SDS PODCAST EPISODE 782: FIVE-MINUTE FRIDAY: IN CASE YOU MISSED IT IN APRIL

Jon Krohn: 00:00:02 This is Episode #782, our “In Case You Missed It in April” episode

00:00:19 Welcome to the Super Data Science Podcast, I’m your host, Jon Krohn. This is an “In Case You Missed It” that highlights the best parts of converzation we had on the show in the last month. In episode #779, I speak to Dr. Hadley Wickham about putting the ‘R vs Python’ argument to bed. Here, the many-time bestselling author and world-renowned open-source developer highlights the relative similarities of the two most popular open-source programming languages for data science, as well as the key differences between them.

00:00:50 For our listeners, if there are listeners out there who don't already use R, why should they be using it? For me, I can actually give one example, which is for me, for data visualizations, I still find I can do things way more quickly, have much more fun making visualizations in R, and get exactly what I want. There had been in the past attempts to create a ggplot style Python library, but the one that I had been using became deprecated and harder and harder to use. It never had all the functionality of your ggplot2 anyway. Anyway, so that's my big example. I don't know if you have big examples of why people might want to use R still today.

Hadley Wickham: 00:01:35 Yeah, I mean on the topic of ggplot2 specifically, I think the best Python equivalent is plotnine, and that's actually by a developer Hassan Kibirige that we've been sponsoring at RStudio, at Posit I think. And I think that's that's the best possible realization of ggplot you can get in Python. But I think there's things about the design of R language that just make certain tasks much easier and more natural to express in R code than you'll ever be able to do in Python. And I think that comes down to at the heart of it, like R is more of a special purpose programming language. It's designed from the ground up to support statistics and data science, and I think that has a lot of benefits, particularly if you've never programmed before. I think you can get up and running in R, using R to do data science, you can do that without learning a ton of programming, get up and running pretty quickly.

00:02:39 And then there's just sort of things about the language that other languages look at R and they're like, "Oh my god, that's a terrible idea or that makes me want to throw up in my mouth." But there's just so many things that are just so well-placed to support interactive data science, where you really want that fast and fluid cycle where you're trying things out. That obviously bends to maybe a little bit of weaknesses on the kind of like, now I've got this thing, I just want to do the same thing again and again and again and again. R tends to be a little bit magical. It tries to kind of guess a little bit more of what you want and that's great when you're working interactively and it guesses correctly. It's not so great when you're working on a server somewhere else and it guesses the wrong thing. But just R, like everything about R I think makes it such a fluid environment for really exploring your data, digging into it, figuring out what's going on.

Jon Krohn: 00:03:38 Speaking of differences between R and Python, I seem to remember, and you can correct me if I'm wrong about this, but I feel like you have a famous tweet from years ago where somebody says something like, and it must've been a famous poster themselves that you responded to and I can't remember, it might've been like Wes McKinney or somebody like that saying that one of the advantages of Python is that it's faster than R, and then you have this super famous reply of, "What is that? And I will make it faster." Do you know what I'm talking about?

Hadley Wickham: 00:04:15 I don't, but I know I've seen things like that in the past.

Jon Krohn: 00:04:19 Yeah, it's a misperception because Python isn't actually that fast itself. I mean whole languages like Julia have come up to be faster than Python.

Hadley Wickham: 00:04:32 Yeah, I think one of the reasons often the biggest, you have the worst arguments with your family and not with strangers. With people who are so similar to you, you tend to have more friction than the people are really different. I think because R and Python are actually really close together in the spectrum of programming languages, it's so easy to see all of the little things that look weird to you as opposed to looking at some programming language that's miles away and it just looks, it's totally different. I just think that, I don't know, I think there's something to that because we're close, you can see these little differences. And certainly when I see things in Python that people are like, "Wow, that's really cool." I'm like, "Challenge accepted. I'll make that better in R."

