paul Meehl comes through on this paper, but I really think that's where we're heading with decision-making, right? Decision-making really is about predicting which course of action you think is going to be best. The more we can start helping people make data-driven and less biased and cognitively sound decisions, I mean, this is just really going to help us transform organizations.

[01:00:50] JM: Okay Mike. Well, it's been great talking to you. Thanks for coming on Software Engineering Daily .

[01:00:53] MK: Thank you for having me.

[END OF INTERVIEW]

[01:00:58] JM: If you are building a product for software engineers or you are hiring software engineers, Software Engineering Daily is accepting sponsorships for 2018. Send me an email, jeff@softwareengineeringdaily.com if you're interested.

With 23,000 people listening Monday through Friday and the content being fairly selective for a technical listener, Software Engineering Daily is a great way to reach top engineers.

I know that the listeners of Software Engineering Daily are great engineers because I talk to them all the time. I hear from CTOs, CEOs, Directors of Engineering who listen to the show regularly. I also hear about many, newer, hungry software engineers who are looking to level up quickly and prove themselves.

To find out more about sponsoring the show, you can send me an email or tell your marketing director to send me an email, jeff@softwareengineeringdaily.com.

If you're a listener to the show, thank you so much for supporting it through your audienceship. That is quite enough, but if you're interested in taking your support of the show to the next level, then look at sponsoring the show through your company. Send me an email at jeff@softwareengineeringdaily.com. Thank you.

[END]