Catalyzing Computing Podcast Episode 1:

Interview with Suresh Venkatasubramanian Part 1

Intro [00:00:10]

Hello. I'm your host Khari Douglas, and welcome to Catalyzing Computing, the official podcast of the Computing Community Consortium. The Computing Community Consortium, or CCC for short, is a programmatic committee of the Computing Research Association. The mission of the CCC is to catalyze the computing research community and enable the pursuit of innovative, high-impact research.

We are joined today by CCC council member Suresh Venkatasubramanian. Suresh is a professor at the University of Utah. His background is in algorithms and computational geometry as well as data mining and machine learning. His current research interests land in algorithmic fairness and more generally the problem of understanding and explaining the results of black box decision procedures. Suresh received a career award from the NSF for his work at the geometry of probability as well as a test of time award at ICDE 2017 for his work on privacy. He joined the CCC council this year.

Interview [00:01:10]

Khari: Your background is in geometry, right?

Suresh: Yes.

Khari: How did you decide to study computer science?

Suresh: The earliest memory I have of being interested in this was basically reading a book about the Turing Test. I grew up in Delhi and there's this British Council library, which is a nice library we used to go to. There was a book there on the Turing Test, so I read that book. I think I read Gödel, Escher, Bach around the same time, maybe a little later. That's when I first started getting intrigued by this idea. And you know computers they weren't really… this was the early 80's, so computers weren't quite a household name yet. But this

idea was out there and my father had actually done some punch card stuff at the time. So one thing led to another, I got a computer and started fooling around and I think things went on from there. It wasn't clear to me that I would be doing computer science. I didn't know what computer science was at the time but everything was very interesting.

Khari: Do you know what the book was that inspired you?

Suresh: No, I don't - it's tragic. It wasn't even a particularly famous book. It was just a book that happened to talk about Alan Turing and the Turing Test. Not even Turing machines — the Turing Test. I am now realizing the strand in my thinking — it was a philosophical thing. How do you think about what it means to have a mind and to have intelligence? The Turing Test was framed in that context. I feel like philosophy has been haunting me ever since then. I do a lot more work talking to philosophers now, but I feel like it's been there in my background ever since then. Those are the kinds of questions that appealed to me.

Khari: So how did you end up in the United States from India?

Suresh: At the time — and this is not so true now, but it was very true then — when you were a student in school in India you had to essentially decide (or at least make some kind of decision) about what you wanted to do with your life when you were in 10th or 11th grade. And the reason is because there were these streams: either the engineering stream or a medicine stream. And those are the only real streams that led to a profitable profession. Anything else and your parents would get worried about what you're going to do with your life, and it would be very full of tension. I know because my sister went through that.

You had to kind of say: am I an engineering person or a medicine person? And really it boiled down to whether you like math or not. If I like math I'm an engineering person, if I don't like math I'm a medicine person. I liked math so I became an engineering person.

By then I was very much interested in computer science. Once you decide to do engineering what are you going to do? You're going to apply to the IITs, which are the biggest and most famous - the Indian M.I.T.s if you wish.

Khari: How big of a school is that?

Suresh: It's multiple institutions. Right now there are fourteen of them (at the time I applied there were five) and there was a common national entrance exam. So it's like two years of coaching to sort of get ready for the entrance exam. Then you took the exam and you get a rank — one out of a hundred thousand or whatever — and then based on that rank you would get a chance to bid for an institution and a discipline. So it's kind of weird, right? Basically if you did well you got to choose the popular disciplines and if you didn't do too well you got to choose the less popular ones which had very little to do with what your actual interests were.

If your interests were in something unpopular that was fine, but if you didn't do well on the exam and you wanted something popular then you'd have trouble. Of course, computer science was a popular discipline. Luckily I got a rank high enough that I could bid for computer science, so then I started doing computer science. Then once you do that, once you're in college, again your options were pretty much circumscribed, at least in the IIT that I was in. You had three choices: you either go to grad school, you go get an MBA, or you go into the Indian Civil Service. These are all sort of solid career tracks. Coming from an IIT you get the stamp of approval that makes you sort of highly qualified for any of these things.

