EPISODE 21

[INTRODUCTION]

[0:00:00.2] JM: With a fast-growing field like a data science, it is important to keep some amount of skepticism. Tools could be overhyped, buzzwords can be overemphasized, and people can forget the fundamentals. If you have bad data, you will get bad results in your experimentation. If you don't know what statistical approach you want to take to your data, it doesn't matter how well you know Spark or TensorFlow or anything else. If you are passionate about the work that you're doing, you're unlikely to finish any of the projects you start, and this is true whether we're talking about data science or anything else.

Kyle Polich hosts the Data Skeptic podcast; a show at the intersection of data science and skepticism. As a podcaster, Kyle takes himself seriously and he's prepared for his shows, which I admire and appreciate. Having met him recently at the Microsoft Build Conference, he's a great guy, he's a great host, and I look forward to doing more podcasts with him in the future.

In this episode, Kyle is interviewed by Sid Ramesh, a data engineering correspondent for Software Engineering Daily. I hope you like this episode.

[SPONSOR MESSAGE]

[0:01:21.2] JM: Spring is a season of growth and change. Have you been thinking you'd be happier at a new job? If you're dreaming about a new job and have been waiting for the right time to make a move, go to hire.com/sedaily today. Hired makes finding work enjoyable. Hired uses an algorithmic job-matching tool in combination with a talent advocate who will walk you through the process of finding a better job.

Maybe you want more flexible hours, or more money, or remote work. Maybe you work at Zillow, or Squarespace, or Postmates, or some of the other top technology companies that are desperately looking for engineers on Hired. You and your skills are in high demand. You listen to a software engineering podcast in your spare time, so you're clearly passionate about technology.

Check out hired.com/sedaily to get a special offer for Software Engineering Daily listeners. A $600 signing bonus from Hired when you find that great job that gives you the respect and the salary that you deserve as a talented engineer. I love Hired because it puts you in charge.

Go to hired.com/sedaily, and thanks to Hired for being a continued long-running sponsor of Software Engineering Daily.

[INTERVIEW]

[0:02:48.3] SR: Kyle Polich is the host of the Data Skeptic podcast. Welcome to Software Engineering Daily.

[0:02:52.8] KP: Hey, great to be here.

[0:02:54.2] SR: It's great to have you in the show. You are the host of Data Skeptic podcast in which you interview professional, personas related to the data science field. Tell us a little bitt about your background.

[0:03:04.8] KP: My background, in academia I started out primarily working on AI with a focus in multi-agent systems. While I was doing that I was also simultaneously kind of starting a little career on ad tech. I spent a number of years there. Left that and have drifted around for a couple of different industries in general working on just software system architecture and system design, how we can integrate data science tools and machine learning. Doing actual implementation, what I call end-to-end from figuring out on paper how we're going to solve a problem to how we realize that as technology. Doing a lot of verification and monitoring and just kind of ML deployment in general.

[0:03:44.3] SR: With so many things going on, what is your main focus?

[0:03:48.3] KP: These days, my main focus is around some key machine learning problems that I'm working on at the moment. For example, there is a site that I'm helping do some spam

detection on. We can build a little bit on the shoulders of giants who've done spam work before but there are some unique things and challenges there that are taking up my time these days.

[0:04:07.9] SR: Now, to get into the topic. People give a lot of definitions to the data science field, but I'm curious to know how do you define data science.

[0:04:16.5] KP: The word I like most in the phrase is science. For me, science and being scientific is following the scientific method and that's generally like three high-level requirements to do that. First of all, whatever you're pursuing is hypothesis-driven, and a hypothesis meets the criteria. That it's consistent with the available data you have. Generally, that's trivial for a data scientist to do.

The second is it needs to be verifiable in some way. It has to make predictions, that can be like an A-B test where you try something and see how it works. It can be hold out sets and k-fold and that sort of thing. Again, data scientist tend not to have too many problems with making predictions. The third and I think most important criteria is that to be scientific something has to be falsifiable. Whatever work you're doing, whatever problem you are working on, it's often easy to come up with some sort of model that seems to work in the use cases you have but the use cases that are working are not the important ones. It's the ones that are going to give you insight or sort of break your model that are important. For me, someone who's doing data science is working on something with data and really following the scientific method and however they're executing it.

[0:05:27.7] SR: Who is a data scientist in your view?

[0:05:30.5] KP: Ha! I mean that's tricky. I think there is about as many definitions as there are YouTube videos answering that question. To me, it's anyone that's understands that to date is what informs us about smart decision-making and follows logical processes and doesn't go with their gut or their instinct but goes with what the data is telling them so long as the data is correct.

You can be someone who is a Ph.D. with a bunch of postdocs and incredibly smart and do that. You can also be someone who stumbled into this and is a whiz in Excel and just knows how to ask and answer the questions.

[0:06:04.8] SR: Interesting that you have to include Excel as a tool for data scientists. You call your podcast Data Skeptic. What is so important about skepticism that you wish to highlight in relation to data?

[0:06:18.2] KP: Yes, what I say skeptic. First thing to be clear, I really mean scientific skepticism. There's a lot of people who misuse that term. For example, people call themselves like vaccine skeptics but the overwhelming scientific majority tells us that vaccines are perfectly safe. They do not cause autism. So people who call themselves skeptics there are deniers. I'm really saying that you want to be a scientific skeptic and it's a very Bayesian idea. It's that you should have beliefs that are commensurate with the evidence available to you.

[0:06:47.8] SR: Interesting. How did your podcast come to be? Can you explain how did it start?

[0:06:53.6] KP: Yeah. I think it's a couple of things that all came together. I've always had a love for the audio format since I was little kid. I was interested in doing a lot of stuff with burning CDRs and getting content to my your MP3 CD player that I had back in the day long before there were podcasts. I've always enjoyed the medium and kind of wanted to contribute something to it.

In my career, I've done a couple of different things in the data science field something I've helped out with on a couple of occasions is doing some M&A activity, like mergers and acquisitions. In case whre a company is thinking, "Hey, should we do a contract with this third-party company? Should we just buy them or should we build something internally?" Asking questions like that.

