SDS PODCAST EPISODE 753: BLEND ANY PROGRAMMING LANGUAGES IN YOUR ML WORKFLOWS, WITH DR. GREG MICHAELSON

Show Notes: http://www.superdatascience.com/753

Jon Krohn: 00:00:00 This is episode number 753 with Dr. Greg Michaelson, co-founder of Zerve. Today's episode is brought to you by Oracle NetSuite business software, and by Prophets of AI, the leading agency for AI experts.

00:00:17 Welcome to the Super Data Science Podcast, the most listened-to podcast in the data science industry. Each week, we bring you inspiring people and ideas to help you build a successful career in data science. I'm your host, Jon Krohn. Thanks for joining me today. And now let's make the complex simple.

00:00:47 Welcome back to the Super Data Science Podcast. Today, we've got the super insightful, crisp and clear communicator Dr. Greg Michaelson on the show. Greg is a co-founder of Zerve, a platform that went live to the public this very day and that revolutionizes experimenting, collaborating on and productionizing data science. They just raised $3.8 million in pre-seed funding.

00:01:08 Previously, Greg spent seven years as DataRobot's chief customer officer and four years as senior director of Analytics and Research for Travelers Insurance. He was a Baptist pastor while he obtained his PhD in applied statistics from the University of Alabama. That perhaps explains some of the variance in how he's such a silver-toned communicator. Today's episode will appeal most to hands-on practitioners like data scientists, machine learning engineers and software developers, but it may also be of interest to anyone who wants to stay on top of the latest approaches to developing and deploying machine learning models.

00:01:42 In this episode, Greg details why his swish new Zerve IDE is so sorely needed, how their open source pypelines project uniquely generates Python code for automated machine learning, why AutoML is not suitable to most commercial use cases, why most commercial AI projects fail and how to ensure they succeed. And he filled us in on this straightforward way you can develop speaking skills as slick as his. All right, you ready for this eye-opening episode? Let's go.

00:02:14 Greg, welcome to the Super Data Science Podcast. This has been a long time coming. We've been planning this episode for over a year. Finally, it's happening. Great to have you here.

Greg Michaelson: 00:02:23 Yeah, it's great to be here.

Jon Krohn: 00:02:25 So where are you calling in from today?

Greg Michaelson: 00:02:27 I live in Elko, Nevada. It's in the northeastern corner of Nevada. It's about as far away from Vegas as you can get.

Jon Krohn: 00:02:35 Does that mean ... Northeastern, so what are the states that are like near there? Is it like Utah?

Greg Michaelson: 00:02:40 We're about three hours west of Salt Lake City. Elko is actually the gold mining capital of the US. So there's ...

Jon Krohn: 00:02:47 Really?

Greg Michaelson: 00:02:48 ... gold mining industry here.

Jon Krohn: 00:02:50 Cool, still today?

Greg Michaelson: 00:02:51 I'm not involved in the gold mining industry at all, but-

Jon Krohn: 00:02:53 All right, and so we know each other from ODSC West. We met in San Francisco, I guess relatively close to you and Elko there. And I'm pretty sure that that was the fall of 2021 or might have been the fall 2022. Actually '22 makes more sense.

Greg Michaelson: 00:03:13 Yeah, I think it was 2022. Yeah. It's a great event.

Jon Krohn: 00:03:15 That's a really good event. And you were speaking there, I think and-

Greg Michaelson: 00:03:21 Yeah, I joined Zerve and we were doing our first testing of the market, to see if our ideas about where the issues in the data science development space existed, if they resonated with the people we're talking to, really good reaction that really encouraged us and gave us some good direction for where we've been going over the last year. So-

Jon Krohn: 00:03:41 Nice. Yeah, I remember you guys had a booth as well. Though I was speaking to you before that, I think I saw Ben Taylor sitting with you guys. You might have been actually the very first conversation. If I am remembering everything correctly, it was like I was at the badge pickup station and I'd just walked in from the airport. And I saw Ben Taylor, whom I love. He's been on the show many times. I guess he goes by Jepson Taylor these days, but he was still Ben Taylor back in the fall of 2022. And yeah, sit down with him and met you. And yeah, we really hit it off, really enjoyed the conversation then and we've been planning this episode.

00:04:20 And the reason why we've timed it to today is because it's today, at the time of episode release, so not at the time of recording, but at the time of release, so the time that very first time that our listeners are hearing this in their ear balls is also the release of I guess the public beta of Zerva, that they can be signing up and trying it out.

Greg Michaelson: 00:04:40 Not the beta, the actual general release. So ...

Jon Krohn: 00:04:42 Oh, wow.

Greg Michaelson: 00:04:42 ... we're going public as of the 30th.

Jon Krohn: 00:04:46 Congrats. So after a long career in senior roles in companies like DataRobot, Travelers Insurance, you have now co-founded Zerve. You're one of the co-founders, and to give a little bit of a taste of what this is, it's a powerful IDE, so integrated development environment, and the key thing about it is different from other ideas that are already on the market is that it allows you to visually design data science workflows. So you can tell us more about that, but my understanding is that you aim to revolutionize the way data science and AI development are approached. So yeah.

Greg Michaelson: 00:05:21 Yeah, the crazy thing about the space is that ... I mean, data science is ubiquitous at this point. Everybody's talking about it. Every company knows they need to be investing in it. There's huge academic programs being created by the universities in the space. But nobody's really built a good place to actually develop those kinds of applications. So really you have two options when it comes to developing data science. You've got your notebook-type environments like Jupyter, that sort of thing. And notebooks are really, really great for exploratory interactive data analysis, right? You get inline outputs, you get instant feedback on what you're writing, so you can write some code, see some results and that's going to impact the code you write next. It's great for inline analysis.

00:06:08 The problem with notebooks is that they were developed by academics to be used as scratch pads in the classroom. So you can't productionize them, it's impossible. And so what all data scientists do is they start in a notebook and they'll get to a certain point where they go, "Okay, now we have to get serious," and then they move to an IDE like PyCharm or a Spyder or a VS Code or whatever it might be. And those are great for writing stable code, right? Those are super great products for developing, but they suck for data exploration. So like trying to get that interactive analysis that you can get in a notebook from an IDE is a horror show.

00:06:49 So you end up jumping environments from one to the other to the other and fighting with the tools to try and make something that works. I guess SAS is really the only company that tried to build an end-to-end-type environment that gave you that code-based interactive feedback but also was super stable. But of course, SAS stopped innovating back in 1992 or something. So that company's dead. So anyway, that's what we're trying to do. We're trying to build a data science development environment that gives you all the interactivity and exploratory capabilities of a notebook, but also gives you stability and persistence and all those things you need to actually take your projects and make them into something real.

Jon Krohn: 00:07:43 It's a critical gap. What inspired you specifically to launch it?

Greg Michaelson: 00:07:47 Well, so I was at DataRobot for seven years and that was a wild, wild journey. I think I was employee 30 or 40 or something like that at DataRobot. When I left, we had maybe 1,500-1,600 people, so it's crazy, crazy growth. And the thesis at DataRobot was that you don't need to write code to build data science, right? That automated machine learning was the future and you can do all this in a low-code or no-code environment. And certainly, the thing that I took away from that experience is that 100% of data science projects require code. You absolutely need code.

00:08:24 And so if you look at the product roadmaps for Dataiku and DataRobot and Databricks and all the all the companies with data in them, they're all integrating notebooks into their environments because they realize you can't do this work without writing code. And the only people that are actually generating value from data these days are coders. So I think we're actually a really long way away from the place where the technology is where it needs to be in order for these low-code, no-code-type environments to really be viable in a really serious business setting for data science projects. And so what we're trying to do at Zerve is build something that is built for experts for coders that's targeted for those people that know what they're doing and they just don't have any good tools to do it in. So-

Jon Krohn: 00:09:18 Nice. Yeah, makes a lot of sense how your experience at DataRobot led you to notice this gap. And yeah, allowing data scientists to be able to write better code, get that into production more quickly, it sounds like a brilliant idea. This is something that my data science team at Nebula, my startup, constantly runs into this exact problem. And most of what we do now is taking, like you say, Jupyter Notebook code and just spending a bit of time, having to spend time that we wouldn't love to be spending to engineer that. There's one data scientist on our team and there ends up being bugs with the way that this happens, but you can also run Jupyter Notebooks from within say PyCharm.

Greg Michaelson: 00:10:00 So there is a plugin for VS Code. It's actually been downloaded something like 40 million times or something. So the fact that it's as popular as it is, even though it has its drawbacks, I think gives you some sense of the size of the market.