Khari: So grad school always seemed like it was the next logical move?

Suresh: I always thought that would be the next step. I didn't think about I want to do a PhD, but it seemed like the obvious next thing to do.

Khari: So you always thought about going to grad school, and you already had family in America…

Suresh: Yeah I was born in England, and I grew up when I was very young in England and America. So even when I was in India, there was always a sort of connection, and I had uncles who are in the U.S. My sister came to the U.S. to do grad school in English actually, so there were people here already. It wasn't unfamiliar for me.

Khari: Okay. So then you went to Stanford?

Suresh: Yeah, so the funny thing was that I came to Stanford thinking I would do AI. Then I went and talked to a professor, and they gave me a proceedings of a conference to read. I read those papers and was like this is not what I wanted to do. So I went hunting around...and my other interest was algorithms, so I went to talk to Rajeev Motwani, who was there at the time. One thing led to another, and I started working in algorithms. This is a thing for incoming grad students — it's okay to not always know what you want to do. It happens; it's happened to me.

Khari: **So how did you get into geometry?**

Suresh: So geometry, for people who don't know this, it's not the stuff you learn in high school with Euclid. I mean it is, but it isn't also. It's computational geometry, which basically means if you're doing algorithm design but your objects of study are not, you know numbers or graphs, but they are, let's say, points and lines and planes — the study of designing algorithms for those objects is called computational geometry. At some level computational geometry is a fundamental tool that shows up in graphics. It shows up in sort of any kind of modeling. It shows up in machine learning. It's a very fundamental sort of operator or set of tools you need to understand the world around you. Anything with geometry has shape. Anything with shape we try to ascribe meaning to. We do this with machine learning all the time now, and you don't realize that underlying all of it is some kind of geometry of the space that we're working with. So it's a very fundamental sort of set of questions. As you can see, I still love it.

[00:06:41] **Khari: Yeah that makes sense. So your dissertation at Stanford was on pharmaceuticals? Making drugs?**

Suresh: Yeah, making drugs. It sounds very cool. [Laughter]

Khari: So what kind of drugs did you make? [Laughter]

Suresh: I didn't make any drugs, sadly. But it was fun.

Khari: Okay. So we weren't making drugs but what did you work on?

Suresh: Pfizer had come to my advisors at the time: Jean-Claude Latombe and Rajeev Motwani. Interestingly enough, this is still a question right? So, you want to design a drug. The way people think about drugs is that they operate in something like a lock and key mechanism, where the lock is a protein, a drug is a key, and with the right key in the right lock you can unlock an effect wanted in the protein. And that's how the drug takes effect. Okay, great. But a drug is a complicated sort of floppy thing. Proteins are fairly rigid. They have a shape like a lock. But think of a drug like a key made out of rubber. It sort of changes a shape quite a bit. So figuring out what part of the drug molecule actually is having the effect...it's like saying well is it the pointy part of the key or the round part of the key.

We don't really know which part is having an effect. So all you can do is: you get a whole bunch of drugs that all seem to have the desired effect, but you don't know why and they all have side effects. Can you figure out what's in common between them? The common key part of all of them that's actually having an effect? Because if you did that you could then design a drug that only had that bit and nothing else right.

Khari: So that's where the computational geometry comes into play?

Suresh: Exactly. You can think of this from a shape. So people had been doing this already, but they were thinking of this by looking at the chemical formula — looking at commonalities in the formula — and then people said well you know when you actually put the drug into someone's body it has a shape and that shape matters. We need to understand the shape and not just the letters and the formula. That's where people like me come in.

Khari: Okay. So after Stanford you ended up at AT&T lab?

Suresh: Right.

Khari: How did that transition happen?