When I've done work like that, often I found that there's like a hot new startup or at least someone portraying themselves that way saying, "You should buy us for a bazillion dollars and our solution will plug-in."

At least in one especially egregious case, we went through this very glitzy demo that looked really good and when we got under the service and start kicking the tires of what was built, there was literally nothing there. We reverse-engineered the whole process and figured out how it was working and built it in about 10 lines of code, “Guys, this is all you have here. Why would we engage in the service?”

It had come really far in these discussions. There were investors involved and all types of finance people looking at this deal and it collapsed when we realized that that company just had sort of this house of cards. I was asking myself questions about, “How did it get this far?” In that situation, it was that you had very smart business people who didn't understand the technical side and people who were good at marketing and kept throwing around terms like machine learning in AI and that got it to this pretty advanced stage before we had to look under the hood and saw that the use case they were telling us about wasn't actually in fact the case.

I’m being a little vague there because I don’t want to be liable about anything, but the general use cases there that we have all these emerging tools and technologies. I could say we’re in like a new industrial revolution of data science in a lot of ways. Just because we have things that are doing really amazing advanced things for us doesn't mean that those same methods are guaranteed to solve any problem. If someone comes up to me and says, “We can cure cancer. We’re going to use machine learning to do.” That sounds likely. Perhaps you will, perhaps you won't. The fact that you just compiled a model doesn't mean that model is correct and useful.

I start to see a tendency for people to say, “Oh! Well, they used data science, or they used this forecasting package. It must be correct.” That's only half the story. The method can be good and the implementation can be bad, or the garbage in, garbage out from the data. I want to be a source where people could learn about my field but also a place that gets people to ask questions because the truth has nothing to fear from scrutiny and we need to not be intimidated by concepts we don't understand.

I think everything in data science is in my opinion pretty straightforward and easy if you have a good teacher or the right book or just something explained in a clear way and there's no reason everyone can appreciate those principles and be smarter about what we trust and what we

believe. That just because something was built with the methods we trust doesn't mean that it's guaranteed to work out.

[0:10:14.8] SR: That's very interesting. Now, a little bit about your podcast. You have a very interesting format where you do a longer interview type of episodes and then shorter mini episodes with your cohost, also your wife; Linda, have your alternate pick, where you take a take a topic and a high level explanation. What made you decide on these two show formats?

[0:10:38.8] KP: I always knew there'd be interviews. In fact, for me, personally that's the most appealing part, getting to talk to all these interesting people or at least having an excuse to do so. I also really wanted to cover more fundamentals as well, so take a week and say, "We're just going to talk about boosting," and explained that in really common everyday terms that any person can understand.

Not that that they can walk away and go implement it, but enough that they say, "Ah! I get what this concept is for or how to use it, some of the pitfalls." I couldn't really figure out a good way to do those more fundamental things until it occurred to me, "Why don't we do these short explainer episodes," and that's how the mini episodes came to be.

[0:11:15.9] SR: How do you choose your topics for the mini episodes?

[0:11:19.6] KP: It's almost totally by whim. It's whatever I have an idea for a nice way to present something. Some of the time I develop an episode that is there to help prepare someone to — Listeners to better understand the next episode. Recently, we've been doing some interviews about generative adversarial network scans. I thought, "Well, this is kind of a new topic in the literature. What don't we do a mini episode that explains it a week before so that people have that in their minds when they go to hear the interview? Maybe they can appreciate it better if they listen sequentially."

I'd like to say every week is planned like that but sometimes we're just deciding to record one afternoon and it's whatever we can kind of come up with that day.

[0:12:04.5] SR: I must say that I often find the conversations between you and your cohost more of a professor and a student, which makes it more interesting, really. Many of your podcast topics are on statistics. What made you decide in statistics?

[0:12:19.1] KP: For me, the core methods of data science are I think things like machine learning, then statistics, then optimization, and then there's a long list that's difficult for me to prioritize what's the most important concept after. Those are kind of the big three.

I can't imagine really someone being an effective as a data scientist without at least very basic statistical knowledge. You could even make a case for needing more advanced statistical knowledge. I also see a gap in a lot of people getting into data science especially people who have a software background. For whatever reason they can often slip through their training without getting much statistics, so I try and be a resource in that way.

[0:12:56.9] SR: Along with your podcast, you also have some interesting projects, like the Open House project, the SNL causal impact. Explain to our listeners, what is the SNL causal impact project.

[0:13:09.3] KP: Sure. That was one episode we did, and I worked with Karen Blakemore on that. Causal impact is this really interesting package in R and it's based on this paper I think some researchers at Google put out that — Let me see if I can give you the basics in a nutshell.

The idea is that if you have a time series, let's say the number of visits to a Wikipedia page that has some pattern, maybe people read it more on the weekends or less on the weekends. It can also have some trends, someone that's rising in popularity. Let's say a brand-new band that's going to be a hit in five years. There's going to be this upward trend. Then there's seasonal cycles, whether those seasons are weekly or monthly or annually or things like that.

Lastly, there's the residuals out of that construction of a time series, the day-to-day fluctuations you can't explain. Naturally, you can't assume that the recent history predicts the future, but an interesting idea is what if we took a bunch of things that are related whether those are stocks that are similar, like IBM and Apple and Google and Microsoft stocks, and we looked at how

they trend up and down together, which does happen. That's why there are things like index funds, things that are similar seem to move together.

Then when something breaks out from that, you can use those other kind of cohorts, or counterfactual they call them, as a way of measuring what was the impact of that difference. An example might be, or the example we were interested in was what if a band appears on Saturday Night Live? Does that give them a big boost in popularity? We wanted to use causal impact to test that theory or really to measure what is the impact of being on Saturday Night Live.

Then as part of the process, we also built the — Or really, mostly Karen built, this R Shiny app where people could interact with a data set. We actually found, in that case, just to jump to the punch line that there didn't seem to be much boost to that appearance. Our result was appearing on SNL either doesn't boost your popularity as a band or maybe the band is already so popular that whatever the boost is is below the statistical noise threshold.