Jon Krohn: 00:10:18 Yeah, for sure. And so I guess for people that are thinking about, they're listening to this podcast right now, they're thinking, "Wow, this sounds like the perfect thing for me," or, "This sounds like the perfect thing for the data scientists on my team," is it like a free tier for people to be signing up right away or how does that work?

Greg Michaelson: 00:10:35 Yeah, that's the idea. We have some values as a company. One of them is to make available a free valuable product that is always free. So for many people, probably 80-90% of our users, our free tier will be everything that they need. And it will give them plenty of cloud compute, plenty of cloud storage, plenty of development space in order to do the projects that they need to do. It's really ideal for students, for people that are learning, for folks that are involved in more run-of-the-mill-type projects. So yeah, there'll be a freemium tier or a free tier and then what we call an almost free tier, which bumps up your compute, bumps up your storage, gives you some more features, that sort of thing, a freemium model for a trivial amount more. So-

Jon Krohn: 00:11:23 That reminds me, it's probably something similar to the Google Colab kind of thing where you pay 10 bucks a month or that kind of thing in the US. And that means that you get allocated better GPUs in the backend.

Greg Michaelson: 00:11:33 Yeah, exactly, exactly.

Jon Krohn: 00:11:34 Yeah. Nice. And then there's probably an enterprise play as well.

Greg Michaelson: 00:11:38 Yeah, yeah, totally. Zerve was designed from the beginning to be self-hosted. So our free tier is going to be hosted in our SaaS environment, in our cloud account, but for the enterprise, you don't want your data and your code going out to some third party, right? You want to keep it all in your environment. So our enterprise play is self-hosted. We're on the Amazon Marketplace, the AWS Marketplace. We'll soon be on Azure and GCP as well. But it's super easy install because we designed it that way from the beginning. Most of the players in this space designer to SaaS and then they have to like backdoor in ways to actually do a self-hosted-type install. But ours is it takes five minutes to set up and then you're up and running. And the way it works is that we operate a control plane that handles things like orchestration, but then all the data and the compute and the storage and everything stays in your cloud environment, in your VPN and so on.

Jon Krohn: 00:12:37 Nice, very cool. That sounds perfect for the enterprise. And another thing that I think would sound good for certainly the enterprise, but maybe also for your free tier is Zerve emphasizes collaboration also from the very beginning. That was a big part I remember from even when we very first met. So can you elaborate on the challenges and benefits of collaborative work in this kind of environment?

Greg Michaelson: 00:12:58 Yeah. So when we were actually at ODSC, last ODSC West where we met, we were talking with a Fortune 50 company, a huge company. And the way they manage their notebooks is they had us they have a Slack channel where they will put different emojis, like different reaction emojis, under the Jupyter Notebook file to say, "Okay, I'm not in the file anymore. Okay, now I'm in it. Nobody else go in," right? And we had to look and see what emoji was there. And that was the way they managed. We had a similar experience a DataRobot. One of the things that we were doing towards the end of my time there was building a COVID forecasting system during the pandemic for the US government. And that was a really, really cool experience, but a lot of the work that we did was in notebooks.

00:13:50 But we're trying to work, follow-the-sudden-type environments, so that we could have people in the US, in Ukraine, in Singapore, in India constantly working on these projects and trying to manage notebooks in an environment like that is a disaster. It's a mess, right? Because most people who use notebooks are using them locally and then you run into all kinds of like dependency problems, "Which version of Python are you on?" "Oh, I'm using NumPy 1.21 instead of 1.23 and that's not compatible with TensorFlow," or whatever it might be right. And so that sort of experience is a nightmare.

00:14:28 And so really, all of the innovation that's happened in the notebook space over the last, say, five or 10 years has been trying to become more collaborative. Google Colab is probably the first example of that, but there's others like Hex and Deepnote and a few others that have tried to solve this collaborative problem, but unfortunately, they don't it right. So if you go into Google Colab and you've got five or six people logged into the same notebook, you've got a crazy lag delay thing going on. And then you also have to do things like locking cells, so if somebody is in a cell then nobody else can edit it. And the big frailty with notebooks is that, if you run the cells in the wrong order or if you run them too many times, then you can get into a bad state, right? And everyone who's ever used the notebook has clicked on the restart kernel button because you've gotten your notebook into a bad state. And the only way to fix it is to turn it off and turn it back on again.

Jon Krohn: 00:15:25 Your business gets to a certain size and the cracks start to emerge. If this is you, you should know these three numbers. Thirty-Seven Thousand. Twenty-Five. One.Thirty-Seven Thousand. That's the number of businesses that have upgraded to NetSuite by Oracle. Twenty-Five. NetSuite turns twenty-five this year. That's twenty-five years of helping businesses do more with less, close their books in days, not weeks; and drive down costs. One, because your business is one-of-a-kind. So you get a customized solution for all of your KPIs - in one efficient system with one source of truth. Manage risk, get reliable forecasts, and improve margins. Everything you need to grow, all in one place. Download NetSuite's popular KPI Checklist, designed to give you consistently excellent performance - absolutely free, at NetSuite dot com slash SUPERDATA. That’s NetSuite dot com slash SUPERDATA.

00:16:12 I do that constantly when I'm working on notebooks, even if it's just for me, even if it's just like a teaching material. I am constantly like, "Okay, I've added a whole new section here. Let's make sure that if I restart the kernel and rerun this notebook, everything will still work." And for the cells ... So that means that even from the beginning, any cells that I know would take a long time to run, I'll have them work with like a trivial amount of data, so that I can just make sure ... And that is like when I get to the end, then I'm like, "Okay, now it's time to like run it overnight or whatever." I change that an index, so that I'm working with the full amount of data.

Greg Michaelson: 00:16:46 Yeah, yeah, yeah, that's a nightmare, right? Can you imagine like when they developed these Jupyter Note, back when they were called IPython Notebooks back at Berkeley when they first developed them 20 something years ago, the first version of these guys, I can't imagine that it had a restart kernel button on it, right? The way that it had had to have happened, and I'd love to hear a firsthand account of this, is that there are people who developed IPython Notebooks and they realized, "Oh, my notebook's in a bad state," and then they've got to close it down, go to terminal, Ctrl C and then restart the notebook and all that sort of stuff.

00:17:23 At some point, somebody goes, "Well, why don't we just put a button there that will do it." Nobody would develop the restart kernel button from first principles. That's never going to be in anybody's design document. It's a Band-Aid fix for an architectural problem. And so that's the big reason why notebooks are unusable in production, but the crazy part is that all the collaboration, excuse me, all the innovation in that space is around collaboration. It's putting them in the cloud. It's making it where lots of people can log in, but the big problem is, if you do have lots of people logged in, it's way more likely to get into a bad state and so you're constantly restarting your kernel.

00:18:04 So think about, Zerve is we don't have a restart kernel button, but you can still have 10 or 12 or 15 or 100 people logged into a project, all of them running code, all of them interacting and you're always guaranteed to get exactly the same output every time. So no restart kernel button. We didn't reuse a single line of Jupyter code. Whereas most of the notebook competitors just forked Jupyter and added some bells and whistles.

Jon Krohn: 00:18:28 For sure. For sure. That is exactly what they did. All right, so walk us through user journey. So what can I be doing in this flow that I couldn't do in Deepnote or Hex or Colab? Yeah, give me a story here.

Greg Michaelson: 00:18:45 Yeah, so Zerve is language interoperable, first of all, so most teams on these data science projects are operating in multiple languages. So you might have some data engineers that are working in SQL, maybe they're pulling data out of a snowflake or something like that. And then they would normally say, "Okay, where do you want me to land this data? They might make a temporary table in snowflake or they might dump it as a CSV file somewhere or store it as parquet or who knows what they would do with it. And then they handed off to a data science team. And maybe that data science team is using ... Maybe you've got some old school folks that are using R. Maybe you've got some ... Because the R guys will tell you that their visualization stuff is so much better than Python and maybe they're right.

00:19:29 And then you've got some data scientists, all the young folks, that are using Python and they're going to do their development stuff. And so they're working in their languages. And, you know, maybe your R guys are in RStudio or Posit like it's called now. And maybe your Python guys are using Jupyter or maybe they're in PyCharm or whatever. And then they get done with that project and then they've got to go to the DevOps guys, right? They've got to say, "Okay, we've got this project and now we need to spin up a server and get you a docker container and do this productionizing whole thing." And so you've got four or five teams that are all operating in different languages. And Zerve lets them all work in the same environment. So in Zerve, you can do things like SQL query, a pandas dataframe. You can write ggplot code to visualize data from some Python data structure, that sort of stuff.

Jon Krohn: 00:20:26 Wow, wow.

Greg Michaelson: 00:20:27 So it's completely language interoperable in a really, really seamless way.

Jon Krohn: 00:20:30 That's cool.