Jon Krohn: 00:05:20 Nice. One of my favorite things that you can do really well, thanks to the dplyr library that you led development of is piping. And so you can extremely easily have functions passive, just like if people are familiar with Unix programming pipes there. Where you have output from one function goes the input to a next function and prior to me discovering dplyr, which was probably around 2010, if that makes sense. Prior to that I would have so many variables in my workspace. It was just such a pain to keep them all straight and you just end up in these weird situations, where should I be investing time thinking about the name of this intermediate variable? Am I going to use this later or should I just name it like intermediate variable 15 and have really ugly code?

00:06:18 And so piping gets rid of all that where you can read the flows like a sentence. You're like, okay, this preprocessing step happens, then this next, and you can just see it so easily. It makes it so elegant to read. Do you think we'll get to a point where, and I have used some kinds of piping attempts in Python, but my experience of that has never been, and I guess it's been a few years since I've tried, but it seems like it's never been as smooth or as easy as with R. And maybe that's related to what you were talking about earlier with data visualization.

Hadley Wickham: 00:06:50 Yeah, the native equivalent of piping in Python is method chaining. If you're using Pandas, you do something dot something.

Jon Krohn: 00:07:03 Yeah Panda's-.

Hadley Wickham: 00:07:05 Dot something. But the big difference between method chaining and the pipe is in method chaining, all of those methods have to come from the same class. They have to live in the same library, the same package, whereas with piping, they can come from any package. And I think the thing that's really interesting about that is that has meant Python has tended to have these fewer bigger packages like Pandas and Scikit-learn, Matplotlib, kind of everything in order to work with method chaining, everything has to be glommed into this one giant package.

00:07:44 Where with R, because you can combine things from different packages, the equivalent of Pandas is kind of like dplyr and tidyr and readr and a bunch of other things. It's way easier to add extensions to ggplot2 than Matplotlib that work exactly the same way because you can just combine them with different pieces. So I think that's just one of these interesting subtle differences in language design that lead to fairly big impacts on the user experience and almost even how the community has to work together and form.

Jon Krohn: 00:08:19 From the impact of small differences, we turn to modernizing the education system with Aleksa Gordić. In episode 775, the creator of a digital A.I- learning community of 160,000 people talks about the recent movement from formal education toward self-directed learning online.

00:08:37 So how do you think the tech industry's perception of formal education is changing? So we talked in this episode a lot about self-directed learning, including just now, but with formal education, do you think with the tools that we have today, with all of the content that there is online, for careers like AI, machine learning, data science, software engineering, do you think that formal education should be changing because self-directed learning can play such a big role?

Aleksa Gordić: 00:09:09 A hundred percent, man. Don't even get me started on education, we could have a whole podcast episode only on this topic. I mean, I definitely think that most of the education systems across the world are still stuck in the late 19th century. The industrial revolution type of a context where you're just sitting down with a lot of people who have completely different interests from you, you're just connected by the geography because you happen to live in the same space. So starting from there, there's so many things that need to be changed about education starting from elementary school all the way to best PhD programs in the world at MIT or Stanford, even those are not really optimal.

00:09:49 And so there is definitely a shift of sentiment happening across the industry, especially now given the latest AI boom, and I see much less people encouraged and motivated to go and pursue the PhD path as opposed to just building because they realize, hey, you can do so much, so much without any PhD or master's or even bachelor. You can learn so many of these things on your own. And that being encouraged by the likes of Elon Musk gives it a lot of gravity, right? Because some of those highly successful entrepreneurs are saying, "Hey, when we hire, I don't actually care that much whether you're from Stanford, man. Show me what you build, what makes you stand out? There is 2,000 people coming from Stanford every year doing CS." I don't know. I don't know the number, I'm just throwing out a number.

00:10:43 And I don't know, I noticed on my own that I definitely have achieved probably much more than a median Stanford person would and I know for sure that they told me already that at MIT and Stanford, they used to have classes where they would watch some of my videos I mentioned to you. The ones that are really in-depth. So from that standpoint, I became a teacher for some of them. Which makes me feel, not saying this to sound arrogant, but I feel proud about that but it also tells me, "Hey, if they're learning from me, that means I've done it myself. I didn't need to go to Stanford or MIT to achieve same level or greater levels."