Now, that all presupposes that our methodology is good, which is really the skeptical part. If you believe that Wikipedia page views are a good proxy for popularity, then methodology is good. I think that's not a perfect proxy but an interesting one. There's a lot of things to still be said about what being on SNL does for you as a musical group, but that's at least what the project was all about.

[SPONSOR MESSAGE]

[0:15:41.2] JM: Don't let your database be a black box. Drill down into the metrics of your database with one second granularity. VividCortex provides database monitoring for MySQL, Postgres, Redis, MongoDB, and Amazon Aurora. Database uptime, efficiency, and performance can all be measured using VividCortex.

VividCortex uses patented algorithms to analyze and surface relevant insights so users can be proactive and fix performance problems before customers are impacted.

If you have a database that you would like to monitor more closely, check out vividcortex.com/sedaily. GitHub, DigitalOcean, and Yelp all use VividCortex to understand database performance. Learn more at vividcortex.com/sedaily and request a demo.

Thank you to VividCortex for being a repeat sponsor of Software Engineering Daily. It means so much to us.

[INTERVIEW CONTINUED]

[0:16:58.0] SR: I believe it's just two weeks ago that you were talking about the Open House project. What is the Open House project?

[0:17:06.5] KP: Open House project, it's open source and open data group aimed at liberating full-grain transaction level real estate data for analysts, perspective homebuyers, data scientist, or we like to say the generally data curious.

Linda, my wife and I, we're about to buy house, a little over, I guess, a year, a year and a half ago. We started getting in the market. I was really frustrated because I couldn't get access to the data I wanted to use my skillset. I couldn't run regressions or do any sort of modeling because the only things I could find online were generally current listings filtered the way some commercial website wanted them filtered. I wanted to look at historical trends and get really granular the ZIP Code levels. When you want that full-grain data, it just wasn't really available anywhere at least in an easily accessible way despite it all being public information with a few exceptions across the country.

After being frustrated and looking around for a while, I thought, "Hey, maybe I can make this into a project." So Open House was created. We've built an API that allows people to push data. We built a crawler and some parsing interfaces and it's actually growing pretty quickly at the moment and the technology landscape is evolving a little bit. In general, we want to be a source where people can come and get access to that full-grain data so they can ask smart questions and get smart answers when it comes to doing any sort of property things like what is often the most important financial decision anybody makes in their lifetime buying a house.

[0:18:34.7] SR: How can SEDaily listeners contribute to the Open House project?

[0:18:38.5] KP: The best way is to go over to openhouseproject.co and then click on get involved. We have a list here of some easy ways to get involved, more advanced ways. One of my goals after we put out the episode is creating ways where people can make what I call the four-hour contribution. Everyone kind of wants to do open source and help out but very few people can commit 10 hours a week for obvious reasons, and it's hard to get into any project from scratch, just jump in and understand the whole code base. We're working pretty hard on our end to make to create accessible tasks anybody can do.

We're in almost like an alpha-beta stage with the launch now. We have really good coverage in a couple of areas, like Los Angeles and parts of Pennsylvania, but it's pretty sparse as well. We're in a data exploration phase. We can use a lot of help there. I would say a few months down the line as the database gets a lot more full resolution of the data across the country, we'd love to have people using this in applications and doing really refined analysis and building stuff on top of it.

[0:19:41.2] SR: Great. I must say the Open House project is definitely very interesting. It did piqued my interest a lot.

Now, to talk a little bit about data processing. In the enterprise world, data processing has multiple steps namely; ingest, store, visualize. Often times, data cleaning is the process which takes the most time, even more so with unstructured data. Why has data cleaning become a separate necessity?

[0:20:08.4] KP: The famous quote is somewhere between 70% and 80% of a data scientist's time is spent doing data cleaning. I actually hope that the data science community can start to use better terminology in the future, because a lot of things fall under data cleaning, some are good and some are bad. Cleaning sound like a chore and sometimes it is, like when a data scientist has to spend a lot of time fixing mistakes developers made in how they logged something or how they unpredictably change the way things get stored.

Cleaning can also mean like removing outliers, or it could be things that just need to be transformed in a very careful way. Those transformations and the data set up are things that aren't so much cleaning as they are structuring. Basically, the algorithms that we use each come with their own set of assumptions and peculiarities. Part about being good at data science is knowing those things really well and how you best prep the data to enable an algorithm to do its best jobs so.

Some of that falls under cleaning, and that's just the normal day-to-day. I think the reason we think of it as such a big process is because so much gets put under that umbrella. Yeah, it's all those steps that take you from the raw data you have to producing some novel result, or at least the step before the modeling, which give you the novel result.

[0:21:27.9] SR: Just to kind of get an idea. What are the tools you use to clean data?

[0:21:32.6] KP: I'm not sure I have a great answer for this. There's like a really long lists of startups and bigger companies that have a lot of tools in this space. While some are great for specific use cases, I don't think there's any clear pack leader there, at least not for me.

To be perfectly frank, I do most of my data cleaning. That is either like specking out what needs to happen or writing the script myself to do those changes, and I'm using a lot of basic tools for that. Just kind of playing around in my Jupiter notebook or something like that. Once I have a picture of what needs to happen, then how do we automate this? Can we use LAMDA architecture, or things like that. Are questions that kind of become — They answer themselves at least for me once we have the plan. If you try and go too quickly to tools, sometimes you get a lot of assumptions and this garbage in, garbage out factor takes place. To be honest, I'm doing a lot of cleanup manually at first until I know what the steps are.

[0:22:26.9] SR: Interesting. To talk a little bit about the statistical relation of your podcast, popular discourse often on data seems to center around knowing how to use computational tools, but computational tools are only one aspect of the three-pronged nature of data science. There is a lot of bias for people, just like you said previously to invest more time on software than on the other two aspects of data science, which statistics and domain knowledge. What is your view on that?

[0:22:56.6] KP: I'm not sure if I know exactly what you're asking.

[0:22:58.5] SR: On popular discourse, you have tools like Hadoop, Spark, Kafka and [inaudible 0:23:04.5] there's a lot of full-stack data science. Often, that full-stack data science is just a bunch of software tools. In fact, often, on your podcast, you have taken topics which are not software, which kind of relate to application. They're just domain knowledge to me. Say, for example, the episode on oceanography, and then you have one — I forgot the name of the person, but you have another episode where you talked more on the application. Then, of course, you do a lot on statistics.