Greg Michaelson: 00:20:31 And we're not doing code conversion on the backend. What we're actually doing is serializing your outputs. So if you create an artifact throughout the course of your analysis, like say a model or a dataset or something like that, we're going to persist it for you, so that you can actually reference it. And we persist it in a really clever way, so that you can interact with those basic data types in any language that you want. And so that language interoperability really opens the door for a lot of collaboration. That's the second key. The first one is stability. So no matter who runs what, when, or how many times, you're always guaranteed to get the same output. And the second one is language interoperability. So no matter who's on my team, they can all interact with all the different parts of the project using the language that they already know in the same place, which is great.

Jon Krohn: 00:21:19 You used the term project a few times. And so I'm guessing that that is quite a different experience from what we would typically call a notebook. It sounds like it's a separate thing and maybe that's somehow how you're able to get this consistency.

Greg Michaelson: 00:21:30 It's canvas based. So if you imagine a DAG, directed acyclic graph, where essentially when you serve to write code, you're essentially linking together blocks or nodes in that graph and then the code executes from left to right. And so your first block might be a query against a Databricks or Snowflake or something like that and then that might connect to a Python block, which where you might filter the data or perform some transformation or start some sort of a machine learning pipeline. And then that might connect to an R Block or whatever it might be, right? It's very flexible.

Jon Krohn: 00:22:10 Nice, very cool. And so if I'm coming in and I want to do some exploratory data analysis, I'm guessing that I start off by creating a node in which I'm importing the data. And then I have that visually flow to a node where say, I'm doing some visualizations, and then in this DAG, can I have a one-to-many kinds of relationships?

Greg Michaelson: 00:22:36 Yeah, absolutely. The other thing that you can do that is really important for collaboration is you can run as many blocks at the same time as you want. So if you look at like a Jupyter, Jupyter is single threaded. So if a cell is running, you might as well get a coffee, or even worse, go have a conversation with somebody, right? And as a data scientist, I tried to avoid having conversations with people as much as possible. But in Zerve, because the way the architecture is set up, I can take my DAG and I could say, "Go from one block to three different forks of that block," and I could execute those all at the same time. So if I want to train, say, 10 models at once, then I can kick those off and let them all do their thing and I can still be working in another block, writing more code or writing more code or whatever it might be.

Jon Krohn: 00:23:24 Nice, very cool. All right, so I could have my left most node and importing data and then I have this visualization node. And then I could have multiple nodes coming off of that visualization node where maybe I'm working on one of them because I've decided that for me and maybe people on some like Team A that I'm managing, we're working on some machine learning model. Meanwhile, there's another group, Team B, that's a whole other group of people, I don't necessarily need to be working with them, maybe we have like a weekly stand-up together and they're more interested in getting insights from the data.

00:24:04 So we have this Team A, is doing machine learning. They've got a node coming off of the import and they're doing all this machine learning stuff and having as many nodes and they're kind of part of this graph. Meanwhile, there's another team doing analysis and insights. And so that's the kind of flow you can have, lots of different groups collaborating in the same code. We're guaranteed to be working with the same versions of software libraries, I guess. Where do you specify that in the flow, that kind of the software libraries, the dependencies?

Greg Michaelson: 00:24:34 Yeah, so every canvas is supported by a docker image that has all of your dependencies in it and that's editable by the user. So if I want to add PyTorch or I want to add in XGBoost, or some Hugging Face libraries or whatever it might be, then I just add those in to rebuild that docker image and then that new environment is ready to go. So you don't have to worry about it, right? So if a new person joins your team, instead of them spending the first week setting up Python and figuring out their virtual environments and all that jazz, you can say, "Okay, I shared with you our default data science environment. You're ready to roll."

Jon Krohn: 00:25:11 Nice. That's great. That sounds so easy. That sounds amazing.

Greg Michaelson: 00:25:16 It's fantastic. It really is. It's really been a great experience to take a lot of the frustrating challenges. We call it Data Science Stockholm Syndrome. Data scientists have been held hostage by ... In fact, we actually said that in Stockholm after we launched our beta in Paris at JupyterCon. We went to Stockholm and we talked about the Data Science Stockholm Syndrome. And nobody really got it apparently that's not a thing.

Jon Krohn: 00:25:46 They just call it the syndrome.

Greg Michaelson: 00:25:47 Yeah. Yeah, the syndrome. So yeah, yeah, yeah. It's been really cool to take all the things that frustrate data scientists about the way the workflow goes and try and build real serious fixes for them.

Jon Krohn: 00:26:02 Awesome. And so we know that you're going into full public release today, January 30th, 2023. Very exciting. What else is coming? What's next in the development roadmap for you?

Greg Michaelson: 00:26:17 Yeah, building software is surprisingly hard. I didn't realize how hard it was when I was a DataRobot because I was purely on the customer side. And so hats off to all the engineers, because yeah, building software is hard. So there's lots and lots and lots of stuff on the roadmap that we want to build. The next big thing for us is a GitHub integration. So we have a pretty basic GitHub integration now that's just a one way GitHub integration to track version history and stuff like that. But one of the things that most data scientists ... Well, really, you have data scientists and then you've got machine learning engineers. And it's not like one or the other. It's a spectrum, right? In terms of how technical folks are. But a lot of the data scientists that I've interacted with are not familiar with GitHub and they don't typically use GitHub as a way to control their code. Like managing ...

Jon Krohn: 00:27:10 Really?

Greg Michaelson: 00:27:10 ... notebooks on GitHub is a nightmare, right?

Jon Krohn: 00:27:12 Oh, yeah, yeah, yeah.

Greg Michaelson: 00:27:15 Yeah, so we want to build an integration with GitHub. That's the first thing that we're going to do, that gives you the ability to do all those things in a really seamless way, handle merge conflicts and protect branches, so when you have a production branch. Because the nice thing about having a stable development environment is that suddenly you have the ability to do more deployment-type things. So instead of having to dump my code somewhere and give it to some DevOps team to build me a deployment, suddenly I can create my APIs inside Zerve and deploy it from there. And I can reference my objects from within Zerve.

00:27:53 So normally, if I build, say a random forest or what, neural network or whatever, and I was working in like PyCharm or VS Code, I'd have to figure out some way to serialize that model and then put it in some sort of a docker container and then give it to somebody to put it in some kind of a Kubernetes cluster. That's super complicated. Anyway, it strikes me as super complicated. But because of the way our architecture is set up, we persist all of those artifacts, so that you can reference them directly without having to worry about how to serialize them and how to handle all your dependencies and libraries and all that kind of stuff.

00:28:31 So I could from like airflow say, "I could reference the model that I trained Zerve and I don't have to worry about any of that stuff," or, "I could build my APIs directly inside of Zerve and then host those guys from within Zerve or download all that stuff, as like in docker and deploy it in my own system." So it's pretty neat to be able to think about deploying in the same place where you develop and that hasn't really been an option very much until now.

Jon Krohn: 00:29:03 Empower your business with Prophets of AI, the leading agency for AI and robotics experts. Whether you seek a captivating keynote speaker, a company workshop host, or even guidance in implementing AI seamlessly into your organization, Prophets of AI connects you directly with the luminaries shaping the future of AI, such as Ben Goertzel and Nell Watson, both of whom have been phenomenal guests on this very podcast. Whether you are a large global enterprise or just beginning your AI journey, Prophets of AI have a solution for you. Their speakers have graced the most prestigious stages around the world, and now you can head to ProphetsOfAI.com yourself to see their full roster or to the show notes where we've got their contact link.

00:29:45 Yeah. And that ties things back kind of, that's where we started off as one of the main benefits of this, but then we talked about so many more since. So yeah, so being able to deploy in the same place that you're playing around, great. Interoperable between many languages, amazing. Consistency, obviously critical and so lacking in so many other of the environments that we play around in, a visual DAG of code blocks to make things easy and to be able to collaborate the running code blocks separately as many times as we want. And each canvas, each project, having its own docker image, making it easy for people to get started and making it easy to ensure that everyone's on the same page, very cool, Greg, congrats.

Greg Michaelson: 00:30:29 Thanks.

Jon Krohn: 00:30:29 All right, awesome. So stepping back a little bit from Zerve in particular and taking a broader view, from your days at Travelers and then at DataRobot, you saw the evolution of automated machine learning or AutoML. And there's this Zerve open source project called Pypelines that promises to deliver AutoML, is it should always have been open, flexible, code-based and targeted. In a recent talk, you highlighted that just feeding data into a machine and having magic happen turned out to be challenging. What are the key misconceptions people have about AutoML and how did Zerve, address them with Pypelines project? And I also need to highlight for sure that Pypelines, that first, well, you think would be an I, the second character in pipelines, is a Y, emphasizing, I guess, a Python route for this AutoML?

