TYLER: This is the healthcare.ai live broadcast, with the Health Catalyst data science team, where we discuss the latest machine learning topics with hands-on examples. And here's your host, Levi Thatcher.

LEVI (L): Hi and welcome to the healthcare.ai hands-on machine learning broadcast. We're excited to welcome you and talk each week about how you can get involved in our healthcare.ai community. I'm Levi Thatcher. We're excited to welcome you. I'm here with Mike Mastanduno. Mike, what's on the docket for today?

MIKE (M): Thanks, Levi. It's great to be here. Today on the docket, we're gonna talk about machine learning in the news. We're a machine learning and a data show, so we're gonna talk about data in the news, we're gonna move on to some questions that we have from our viewers from the last couple weeks, and then we'll really get into the meat of the presentation. So, the number three thing we're gonna cover is gonna be a tutorial on how to get from our website installing healthcare.ai and R and RStudio and then get a model up and running. Then we'll close with answering your questions that come through the chat.

M: I'd like to mention that the YouTube chat is not working on our homepage right now, but you can get it on the YouTube page itself. The link to that is in the lower right-hand corner of the video screen. Also, we're gonna be showing a lot of high resolution text that's kinda small, so you might want to adjust your YouTube resolution using the little gear in the lower right corner. And then you know, we really want you to be able to participate and interact with us, so please ask us questions in chat. We'll try to address them as the hour goes on and at the end we'll make sure to get to your comments and questions.

L: Exactly. The whole idea is to be interactive, educational, so we're learning from you, you're learning from us. Really, it doesn't work without that.

M: You can get us on twitter at @LeviThatcher or @MikeMastanduno. So, let's take it away, Levi.

L: Yeah yeah. Why don't we start out? We're gonna talk about machine learning in the news. Always in the news these days, if you're looking at the right websites. So, what's most exciting this week?

M: So, this week, there was a really great app that came out as a Google Chrome extension. And it kind of goes through the idea of *what can your online data presence, what can a company learn from that?* We like getting Netflix recommendations, but it can be a little creepy. Don't you think?

L: If you take it too far.

M: Yeah. Maybe you're shopping and then you go to a new website and there's all the sudden ads for those products popping up on your websites. That can be a little weird. So how far is too far, really? There was an article in the New York Times, maybe two years ago, 2014, where Facebook was doing an experiment with users newsfeeds to try to manipulate emotions. A lot of academics and ethical experts decided that was definitely too far. But this app that we're talking about tries to answer the question of what can Facebook really learn from your data? I thought it was just fascinating, the wealth of information you can get. The way this app works is you install it in your browser and it runs in the background and it just keeps track of things that Facebook would keep track of, like: what you like, what you look at, what you type, how long you spend on each section of content.

L: So any way you're acting with the app, basically.

M: Really, just, you're interacting with Facebook and the app is tracking you, just like Facebook servers do.

L: Wow that's amazing.

M: If you don't like that idea, then maybe stop using Facebook.

L: Yeah, but this will give you an idea as to what exactly they collect the data on.

M: Exactly. So, once you have all that data aggregated, they can do machine learning and natural language processing to figure out things like, whether you're political or liberal, or liberal or conservative, what your religious preferences might be. They can even get into how you like to shop and spend your free time. Do you eat fast food? Things like that.

L: Yeah and it's all about delivering a better experience, typically—

M: Yeah that's how they would view it.

L: or serving more appropriate ads.

M: Yeah exactly. But for some people who are worried about their online data presence, that can be concerning. So, this project is open source, which means that nobody sees the data it collects, except you, unless you publish it. It's a reminder that that data is being collected everywhere you go.

L: Yeah, so if you're concerned about what people are able to gather from you on Facebook, this allows you to really see *okay well where is the line where it starts to get creepy?* So, it's called data selfie. Is that right?

M: Yeah, you can find that on GitHub and then if you Google for it, there's a few different little videos that give an intro. I thought it was really cool.