Suresh: You do your PhD, you apply for jobs, you get offers, and you decide. I had an academic offer, I had an industrial lab offer, and I talked to a bunch of people and they gave me all kinds of advice — none of which made any sense. So I spent three months agonizing over this to my girlfriend at the time (now my wife). She got sick of me and said she doesn't want to talk to me about this. And finally I decided to go to AT&T. For me, at the time, it was a tough choice. I think like most choices in retrospect it makes sense, or it makes sense in the sense that you rationalize it so it makes sense. [Laughter]

At the time it was difficult, but I'm glad I did it.

Khari: Yeah. What kind of advice — for people that might be listening and struggling with academia or industry or two different job offers — what kind of advice did you get, and why was it not helpful or helpful?

Suresh: I think the conditions right now are so different from the way it was then. I don't know if anything I relay now would be useful, but the advice on the side of going to industry: if you want to end up in academia then one way to think about the industrial job is to treat it like an extended postdoc. You do a lot of work. You have very few responsibilities. You can do all kinds of research, set up a direction, and then you can get to academia.

This is good because now you have a nice research track going. So once you get to academia and you're slammed by the thousand things you have to do as a professor, at least your research is chugging along. That advice I think is correct even today, and you see people doing this a lot. Like people will get faculty jobs and they'll still take like a one year or two year postdoc to do something else. And I think it helps them, because you get that level of maturity and you get that level of seasoning of how to think about your own questions.

So that's I think solid advice. I don't know if the companies are very happy about me saying this, but I guess this is conditioned on if you want to go to academia eventually. If you're still dithering about whether to go to academia then I think it's a different set of questions right? So maybe go to industry, try it out for a bit, see what it's like. You know what the academic life is like, go see what the industrial life is like and then see what happens. My worry was you go to industry you get sort of stuck in a particular position for a while — it's

hard to maintain your research viability and you have to be careful about that. I think it's very easy to get sucked into whatever projects you're working on in a company and then forget about the fact that you got to maintain external viability if you want to go back to academia.

Khari: So what's different about the research you're doing in industry that would limit your transition to academia?

Suresh: Big companies, small companies, startups - they're all very different, right?

So let's say if you're working at a big company like Google it's possible you can spend time just working on sort of standard academic type research where you write papers, but of course you may be working on very cool stuff that you can't write about, you can't talk about. You may be working on very fundamentally important things but they don't involve the certain kinds of novelty that are prized by the academic environment. And sometimes the right thing to do is not the novel thing to do — the right thing to do is just take something someone did 10 years ago but just engineer the heck out of it and make it work. That's very valuable but it's not — I want to be careful here — it's not always viewed as academically valuable in the same way. And it really depends on the perspective. The default mode is that novelty is what's prized. But there are a lot of caveats here. I don't want people yelling at me.

But I think that's often a challenge. I think the reason why people often go to industry from academia is that they want to have that experience of really deploying, engineering, building something concrete and get the value for doing it, which you may not get value for or be able to do with academia.

Khari: Okay, that makes a lot of sense.

Suresh: Yeah. So the advice I usually give students now is: think about what is it that excites you. If you're building things that are exciting, working with a team, then you know, maybe some kind of industry environment is good for you. If sort of thinking about long-range problems, thinking about hard questions where there's no easy solution, thinking in that

timeframe is important to you then maybe academia is the right place. But these are all maybes and really it's a very personal decision, and that's why it's hard to give advice.

Khari: So what kind of things did you work on while you were at AT&T?

Suresh: Just a whole bunch of things. You know one thing...at a company like AT&T at the time in the research lab, one of our jobs (among other things) was to be kind of the consultants for the internal business groups. So they wouldn't come to me directly but they might come to my boss or my boss's boss and then that would eventually route through to me. There's this question, maybe you can help.

There was one particular project that lasted a couple of years where they came and talked to us about something and then we had some algorithmic solutions. And that led to a lot of engineering work and that was quite helpful for a long time. I mean I'm not sure I can say much more than that, but it was definitely a fruitful collaboration. There are a lot of smaller-scale collaborations along those lines.