Greg Michaelson: 00:31:23 Yeah, you'd have thought that name would have been taken, but-

Jon Krohn: 00:31:27 Yeah, that's actually amazing. Do you pronounce it Pypelines? Pypelines?

Greg Michaelson: 00:31:32 Exactly. Yeah. So I think you can make a pretty strong argument that DataRobot invented automated machine learning back in 2012, 2013, 2014.

Jon Krohn: 00:31:41 What about H2O?

Greg Michaelson: 00:31:45 H2O came along after DataRobot did actually.

Jon Krohn: 00:31:47 Oh, really?

Greg Michaelson: 00:31:48 Well, they're their automated machine learning stuff did anyway. H2O started out as an open source project and I think, if I remember right, their automated machine learning product is called Steam, something like that. I don't recall, I never really got a chance to play with the H2O stuff.

Jon Krohn: 00:32:09 I really only know where they are today, and today, they make it seem like AutoML has been their thing all the time. That's like the whole platform is based around.

Greg Michaelson: 00:32:20 Yeah, yeah, yeah. No, because I remember at DataRobot, years into the journey there that we started finally talking about what H2O was doing to copy what we were doing, which you know how it is, but that's obviously a biased perspective since I was at DataRobot and not at H2O. And I know a lot of good guys in H2O and they're doing some good stuff. So that is what it is.

Jon Krohn: 00:32:41 Yeah, but anyway, pioneers in AutoML no matter what.

Greg Michaelson: 00:32:44 Yeah, so DataRobot got its start using automated machine learning to win Kaggle competitions. So that was how ... This was back when Kaggle was fun and now it's just like a targeted search for target leakage. Kaggle has gone way downhill in the last several years, but back in the day, that's how DataRobot got started, is they won some Kaggle competitions just using pure automated machine learning. So the thing about it, I think about really any low-code or no-code-type tools, which automated machine learning definitely falls into that category, certainly in the way that it exists today, is that they are really great for happy path problems.

00:33:29 So if there's nothing weird about the problem that you're trying to solve, then automated machine learning is going to be great. You can build a really basic customer retention model using an automated machine learning solution really easily. And it will probably be better than one that you could build by hand. The technology is really good. The problem is most of the problems you face are not happy path problems. Like for example, when we built the COVID forecasting machine for the US government, we couldn't use DataRobot at all for that project. We literally coded it from scratch, which seems crazy that a company that exists to build machine learning models couldn't use its own software really to do that forecasting, which is crazy. That's neither here nor there.

00:34:24 The thing about low-code, no-code is that, most problems, you run into the flexibility problem really fast. So like if I'm building a low-code, no-code solution and somebody says, "Hey, I would really like to be able to merge datasets," let's say, well, then I need to build a merged datasets feature. And if somebody says, "Well, I really need to be able to do like, I don't know, this feature X." "Okay, I've got to go build ..." You got to build every possible feature, whereas with a code-based environment, you don't. We just expect that people are going to know how to merge tables or visualize data or something like that.

00:35:03 And so I actually think it's the wrong approach to try and build every possible thing that anybody could want to do using like a template or a point-and-click-type solution because the people who are generating value here are the ones that are doing it with code. So anyway, I think that's the main drawback with the automated machine learning stuff, is that it's just so hard to build for every possible eventuality when it comes to like all the different types of models you could build and all the different kinds of tweaks that you might want to do. And it really is ... It's both a science and an art to do that kind of stuff.

00:35:40 For your most basic problems, like I said, it's great, use it all day long, but a lot of times, you're going to need a code-based tool to go alongside of it at the very least for the integration piece. Because 100% of projects, even your most basic ones require code. You got to integrate that with Salesforce. You've got to integrate it with some system for deployment. You've got to write an API for it or whatever it might be. You're going to need to write some code at some point. And at the very worst, you're going to just have to start from scratch. You're going to get two-thirds of the way down the road and realize, "Oh, this tool won't do that," and then you have to start over from scratch in code.

Jon Krohn: 00:36:14 Yeah, one of the things about AutoML that I think managers who don't have hands on data science experience think that there's going to be a magic bullet with AutoML because it can do so many different kinds of things. But when you are hands on in a data science problem and you've moved beyond like Kaggle datasets or toy problems, when you get into the nitty-gritty data science problems that tech companies are facing, typically, they're solving problems that have nuances that have aspects that no one has ever dealt with before. And it's not like you just have this perfect, "These are the labels. Here's our inputs. Let's just find some function that predicts the labels as best as possible." It's almost never that simple in real life. In real life, there's tons of ugly things about the training data that you've collected from your users. That means you need to be artificially putting in barriers in some way in order to have the machine model do what you really want as opposed to what your labels will get you on their own.

Greg Michaelson: 00:37:26 The other thing that I learned about automated machine learning is that the people who actually want to use it have no idea what they would use it for. So like one of the things that I did when I was chief customer officer at DataRobot was we hired about 100 former McKinsey folks to do nothing but go around to do use case workshops with our customers all the time. Because the people who want to use automated machine learning, who don't know how to code, they also have no idea what to use machine learning for.

00:37:55 And so we developed these use case workshops, so we could go around and say, "Okay, tell us about your business. Let us understand the oil and gas industry and then we'll come up with some reasons for you to actually use this software that you bought to make your business work better." And I'm not exaggerating, we had 100 guys and gals and that's what they did. They went around and did use case workshops to help our customers figure out what they needed machine learning for in the first place. So that's another thing, is that the people who know what to use machine learning for hate automated machine learning. So there's a retention question there as well. If the customers that could use your tool don't want to and the customers that want to use your tool don't know how, it's a weird situation to be in from a [inaudible 00:38:45].

Jon Krohn: 00:38:46 Yeah, you can imagine executives that don't have technical experience, seeing AutoML and thinking that this means that it can solve any kind of problem they have.

Greg Michaelson: 00:38:58 Yeah, yeah, exactly. And that's how the sales cycle works here in these types of ... It's a top-down enterprise sales cycle. So you go to the chief data officer or the CIO and you say, "Hey, you could do this more efficiently, you could do that more," and then they buy it for $1 million or whatever it is and then they send the message down, "Hey, this is the tool we're using now," and then, that's a change management issue in and of itself.

Jon Krohn: 00:39:29 Yeah. And a lot of the time, it's going to end up being the case that then the people who are hands on actually solving the problems are like, "Why did you buy this tool for us? It doesn't, in any way, solve the problems that you were hoping it would. Yes, it does AutoML, but the problems that we're tackling, you can't subject them to that kind of system."

Greg Michaelson: 00:39:46 Oh, so Pypeline, so you're asking about Pypelines?

Jon Krohn: 00:39:49 Oh, yeah.

Greg Michaelson: 00:39:49 Yeah, so what I thought all along in terms of like what would make an automated machine learning solution actually really useful is if it could be targeted at experts. So the people who know how to write the code, who have written machine learning pypelines in the past, 100s or 1000s of machine learning pypelines in the past, but if you feed it ... So if you feed pypelines a dataset, then what it returns is not trained models, but rather training code that then you can take and do whatever you want with.

00:40:24 So, "Okay, I don't want to do median imputation. I want to do mean imputation," or, "If I want to discretize this continuous variable in this very specific way, well, I've got all the code to do it. I've set up all the column transformers. I've done all the variable grouping. I've handled all the pre-processing, all that kind of stuff and so I can save to three hours on any given project just by feeding it into this open source tool, that gives me boilerplate training code. And then I can customize it to be whatever I need it to be for the particular project I'm working on." So that was the vision anyway when it came to Pypelines.

Jon Krohn: 00:41:03 Nice, very cool. So let's do the same kind of thing as I had to do earlier, take us through a user story. So now I am a data scientist, I work in Python and I guess I can go to get up to GitHub the Pypelines code?

Greg Michaelson: 00:41:18 Yup, you go to GitHub and then you import some packages. You could do it from a notebook if you want or-

Jon Krohn: 00:41:25 Or from Zerve.

Greg Michaelson: 00:41:25 Or from Zerve absolutely. It's integrated into Zerve. So you don't even have to import anything.

Jon Krohn: 00:41:30 Oh, nice.

Greg Michaelson: 00:41:31 It's built right into Zerve. But you import those packages and then you create an instance of a pypelines object by specifying the dataset and the target and how you want the cross validation set up and all the kinds of stuff that you'd normally have to do. And then I think this function is GitCode and it just outputs the code. You can output it directly to your Clipboard, so you can copy/paste it somewhere or whatever. And it just gives you that code ...

Jon Krohn: 00:41:58 Cool.

Greg Michaelson: 00:41:58 ... that you can then take and use. So it's very simple to use.