How much should one who's wanting to become a data scientist, like, say, invest on software? Because, oftentimes I see the discourse a lot related only on software.

[0:23:53.0] KP: Yeah. I mean, tools will come and go, and there is a lot of good information available in terms of blogs and what have you that you can find out like the latest about Spark. I barely cover spark at all anywhere on the podcast. Partially, that's because you can find resources out there, but it's also because that's not very evergreen. New releases keep changing things. Also, to really understand the tools, you need something visual. Most cases, as an audio podcast, it's hard to tackle it.

I think the other half your questions is like where should a person put their energies. For me, personally my advice is to focus on methodologies over the tools. Now, you have to know the tools to get things done, but domain knowledge and knowing the methods will kind of, in my opinion, lead the path towards accomplishing something. In fact, domain knowledge can often be the really big differentiator.

Once you have a plan figuring out how to execute it usually kind of falls in place, at least in my experience. That's why even though I'm constantly trying to keep up with tools and see what's new and what other people are doing, once I know what I want to accomplish, usually the tools are secondary and easier.

[0:25:04.4] SR: I want to center now a little around statistics here. You do a lot of topics and many episodes expanding machine learning and statistical methods. Now, why do you see a need for that?

[0:25:16.3] KP: Everything I know about machine learning is really really simple. You can either assume I haven't learned very much or you could assume that I'm the most brilliant person and I can understand it easily. I don't think either of those things are true because I don't think I'm necessarily all that smart or special.

The thing about learning, and this goes beyond data science, but especially within my field is the literature can be really obtuse. Some of that is because it's moving fast. Some of that is because it builds on other literature that assumes you know a lot of things. I'm at the point where I can pretty much pick of any paper in my field and I don't struggle so much with it, but I remember lots of days when I would. I'd read something eight times and I still wouldn't get it.

I wanted to be one avenue by which people could come and learn about stats and machine learning and those things in more accessible terms, because I think the core intuitions are really what's key to being successful. They're all pretty simple if there explained well. The problem becomes when the way they're described is just not accessible to the person trying to learn them.

[0:26:21.9] SR: Now that you have touched on the clearing the paper topic, I do have a question on that. Before we go there, how important is it to know statistical learning as a topic. To call one a data professional, you have the Travor Hastie's, the infamous introduction to statistical learning, and then you have the Use R! series in Springer. How important is that, really, to call one a data professional?

[0:26:50.5] KP: Data professional to me sounds like a much broader term. That can include people like database administrators or ETL engineers. Realistically, no stats is probably required there. For someone to be a data scientist, you have to know at least the basics of what's covered in — The Hastie book is a great one. If you want a more introductory source, I would say Open Interest Statistics for sure is a required — You have to have everything covered in there. It's a free PDF or a very cheap book if people are looking for a resource. Fundamentally,

most of data science is rooted in some way in statistics. People have got to get comfortable there.

[0:27:31.7] SR Leading figures in data science have Ph.Ds, some in computer science, some in other fields, but they don't have a doctorate level accomplishment, and you did touch on this topic before and I often do find reading papers difficult to say the least. Really, the word I was looking for was intimidative. Really, task which is out of my reach at least at the present moment.

How important is it to comprehend academic papers, because there's a lot that's coming on the topic for aspiring data scientists.

[0:28:04.4] KP: Yeah, that depends really on what you want to do with your career. For the foreseeable future, a person could have a really good career knowing one technique, even a very simple technique, using it well and migrating from business-to-business just repeating the process. You could get really good at A-B testing and help dozens of companies in your career or figure out how to improve the sales funnel on their website, something like. You could do that successfully without learning too much because that's a good skill that the market needs. Like I said, at least the foreseeable future until some AI figures out how to do that and make sure that task obsolete.

For me, I think, one of the joys of life is learning, and I see that very strongly in people who want to call themselves data scientists. It's a love of learning, finding new techniques and growing as a professional. For that, at some point you're going to hit a ceiling where you've learned all the literature that is old and you're starting to be ready to take on more cutting-edge things or newer ideas that haven't yet made their way into textbooks or very simple YouTube explainer videos or on the data skeptic or things like that.

At that state, that literature goes into the archive and into other journals and things like that. Maybe not day one, but as you grow as a professional, at some point people need to develop an ability to read academic journals for sure.

[0:29:29.2] SR: As a starting point, are there any papers that you would recommend? Say, you did have a specific recommendation when you are doing the board cloud topic on visualization. In general even, you do a lot of topics on papers. A lot of many episodes, sometimes even all the longer episodes. Are there any recommendations that you have for people wanting to get started on this to experiment?

[0:29:56.4] KP: Yeah. I think you have to fall back on knowing where you are already. Everyone comes from some academic background, I presume, even if that's as simple as like you have a high school degree and you went to some coding boot camp for two weeks. That's great and that's a good start. It's different from someone who spent 10 years in school and has a Ph.D. and that doesn't make the boot camp person inferior. It just means that they're on a different place in their journey. What do you already know the where do you want to go next are your key questions?

For example, in my academic work, a lot of what I did started to rub up against the game theory literature. I was totally unfamiliar with it and I really struggled reading econometrics papers. In fact, I have a lot of criticisms of the way econometrics people write papers. I think they do a bad job often in how they write equations and things like that. They could be much more clear. Of course, I say that as someone with a computer science degree. They may say the opposite about the literature in my field. Overtime, I develop that muscle.

On the flip side I could pick up a computational theory paper and I had the sufficient background to understand that and appreciate it. If you know where your strengths are, read the literature close to. Ideally, it's kind of like on the path you want to be on, but you'll find that when you start to know a field, you know maybe half the people publishing it and publishing in it and you start to figure out authors that you like the way the write, and that's a good introductory way to kind of, I guess, get familiar with reading, growing and being able to read papers better and stuff like that.