Khari: What was the time frame that you were doing all of this? When did you go to grad school and then when did you go to AT&T?

Suresh: Oh now you're going to ask me an awkward question. Okay, so I graduated in '99, almost 20 years ago. This is my three-year plan that became a seven-year plan. I had a three year plan to get to AT&T and then go to academics. That became a seven-year plan because I was lazy and I was having fun. Then that's what I moved to Utah. I've been there since.

Khari: What inspired you to move to Utah? Just you got a job there or had you always liked the West?

Suresh: I didn't know anything about Utah but I was looking for academic jobs, and so I had done my interviews. I liked the place - I came here and it was nice.

Khari: You live in Utah? In Salt Lake City?

Suresh: Salt Lake City.

Khari: How's that?

Suresh: It's great. It's in the mountains. People don't realize how...you know people talk about Denver right, but Salt Lake is basically the same height. So you're 5,000 feet above the ground and the mountains are right there. I'm 20 minutes from the slopes.

Khari: Do you ski?

Suresh: I'm literally 22 minutes from the best skiing in the country. It's a good place to be. We're all waiting for right now - some resorts are already opened up in a couple of places. Usually after Thanksgiving they're all open.

KHARI: Do you ski at Snowbird?

Suresh: No, I don't go to Snowbird myself but I could. I mean there are basically four right next to my house, and I go to another one which the locals know but the tourists don't...
Khari: Can you say it or...

Suresh: ...so I'm not going to tell you what it is? [Laughter]

Well actually it's called Solitude, appropriately enough. But I have been to Snowbird a couple of times. It's just a question of access. Snowbird and it's partner are in a canyon that tends to get more snowed in, it's more narrow. So sometimes it will be closed up, whereas the other one is a bit more opened up.

Khari: All right, well that's nice. Of the classes that you teach now what has been the most interesting?

Suresh: I've taught a bunch of stuff. I've taught a lot of these standard theory classes such as graduate algorithms, geometry, randomization. I taught a seminar one summer on different failed approaches to proving P vs. NP which was fun just for me to learn. Sadly the most fun

class happened to be the one where I didn't know anything about the topic and tried to learn it. So it was great for me but not so much for the students. [Laughter]

I think all these classes were challenging initially because I didn't want to just...I couldn't bring myself to take a standard course and copy it. I had to be annoying and do my own thing. Trying to articulate my own vision for what I wanted out of a class like algorithms. How do you come up with a brand new interpretation of a Beethoven sonata, right? I mean people have played that one hundred thousand times. The same thing — if you're gonna teach an algorithms class. There is basically the standard way to do it. How are you going to bring your own interpretation into this? So that's something I struggled with for a bit trying to figure out what is it that I wanted to say when I taught algorithms to students. I had to come up with that form. That took a couple of years to come up with a formalism for that.

Same thing with geometry. I have a way of teaching it that I think is not standard and articulating that clearly in class was also a bit of a struggle. Probably because the hardest class I taught though was this recent class on the ethics of data science, which is something I was very ill-equipped to do but I did it anyway. Mainly because I'm not a philosopher, I'm not an ethicist, I'm not trained to talk about it. The way you teach those classes is very differently than the way you teach an engineering class. Engineering class is like here is the problem here is the answer, here is the problem here is the answer. In ethics the solution is to discuss and look at the perspectives, the multiple perspective. You're not coming up with an answer - you're learning how to apply these perspectives and students find it very difficult. I found that very difficult, but it was fun and it was very illuminating for me also.

Khari: I guess ethics is really about the framework — the process that you arrive at the answer not so much the answer itself.