Jon Krohn: 00:42:00 So cool. Yeah, AutoML, but you're outputting Python code instead of outputting results ...

Greg Michaelson: 00:42:06 Exactly.

Jon Krohn: 00:42:06 ... which means that you have all the downstream flexibility that you would with a code project, but it's skipping a bunch of steps for you.

Greg Michaelson: 00:42:12 Yeah, and the license is super open. So you can take it and use it for whatever you want.

Jon Krohn: 00:42:17 Fantastic. That's really cool. Let's talk about AutoML a little bit more as opposed to just in the context of pypelines. So do you think that in the future we will have these "citizen data scientists" that are able to run machine learning on autopilot in all kinds of different places?

Greg Michaelson: 00:42:37 I don't know, man. I think that I once saw a grainy photograph of a citizen data scientist in a national park. I've never actually met one. Citizen data scientist ... DataRobot may have coined that term. I don't know. It was part of our spiel. And there are a couple of reasons why I don't think that they exist today. And one is that you actually have to have a significant amount of knowledge to frame a problem in the first place, and then the other one is, you're going to need code at some point along the way. And so you're going to have to have someone to mentor you or someone to hand it off to once you get past the most basic step.

00:43:19 So if anybody knows a citizen data scientist, I'd love a LinkedIn introduction. I'd love to have that conversation. I've never met one and I look for him for years and years along the way. So in the future, I don't know that ... Some of these generative AI models are ... ChatGPT writes pretty good code, which is exciting, but there's a lot of nuances and a lot of ways to get it wrong. So I'm still pretty wary about a lot of the ChatGPT-based solutions that get built out there. So yeah, I don't know. I don't want to say never, but probably not for the next five or 10 years will you have something that is a reliable coding environment that won't do dumb things like set up your cross validation wrong or bake in some target leakage into your models or overfit or whatever it might be. Just because there's a lot of subtleties there that it'll be awfully hard to design a prompt to communicate.

Jon Krohn: 00:44:19 Large Language Models are revolutionizing how we interact with technology. With companies rapidly adopting models like the GPT and Llama series architectures, the demand for skilled LLM engineers is soaring. That's why Kirill & Hadelin, who have taught Machine Learning to millions of professionals, have created the 'Large Language Models A-Z' course. Packed with deep insights on Tokenization, Input Embedding, Transformers, Self-Attention and LLM Tuning this course will help you gain hands-on experience with LLMs and stay competitive in today's job market. Enroll at SuperDataScience.com/llmcourse for your free 14-day trial. This course is exclusively available in the SuperDataScience Community, you won't find it anywhere else. Once again, the link is SuperDataScience.com/llmcourse

00:45:08 Yeah, I think that tools like GPT-4, they make it conceivable that a citizen data scientist could exist possibly. But I think it's getting into the nitty-gritty and being able to solve problems, be able to know that assumptions that you're making are going to pan out in the real world once you're in production. And I guess it's theoretically possible that somebody could develop all that insightfulness that data scientists acquire without actually writing code themselves. But I think what's going to end up happening is that citizen data scientist is iteratively making himself less citizen all the time.

00:45:50 Maybe they can get started using GPT-4 to generate some example code, but then once they start running into issues, they're just going to end up starting to have to pick up an understanding of how that code works. And that can be made easier by asking GPT-4 questions or similar kinds of tools questions, but I think they end up becoming less and less of a citizen as they get more and more of that experience. I think that the idea ... It's hard to imagine how somebody could start without coding experience, get into designing and then deploying machine learning models and somehow along that way, not picking up coding skills is hard to measure. So I think it makes it maybe easier to get going, but I think you still end up getting the skills by the time you're doing it at a professional level.

Greg Michaelson: 00:46:45 Yeah, there's probably some sort of a training path there somehow. Maybe in the university, there's some way to incorporate these AutoML type tools to teach students.

Jon Krohn: 00:46:58 I think all the students are doing it anyway.

Greg Michaelson: 00:47:00 Yeah, right.

Jon Krohn: 00:47:00 I think it's making students' lives easier. We're in an interesting gap right now probably a few years where instructors can delude themselves at, "Ah there's probably not that many of my students using GPT-4 to generate the code for this project."

Greg Michaelson: 00:47:17 Exactly, exactly.

Jon Krohn: 00:47:18 Okay, so we stepped out a bit in the last section of questions where we moved away from just Zerve specifically to AutoML. We talked about pypelines. Let's take another step back now and talk about AI applications in general. So we got into this a little bit a few minutes ago when you're talking about McKinsey consultants at DataRobot being sent out as this army of people working on use cases. So with your role as a chief customer officer at DataRobot as well as your senior director of Analytics and Research role at Travelers before that, you've been in the deep end of AI development for a wide array of businesses and use cases. Greg, why do so many AI projects fail?

Greg Michaelson: 00:48:02 That's a good question. Part of it, I think, is a framing problem, specifying a project in a way that will give you a good outcome, right? So I've specified what my goal is, that the data exists, that the business agrees, that that problem is actually the problem that needs to be solved.

The whole project origination, project framing kind of thing is really important. And a lot of times, what you'll see is that there are really big disconnect between the data science teams that can actually frame problems and the business teams that actually know what the problem is. You know what I mean?

00:48:43 So you sometimes will see a data scientist or a team of data scientists building solutions that, if the business team knew what they were building, they would know from the get go that that would never work, right? It's not answering the right question or the right data is not going to be available or the timing's not going to work out or it's providing a trivial answer. That happens an awful lot, is that somebody builds a model to predict something, but the prediction is obvious. It's not better than what a person could do and the volume is not big enough to actually warranty any type of automation.

00:49:19 The implementation is also a hard part around the success of these projects. So we worked with a bank at DataRobot to implement a solution. It was actually really novel that it was it had to do with the foreign exchange market and I don't think I can go into details for what the particular problem was, but what this bank did is they uncovered almost like a glitch in the system that allowed them to be really smart about how they did currency transfers and uncovered almost like a half a billion dollars in annual revenue, new revenue on this new product, a massive amount of revenue. It took them three or four years to get it implemented, which is crazy. And the amount of bureaucracy that this bank was facing when it came to actually implementing this project was remarkable. It was-

Jon Krohn: 00:50:10 It sounds $like 2 billion in losses by just not getting that up and running.

Greg Michaelson: 00:50:15 Yeah, exactly, bonkers. But the thing about it was, is in order to implement this model, they had to touch core systems. And the system that actually did the currency transfers, that system can't have zero downtime. You know what I mean? So everybody was super nervous about making changes, because if that system goes down, the losses are way, way bigger than the gains would have been if the project had actually been implemented. So there's a lot of bureaucracy and change management and implementation challenges and stuff like that that you encounter as well.

00:50:48 So then the third thing I would say is that, and this may be a little controversial, there's not that many out there. A lot of people will go out and say, "Look, every business should have thousands of machine learning models in production," and "There's trillions...". I think I remember our CEO doing an interview in a magazine where he said, "There should be trillions of AI models out there that are constantly being updated and used and making life better and all that kind of stuff." I don't think there's that many, right? I think, in an organization, you might have dozens, at a mature organization that has been doing machine learning for a long time. There's just not that many opportunities to do.

00:51:32 Now the ones that you actually get deployed, those are going to generate tons of value. By all means, organizations need to be interacting with their data in important ways, whether that's building dashboards or training models or whatever it might be. But how many are there really? Because in order to find a project that is actually deployable, the stars have to align. You have to have the data, you have to do the work to build the models, but then you have to have business buy in. You've got to have IT support to actually do the deployment. So you've got to have ... All the cloud stuff has to be all set up in terms of infrastructure. You've got to have all the connections. It's a lot of work to get these things built.

Jon Krohn: 00:52:12 Yeah, and maintaining them in production to make sure that there's not the various kinds of drift that you can encounter. That's something that ... Yeah, even in my own data science team, we're relatively small, including myself, which is generous because I'm not ever writing production code, but including myself, there's five data scientists in our company. And so how many machine learning models can we get into production before the entirety of our data science resources is consumed by keeping machine learning models live without even doing any new development? It's not very many.

Greg Michaelson: 00:52:46 Yeah, yeah, totally. And there is that whole industry of MLOps that has come about, but I think, arguably, the markets not really ready for a lot of that MLOps stuff. Most organizations probably don't have any deployed machine learning models, all but ... I mean, your biggest ones certainly will, but your medium-sized companies, how many of those are actually doing that kind of work? Well, I think we're a while away from having all those implementation challenges solved, so that we can actually get there with these projects.

