Narrator: 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 example. And here's your host, Levi Thatcher.

Levi Thatcher: Hi, everybody. Thanks for joining us for the hands-on healthcare machine learning broadcast. I'm Levi Thatcher and we're excited to have Taylor Miller join us.

Taylor Miller: Hello.

Levi: And Taylor, what's on tab for today?

Taylor: We're going to hit the Mail Bag and see what kind of questions we've been asked this week. And then we're going to look at open source tools for analysis for team collaboration tools - kind of explore some tools today.

Levi: Awesome. Very exciting.

And Taylor is from the data science team at Health Catalyst. And we're thrilled to have you with us.

And just a couple of minor things before we get going.

Taylor: Yeah.

Levi: So if you want to adjust your screen resolution, if you can't see us super clearly. You know, pick it up to HD if you want - there in the bottom right-hand corner of YouTube. And we'd love for you to subscribe so you don't have to worry about, you know, finding these each week, so you get a reminder. And also, login to YouTube if you want to chat and send in comments during the broadcast because we thrive on that. And if you want to help healthcare.ai, you could subscribe and become part of the community. You get the blogpost, make it so you don't have to worry about finding these things manually.

So, starting with the Mail Bag, Taylor.

Taylor: All right.

Levi: And the first question we have this week is, you know, it’s kind of broad but we love it. “How do you frame a business question for machine learning?”

How do you start down that path?” Let's say your business kind of knows what it wants to do. So, what's the direction you take from there?”

Taylor: Sure.

So I think the first thing to look at is, is to think about, “All right, what kind of a question is this? Is this a purely exploratory question where we are trying to find signal in noise and we may not know what that is? Or is it a question of predicting a specific thing where maybe we have an outcome? Is this widget green or red?” And if it's that, the next thing we’d want, we would call that supervised. With machine learning, there's supervised learning and unsupervised. Supervised is where you have a target that you're trying to predict--

Levi: Awesome.

Taylor: --and you have that.

Levi: So we're not always predicting at machine learning. That’s a good point that we've emphasized enough.

Taylor: Yes.

Levi: And so, let's say that we do want to predict something.

Taylor: Sure. So, if we want to predict something, first question is “what are we predicting?”

Levi: That's a great question, right?

Taylor: So track that down. Get a spec around that. Find that column, or that feature, or that outcome that you're trying to predict and then we have to isolate that.

Levi: Awesome.

So, hopefully you have data around what you're trying to predict. That's an important step.

And let us think of a couple of different examples of this. So, in healthcare, okay maybe you have a heart failure cohort and you're wanting to predict whether these guys will come back to the hospital within 30 days. This population - are they coming back to our hospital?

When you get to that point, you're going to want to split these guys into a train and test that. That’s something you’ll hear come up a lot in supervised machine learning. The idea is that, “Okay, we're going to predict on half of these folks and

we're going to train the model on the other half." Maybe it's not half but that's the idea. You split it into two populations. And it gets a little bit tricky with different types of questions.

So let's imagine that you're predicting a population health level question. So let's say that for everybody that comes into my hospital, I want to know— or, let's even go broader than that - for everybody, in my health system, who's likely to come into my hospital this year. So, how do you split that into a train and test it?

Taylor: Well, that's tricky. You can do that. You can start to think about it in a few different ways.

So, the ideal way to have a train and to test it is that the training set represents the population that you're trying to build a model on. So there's lots of different ways you can slice that. Maybe you want to say, "All right, well let's take all of the past 10 years data, or one year of data, or last quarter's data and let's use that as our training set." And then we can apply that to all of our test set as it were or are real live data we want to predict which would be, in this case, everyone in that system.

Levi: The test set are those folks who're getting predictions each day or however often you want to give them a prediction. Great explanation.