[0:31:29.8] SR: Interesting. So I've come across a lot of talk on using deep learning models, informing networks for feature extraction with examples in [inaudible 0:31:38.6] recognition, text classification, but it's not very apparent if those applying deep learning really understand it well

enough. You've done many episodes explaining deep learning and related concepts. How important do you think is the understanding of the underlying mathematics before applying it?

[0:31:58.2] KP: I'm going to say something that's potentially a little controversial here. To be frank, I don't think you necessarily need to understand a lot of the underlying theory in deep learning to use it. There's a lot of really cool work going on for how we can make deep learning models more interpretable or sort of probe them.

I read an interesting paper a while back, the basic idea they had was to understand how image recognition system is working that you could randomly occlude, that is block out or blur parts of the image, and see how that harms the accuracy of the predictor. That's a very big black box style of approach which is kind of something I want to promote here a little bit.

We are coming to the point when these networks are so big, it is really difficult to figure out how they're working. Now, there're some cool tools out there for sure that let us inspect different layers and kind of see what areas they're working on. There should definitely still be work in interpreting models. We shouldn't give up on that.

For a moment, let's just say we give up on it. The model is so vast in size and compute cycles that it's almost hopeless to identify how the intuition is, an abstract level of how it's working. That doesn't mean we can't learn things about it or test it. At that point, that system is very much like another human being. I can't necessarily know all your motivations or how you're going to think or strategize, but I can test that scientifically like an experiment.

Let's say you have a black box model that's using deep learning and does some important task. Let's even say a medical task where it's critical, there's life or death situations here, and all the training looks really good. The K-fold validation looks great. We don't for sure know if it will generalize.

In a perfect world, yes, we'd like to know exactly how it's working, but if for some reason that's off-limits or too hard we can inspect it like a black box. We can try and fool the machine, and there's good work going on in the fooling images side of deep learning. Basically, we can kind of

go back to the falsifiability, that third important principle of being scientific. How can we break the machine and that we usually tell us where its edges are.

If you can build a black box of some kind that does something useful and you invest a lot of time trying to show that it doesn't work and you fail, that's good evidence that you can trust that system. At the end of the day, at least on the practical side, I'm really concerned about solving real world problems and implementation. If you can do that is a black box, then so be it.

[0:34:25.4] SR: Interesting. Now, coming to trends. Now, being in the data field, I'm guessing that you live the data field as supposed to working just in the data field. What trends do you see in different data-intensive industries? You have retail, you have healthcare. Then in finance, you have automated trading bots and then have the fin-tech phenomenon. What other trends do you see in the data field?

[0:34:52.1] KP: I don't know that I have in my career touched on enough industries to speak really broadly. I've never worked in fin-tech or healthcare for example, but I've worked a lot in e-commerce. I know something about the trends there, but maybe I can just talk more at a macro level and say that, obviously, deep learning is a prominent tool all the time. Spark is becoming more the platform of choice.

I actually have two predictions that's I don't know if these are controversial, but I don't hear as many people saying these as I think would be commensurate with how strongly I feel that they're correct, so I'll give you two predictions here. I suspect that serverless architecture or LAMDA architecture, if you use the AWS lingo, is going to get more and more popular especially as there's better tools to do change management and stuff on it like that.

I also think — This may be my more bold claim that chat bots are going to be ever on the increase and that 5 to 10 years from now chat bots will be the new user interface. For that to happen, we're going to need smart data scientists working behind the scenes to make them operate.

[0:35:53.6] SR: Can you elaborate a little bit on chat bots. This is a topic who's just taken on a like a fire in the past six months. You have so much being written on it like never before, and it's a very interesting topic too. Can you talk little bit more on chat bots.

[0:36:09.1] KP: Absolutely. What has changed at least in my opinion in recent times is all the plumbing has been worked out. There are a couple of places to do this, so I don't want to give my vendor preference. I'm agnostic to all of that, but there are many tools out there that will get you up and running and building a hello world bot in about an hour, where if all you bot does is echo back what the person says, obviously it's dumb. It doesn't do anything. Being able to create that, deploy it to Skype to run over SMS, to put a plug-in on your website, basically, to go into Slack to communicate over every possible communication channel, that problem has been solved. All that technology, all the plumbing is available. Now, the question is what do we actually do with it?

If people aren't familiar with the ELIZA model, that's a milestone in AI in speech stuff. That's an old one and kind of a funny story that I encourage to look up. Since I think it's been talked about enough, we can skip it here. The hard part next is going to be how do we put intelligence behind those bots?

Another trend that is important to why chat bots are growing in popularity are API marketplaces and cognitive services. Basically, people are building models and providing them at very low commodity cost for use. For example, facial recognition. Even though it might be fun, there is literally no reason for anyone to start from scratch trying to build facial recognition today. You can go out and use these services that will do that for fractions of a penny. It makes no economic sense to try and build your own model, and some of that is also because the cost of doing that, spinning up a deep learning hardware and running all the trading, someone's already done that.

Now, there's interesting things to be said that we can get into if you want or not about how do we use transfer learning, allow some those basic models to be extended for my specific use cases. By and large, there're all these tool sitting next to the bot frameworks that I can pick up and do natural language understanding and parsing and sentiment analysis that works pretty good in the general use case.

The availability to integrate that in lots of places and spin things up fast as I think what started this revolution. What's going to really make a change and take off? Of course, obviously, you got to given out to Siri and Google Assistant and these sorts of tools that are doing great, and they're doing great not because they're AI, but because they solve very specific tasks really well. I'm always telling my phone like, "Oh, remind me tomorrow at 3 PM that I've got to do something , or remind me that Sid is going to do the interview tomorrow." It knows these — Even though I can say those things in lots of different formats, it's not trying some silly like regular expression way of matching. An intelligence is able to parse those basic domain specific tasks.

If we can do that, businesses can start solving problems that are specific to their domain. I imagine some place like a bank, 80% of the visitors to the site just want to know their balance or something simple like that. They have these very basic questions that the chat bot can probably handle. We'll see industry adoption because of that. Also, the ability for smaller teams to do deep learning work and to take advantage of some of the advancements that are going on in NLP is what's really going to ignite the trend I think.

[0:39:28.0] SR: Do you also see this trend continuing on to the finance field? Even more so, in context of the automated trading bot, or the financial advisor bot. Do you see any revolutionary disruption here?