Suresh: Well, I think the answer is very important. This will get me in trouble again, but I think it would be nice if the social sciences spent more time thinking about the answers as opposed to the perspective. I think one of the fundamental conflicts between computer science and social science is that we are always looking for answers. And I think that many in the social sciences are looking for perspectives. You need perspectives otherwise you come up with the wrong answer, but you need an answer or else you don't know where you're going. So I would say that yes, ethics is a lot about process and perspective but you

also want to get somewhere — especially when we're having these hard questions about the way we see algorithms deployed in the real world. It's important to recognize the problem and I think that's what computer science is not always very good at. Having multiple perspectives and ethical frameworks allow us to recognize what the problem is, and articulating it clearly is important but then you need to know what to do. So I think this constant tension is there between recognition and solution.

Khari: I mean, that's definitely true. As someone with sort of a social science background...

Suresh: Oh you do?! I hope I didn't offend you. Or if I did, I hope you'll forgive me. [Laughter]

Khari: Well I studied international studies and economics. The challenge is always if two people are in conflict, resolving that conflict involves someone losing so...

Suresh: You must be an economist. [Laughter]

Khari: Yeah, yeah.

Suresh: But things are not always a zero sum game, right?

Khari: They're not always zero sum but they're frequently perceived that way by people which is what matters.

Suresh: Right, I think getting beyond this...that's the thing if you if you only have one perspective then everything's a zero sum game. I mean they've done this in negotiations, in academic negotiations, if everything is about your salary then there's really nothing you can do. You'll ask for something and they'll give you something and you either decide if you want it or not. But if you make your space of negotiating more multidimensional, you have different perspectives, then you can give somewhere and get somewhere else. And I think understanding that in the context of algorithms is also helpful. It's not just about the accuracy, it's not just about the buyer - it's about a bunch of different factors that you can

understand. That's where some actual, I won't say compromise, but real agreement can come from.

Khari: Are there any examples of algorithms that you can mention that resolve things in multiple dimensions? When you hear about algorithmic bias you hear about, say, sentencing in the criminal justice system, which obviously seems pretty zero sum. But maybe it's not or maybe there are there other examples?

Suresh: But it's not; so even there it's not. The most obvious example I think is cake cutting. This classic 3000-year-old problem of how to cut a cake. This whole "I cut, you choose" thing. So the idea is you have this cake and it has a bunch of stuff in it and two people want it. How do you decide to apportion it fairly among the two people? The goal is not to divide it up equally because maybe one person loves icing and the other person hates the icing. The goal is to divide it so that every person feels they've got what they wanted. And so the whole fair division problem is how do you do that. The "I cut, you choose" idea basically says look if I cut it and you choose then if I cut it badly then it's my fault. If I cut it according to what I want and you take what you want, we're both happy. That's a very simple example of how, in fact, the more people disagree on what they like the easier it is to solve this problem.

It's a weird paradoxical thing where you know if you hate cake and love icing and I hate icing and love cake, this is a very easy problem.

Khari: Yeah, you just give me all of the icing and you eat all the cake.

Suresh: But if we both love cake and hate icing then we have a problem. So when we agree on the value it's harder to divide than when we disagree. So this is one of those things where disagreement is a good thing. But you have to identify the multiple dimensions along which you might agree or disagree.

Khari: Okay.

Suresh: Now in the case of the sentencing system. First of all, I'm not saying that it's an easy problem — it's a very complex problem and I don't think algorithms are the solution but

they're part of a larger solution. But even there you have to be more careful about saying what are your goals here. If you're trying to do a pretrial risk assessment, do you release people, for example? It's a big topic right now in the world and there are multiple things you could say. You're worried about people not showing up in court, you're worried about people committing a crime while they're out before trial. You have different ways to keep track of whether people can show up in court. People may not be able to shop in court because they have a job and they can't afford to leave their job and so there's all these factors. If you start considering it, the problem becomes more nuanced and more complex, but also the solutions start appearing. And I think avoiding sort of a single binary release or not and allowing yourself an option space that's broader means that you can actually find a way to get somewhere that keeps all the stakeholders a little bit more satisfied than they would be.

Khari: So would that involve, say, probationary periods of various kinds or...