00:11:18 So it's possible, but again here, self-awareness matters a lot. You have to know whether you are that type of person who can be that self-directed and prospering in that multitude of choices, as you said before. And having that, it's not an easy thing for everyone. Because when you have a strict curriculum, you're going to Stanford. You know that every day you're getting closer to the credential of being at Stanford or a Stanford alum. And so it's easier than, yeah, I'm doing this and at the end, there is no credential unless you build some public artifacts, but you have to be much more self-confident and self-directed and build your own curriculum and execute on it. So it requires different type of mentality.

Jon Krohn: 00:12:00 Nice. Yeah, we'll include some of those Medium blogs in the show notes so that people can check those out, and blending some things that we've been talking about already in this episode. So you've been talking about learning now most recently, but are there ways that you envision AI transforming education, perhaps making learning more personalized, accessible to people with different backgrounds, interests? We talked about this in the language context already, specifically where somebody who only speaks Serbian could be learning entirely from English documents in the very near future. That is not science fiction, that is science today. But yeah, are there other ways that you could envision AI transforming education?

Aleksa Gordić: 00:12:42 I mean, 100%. AI tutors are the future, and that's the only way we can scale this up because I complained previously in the episode that we still have this industrial revolution legacy of putting 30 people in the same classroom, which doesn't scale because you cannot personalize or have that one professor teacher give the same level of attention customized to the learning style of every individual pupil in that class. It's impossible to scale that up.

00:13:11 And also due to incentives, you basically don't have the best teachers in the world, right? Because if somebody is that good, they'll probably not go and teach at elementary school, they'll go to MIT. And so there is also that incentive moment there that prevents us from having the best possible education. So the only thing that can scale are algorithms, software can scale. And so AI tutors are definitely the future and I see already myself using on a daily basis definitely ChatGPT and coding assistants. I think those were the two most important AI products for me personally, like code assistance, so Copilot, which I can use for free by the way, as an open source contributor, that's a very nice gesture from OpenAI. And then secondly, I just use the chat assistant and mostly

ChatGPT. I mean, 100% of the time actually I use ChatGPT. It just works.

Jon Krohn: 00:14:05 I love them. They're so powerful. They've transformed how I do everything. And it'd be crazy if you're listening to the show and you aren't paying the $20 subscription for access to things like GPT-4 or Claude 3 or Gemini Ultra. These algorithms, it's amazing how much more quickly I can learn topics instead of especially writing code. I think that's where it's most useful because I used to spend so much time getting stuck on small issues that it's not a learning experience where you're getting stuck on these trivialities of code semantics, but having to spend time digging through Stack Overflow, I mean, I guess before the internet and Google searches, it'd be even more laborious having to go through textbooks to figure out how to solve some problem in your code. And now you can focus so much more high level on the problems that you're solving as opposed to getting stuck on the syntax, which is so nice.

Aleksa Gordić: 00:15:07 100%. Everything that's repetitive, you as a human should just say, "Okay, go execute this for me N times." You don't want to be the for loop. We're literally being, well, the history of civilization is us going from being calculators and dumb machines to being more and more free to do high level cognitive work, right? Because you previously literally had people who were computers. In Ancient Greece, you had people who were acting as memory sticks because they were learning and memorizing every single transaction. And that's why you had so many memory techniques being developed back then, like Roman memory room, memory palace or whatnot. Everything happened back in Rome and Ancient Greece probably before that because people had to memorize, had to compute. So all of these methods were developed and all of a sudden we need less and less of that.

00:15:58 And now finally you're getting freed up to do just creative stuff, hopefully. We'll see. I mean when you get to superintelligence, it is just, all bets are off how the future society looks like and where do you find purpose and meaning. And one could make an argument that, hey, take a look at chess and what happened with chess or Go, it's not like humans stop playing the game just because they're not the best in that game. It turned out that it's more of a symbiosis and humans became much better and are using AIs to devise new techniques and moves that they've never done previously, but with a caveat that these are not the AGI's. That's why I say it might be like all bets are off when you get to AGI, not just a very constrained type of specialized AI such as whatever AlphaGo or Stockfish or some engine of that sort for chess or those games.