Jon Krohn: 00:53:19 Yeah, this highlights ... I sometimes get questions from listeners via social media, my YouTube channel, where people ask things like, "Is there any point in getting started as a data scientist still, or are tools like GPT-4 or GPT-6 that's coming or AutoML, are these just going to replace data scientists?" And I think you're hitting on exactly why there's going to be even more demand in the future for data scientists, which is that there are so ... Except for the biggest organizations, the biggest big tech companies, they have for years now figured out how to be capturing in data operational processes within their company as well as involving their customers, their users.

And so they have been able to stand up dozens maybe in the largest cases like your Googles, your Metas, maybe there's over 100 machine learning models.

00:54:11 But most companies, huge companies, billions and billions of dollars of revenue, they might have a handful of machine learning models in production. And so there's so much opportunity for them to be capturing so much more data and to be implementing so many more models. And so I think demand will continue to grow and grow and grow. The previous ... So we released two episodes a week. We have episodes on Tuesdays and Fridays. The Tuesday episodes are always long, so they're pretty much always at least an hour long. And so a couple of long episodes ago, I'd say two weeks ago, we had episode 749 where Kirill Eremenko, he actually founded the Super Data Science Podcast and he was hosted the show for the first four years, I've just been doing it for the last three and a bit now.

00:54:59 And Kirill was on the show as a guest and he was talking about large language models. It was an intro to LLMs episode and he was making the case at the beginning of the episode that, "Right now, people who are experts in LLMs can command huge salaries and that will probably go down over time, as it has for data scientists." He's saying, "10-12 years ago, data scientists could have these huge salaries," but everything that I see, I've been thinking about this a bunch since we recorded this episode and I don't think I articulated it very well then. I was surprised by the argument that I was making because the median data scientist salary has been continuing to go up over the last 10 years.

00:55:36 And it's because of even ... We create even more demand. The more data scientists there are creating machine learning models, getting them into production, the more opportunities there are for organizations to be capturing data within their organization from users, from counterpart organizations that they work with, and therefore, it's this constant blossoming of opportunities for automation.

Greg Michaelson: 00:56:01 Yeah. No, I think that's absolutely right. I also think that we need to expand our definition of what data science is. Because in school, data science is like training models, but really, there's so much more to it than that. I'm always surprised when I run into data scientists that don't know how to write SQL code because it's like, "How are you going to get the data in the first place?" And there's a ton of value in just being able to go and get your own data. You know what I mean? And then on the other side of the coin is this whole machine learning engineering kind of field.

00:56:35 So let's say I have a model. Let's say I need to figure out what to do with it. Well, there's a whole pile of technologies there to set up like, if you want some sort of a scheduled job or if you want to create an API, if you want to host that thing, there's an awful lot of stuff there that you need to know in order to really generate value. And so I think the window is actually opening bigger for data scientists, but also the bar is getting higher because the amount of things that you're needing to do, now that the technology is more mature and the amount of data is bigger and bigger and bigger, the amount of things you have to be able to do is just more intense.

Jon Krohn: 00:57:20 Yeah, it's interesting. There are certainly ... There's a wider range of problems that people can be solving, but I think also something interesting has happened and I don't want to get dragged into this too much, I've got other more specific questions for you, but just this kind of an interesting conversation to be having, and hopefully, lots of our listeners find it interesting because they are either thinking about getting to data science or they're at some point in the data science career is that I think that data scientists, it seems like, while median salary for data scientists has gone over the last 10 years, 10 years ago, it was typically a requirement to have a PhD in a quantitative subject or a field related to data science ... Well, there was nothing called data science 10 years ago. You can get a degree in that, but like a math degree, a stats degree, a programming degree.

00:58:10 You typically have to have a PhD in one of these kinds of areas and now it doesn't seem like you do, but I guess that's interesting because I guess it's the shift. Maybe 10 years ago, you needed the PhD because the assumption was you were going to be the kind of person that was building models, which is something ... That working independently, having hypotheses, it's PhD kind of stuff. But now 10 years on, it is the kinds of these, either, are much broader kinds of problems that we're tackling, "Maybe you don't need to be that much of an expert in modeling itself, because maybe if you get the data inflows right, and you've got the downstream processes engineered right, maybe it is just an AutoML problem in the middle and you don't need a PhD to be tackling it." But simultaneously, there's pressure to be learning tons of additional software engineering skills around the core data science modeling work.

Greg Michaelson: 00:59:06 Yeah, I think about a lot of the data science work almost like being a plumber or an electrician. In a lot of ways, it's a trade that you might go to trade school for if such a thing existed here in America. Learning to interact with data and build data pipelines and do that and build deployments and that sort of thing, a lot of that is boilerplate and something that you just learn and figure out. But you're right, there's tons of research going on there and the capability to take what you know about, say, how to train a random forest and extend that knowledge to include things like transformers and large language models and stuff like that, that's not straightforward. And it requires the kind of person that is a lifetime learner and is really passionate about this stuff, the kind of person that has models for Fantasy Football and does data science in their free time and all that kind of stuff.

01:00:01 And I think that will probably go away as some of these concepts get incorporated more into the education system, where kids are going coming out of college, knowing about transformers and LLMs and all that kind of stuff. But we're definitely not there yet.

Jon Krohn: 01:00:19 Yeah, yeah, so super interesting conversation. We've gone off on a piece different from what I was expecting, gone off piece, I guess. And so back to my list of topics that I was hoping to cover with you, we talked a few minutes ago now about why AI projects fail. But let's kind of flip that on its head. So for our listeners, whether they are technical or not, given all of your experience at DataRobot, at Travelers and so on, maybe even now at Zerve with use cases, what do you think are the low hanging-fruit today? Maybe this is a tricky question and so there may not even be a good answer, but given your experience, you're exactly the person to ask and so you might be able to pull it out, are there some kinds of areas in an organization where there might be low-hanging fruit for automation?

01:01:13 What kinds of things would need to be right? How would the stars align for there to be opportunities for low-hanging fruit for using AI to automate some processes and get some quick wins, get some profitability, some revenue for the company?

Greg Michaelson: 01:01:31 That is a good question. I don't know if there are any. It's hard, right? Quick wins, that's the dream, right? "Let's go get some quick value." And every organization is going to be very specific. The answer to that question is going to vary pretty dramatically from company to company, but to me, the answer is going to be to focus on the business problems. The wrong approach is to say, "How can we use a random forest to make my business more profitable?" That will almost certainly fail. I think the right approach is to take technical people and make sure that they understand the specificity, the idiosyncrasies of the business, so that they can ... And then they'll figure out all that other technical stuff, right?

01:02:24 So if you work at a publishing company and you make money by selling ads on your websites across your different publications, then that's a complicated business. And do the data scientists understand how that bidding process works and how the pricing models work and all that kind of stuff? I don't know. So I think that's the biggest way to get value, is to make sure that your technical people know as much about the business as possible.

Jon Krohn: 01:02:51 That was such a great soundbite. You really made that happen there. I put you in a real pressure cooker.

Greg Michaelson: 01:02:56 You did, I wasn't expecting that.

Jon Krohn: 01:02:58 Yay. They came out with a gem, so nicely done.

Greg Michaelson: 01:03:01 All right.

Jon Krohn: 01:03:01 That's a beautiful ... We take portions of the episodes and turn those into like standalone YouTube clips that might last few minutes and that sounds like a perfect one.

Greg Michaelson: 01:03:11 Nice.

Jon Krohn: 01:03:13 Awesome. So let's now zoom back into your background. So we've talked about Zerve, we've talked about AutoML

Pypelines, we've talked about data science innovation and AI projects in general. So let's now move back to the kinds of things you were doing before you were in AI. And we're going to tie those into what you are now doing today in data and AI. So you have a super interesting background. I am pretty confident, although maybe a guest of mine has had this and it just wasn't like on their LinkedIn profile, but I'm pretty confident you're the first pastor that I've had on the show.

Greg Michaelson: 01:03:55 Well, you were talking about Ben Taylor earlier. I think his title is data science evangelist or something at Dataiku, right?

Jon Krohn: 01:04:02 Yeah, yeah, yeah, exactly. He's moved on from Dataiku now as well.

Greg Michaelson: 01:04:07 Oh, really? I didn't know that. I haven't talked to him.

Jon Krohn: 01:04:08 Yeah, he's doing his own thing. He's got his own startup growing ...

Greg Michaelson: 01:04:12 All right, good for him.

Jon Krohn: 01:04:14 ... related to what he calls #GoalEngineering. But yeah, these evangelists, you see these roles, they've all made me cringe. I've gotten cold reach-out emails about, "Would you like to be data science evangelist for a company or AI evangelist?" and I'm like, "Who wants that title?" But yeah, you were a Baptist pastor.

Greg Michaelson: 01:04:40 Preach.