[0:39:43.1] KP: I'm absolutely open to being wrong, but I don't see any disruption there. I think some of that comes from me being very cynical about finance. I have a very game-theoretic perspective on a lot of finance, and I talk about this once in a while on the Data Skeptic blog. My basic premise is this, if someone knows something, like let's say there's a breaking news story that says Apple found a bunch of gold bars in one of their warehouses they didn't know they had. Immediately, Apple is worth more money, so the stock price is immediately going to shoot up.

The only way you make money is if you're first to that knowledge, and that's why we have high-frequency trading and all these kinds of things, and the barrier to entry is so high, no small group can get involved in that. You have a couple of big players that are really fighting amongst

each other eking out the fractional percentages of some advancement they've made in some mathematical model.

Unless you have private knowledge, which doesn't have to be insider-trading, but means you know something no one else knows, then the market is more or less noise in my opinion I think. I invest almost exclusively in just index funds and things like that. If the market doesn't admit any signals that are predictive, then nothing can be predictive. I guess my core thesis here is that the market doesn't admit very many things.

[SPONSOR MESSAGE]

[0:41:04.2] JM: Your application sits on layers of dynamic infrastructure and supporting services. Datadog brings you visibility into every part of your infrastructure, plus, APM for monitoring your application's performance. Dashboarding, collaboration tools, and alerts let you develop your own workflow for observability and incident response. Datadog integrates seamlessly with all of your apps and systems; from Slack, to Amazon web services, so you can get visibility in minutes.

Go to softwareengineeringdaily.com/datadog to get started with Datadog and get a free t-shirt. With observability, distributed tracing, and customizable visualizations, Datadog is loved and trusted by thousands of enterprises including Salesforce, PagerDuty, and Zendesk. If you haven't tried Datadog at your company or on your side project, go to softwareengineeringdaily.com/datadog to support Software Engineering Daily and get a free t-shirt.

Our deepest thanks to Datadog for being a new sponsor of Software Engineering Daily, it is only with the help of sponsors like you that this show is successful. Thanks again.

[INTERVIEW CONTINUED]

[0:42:24.5] SR: Oftentimes, data tools include R, Python, Scala, and another programming languages, and SQL doesn't get a lot of place except for an ATL applications and that it's still

being used and relational database is still the model. What's your opinion on the importance of SQL in the age of R and Python?

[0:42:45.0] KP: You're making me feel old here, because I'm now the cranky old man saying, "A SQL is super important. Everyone has got to learned it," just the way people gave me that same advice about Fortran 15 years ago. Maybe SQL will go away, but I highly doubt it. I think relational databases will never go away, they're too useful. They solve specific problems very well.

What happened was we had a decade or more of the relational databases being really awesome and a great persistence layer and everyone used them because they were reliable, but not every application is a banking app. You don't need acid compliance and all these things. As people started to realize you could relax those constraints, that's when I saw this emergence of all these other tools. Especially, with big data, they become more important. How we're going to deal with a world in which the CAP theorem is a reality and those sorts of things.

SQL, regardless of what the underlying technologies, it's a wonderful and powerful language. It's very expressive, and even many business people, non-data scientists with no technical background can do amazing things with just the basics of SQL knowledge.

While data is not inherently relational to begin with in most cases, it seems to end up that way in a lot of applications and it would be truly surprising to me if more than 50% of the world's structured data and data that goes in databases was ever not something that could be queried by SQL.

For that reason and for the fact that it's still so prevalent in business, I think everyone should learn it. Also, if you're going to be a data scientist, SQL is the least of your worries. It's pretty easy to learn, and you get a lot of big bang for your buck out of it because that is the keys of the kingdom of understanding data in most companies.

[0:44:28.5] SR: Now that you have actually touched on a really important topic, I just want to kind of digress and ask about — In the list of worries that an aspiring data scientist should have, you've already put SQL on the bottom of the list. What would you have on the top five top five?

[0:44:47.0] KP: Top five. I think, first have a plan for your career, and that's good general advice, but just saying, “Oh, I want to be a data scientist.” That's great. That's like saying, “Oh, I want to be in technology.” What do you mean? You can be a front-end designer, or you can be a systems architect. Even data science is now an umbrella term. I think you need to plan ahead what do you want to accomplish in your career and that should be a combination of the methods that you think are really interesting and you want to work with and the domains you'd like to work in.

Because while you can get by with only one of those skills, being really knowledgeable about healthcare can get you may be a good job and catch up on the methods later or vice versa, but combining those makes you key asset some place.

I mean top five skills really depends on what you want to accomplish. If you're interested in text, audio, video images, you do need to go learn deep learning. If instead you're interested in — I don't know. More like customer modeling and trend analysis and improving conversion funnels in e-commerce, deep learning probably is not to play a role in that. May be some small role, but fundamentally there's going to be a lot of domain specific feature engineering to do and you want to understand the algorithms, like XG boost and random forest and logistic regression in those to solve those problems. The problems you have should drive the tools you use, not the other way around.

[0:46:15.7] SR: In terms of names, at least, what would you put as the number priority for an aspiring data scientist? Would that be Hadoop or would that domain-related knowledge? Would that be statics? What’s your pick?

[0:46:31.0] KP: That's really hard, because data science is such a broad umbrella term. I guess I would say if you’ve got to put one thing on top, it's probably statistics because so much emerges from that. Learning basic statistics isn't going to teach you anything about map reduce, but there are things about map reduce that operate probabilistically. It seems like stats are kind of fundamental and that's why guess I’d put those at the top.

[0:46:56.6] SR: Interesting. Now, to break the glass ceiling of this field, what does it take? Who's qualified? What does it take?

[0:47:09.1] KP: I think this is kind of a different answer for every person given their background. I'll try and give maybe the most general-purpose answer I can give, and that's figure out what you're good at and then go on the trajectory. Wherever you'd like to be, there's a skill gap between what you need to do those things and where you are now, so start applying.