Suresh: Yeah, supervised release, or monitoring systems - all kinds of things. I'm not endorsing any of these, but I'm saying that the more options you have on the table the more wiggle room we have to play with possible solutions that can satisfy different stakeholders. That's something that's hard to do in algorithms sometimes.

Khari: That's definitely good food for thought. You're also involved with the ACLU. Can you talk a little bit about those projects?

Suresh: I think sort of spiritually I've been a member of the ACLU all my life. In the sense that, ever since I came to this country I kind of felt that being an immigrant, being a brown-skinned immigrant, that the ACLU is the only entity that is going to have my back no matter what. You know political waves come and go. An organization like that that's built on principles, it's….they're always going to support or help. I've seen this happen time and again.

So I recently, 2017, I got on the board of the ACLU in Utah. I'm very grateful for that chance to be there. The ACLU in Utah does amazing stuff and I think that what people don't often realize (because they only see the court cases) is that the ACLU is not just about court cases — it's only one of the many things they do. Often times (and we see this in Utah) the ACLU

will work with the politicians and they'll work with the legislators to make sure a bill looks better or make sure a bad bill doesn't get through. You won't hear anything about this. It will all be under the cover but it's a large portion of what they do. So lawsuits are kind of a last resort when other things fail. So I think this idea of you work with the government agencies and other groups to make something happen: you find points of agreement, you find points of compromise, and ways to make something work, as opposed to just sort of flame throwing. That's something that I didn't know about the ACLU and I'm very impressed. I'm very glad that I'm a part of that group and what they're doing there.

Khari: You're also on the research advisory council for New York City's effort to build a failure to appear risk assessment tool. How does that work?

Suresh: Basically they're building a new tool to do risk assessments and they've brought in a bunch of people who have expertise in different aspects of this problem. It's not just computer scientists, there are criminologists, lawyers, public defenders, people like that to see how they're building the tool and get our opinion and have us weigh in on some of the things they're doing.

That's been a very interesting experience to see how things get built from the inside.

Khari: Very cool. Has getting more involved in how the political sausage gets makes you want to run for office or even consider a thing like that?

Suresh: [Gasp and sigh] No, no, no, no, no, no, no, no, no. Having said no enough times. [Laughter]

Khari: Not even city council or…?

Suresh: I don't think I have the skills. I mean I can see how it gets done, but it's not that I don't want to because it's distasteful. I think it's very hard work and kudos to people for doing it. I don't think I can.

I don't have the kind of personality that goes along with actually being in politics. You have to like people at some point. [Laughter] You have to be able to interact with people and I'm

much more comfortable in the academic sort of the research-world thinking. One never says never with these sorts of things, but I think my skills are best where they are. Just being a researcher who thinks about these questions.

Khari: Could you maybe pitch people on getting more involved? You know people who are listening to this who've maybe thought about getting involved in their community local government, using science to materially effect and improve people's lives.

Suresh: Yeah definitely. I mean there's no question that we are heading into a world of data science and algorithmic thinking. We're already there in many ways. And people like me, people who are trained in computer science, have skills we don't realize we have: to model, to compute, and to understand how the broad tools of machine learning and algorithms work in ways that others just do not have. Just like many years ago lawyers decided that being poor was not an excuse to not be able to have good representation in court and they built a whole pro bono system around defending people who could not defend themselves with the whole public defender system. I think we need data scientists to do pro-bono work, especially in local governments.

You see a lot of problems with local governments where they would like to do something useful with data they have, they just don't know how to do it. They don't have the technical know how or the resources to do it. Especially if you work in data science you can make a good living working at your job, but you have skills that could help really make things better in a careful nuanced way. I'm not saying wield the algorithmic hammer and sort of claim you can use technology to solve everyone's problems. I don't think that's true, but I think you can work with the local governments. You can work with local authorities to understand *their* questions, come to them on *their* terms, and be useful in ways in which, in places where they may not have thought about or may not be able to articulate the ways in which technology might assist in what they're doing.