L: Yeah, that's fantastic. Data selfie. Thanks, Mike. So, that's our first segment on "In the News." Next up, we want to hit the mail bag. Let's do it.

M: Yeah so, we had a lot of questions from users over the past couple weeks and I think it will do a nice job to kinda segue into our main program because we had a lot of questions about installation, the package, why we want to use the package, so maybe—

L: *Why are you doing live broadcasts?* All sorts of weird stuff that we're doing.

M: Yeah exactly. So maybe, Levi could you just kinda give an overview of, you know we've said this package is specific to healthcare and that can be kind of a vague term, could you give an overview of why [healthcare]?

L: Yeah for sure. So, when we wanted to get started with machine learning in healthcare, we looked around and there weren't really any packages designed for healthcare. So what we did is we built our own. You might have heard about this: healthcare.ai. You might ask why is this special for healthcare? But, really it comes down to the algorithms being appropriate for typical healthcare questions. A lot of the times you're dealing with longitudinal data in healthcare that you wouldn't in other industries. We provide you connections to files and databases that are common in healthcare and we provide metrics as well. So, when you're creating models, you often have performance measures, and we provided performance measures that are particular for healthcare-type models that you would build using the package.

M: So healthcare is kinda the only package that does all of that in one place, right?

L: Yeah, so ours is kinda the first one to try to automate a lot of the steps. Yeah, exactly. So, you might have heard of caret or scikit-learn, which are R and python packages. Really what we've tried to do in healthcare.ai is get rid of a lot of these different questions that analysts or data architects might have, as to: *What algorithms should I choose? What connection should I use? How should I do my feature engineering and decide which features I should keep in the model?* So we're trying to provide a clear pathway for people to actually [interruption in program]

M: You know, maybe you don't need a PhD or extensive technical training in machine learning, right?

L: Yeah. Exactly. So, data scientists are hard to find, they're expensive. It really shouldn't be that difficult for people to create models and to use [interruption in program]

M: Great, thanks so much. And then I guess one other question is, a lot of people are asking where they can get help and how to join the community. So we've got a number of ways to do that. We've got a stack overflow tag, healthcare.ai, for technical questions and the whole community can help with those. And then we have a blog you can view. And you can subscribe to the email list as well. We really want to make sure we're giving back to the community as well, so that's one of the reasons we're doing this broadcast. So, with that, we've also had a lot of excitement about the website and the product, so I think Levi's gonna move into the meat of the program now and start with an overview of the website and then kinda get into the—

L: The nitty gritty here.

M: the nitty gritty, yeah.

L: Yeah yeah. Thanks, Mike. Fantastic. So, we wanted to give you a little tour of the website. It's brand new, it's looking good, and we're excited about it. Really, it's based around community education for healthcare machine learning. So, if you're getting started, that's fantastic. If you're just looking to survey what we're doing, this is the place to go. The url is literally "healthcare.ai" so check it out. So, you'll notice that we have a blog, which we've been trying to post about weekly. I think we'll keep it at roughly that cadence. And this is where we talk about models that we're building into our tools, talking about how to prepare data, such that you can create models yourself, talk about different metrics that are common in machine learning, talk about the algorithms, really it's free flowing, it's free-style. So, whatever we come up with that week, it'll go in there. And please, we want feedback from you all as to what we should put in the blog, so please reach out via twitter or email. And then on the website you're gonna see "weekly broadcasts," which is probably how you found this broadcast.

M: And we've just been told that the chat is now working on the healthcare.ai page for the weekly broadcasts, so feel free to contribute there.

L: Yeah please throw in your comments, good or bad, we want to improve. And please subscribe to the channel as well. So, you have the "subscribe" button up here in the top right of healthcare.ai. "Contact" which has our email and twitter handles. And then "Packages" is what we're gonna focus on for the next couple of minutes. And you might say *what is a package?* Mike, you've been around the python community for a while, what's a package?

M: Software package. It's a collection of tools and functions and code that you can download all in one line and pull it onto your system. It'll come with examples. It'll come with help. It'll come with a community that helps you use it. And you'll get functionality that you wouldn't necessarily get by kinda cobbling things together on your own.