The great thing about data science is you don't — It's not like the chemist where maybe you need hundred thousand dollars-worth of equipment to do any nontrivial chemistry experiments. You can do data science with the technology you already have, so pick a data set, a problem. Find a charity you can help with or better yet a small business; your cousin's carwash, or someplace that has some data that you can volunteer or work on the cheap and just start working. Figure out what the problems are and then look for methods to solve them.

You're going to get a lot further just by doing than by worrying about what to learn, because the doing will create the necessities of invention and you'll also have something to show that will break the — As you call it, the glass ceiling, which I presume means getting a job and getting into the field. That's going to be a lot easier with a portfolio of work than a claim about how many books you've read.

[0:48:23.2] SR: Interesting. In your opinion, what would be a good data science product, or if you have a list of — I'm easily guessing that you would have a lot of opinions or ideas on this. What would be a good data science product?

I just want to set the context here. You have a lot of open data available. For someone who would want to get started on this of what could be something of value.

[0:48:50.7] KP: The most important thing about project is that you finish it, and most projects are a lot harder than anyone anticipates when you start them. The key is to be practical. I think also the best piece of advice I can give is that you have to do it about something you're passionate about. If you open up your city's local data portal — I live in Los Angeles and our open data portal is ranked number one in the country by some survey, and there's a lot of just

kind of stuff in it, like I pulled on the dataset recently for street sweeper data and I played around with it for a few hours and it came to nothing, because I didn't have a question. I didn't know what I was doing with it. I was just kind of looking around, and that's okay. Maybe I would've found something and maybe if I had been more clever that day, something might've come out of it. Have a real problem that's interesting to you.

If you're interested in jazz, get a dataset on jazz and do clustering or do whatever. Explore it. Ask interesting questions about it. If you're not passionate, that project is probably not to get finished. It also helps to have something that isn't necessarily clickbait but is interesting to get people to look at your project and give you feedback on it. Something on the Olympics, when the Olympics comes around, is probably a lot cooler than — I don't know — The best place to buy a hard drive or something like that.

I would also say you want to get a dataset if you're — Earlier on in your career, get a dataset that already prepared. It's easy to have a big dream about crawling some site and parsing out a bunch of data, but those things will slow you down pretty fast.

[0:50:20.9] SR: I'm guessing that you know on the different courses that's available, in Coursera, there's a lot on Edx, there's a lot on Udemy. In your opinion, how would you rate them? Again, someone who's wanting, who's eager to get into this, just enroll in any of those courses and get significant tools that he could get something with them.

[0:50:46.8] KP: To be honest, I'm not sure. I haven't taken too many of those. I mostly kind of have created my own syllabus whenever I'm learning things, and I found my way that way. Obviously, I get a lot of benefit out of watching YouTube videos, so I stumble in and out of Audacity courses that way.

I think overall, the real thing is to find the medium that works for you. I learn pretty well through audio and I listen to a ton of it while I'm doing chores at my desk or just about anything, so I get to enjoy lots of podcasts and learn that way.

I tend not to do as well with textbooks. I buy them aspirationally, but they're intimidating. 500 pages are going to sit there a while, so I tend to bias towards, "Let me read a few papers. Let

me watch some videos I like." Even though I don't have any direct recommendations of the top of my head, it's about finding the channel that works for a person to their learning style.

[0:51:44.3] SR: I think it would be a great disservice if I don't ask you the question. What are your favorite data science related textbooks? Or books even?

[0:51:52.5] KP: Okay, books. Let's see. Recently, I'm almost done with the deep learning book by good fellow, Bengio and Courville. It's excellent. I take issue with actually a few parts being a little unclear. Overall, that's the premier textbook. Definitely pick that up. Russell and Norvig for AI. I've already mentioned Open Intro. Elements of Statistical Learning is sort of a gold standard. Those are the main books that are coming to mind off the top of my head.

[0:52:19.5] SR: Okay. Now, we are coming to the end of the interview, and I would just want to ask some open-ended questions. I've sometimes come across the idea that quality of data really is nice to have. If you have enough data, then you can make something of it. What's your opinion on that?

[0:52:40.5] KP: Yeah. Quality of data is the key. If you get all of the historical data on every lottery that's been running in the United States for the last hundred years, it's very unlikely you're going to build a model to predict the lottery. There's just nothing in the training data that is predictive of the output.

I think maybe that's what separates in my opinion, like a junior data scientist from a senior one. Someone who can step back from a problem and ask questions like, "What do we actually expect here? What's the upper limit on predictability? How well will our training data correlate with our objective function?"

Knowing those sort of glass ceilings, if you will, in your data set, will often tell you, "Do I have enough data here to learn a useful model? Is the data even accessible?"

I like to think of data often with analogies in either biology or in like industrial settings. In biology, we have all our five senses, and they're pretty good, but they're not perfect. For me, hearing for

example is one of the more fallible ones. There are tons of times I think I heard something in front of me and it was behind me and all those sorts of things.

In an industrial setting, you have veto sensors all over the place taking temperature or whatever measurements they need, and the manufacturer of the sensor will tell you the precise conditions under which it can work. The sensor fails below a certain temperature and its measurements have a very specific low standard deviation and all these sorts of things. That is the way we have to look at our dataset.

If you're working for some eCommerce company, one problem will be that people create new accounts. Can you link those two accounts? Sure to some degree, but not perfectly. You can cookie them and do all that sort of stuff, but they can be multi-device. At some level, the problem is not solvable perfectly, so you have to know kind of how good can we get it, where do the trade-offs lie?

If the business needs to invest a half a million dollars to improve the tracking, you have to be able to prove out that that investment will give you data that allows you to make smarter decisions and get something out of them.

In fact, there's a nice little mnemonic for this, the value of information equations says, "What is the value of some information?" And it's all about the decisions you make. Let's say you're going to buy a car and you have the option of getting an inspection or not. You can choose yes or no to buy the car with or without the inspection. Hopefully you make a more informed decision given the inspection. Take the utility you'd have from making the decision with the inspection, subtract the utility you'd get making the decision without the inspection and then subtract the cost of the inspection and that gives you the value of that information.