So with the heart failure cohort, if you want to predict a 30-day readmission likelihood. Anybody in the past that's come into your health system, that's been out of your hospital for 30 days, they're used to train the model. And then folks within that 30-day window, let's imagine anybody within the hospital or that's been released, you know, within that 30-day period, they're getting a prediction each day. And it's nice because you get to see their risk profile change over time. We can also stratify that patient set as to "who's most at risk in your hospital today?"

So, two basic examples, but let us know in the chat if you have further questions as to some of those details.

Second question in the Mail Bag this week, "What are the security implications of using healthcare.ai?"

Great question. And we get this a lot? So want to run down kind of just the basics of that?

Taylor: Sure. Sure. So, I mean, obviously, when we're dealing with healthcare data and PHI, we have to be extremely, extremely careful. There's piles of regulation that help us be careful with that data because that is data about us. That's our most intimate data.

So we thought about this a lot. In some places, you used to see people passing data around in csv files and that's just not a great way to go. So, we approached this problem as "Okay, we have healthcare.ai as a tool. Let's bring the tool to you and your data." So, it's definitely not a cloud service. You're sending data anywhere. This is a tool that you download, you install, and you use on your data - in your environment, protected by your own data policies.

Levi: I like that. The tools come to you. Yeah, the data's not moving around.

We actually have some chat coming through. And Brent wonders, Taylor, if you could give a little bit of background about yourself. That's a great idea, Brent. Tell us a little bit about you, Taylor.

Taylor: Oh, sure. So, I'm a pharmacist by trade. I practiced primarily in a community setting. And I've done a fair bit of software engineering for quite a while. And I'm new to the team here, so it's pretty exciting I'm sure.

Levi: Happy to have you.

Taylor: Yeah. It's a pretty exciting merger of my health background. And with that, comes a lot of the frustration of the lack of good tech and good tools in healthcare as a healthcare professional, so I kind of bring that perspective to the table. And yup, stoked to be here.

Levi: Fantastic. Pharmacist - data scientist. We're thrilled to have him.

So that's the Mail Bag for today.

Now, please send a chat across if you have any questions. We're excited to make this interactive and to learn what problems you're facing? What details are not coming across these broadcasts? So please let us know.

And we're going to be a little bit shorter today in our broadcast. We're going to talk about tools in data science – specifically, open source data science tools and kind of break it down into a couple of different categories. So, what's first on the docket?

Taylor: Yeah. So first thing on the docket, we're going to talk about some analysis tools - some very basic analysis tools. We're going to look at survey of a few machine learning tools. And we are going to look at a few collaboration tools to help teams collaborate. And then we're going to talk a bit-- a little bit deeper dive, still pretty shallow, but about open source tools and version control and some interesting things like that.

Levi: Awesome.

So analysis. If you're wondering, “How do I do analysis in healthcare?” Yeah, that’s a pretty simple question, so you don't want to buy any tools. You want to get started today on your laptop. It’s what we're all about. So, what are some of those popular tools, maybe in Python and R?

Taylor: Yeah. So let's talk for just a moment about-- so Python and R are programming languages. And R is a statistical base language. Python is a general use case language that’s widely used in data science. And there's things you can use. Some people call them packages or a library. But it's a suite of software that you download into your environment. You use that. Healthcare.ai is one of those tools but we're going to talk about some other ones today.

So let me show you a couple of tools. Let's see. One of the ones that we like for Python is called Pandas and we've written about this. You can find some posts on our blog. It's a really, really optimized and highly-focused tool about data analysis. You load in your data. It can talk to databases, flat files, Excel files, CSVs – anything, you name it. You can get it in there. And you can do very easy summary statistics on it. You can do visualization tools. So a lot of—

Levi: Yeah.

Taylor: So a lot of things like subselects, and queries, and grouping, and pivoting.

Levi: So, learning about your health data. Is it specific to healthcare?

Taylor: No. No, no. Nope, you can use it for any kind of data.

Levi: Yeah. I believe it started out perhaps in the financial realm. So yeah, it comes from the financial realm. I guy named, Wes McKinney, I believe was the developer behind it.