L: Yeah yeah. So, it's a nice package of code that gives you awesome functionality. So, that's why we were excited to use packages and R work here, to distribute code amongst the team and to help the community get involved in model building in healthcare as well. So, if you click on "packages," it describes: *Well what can you do with healthcare.ai? Why am I even going to this site? Why do I care?* Then *why is it specific to healthcare?* (which we talked about). More importantly, *how do I get started with these tools? I have some data, maybe it's on a financial process in healthcare, maybe it's clinical, so how do I create a model using my data?* If you keep scrolling down, you'll notice that we have an intro demo, which you can watch, and then we have the links to the R and python packages. Now, Mike, R and python, why? You know, weird, it's a letter and it's a snake. So, for the end initiated, why?

M: Two programming languages and you know, both languages have kind of emerged as the leaders in data science and machine learning applications. Mostly because they're free, they have a large library of statistical and machine learning tools, and a lot of tech industry companies are using them to develop their products. So, it's kind of the current, the wave of the future, and neither have really edged out the other in terms of capability or functionality. They have their pros and cons, but for the most part, it's still pretty much neck and neck. So, we decided to just develop both packages. They're pretty equivalent in functionality, and that way, users can get up and running in the tools of their choice, as opposed to being forced to learn a whole new programming language.

L: Yeah exactly. I used python in grad school. Did you use python at Stanford?

M: No I didn't, I was a MATLAB guy, but I converted to python and I'm never going back.

L: haha Yeah, so I was a python guy, and have gotten more into R in the last year. So, now I have been won over to the R side of things, but still love both. Cutting to the chase, if we click on the "R package," we'll go to the documents, which is how you actually download the tools. So, if you notice, here, we have a link to our "Github Repo." So one of the benefits of being open source is you can see all the code that lies behind what we're doing. And you might say: *Well okay doesn't that expose your innovation, your trade secrets?* And the idea behind that is that we want to be learning from the community. If we've done something wrong or incorrectly, we want to hear about it and we want to enable you to learn how to do these things as well. So, if we click through and actually look at the repo, if you guys click through at home, you'll notice that it's on Github. So Github is actually the server where the code lives. Notice we have the star functionality here at the top. So, if you like what we're up to, it wouldn't hurt to give us a little star there. And you can browse the code there when you get time. But, it's totally open and we welcome contributions, so please make it better, if you're interested.

L: But if you scroll down, you'll notice that it describes, *okay well what do you do with this tool?* And *how do you install the latest on windows and macOS?* But before that, *what are you actually doing when you install this tool?* Well, first off, you need R and RStudio. So like Mike mentioned, R is a statistical language. It's been around for a decade or two in various forms, it came from the S language, and it's gotten really popular in the last couple of years. And so we're gonna help you, today, install R and RStudio, and it just takes a couple of moments. So, we're gonna jump to our virtual machine and follow the instructions on the website that we just pointed out. And so what we do here is, we're gonna go to the R site that's linked, then we're gonna look for the windows instructions as to how to install because we're using a windows machine. And you know, this could be on your desktop, it could be on your laptop, it could be on your server, so wherever you might be using your computer, you can totally download R—if you have permission—and get started. And then we're also gonna pull up the RStudio site as well and download that.

M: What's the difference between those two, Levi?

L: Great question, Mike! So, RStudio is the way that most people interact with R. It has fantastic tools, in terms of package management, file management. So, if you're gonna download R, we recommend downloading RStudio as well. And so, we have those downloaded from before, but you can see on the RStudio page, you can simply scroll down and download the free version yourself. Note that RStudio has a professional version, but we don't think you'll need it, 'cause the free version is pretty fantastic as it is. So like I said, we downloaded these before. So, if we go ahead and run the R installer— [starts the installation process]

M: So, just to reiterate, R is kind of the language and the nuts and bolts, and then RStudio is the nice presentation to make it all look good and easy to use, right?