Being able to apply that will tell you a lot about like knowing if the dataset needs to be cleaned more or improved more and if it can be. I think it always comes back to that, because I've rarely worked on a dataset that didn't have some issue in how it was recorded that came from choices human beings made. Not to say that people made mistakes, but especially at startups, you are constantly changing and doing what you need to do to get things out the door. That will leave

you pretty scarred dataset in general. Do we want to try and repair it, fix it, move forward? Asking those sorts of questions are really what are going to set the bounds of your success.

[0:55:54.7] SR: Great. Now, to wrap the interview, what's the best advice that you would give to an aspirant?

[0:56:03.1] KP: An aspiring data scientist?

[0:56:05.0] SR: Aha!

[0:56:06.6] KP: I would say there, it's figure out what you want to be doing. Data science, like I said, is so broad. It's kind of like saying, "I want to be an athlete." Well, "Okay. You want to be an athlete? Pick a sport." Once you know your sport, figure out who are the leaders in that sport. How did they get there? And that your roadmap. Go do everything those people did. As you get about halfway there, start to ask questions about if they were on the right path are not, or if their path is the right path for you.

[0:56:34.7] SR: What is the worst advice someone ever gave you?

[0:56:37.3] KP: Worst advice someone ever gave me. To be totally honest, I don't really know. I don't tend to tabulate mistakes. I just try and get back on course as quickly as possible.

[0:56:51.6] SR: I suppose that's the best advice.

[0:56:54.4] KP: Maybe.

[0:56:55.3] SR: Okay. What's your favorite editor?

[0:56:58.1] KP: Audio editor?

[0:56:59.1] SR: No. Text editor, like coding editor.

[0:57:03.1] KP: Oh! Sublime.

[0:57:04.6] SR: Oh, okay. Yeah. I have Atom, but Sublime is set up too. Would you have any shout outs for not just books, you've already done books, but for YouTube videos, or podcast, or courses, or audio books even that you would want the audience to kind of take a note of.

[0:57:27.8] KP: Yeah. All right. Let's do podcasts first, because there's a lot of us data scientist podcast out there. I know I'm going to leave somebody out because the list is long, so I apologize to all my peers. Data Stories, Partially Derivative, Talking Machines, Learning Machines 101, Not so Standard Deviations, Linear Digressions, Becoming a Data Scientist. I think those are pretty much the coverage of it. I listen to all of them. They're all great each in their own ways, so figure out the ones that suit you as a listener. Then as one off, because this is a new show that just came up that I'm liking, there's something called Science Solve It, which has been really good.

Then videos and courses and stuff. I wanted a refresher recently in computational complexity theory, so I went through a course on YouTube that's from the MIT OpenCcourseWare, and the lectures were excellent. I definitely recommend that if you're interested in computational complexity.

Videos — There's a lot of stuff out there. We've talked about some of the bigger names. There's one that's newer that I just discovered called — And I don't know how to say this. Something like Kurzgesagt, and the second tagline which is probably easy to search for is In a Nutshell. It's not so much data science, but it's fantastic science content. You don't really need any background. There are well told good animations and whatnot. I guess that's pretty list of recommendations.

[0:58:50.5] SR: Do you have any recommendations of any books, like nonfiction books or audiobooks even?

[0:58:56.7] KP: To be honest, I don't read too much nonfiction, but if you're looking for more, like armchair reading or the kind of stuff I read on flights. Jena Levin, the physicist, is one of my favorite authors. She's written three books, the first was called *How the Universe Got Its Spots*. The most recent one is called *Black Hole Blues* which is all about Ligo, and the one in the

middle which is a work of fiction that's very — Like breaking the fourth wall at times and really good is called A *Madman Dreams of Turing Machines,* and it sort of follows the fictionalized versions of Kurt Gödel and the Alan Turing and some of the other giants of the scientific literature and whatnot. I guess for that armchair reading, yeah, go check out Jena Levine.

[0:59:40.5] SR: Now, for your pitch on thinking skeptically off and with data.

[0:59:44.7] KP: Yeah, that's my tagline that I try and end all my shows with. I guess it comes down to this, data is really bringing us, what I think of as a new industrial revolution in many ways. It's almost magic in some cases. Anyone who — I don't know. Maybe 25 or older like myself can remember a day when search engines were absolute garbage and then suddenly Google. It's easy to kind of gets — To be trusting and say, "Oh! Technology is going to solve every problem, because it's solving some really hard problems and doing them in really interesting and very effective ways."

What I don't like is when somebody says, "Oh, we solved the problem and we did it with machine learning and that's the end of the sentence." As if that just made it intrinsically correct, say, "Because machine learning or because AI." Having a useful tool does not mean that what you did with it was the correct usage or is effective in some way.

I think data scientists need to ask inquisitive questions and go beyond the exaggerated headlines and talk about how we can empirically test things and measure our claims. Data is both the tool by which we — And the methods of data science are the tools by which we can analyze claims, and it's also something we want to be questionable of when someone hands us data and says, "I've come to this conclusion based on this data. Does the data really address the problem? Were the methods implemented in the proper way?" I think it's two-sided. Data the most powerful thing we have or one of the many powerful things in our arsenal for solving problems, but it also is not to be used without some forethought and skepticism.

[1:01:25.8] SR: It was great to have you on the show. Thanks for coming. It was great talking with you, Kyle. I really appreciate it.

[1:01:33.1] KP: My pleasure. Great chatting with you. Thanks for having me. I'm a big fan of the show, so it'll be weird to hear myself on SEDaily one of these days.

[1:01:41.9] SR: Thank you, Kyle.

[1:01:42.1] KP: Thank you.

[END OF INTERVIEW]

[1:01:48.0] JM: Artificial intelligence is dramatically evolving the way that our world works, and to make AI easier and faster, we need new kinds of hardware and software, which is why Intel acquired Nervana Systems and its platform for deep learning.

Intel Nervana is hiring engineers to help develop a full stack for AI from chip design to software frameworks. Go to softwareengineeringdaily.com/intel to apply for an opening on the team. To learn more about the company, check out the interviews that I've conducted with its engineers. Those are also available at softwareengineeringdaily.com/intel. Come build the future with Intel Nervana. Go to softwareengineeringdaily.com/intel to apply now.

[END]