Jon Krohn: 00:16:49 Exactly. It's really mind-bending. It is difficult. I mean, I can't wrap my head around what this future will be like when we are no longer... Humans have enjoyed for some time now being by far the dominant intelligence on the planet. And when there's something else around, it's like asking a chimpanzee to do calculus. The chimpanzee is very smart. It's one of the most intelligent animals on the planet, but you'll never be able to get it to graduate from a Stanford degree. And so when there's something else around that, we can't even, in the same way when the chimps sees us writing the equations on the board, it's hopeless. And for us, we could be soon encountering this intelligence where it's hopeless for us to try to understand it.

Aleksa Gordić: 00:17:48 Yann had a take on this and he said that, "Take a look at the current society and you'll see many examples of greater intelligence is being controlled by much smaller intelligences." And you see this across the companies. You have dumb CEOs who just grit and had the luck or whatnot, or I don't want to diminish them, but oftentimes they're not as smart as many of their employees. And that happens. But the thing is, the thing that remark of Yann's, it doesn't do it justice because we are not talking about small difference, a couple of points or tens of IQ points. We are talking about something that can exponentially then improve itself and you can scale it up and it can be much smarter than humans. So we are talking about cat compared to human. Cats never controlled humans. I mean, well, that's maybe a [inaudible 01:24:33]-

Jon Krohn: 00:18:42 You picked the wrong animal.

Aleksa Gordić: 00:18:43 I picked the wrong animal. Maybe pigeons. Let's take pigeons. They're like less high agency. Cats are like the apex predator of this world.

Jon Krohn: 00:18:51 Exactly.

Aleksa Gordić: 00:18:53 I mean, but you get the point. It's not going to be the same qualitatively speaking, when you have something that's alien intelligence, that's intelligence that makes Einstein look dumb. And then as I said, all bets are off. We don't know what happens, how that dynamics plays out.

Jon Krohn: 00:19:10 We may have had a couple of fact-checks from cat owners about that last comment! We can probably agree that our pet cats are more interested in control than collaboration. And, when it comes to working with AI, many of us worry about these tools 'controlling' how we work and think. How can AI work with us and not over us? Or, to put it bluntly, how can we ensure AI doesn't go the way of the domesticated cat? That's what I ask the world-renowned futurist Bernard Marr in episode number 777.

00:19:40 In chapter five, you talk about powerful ways that organizations can harness GenAI that highlight the potential for human collaboration with AI as opposed to replacement. Do you want to talk about that at all?

Bernard Marr: 00:19:54 Yeah, I get this question asked all the time. What will this mean for jobs in the future? I think there's a lot... I have three children. I've got three children between the age of ages of 12 and 18, and I worry about what that might mean for them in the future. My hope is that AI will not replace us, but augment us. What I'm also hoping for us that AI will make us more human instead of less human. Sometimes we position AI as men versus machine, and I completely understand why, because it sells newspaper and papers and magazines and people click on articles that say, "Okay, AI is coming for your job." But my hope is, in practice, that will augment our jobs. I have actually written an entire book on future skills because I get this question asked all the time. What skills will we need? How do we compete with machines in the future?

00:21:02 And out of 20 skills I talk about in the book, three are technical skills. I need to have some technical understanding and understand the capabilities of all of these technologies we've touched on. I need to have some data literacy to some extent. I need to understand some cyber threats that are coming along. Beyond that, rest of the skills are the ones that make us truly human. They are our creative problem solving. They are our interpersonal communication. Our complex decision-making. Our ability to understand whether data is true or not. Contextualizing things. All of this stuff that really makes us human. My hope is that we will outsource some of the things that, in my opinion, we waste our immense human potential on doing. Just a tiny little example. Whenever I write a Forbes article, Forbes ask me to basically capitalize every word in every sub headline.

00:22:12 And this is very often not how I write. I just write and then I need to go back into the article and make sure that every word is capitalized. I can now give this to ChatGPT and say, "Hey, please just capitalize this." Takes a second and is done. This is for me such a... It was a waste of my capabilities because now I can spend more time being creative thinking about how do I want to tell my stories? My hope is that we would just get more of this. A perfect example is radiologists. If you are a radiologist in a hospital that analyzes X-ray images or CT scans, AI can now do this very well. I remember I took my son to the hospital recently, and we suspect that he broke a bone in his hand and the doctor came back and said, "Okay, we can see there's a hairline fracture in the middle of your hand, but the AI's also suggesting that you might've broken your finger as well."