There are lots and lots of places where this can happen. Tons of them, and I think that's a really great chance for data scientists, even if you feel you don't want to get involved in politics. You don't have to! A lot of what happens in local government is not politics it's just people trying to get stuff done right. We see national politics at that level but a lot of local

politics isn't like that. It isn't as kind of red or blue or whatever it is, it's really just you have stakeholders. They have different ideas on what needs to happen but they're not fighting each other they just have different points of view, and if you can bring in technology to help understand the different points of view and help come to some solution then everyone is satisfied. No one no one walks away thinking it's a zero sum game. And I see this happening in Utah a lot. Even though Utah is perceived from the outside as this red state, I think internally there's a lot of understanding that people need to work together to make things happen and you see that there's opportunities to do stuff there.

Khari: Yeah I think general local politics tends to be less divisive because there's a road we need to fix. How do we get the money or hire someone to fix the road?

Suresh: I don't want to minimize the importance of policymaking at the national level. That really governs the big money and the big enterprises, but a lot of stuff happens at the local level as well and that has real effect. I mean if you come up with a better system, like you said, to help manage resources for fixing potholes people really get affected by that. I mean that's a real thing. It might not be a big thing that splashes on the New York Times, but it's an important thing. If we are willing to say OK it doesn't have to be high-profile but it has to have an impact.

You know you're a database person, you know how to manage data, that's a big thing. A lot of what local government struggle with is how to collect data and how to manage it. There are a lot of places where computer scientists can help with local government and do good things without it being political or controversial in any way.

Khari: So thinking like an economist, what kind of incentives would you suggest for governments to encourage more data scientists to get involved? I don't really know how the pro bono system works for lawyers.

Suresh: So the pro bono system works for lawyers because law firms and law schools enforce it. In other words, law school...not that they enforce it, but there is a culture built up around it. It's understood that if you work at a law firm that you'll do some pro bono work and at that time it's kind of like the old Google 20% time used to be. I mean there's like a cultural built-in expectation that you will do this. I would like to see any department that

teaches data science to strongly encourage students to look for opportunities to do pro bono work, to help them find those opportunities by building connections with local government, and then telling local authorities “hey you have some data problem. We have students who can help. We can do some consulting for you and it's pro bono work. So hopefully that will not cost you too much. We just need your time and your expertise to understand the problem.” I think it's one thing to do and if we can pitch local governments this would be great.

But again people are very busy and there are lots of issues with data privacy. There are a lot of issues, it's not that simple. It'll take effort to do it but you can even do this across a university. So one thing we do at Utah — we've been building up our data science program — one of the ideas of the center we're building is to have people from across the university who have data questions come ask us and see if we can help out with some of these things. I think there are lots of places like that where you can start and then that can lead to one more thing. Now we've had people from the local government come and talk to our data science students about some of the questions they are looking at. I think we have to start there.

Khari: That sounds great. I mean maybe this is something you could pitch to CRA, like some kind of endorsement of a pro bono system.

Suresh: Yeah

Khari: We don’t have to come up with a solution, but just…

Suresh: Now that you're making me think about it maybe should write out.

Khari: Yeah, just a memo that says departments should do this because it would be good.

Suresh: Yes, because CRA has a lot of clout. I think when the CRA speaks people listen. I respect CRA and the CCC a lot. For many years, before I was on the board, I’ve always thought they were doing an excellent job. If they were to come out and make guidelines this would be helpful to people. I think we need some examples to say look it can be done

otherwise no one will believe it. So I think I need to put my money where my mouth is and make this happen in Utah. Then say look we can do this, let's see if we can do this somewhere else.

Khari: - It could be an interesting initiative.Yeah that's something to think about later.

Suresh: Definitely.

[00:29:17] Outro

Khari: That's it for the podcast. I hope you enjoyed it. Tune into our next episode where I'll continue talking to Suresh about algorithmic accountability and fairness. We'll also discuss how to get involved with the CCC and other similar organizations.