L: yeah yeah. Exactly. That's a great point. So, we'll install R first, so we're focusing on the language installation first. And this is the thing, like Mike said, that allows you to do calculations and create models and work with statistics overall. And so this will just take a second to install.

But, Mike, when was the first time that you heard of R? Like were people in grad school with you or in your post doc [using it]?

M: Yeah, people were using R when I was in grad school, and I thought it was mostly a statistical language, but I didn't realize that you could do so many different things with it. You could do web apps, web scraping. We've had some comments talking about how python really excels in those tools and I think that's probably true. Python is more of a language that you can do versatile things, but you know, you can't argue with R's statistical foundations.

L: Yeah, yeah. If you find professors around the world working on a new math technique, what they'll do is they'll write an R package and they'll put it on the server for the whole world to pull down, so it's really fantastic for math and statistics. So, that went through kind of fast, so if you're installing R yourself, you can click through the defaults and decide where you want to put it on your machine and just takes a couple seconds, like you saw there.

L: Now we're gonna install RStudio, and again, the defaults work fantastically, so we'll just click through here, and if you need to put them in a particular place on your computer, you can do that. It just takes a couple seconds to extract.

L: So, I remember when I was in grad school, I had a bioinformatics friend, and it sounded like the biology department, that whole field was a lot ahead of the curve, in terms of R use. He mentioned it maybe 2010 or 2011 and I was like "R?" It was a foreign concept, but in the last couple years, in tech, it's gotten a lot more popular.

L: Okay so RStudio is installed. If you noticed, the defaults were great. So now, if you want to actually go ahead and start using it, open up the Windows key and search for RStudio. And like Mike said, this is how you interact with the language itself. So it has fantastic tools. And we'll give you just a brief intro here. So, you'll notice a couple different tabs. So, we, to first start off, let me go ahead and increase the font so you all can see it. So, we'll go down to appearance here, and [change the font size to 18]. There we are. Okay. So, a couple different windows here, so you can either interact directly with R, type fancy math like that [*typed 1+1* and RStudio produced the answer], and that's in the console window in the bottom left and maybe in a different spot on your machine. But the idea is here you're doing various math equations if you want, figuring out the cosine of something. Haven't heard that in awhile. Not used as much in our field as it was back in grad and undergrad math classes.

M: It's big in physics though. In physics undergrad, cosine is very important [our third year?].

L: Yeah that's right. Exactly. So, that's the console where you're doing these interactive-type commands. And if you want to run a bunch of these commands in a row, maybe you want to compute some computation and then take the cosine or then take the sum or mean or median you do it up in the script window. So you open up a new script, you just simply go [*press the small white square with a +, then select "R Script"*] right there and that's pretty straight forward. But, what we want to do here is install the healthcare.ai package. So you [type] "install.packages" and in text or in single quotes, you type "healthcareai" and this takes a second to install. And it's up on cran, so that's where we're pulling it down from now. So Mike has some great experience with cran here. Do you want to talk about what cran is?

Mike: Yeah sure. So cran is a package repository for software packages written by people around the world where they can submit to it and our code lives on cran. And it has to pass a strict set of guidelines and rules to be allowed on cran and there [are] a lot of benefits to being on cran. You can do one line installation from your console to get a package on cran. All the packages that use it are gonna be, it's gonna be documented, the versions are gonna be easy to find. It's just a great way to make sure that your package is available for users to get.

L: Yeah so we consider ourselves somewhat self-respecting now that we're up on cran. That's kinda the idea is that you have a stamp of approval on your package. So, you'll notice in your window there after you type that "install.packages('healthcareai')," it'll run through and install a bunch of prerequisites for you and pull actually, pull the code down from that cran server so that you can interact with it. And so, what we're gonna do here is just briefly pull up okay well if you type "library(healthcareai)" that actually pulls up the functionality into your memory in RStudio. And then you can notice that anything in RStudio can be accessed by the "?healthcareai", or "?median," "?mean." So the question mark brings up the documents for whatever function you're dealing with or whatever package you're dealing with.

M: Levi, can you just talk about the different panes of RStudio? Like there's four different windows here, what do they all do?