00:23:16 She was saying, "We can't actually see this with our eyes, but the AI usually is right." The AI now is able to do this. For me, we come to biases and we talked about biases. Humans also have huge biases. If you are radiologists and someone has come in for a potentially broken back and they've been through a CT scan, our bias is that we will only look at the broken bone and look at have they potentially broken their back? But because the CT scans scans the entire body, there might be secondary or tertiary diagnosis that might be relevant as well that we as humans would never look for. The AI can do this. We are now at a stage where the AI can analyze images and CT scans much more consistently and probably at the same level, if not better than humans can.

00:24:20 What does this mean for radiologists? It will change the way they work, because if you think about this, is it really the best use of this amazing human potential that we have where someone sits in a dark room for eight hours a day trying to understand is this bone broken or not? Or would it be better if we spent more time on maybe talking with a patient about what this all means? On doing research to further the field of radiology and make it better. All this exciting stuff.

00:24:51 My hope is that this is what AI will do. In the short term, I have some concerns about how ready we are as a world to move to that, because there are so many people happily or very often not happily working in jobs where they earn a living doing stuff that actually is a waste of their potential, but it's a necessity to earn a living. This transition will be difficult, and I think it's really important for governments and for businesses to understand that we are seeing this huge transformation and it means we need to retrain people. We need to augment their jobs, and we need to help them in making this happen.

Jon Krohn: 00:25:37 “We need to retrain people”: This is a sentiment that my next guest 100% shares. SuperDataScience founder and educator of literally millions of people online, Kirill Eremenko is dedicated to helping us work confidently with data and AI. In episode number 771, he helps us explore gradient boosting and why the machine learning technique is so powerful for making informed business decisions. Here’s a short clip from the full gradient-boosting workshop that episode 771 is.

Kirill Eremenko: 00:26:06 Now, we can move on to Gradient Boosting. Gradient boosting was originally proposed by Jerome H. Friedman in 1999, and there's two papers you can find online. One is called Greedy Function Approximation: A Gradient Boosting Machine, and I think that was more of a lecture that he gave, because it's got 40 pages or something like that. The second paper you can find is Stochastic Gradient Boosting. This is the person who created it. What is Gradient Boosting, and how is it different to bagging, bootstrap aggregating, and AdaBoost?

00:26:43 The main thing with Gradient Boosting is that this time, we're not just going to adapt. We're going to actually be changing our sample. We're not going to be doing any bootstrapping. We are working with the original sample all the time. That's very important to understand. There's no bootstrapping in Gradient Boosting. What you do in Gradient Boosting is, we're going to look at Gradient Boosting for regression first. You have your 1,000 customers. I can't believe this example stuck. That was just a random thing I wanted to do for the trees.

00:27:16 You have these customers, 1,000 customers that bought candles from your store. You want to predict the future spend of customers. Your target variable is your dollars spent. What you do is, you take as your first step... again, Gradient Boosting is again going to be an ensemble of models. Your very first model is just an average. It's a simple average. You take the average of all of the dollars that all of your customers spend, and let's say you get something like $57 for simplicity's sake. That's the average of all your customers' spend. Next, what you do is you calculate the errors. You look at, "Okay, $57 is my average," that is, of course, a terrible prediction, a terrible model. You just took an average. For some observations, you'll have an error, some observations will be lower, some observations will be higher.

00:28:11 You basically calculate the error for each one of your 1,000 samples, and then you take those errors, whether the error is $2 or $20 or minus $100, you take all of those errors and you build a decision tree to predict those errors. The first model, it takes the average, works with a sample. The second model, which is our first decision tree, it'll work with all of the errors that you got as a result of the first model. Now, this decision tree will be structured in some sort of way. It'll make its own predictions, and now you will have, again, errors.