Jon Krohn: 01:04:41 A real evangelist, yeah, for six years and you're doing that after or maybe I'm not getting the timings exactly right here, but you did a bachelor's degree in philosophy, a master's in theology, you're a Baptist pastor for six years and it looks like you did a second master's in statistics as well as a PhD in applied statistics while you were doing the pastoring.

Greg Michaelson: 01:05:03 Yup, that's right.

Jon Krohn: 01:05:03 Am I getting that roughly right, yeah?

Greg Michaelson: 01:05:05 Yeah, no, that's totally right. I went to Southwestern Seminary and I got a master's in divinity in 2004. And then I had a little church that I was the pastor of in West Alabama, a little town called Vernon, Alabama and Lamar County. And that was a really great experience. I'm an introvert naturally, which is not-

Jon Krohn: 01:05:31 Really?

Greg Michaelson: 01:05:32 Yeah.

Jon Krohn: 01:05:32 That is surprising. The only thing that tipped me off to that was you talking to like data scientists wanting to not talk to people, which for me is funny. Obviously, maybe I'm in the minority as a data scientist because I'm a podcast host data scientist. So I really do want to be talking to people and I constantly want to and that's my management style. That's what I want doing. It's a majority of my day that I'm alone and I don't even really love that. I always want to be working on problems with people like pair coding, that kind of thing. I'd much prefer over being alone, but yeah-

Greg Michaelson: 01:06:05 Yeah, after this podcast, I'm definitely going to need a nap. No, no, but seriously, actually, I get that a lot, but whenever I take these personality tests, which I don't love, but whenever I take them, I always score introverted. So being a preacher, I've just been ... The story of my career is being thrown into situations that I'm not remotely prepared for and then just trying to have to find my way. The first wedding I went to, I officiated.

Jon Krohn: 01:06:36 Oh, wow.

Greg Michaelson: 01:06:36 The first funeral I've ever been at, I was officiating. All of those things were first for me, but I was in charge. And so I'm like YouTubing funerals and weddings and stuff to try and figure out how you're supposed to do these things. It was a great experience for me. I learned a lot about how to work with people, but at the end of the day, I thought it was going to be more about studying and teaching and that sort of thing. And it turns out it's just a PR gig. Being a preacher is just a PR gig, trying to convince people to do things that they don't want to do, which is great. There's value in that, but not how I wanted to spend the rest of my life. So yeah I actually saw the show NUMB3RS. Did you ever watch that show?

Jon Krohn: 01:07:20 I am aware of it. I think if I remember correctly, it's all capital letters in the title and E is a 3.

Greg Michaelson: 01:07:28 Yes, exactly, yeah. It's the stupidest show in history, but I used to love that show because it was this mathematician that would like solve crimes using mathematical magic. And I was like, "Why don't I have a job like that? Why don't I have an interesting job? Why am I trying to convince people to give money to their church kind of thing?" So I actually went back to school to ... I googled like, "What are the finest jobs or what are the best jobs?" or something like that, something that your typical high school senior is going to be googling. And it turns out actuary was the number one voted job at the time.

01:08:05 I went to the University of Alabama and I applied to their School of Mathematics because I wanted to get a degree in math and start taking actuarial exams. But I didn't. What I had at the time was a bachelor's in philosophy and a master's in theology. And they wouldn't admit me. They were like, "No, you need to get a bachelor's in math before you can enter our math graduate program. And so the School of Business there had an applied statistics program and they were like, "That's dumb. Why don't you come and join us?" So I entered the I started the applied statistics program at the College of Business in the University of Alabama, and yeah, the rest is history. It was great.

01:08:47 I kept preaching for ... I think it took me two or three years to do that whole program from start to finish, and yeah, I kept preaching that whole time and then my church was great. It was a really good experience. I still keep in touch with some of the folks from there. And yeah, I went from there to Regions Bank in Birmingham where I was building credit scoring models, which is not remotely related to being a preacher, but such as it is, the twists and turns of life.

Jon Krohn: 01:09:15 I guess you could think about your divine credit, some kind of ledger of whether you're going to heaven or hell. So yeah, so with that kind of background, theology, this seems to me and you could correct me, but theology seems to be related to ethics in a way, right? That must be part of it.

Greg Michaelson: 01:09:44 Yeah.

Jon Krohn: 01:09:44 It means doing the right thing.

Greg Michaelson: 01:09:45 Sure.

Jon Krohn: 01:09:46 And so while you are a DataRobot, you witnessed some of the shocking, unethical behavior from leadership, which you've posted on and so I will include in the show notes for this episode your original LinkedIn posts, which is why I feel comfortable asking you this question because it seems like something that maybe a lot of people wouldn't want to open up about. And maybe that's something related to having that background, being very much in the public eye about ethics and doing the right thing that you felt comfortable doing this. So yeah, so what happened to DataRobot while you were there?

Greg Michaelson: 01:10:19 The end of my time at DataRobot was really interesting because there was turnover in the CEO. So the founder of DataRobot became ... I don't know the details of how that switchover happened, but it happened that he was no longer the founder and the COO became the new CEO. I don't know all the politics. I wasn't that involved in board meetings and that sort of stuff, but for whatever reason, the board decided that they wanted to have new leadership in the company. And The Information posted an article, which I think is what you're going to link in what you just referenced, where it apparently came to light. And I don't have any firsthand knowledge of this. I'm just going off of what it said in the article, that some of the executives at DataRobot had figured out a way to sell some of their stake in DataRobot on the secondary market to the tune of millions of dollars, eight figures' worth, which looks-

Jon Krohn: 01:11:23 Like per person kind of thing? Like those kinds of numbers are the total?

Greg Michaelson: 01:11:27 Yeah, between $4 and $10 million per person kind of thing, according to the article. Like I said, I don't have any firsthand knowledge. But I think that story was deeply unsettling for a lot of the folks that had been at DataRobot for a while, who had been looking for ways to get some liquidity for their stake in DataRobot along the way. And it turned out that point in time was the peak valuation for DataRobot. And so that handful of executives made a killing, while the folks that had been building DataRobot for the last six, seven, eight, 10 years were left with stock or with options, well, that they couldn't find liquidity for.

01:12:19 And so I know that created some morale problems and I think it actually ended up leading to another turnover in the CEO role at DataRobot, which was disruptive, I suppose, is the word for it. But after some reflection, it's been a couple of years since all that happened, there's a lot of risks when it comes to founding or joining a startup. Actually, when I left to Travelers to join DataRobot, I told people that joining a startup, it's like a job and a lottery ticket because there's enough risk in that whole space that your options may never be worth anything or they might be worth potentially millions of dollars. And so you have to be comfortable with that level of risk and not be too salty when things go awry.

01:13:16 Because at the end of the day, nine and 10 of these things are going to fail outright and the ones that are like DataRobot become unicorns, well, those get really, really complicated when it comes to, "How do you pay?" Because DataRobot raised over a billion dollars in venture capital and so that's a massive amount of money.

Jon Krohn: 01:13:34 Crazy. Yeah, yeah.

Greg Michaelson: 01:13:35 And so, "How do you ..." Getting liquidity out of that, unless the company has a structured program for it, which I would argue DataRobot should have had, and towards the end there, I think they probably created some programs like that, but certainly, for the first five or so years that I was there, there was no way for employees to cash out some of that liquidity, take something off the table and get rewarded for the work that they put in because people were working insanely hard at DataRobot to build something that was really, really cool. And so yeah, that was definitely a turbulent time. And like I said, I don't have any firsthand knowledge of any of that stuff. I'm just going off of what was in the media. So anyway, that's my take.

Jon Krohn: 01:14:21 Yeah, so it's a very interesting story and I'll link to the article, as you mentioned, and that will be in the show notes, so people can dig in in more detail. I think when people read that article, I think they'll see there's even more elements you probably don't feel comfortable speaking to because you don't have the firsthand experience, but it really does seem like a scandalous situation. And so I guess a takeaway for our listeners, if they're thinking of getting involved in a startup or they are involved in a startup and they're worried about being able to get some liquidity, I guess that's the lesson here, is that you should be pushing management to have some kind of liquidity, so that there's some way for employees to cash out at least part of their equity options, as opposed to being locked in for potentially a decade or more and not knowing how cap table changes are going to impact that lottery ticket that they have.

Greg Michaelson: 01:15:15 Yeah, and also talk with a CPA because there's a lot of funny tax things when you exercise stock options that you have to be really aware of. And so yeah, there's a lot of complexity to it. It's not for the faint of heart, but certainly, it's fun. It's a wild ride and people ... Certainly, I'm a person that wants to create and is comfortable with a pretty high level of risk in my life, so startup life seems to work for me, but it's not for everybody. So beware.