L: Yeah, so there's a lot going on, that's understandable. So, on the right side, if we focus over here on the bottom right, you'll notice that we have a help tab, which we just focused on after typing "?healthcareai." In this help tab, you can find the documentation for whatever function, or whatever package you're working with. You also have a packages tab [in the same bottom right quadrant], which is fantastic as that shows you exactly what you've downloaded. So if we scroll down [through the packages tab], you'll notice that we have healthcare.ai installed, just as we have any other R package installed. And then you have a plots tab, where your plots will show. And what's fantastic is that you have an environment tab on the top right. So if you do things like "a equals 1," that shows up in your environment tab, and you know, not super fantastic as of yet, but you get the idea. As you work with variables in R, or more specifically in RStudio, you'll see them on your environment tab and you can go inspect them quite easily.

M: That's kind of a way to transition from excel into R without too much pain, right?

L: Yeah, exactly. So, if you're used to excel and you're transitioning, you're probably used to looking at the data, you know, looking at this tabular format of data. And so, as you work now in RStudio, you'll find the environment tab's pretty helpful as you able to actually look at the data and say *okay well what do the columns hold and what do these rows hold?* That will kinda help your transition.

M: Great.

L: Yeah, so now we've installed the package. Let's go ahead and pull up those documents once more. We'll just give you a brief tour. We're gonna keep this first broadcast brief. And what you do when you see these documents pull up is you'll notice that there's kind of a two-step process here when you're creating a model. And we'll go into this model development process a little more in coming weeks. But what we have here is, in the first step, you're actually trying different algorithms, trying different columns, and seeing how accurate your model is when you've done that. And it just takes a couple of seconds actually, so let's dive in, you know, we have a second here, so we'll go into "RandomForestDevelopment." Now, RandomForest, that's kind of a funny term. A lot of times when we're creating documents, some people will say like "RandomForest? Is that a typo?" But can you describe RandomForest to us Mike?

M: Yeah sure. So, RandomForest is an ensemble algorithm, kind of meaning that a forest is made up of trees. And a decision tree is kind of like a "if a value is greater than a number, than x, go left, otherwise go right." And then you get to the next fork in the road and decide based on some other variable. And eventually you can classify your data as either a yes or a no, or a zero or a one. And so that's kind of one decision tree. In a random forest, you use hundreds of decision trees to kind of vote on the proper classification of each data point, and so you get a sort of strength in numbers type improvement.

L: For sure. That's a fantastic explanation. You often hear people when you're describing math and getting degrees in math—I don't have one—but there's obviously people doing math research all over the place, and you're like well can't I just use the same math textbook that was out there ten years ago? Why do I have to keep buying new math textbooks? You might hear that complaint here and there. Really, with machine learning, I've started to realize more and more what math innovation means, in terms of the algorithms that are coming out over time. And RandomForest is a good example of that. So really, it got popular in the 90's and I believe the main paper for it came out in the mid-90's, and the same with Lasso, which is another algorithm that we have here. So, let me just back up and show you Lasso. So, Lasso and RandomForest are our two algorithms, and these are recently relative developments, relatively recent developments in the math and statistics community, which is kind of mind blowing. Like, we're still discovering things in math that are—

M: Of course, the statisticians are gonna tell you that they discovered them ten years ago. Haha

L: haha There's kind of machine learning vs. statistics. They're kind of at loggerheads sometimes and sometimes work together well. So, interesting exploring those dynamics. But, to be brief here, we offer two algorithms and that's the last on the RandomForest. And we'll go a little bit more into those in other episodes, but today we wanted to focus on getting you started on R, get you working in RStudio, give you a tour around RStudio, kind of show you how to install the package into RStudio from cran, and show you the help documentation. So, again, just to reiterate, if you ever get stuck, what you want to do is load in the package healthcare.ai and then type "?healthcareai." and you'll have the help pop up on the help screen. So, that's kind of the [end to end] here.

M: Levi, I noticed when you were just typing healthcare.ai out, you managed to get, you typed like the first half of the word and then the rest came up.