00:28:52 You will have errors of this decision tree's prediction, and some might have $5 error, some might have a minus $50, minus $100. Again, you look at all the errors of the predictions of this second model, which is a decision tree, and you use those errors. Again, you'll have 1,000 errors, in some cases it might be zero, but you'll have 1,000 values, and you use those errors, and you make another decision tree. Your third model will also be a decision tree, and it'll predict the errors of the second model. Then you build a fourth decision tree, which will be predicting the errors of the third model. Then you build a fifth model, which is also decision tree, and you predict the errors of the previous model, and so on. You chain them together. The key word here is, you're chaining models after each other.

00:29:41 The first one is an average, and then it's decision tree, decision tree, up to 100 times, however many decision trees you want. Each one is just focusing on predicting the errors. What's the point of that? Well, guess what? Now, as our final model, we're not going to take the average, we're not going to take the weighted average. As the final model, we'll take the sum. You'll take your first model, which is the average. You'll add the result of the second model, which is whatever the decision tree predicts for this kind of variable. Let's say you have a new customer come into your store, and you want to predict how much they will spend. The answer will be the average, which was $57, plus whatever the next model, model number two, which is a decision tree, whatever it predicts, plus whatever the next decision tree predicts, plus whatever the next decision tree predicts and so on.

00:30:30 You add all of that up, and because each time you are predicting the errors, now, your prediction is the average plus, what would the error be for this person that just came in? Okay, the error for this person is 57? Okay, based on their age, based on the income, whatever the decision tree is looking at, the error of this second of this initial model would've been minus $101. You need to add that. You go from 57 minus 101, I'm not that great. What is it, minus 44? Is that minus 44? Yeah, minus 44. Then the next model will be, what would the error have been based of that prediction, the previous one? The error would've been $50. Now you go up to $3, and then the next model says, the error at this point is about $27.

00:31:17 Now you go up to from $3 to $30, and so on and so on and so on. Then the final result is, this customer, based on their features and based on what the model predicts for them, this customer will likely spend $39 in our store. That's how Gradient Boosting works. You're basically chaining models to constantly just predict the errors of the previous one, and that means in the resulting model, you need to just add them up. Each prediction will be a prediction of the errors, and in the end, you will get your final result.

Jon Krohn: 00:31:58 Nice, very well explained.

Kirill Eremenko: 00:32:01 I just had this idea, it's probably good to call the first model, the average, call it model zero, because otherwise, it's confusing. The first decision tree is the second model. The second decision tree is the third model. Model zero is your average, and then model one is a decision tree. Model two is your second decision tree, and so on and so on and so on. The final result is the sum of this chain.

00:32:25 As you can see, it's very different to what we had previously, in the bootstrap aggregating methods, which were bagging, basically a random forest, the way we took the average. It's also different to the boosting method of AdaBoost, where we took a weighted average of the models. AdaBoost is in between. It's used as bootstrap and it's used as aggregating, so it's a bootstrap aggregating method from the sense of how the samples are built, but it's a boosting method based on the concept. AdaBoost is a transitional, whereas Gradient

Boosting is pure Gradient Boosting. There's no more bootstrapping. Straight into, use the same data set all the time, but you focus on improving, improving, improving, improving.

Jon Krohn: 00:33:12 To summarize back, the random forest is random. This went into your democracy versus meritocracy example. With a random forest, you are randomly creating a whole bunch of decision trees, and the more that you create, you get this slight marginal improvement. When you go from 1,000 random decision trees to 1,001, there's this very marginal improvement. The core idea with adaptive boosting was to not be randomly creating these decision trees, but to use some information, like which data points were misclassified previously, and let's overweight those in the subsequent model so that we're consciously iterating in the right direction adaptively, AdaBoosting. Gradient boosting takes us another level further by not just saying, "Let's focus on the data points that were misclassified. Let's look at the residuals, the specific delta between what the correct answer would've been and what the model predicted, and let's fix those residuals." You're focusing on where the most possible opportunity for improvement is, and that's why Gradient Boosting is so powerful.

00:34:34 Kirill’s courses are always full of these analyses of practical business questions. And this is where we’re headed in our final clip, from episode number 773. Here, Dr. Barrett Thomas emphasizes this need for businesses to think about all of the variables that impact when, why, and how an AI tool is used. I ask Dr. Thomas, an Iowa University Professor of Business Analytics, about the questions that a delivery company might need to ask about using drones.