Jon Krohn: 01:15:51 All right, nice. Well, it's been a great journey with you here, Greg. One last question for you that ties in all of your experience and also might give our listeners a little bit of extra insight, so we just had a tip for them on how they could be trying to ensure they get good value and they avoid tax implications when exercising stock options. Another more technical tip for tech professionals, so in an interview five years ago, you said that a great communicator has some technical capability. You said that a great computer that has some technical capability is hugely valuable. So what tips do you have for tech professionals looking to improve their communication and storytelling abilities when it comes to explaining technical concepts? And I got to say, you are a perfect example of this. Should they spend six years as a pastor? Is that you recommendation?

Greg Michaelson: 01:16:52 Yeah, no, I don't recommend that. I think there's a few things. One of them is you've got to ... How do you train somebody to be self-aware? Because at the end of the day, being able to communicate is about figuring out how you're coming across and how people are experiencing the things that you say. So it's hard to be willing to get that feedback, right? So having people in your life that can actually tell you, "Hey, this is how that came across," or, "That was really boring," or "Nobody understood what you were talking about," finding those kinds of opportunities for feedback I think are really important. But then just practicing way outside of your comfort zone. I just did a training with the Intel Ignite program. So Zerve is a part of Intel Ignite, which is a startup accelerator that Intel does.

Jon Krohn: 01:17:48 We had some items on that. It was hugely competitive. We didn't get into these questions because there's only so many things I can ask, but our research did dig up. So Zerve was actually founded in Ireland ...

Greg Michaelson: 01:17:59 Correct.

Jon Krohn: 01:17:59 ... which is, I think, probably obvious to listeners given your accent. No, I know it's your co-founder Phily that has that Irish accent-

Greg Michaelson: 01:18:08 And Jason. So there are three co-founders, Jason, Phily and I. Both of them are Irish.

Jon Krohn: 01:18:14 Got you, got you. Yeah, I hadn't met Jason. So Zerve was founded in Ireland and it was one of just 10 European startups chosen to participate in this Intel Ignite accelerator program. And so congratulations, it's a big deal. And yeah, so tell us more about that.

Greg Michaelson: 01:18:30 We did this training on ... Well, the first session that we did together was about pitching, right? Because one of the big focuses in this program is on raising money and finding the business and all that sort of thing. And so we had this guy come in, who's fantastic, to do some training on how to communicate. He's a big fan of toastmasters.net, toastmasters.net. And so he had all these kinds of tricks and things that you could do to be a really effective communicator and we had to get way outside of our comfort zone in order to really have something magical happen when we were telling stories or talking about our products or whatever it might be.

01:19:11 So I think having those experiences, even if it's not technical communication, even if it's more like personal communication or storytelling-type stuff, really the only way to learn this stuff is to just throw yourself into the deep end and try and wade through the river of blood as it were in order to get to a good spot. So just practice, I think, is probably the biggest thing I could say.

Jon Krohn: 01:19:34 That's a really great tip. Having something specific like that, a structured program like Toastmasters, that makes a lot of sense. People can actually just look that up, do it. There's probably online things, but I think Toastmasters also has ... If I remember correctly, they do have like in-person components. So you can go and practice in person, which I think-

Greg Michaelson: 01:19:51 Yeah, it's fantastic. I did it some in, I don't remember if it was college or seminary or what, but it's a great opportunity. If there's a Toastmasters group in your area, it's definitely worth joining. It's a lot of fun. It's also scary as all [inaudible 01:20:04].

Jon Krohn: 01:20:06 Yeah, and I absolutely agree. So we used to ask on the show constantly, it was like my go-to question that asked the most, was like, "What do you look for in data scientists you hire?" or, "What's the most important skill with data scientists or technologists?" And by far, the biggest answer was communication skills. And yeah, so I think this hits it on the head, it gives people now a tool. Toastmasters is a way that you can be becoming more effective at communicating effectively and it'll be hugely valuable in your career because that's what allows you to get buy in, whether it's with people that you manage or people that are at the same level as you or that are more senior to you. Communication is the key to being able to lead effectively or even if be able to be an IC effectively, because it allows you to make your case to your manager that, "This is the way we should be doing on this project for X, Y and Z."

Greg Michaelson: 01:20:56 Totally, completely agree.

Jon Krohn: 01:20:58 And then very last quick question for you is, on your LinkedIn profile, it says that you are a cereal entrepreneur, spelled C-E-R-E-A-L, like Wheaties and Cheerios. So what's up with that?

Greg Michaelson: 01:21:09 Weetabix. So I had a little passion project. Me and my partner started a company called Cerup, C-E-R-U-P. And we if you go to cerup.com, you can see our product, it's syrup for cereal. So you can put different types of flavorings on your cereals. You can put lemon cream topping on your Cheerios or peanut butter topping on your Apple Jacks or whatever it might be. I think we have 12 or 13 different flavors that we've developed, and yeah, it's a fun little side project, syrup for cereal.

Jon Krohn: 01:21:46 That is super fun. I'm on the website right now and I'll have it in the show notes. Yes, cinnamon toast, banana cream, peaches and cream, strawberry, coffee, apple cinnamon, lemon cream, raspberry and orange marmalade syrups.

Greg Michaelson: 01:21:58 It's really good. It sounds weird and they call them cereal drizzles. That's the name for it now. So yeah, that's a line of cereal drizzles.

Jon Krohn: 01:22:08 Nice, amazing. Fantastic, Greg. Before I let my guests go, I always ask them for a book recommendation. Do you have one for us?

Greg Michaelson: 01:22:17 I just read a book called Habeas Data. It's really interesting. It's about the legal implications of mass surveillance, so things like, if you could set up a computer to record every license plate on every car that passes a particular intersection, one point of view says, "Hey, that's all public. Anybody could look at that license plate and write it down on a piece of paper." But the other perspective is, "You can learn an awful lot about someone by tracking where they go all the time and computers give you that capability to do that now." So Habeas Data is from a legal perspective, "How do you think about those kinds of things and what sort of court cases have happened and where do we think the future is going?" So that's a really interesting one that I've been looking at.

01:23:05 And then there's another one called BS Jobs. It spells out B-S, but I know we're not supposed to say bad words on your podcast. And it's a really funny little book on jobs that are legitimately completely BS and why. And so if anyone is fed up with their current career, that might be a fun job to think about or a fun book to think about reading, so those are two good ones.

Jon Krohn: 01:23:30 Nice. The second one sounds like a lot of fun. And the first one sounds like a great resource for digging into some of the thorny ethical issues that we see on this show or hear about on the show very often. Greg, thank you so much for an amazing, insightful episode. I can't wait to check out Zerve and I'm sure a lot of our listeners will as well. It's nice to be able to have products on the show like this where I hope it doesn't feel like an infomercial at all for people to hear from somebody creating an amazing tool like this, that can genuinely transform the way that they're working. I can't imagine it does and it's nice to be able to get into the technical detail on some of this stuff. And so for people to be able to follow you or Zerve after today's episode, how should they do that?

Greg Michaelson: 01:24:13 Our Instagram is ZerveAI @ZerveAI. So that's probably the best place to follow us. And then we're on LinkedIn as well. We're Zerve on LinkedIn. So give us a followup. We're going to be big.

Jon Krohn: 01:24:26 Nice, no doubt. And so yeah, so we'll have those links in the show notes. Thank you so much, Greg, for taking the time and it's been such a treat catching up with you. Maybe we'll catch up with you again in a couple of years and see how the Zerve journey is coming along.

Greg Michaelson: 01:24:38 Love it. I look forward to it.

Jon Krohn: 01:24:39 Well, I hope you enjoyed that informative and insightful episode. In it, Greg filled us in on how his Zerve platform, which is free to use, means we can collaborate without headaches in any programming language and then deploy to production without needing to rewrite our code. He talked about the open-source Pypelines AutoML project that outputs code instead of just results, how we should focus on business problems, not cool new hammers to ensure AI projects are a success and how Toastmasters is an effective and secular way to learn how to communicate like a pastor.

01:25:16 As always, you can get all the show notes including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Greg's social media profiles as well as my own at superdatascience.com/753. Thanks to my colleagues at Nebula for supporting me while I create content like this Super Data Science episode for you. And thanks of course to Ivana, Mario, Natalie, Serg, Sylvia, Zara and Kirill on this Super Data Science team for producing another eye-opening episode for us today. For enabling that super-duper team to create this really super-duper podcast for you, we are deeply grateful to our sponsors of course. You can support this show by checking out our sponsors links, which are in the show notes.

01:25:55 And if you yourself would like to sponsor an episode, you can get the details at jonkrohn.com/podcast. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until next time, keep on rocking out there and I'm looking forward to enjoying another round of the Super Data Science Podcast with you very soon.