L: Yeah, yeah. Exactly. So, let's do that again. So, was that with the question mark or was that with the..? There we go. So, that's one of the beauties of using RStudio, whereas if you were just working in notepad, or noteplad++, you don't get this sort of tab completion. That's what it's called, right?

M: Yeah yeah. So, you can just press tab and the word kind of finishes, right?

L: Yeah. There we go. So that's one of the benefits of the IDE, besides being able to have an environment tab and files tab and help tab, you know, tab completion is fantastic. So, that's sort of the [end to end] with installing healthcare.ai, and to finish up, we'll just look at the Q&A and see what users have been sending across and talk a little bit about what people want to discuss before we end.

M: Yeah, sure. So we've got a couple of questions just kind of on real-world examples. *I guess what has healthcare.ai been used for already?* And, you know, I'd love to speak to that. We've had fifteen to twenty models built in healthcare.ai already and they span clinical uses because those are the most common ones. So, hospitals are trying to address their biggest problems, like hospital-acquired infections or readmissions. Things like that, that just affect a wide array of patients and end up costing the health system a ton of money and outcomes statistics. So, we're looking at sepsis models for readmission, CLABSI infections (which is central line infections), and then we're looking at things like heart failure risk. There's a lot of places where the current standard of care is to use this rule based system that's very wide net and it's not customized to a population or to a health system. So, there's a lot of room for machine learning to make improvements like that, and so we're definitely going to see machine learning kind of moving towards detecting cancer or stroke victims, things like that.

L: Yeah, we're starting with the base cases, the low-hanging fruit.

M: Exactly, yeah. The low-hanging fruit is what we're starting with. So, those are kind of the clinical applications that healthcare.ai has been used for, but then it's also been used for operational and financial things. Like there's a propensity-to-pay model at a few clients we have, that it gives a prediction of whether or not a patient is going to be able to pay their bills. And that can be really useful from the standpoint of being able to allocate social resources or financial aid, charitable services.

L: Yeah exactly, so a lot of people hear that and they're like "ah that's kind of mean of the corporation to treat people differently that can't pay their bills," but it can actually help. So, what happens is, rather than a person having a $6,000 balance and not paying anything, what happens with this prediction is that the health system can say "well, let's maybe mark down this balance to 3,000 and that actually gets paid and the person feels good about paying their bill and the health system actually gets some money, rather than zero money.

M: So, yeah. It takes care of unpaid for care.

L: Yeah exactly.

M: And then we also have things like length-of-stay models, so you're trying to predict how many beds you're gonna have available in a week. You can look at the current patient population and get an idea of how many are gonna go home, how many are gonna be here still. And then, even stuff like staffing, you know. It's hard to staff an ER with the right number of doctors, nurses, and techs.

L: Yeah so things like census. And then no-shows is another operational-type prediction that we've been doing. So a lot of outpatient clinics, and even inpatient scenarios are really interested in seeing: *okay well we have this scheduled list of appointments today and we know some of them aren't gonna show up. How can we intervene, you know, maybe with a phone call or an email or a text to get more people to show up to their appointment? Or, on the other hand, to double book particular slots such that the clinician is busy all day long and his or her time is efficiently used.*

M: Mhmm. Great. So then, we also kind of have a, you know, we've had a lot of concerns with people who are trying to get data and get clean data. So, how does the package make basic data cleaning and manipulation easier?

L: Yeah that's a fantastic question. So, one thing that I'll just say out of the box is through imputation. Now, I actually, that's more of a recent term for me coming into data science. I came from the atmospheric sciences. So, you might not have heard of imputation, but it really means getting rid of empty data. So, you might have these empty cells and sequel server on your csv, what are you supposed to do with those because the algorithms don't really like them. So in the package, you have the ability to flip a switch and have the code essentially handle the imputation for you. It does that by filling in that empty cell with either the column mean or the column mode, which is the most frequent value. And that just is handled behind the scenes for you, so that's one way that the package helps is through handling null values. And maybe in a future episode, we can go ahead and talk a little more about the processing steps.