00:35:00 So we've been talking about same day delivery and drones come up as something that could be super helpful in having same day delivery. I mean, I guess it's conceivable though, I don't know of this yet, or at least I don't see it in New York, of either a sidewalk drone that just drives itself or a flying drone bringing me a meal or bringing me some small Amazon order or something. Where are we in terms of drone deliveries happening? Are there places that that is happening regularly, and what's the benefit once we get it to work? How does that complement existing systems?

Barrett Thomas: 00:35:44 Companies have been piloting drones, whether it was Google had a pilot on this, Amazon has been trying to do this. You do see JD.com in China has used drone deliveries. Where I think it's most successful at this point is not so much in rural areas or in cities, but when we're delivering into rural areas and we might send that drone and we're going to a more rural area, it drops off a set of packages in the backyard of somebody who then delivers them to the individuals. But when we're doing research, we also want to look into the future and we want to try to understand what could that future look like. Should I be even in trying to invest in these technologies? Could there be any advantage to doing so?

Jon Krohn: 00:36:42 Yeah, you might discover that actually your cost ballooned. It seemed like a cool idea, but for some reason...

Barrett Thomas: 00:36:47 Yeah, that's exactly right. And so, one of the papers that I have with a co-author from Germany, Marlin Ulmer, and then a former PhD student, Xinwei Chen, we were looking at this question because we wanted to know, wow, if you're Amazon and you're doing your same day delivery in an urban environment, would I ever want to use a drone? Would I just want to stick with trucks or maybe I want to use some combination of the two? And in fact, what we found is that at least at that time, if you considered that the drone technology, it could carry one package. And so, it would take a package, it would go out from the delivery depot, it could deliver it, then it had to come back and pick up another. But if you're using a vehicle, well, a vehicle, I can have multiple packages on, so I can put multiple packages onto it. It can go out and it can do a delivery route and then return and pick up the next set of packages.

00:37:57 So each of those might have an advantage. The drone moves faster. It's not affected by traffic, but it's only doing one at a time versus the truck that's affected by all those things but I can carry multiple packages. And so, maybe the result wasn't that surprising, but it turns out you do want to use both and you would deliver things that were further from the depot, you would deliver those using the drone. And I would use trucks as we were closer to the depot because I could put more packages on it and I could take advantage of the fact that it wasn't doing those back and forth trips to the depot.

00:38:36 Now, the question is could we ever use those in a city? There are tremendous challenges in a place like New York in the fact that you have really tall buildings. Once you have tall buildings, you have the effect that might have on winds moving through the buildings. Do we really then want drones that maybe these are small drones, but they still weigh something that could have challenges and be knocked out of the sky. So subsequent research is looking at, well, okay, maybe I don't want to have drones delivering those packages, particularly in urban areas like that. But maybe what we want to do is instead of having that truck go back and forth, maybe I want to use a little bit larger drone and resupply the truck closer to where it's going to do its delivery. So maybe it comes out to a loading zone of some kind, an area that is a little safer, and maybe we do that.

00:39:43 So we didn't do that work, but there's been folks starting to look at that, and I think that's a really promising idea to think about. And particularly when you think about many of the dense European cities where bringing in delivery vans can be really difficult, they're already doing things like cargo bikes in many of those cities. Well, those cargo bikes have a pretty limited capacity anyway, and they're not going, you're going to have to ride the bike back. And yes, they're electric bikes, but ride them back and forth to these depots. Maybe we want to do something in between so that it's just a lot more efficient.

Jon Krohn: 00:40:23 That's cool.

Barrett Thomas: 00:40:24 And we're just getting more productivity.

Jon Krohn: 00:40:26 That's a totally new idea to me. It never occurred to me and makes so much sense.

00:40:29 All right, that's it for today's In Case You Missed It episode. To be sure not to miss any of our exciting upcoming episodes, be sure to subscribe to this podcast if you haven't already but most importantly, I hope you'll just keep on listening! Until next time, keep on rockin' it out there and I'm looking forward to enjoying another round of the Super Data Science podcast with you very soon.