M: Yeah it's probably also worth mentioning that the package, it will, drop a column if you've got a certain percentage of null values. Just so that maybe that column is not that, there's not enough information in it to be worth keeping or the imputation is gonna be suspect, so it's better to get rid of it.

L: Yeah that's a great example. So, we've tried to make it really such that the tool automates a lot of the job that you probably wouldn't want to have to deal with, but also giving you the flexibility to go in and tweak things if you do want to. But that fantastic default's out of the box.

M: I think it'd be—you know we had a question about RandomForest and how prone that is to overfitting. It might be worth talking about the algorithms that we've chosen to include in the package and how they deal with overfitting in general.

L: Yeah so that's one concern, right, that pops up pretty frequently in machine learning is okay well maybe this model is good on this training data, but how do I know that it will generalize well and predict well when it sees new data. And so again, trying to automate things as much as possible, we've chose the RandomForest and Lasso algorithms because they're fairly good at generalization and not overfitting. So, with Lasso, it actually does a normalization, so it actually looks for a simpler model than it otherwise would offer by shrinking the coefficients and actually pushing certain coefficients to zero. So, with Lasso, you may start out with forty variables and it will say *Nope, twenty of those variables aren't any good*, and it will make those coefficients zero for those twenty. Or maybe ten out of thirty, or something like that. But, it helps you decide *okay well what variables should actually remain and go in the final model?*

M: And then it does a nice job of smoothing the variables that are in there so that one variable isn't allowed to dominate the whole model, which leads to better generalization.

L: Yeah so the smaller the variables, the less likely you are to overfit.

M: Yeah you're pulling a small amount of information from a large number of variables.

L: Yeah exactly. And RandomForest is similar in that with the way these trees are formed, remember that the trees make up the forest, so what happens is that in each tree, at each split in the tree, you're actually going to look at a few of the variables out of the entire subset. So, maybe four out of twenty, or something akin to that. And so it's hard to overfit when you're just looking at that subset. And we can get into that a little bit more in a coming week, maybe we'll have an algorithm show here coming up.

M: Yeah that'd be great. We have one more question before we talk about next week. We had someone ask about adoption, so *how do you convince unfamiliar clinicians of the credibility of the algorithms and the predictions?* And so, one way we've done that is to include guidance about the predictions. With each prediction, you get a couple of columns that describe how the prediction was made. Meaning for instance, if age is a really important feature in heart failure risk and higher age corresponds with higher risk, an older patient is gonna have age pop up as a feature that drove that prediction. So, it kinda follows the clinicians thought process, where they say: *okay the patient is maybe overweight and a smoker*. and if they see the algorithm saying *okay maybe the patient is overweight and a smoker*, they're gonna be like *Oh I agree with that.*

L: Yeah it's definitely important to help clinicians build trust in your models or else they'll just sit there and not really impact care. So that's been critical in our work thus far.

M: Mhmm so that's been really exciting. And then we're hoping to include more functionality of that type just to make it easier on clinicians and people who are trying to use the algorithms. And so one thing we'd like to actually include is even a way to say *okay Joe's heart failure risk was a point eight out of one, which is pretty high. He's not doing so well. But, if he were to lose twenty pounds, that would lower his risk by twenty-five percent.*

L: Offering guidance to the clinician's to goal-setting.

M: So it goes beyond the prediction and offers guidance.

L: Yeah and that's an active area of machine learning research. We're learning with the entire community and we'll be presenting as to exactly what we're finding each week.

M: So, speaking of learning, we've gotten a basic tour and maybe people want to get a little bit of foundational knowledge, do they need a PhD in machine learning?

L: No no no. Exactly. And that’s a fantastic segue. So, next week we’re gonna be talking about what other educational resources are out there that are fantastic, specifically, MOOCs. And so we’ll go over how we can use data science to find the best MOOCs out there for data science.

M: It’s a little meta, but I think—

L: Yeah that’ll work.

M: that’ll be a great one.