But, of course, broadly applicable. I believe you have a blogpost using Pandas up there, don’t you?

Taylor: Yes, that's correct. Yeah, it goes over some of the intro from assuming you know nothing, right up until some neat manipulations and you can do that in half an hour or an hour.

Levi: Awesome.

So since we're excited about is providing these tools to you for free. And not only saying where to go to check out and download the tools but also how to use them on your data.

So if you go to the healthcare.ai blog, you'll find that from the main page. Taylor has a post from just a couple of weeks ago, I believe, where we've actually stepped through a particular data set. I believe it's a free open data set?

Taylor: It was, yes.

Levi: And you can see how to do that yourself. You can follow along in the actual Python code. And we'll keep trying to do that every couple of weeks on the blog.

Taylor: Yeah. So that's one of our favorite tools for Python.

If we go to the R language, one of our favorite tools is DEEP flyer and maybe you can talk a bit about that?

Levi: Yeah. So R, like Python, is great for data manipulation, data analysis. And the idea what DEEP flyer is very similar to Panda's.

So, let's say you have some data set on some, you know, a quarter of your population. You're looking at heart failure patients and you're wanting to learn, "Okay. Well, how is their treatment effective in department A versus department B?" So you can do subselects. You're going to filter group by's. Maybe those terms don't mean much to you but the idea is that you'll be able to slice and dice your data and compare how maybe one department's doing against the other in terms of mortality rate or readmission rate and all those sorts of things.

And again, on the website, on healthcare.ai, on the blog, you'll find a couple of articles where we break down some healthcare data using R DEEP flyer. And we have the code up there so you can actually follow along using a free and open data set.

Taylor: The other thing worth mentioning about both of these tools in particular is that these are heavily used in the data science community. They're very, very optimized for speed. So this isn't something that you're going to be fiddling with,

and waiting, and waiting. I mean, you press go and depending on the size of your data set, it's very fast. You can get some insights very quickly.

Levi: Exactly. And I think open source shed its reputation as being unchecked or untested. I think, ten years ago people would have a hesitancy to use open source. But now, even though there's tools like, you know, SaaS and other expensive products out there that you can buy, you really shouldn't hesitate too much in terms of the user interface and usability of these tools.

Taylor: Absolutely.

Levi: So that's the analysis portion - how do you do analysis in healthcare data.

What's next?

Taylor: So the next thing, let's talk about a couple of machine learning tools.

Levi: Yeah. Exciting.

So, okay.

Taylor: It sounds like we lost our audio. We apologize. Those people have already been fired. But we sure are back.

Levi: We are back.

Awesome. So apologize for that.

Let's go into the machine learning section of the toolkit. And again, we're going over free and open source tools that you can use on your data to accomplish awesome things. So, let’s just start with Python here.

Taylor: Yeah. Let's go with some Python. Let me show you one of my favorite packages for machine learning that's widely used in data science. It's called scikit-learn, a tremendous library. The most incredible thing about this library is that it has in it, piles and piles and piles of different machine learning algorithms.

But the beautiful thing is that you don't have to learn the intricacies and the trickiness of this. They've done a ton of work to make using any algorithm very, very, very similar. And that's one of the most amazing things about scikit-learn.

Levi: So you don't have to be a mathematician or a PhD to do data science.

Taylor: Absolutely.

Levi: You know, scikit-learn has this fantastic documentation. Like, we really recommend you check out this site in particular. Great tutorials, visualizations. Machine learning can sound scary. But really, they do a fantastic job of breaking it down.

Taylor: They do.

Levi: And so, that's the Python side of things.

Taylor: All right.

Levi: Yeah, should we go to the R—

Taylor: Yeah.

Levi: The R equivalent. Is there an R equivalent?

Taylor: Let me show you the R equivalent to this. And there's a few. One of those is – oh, my goodness. It's called caret. I forgot the name for a moment.

Levi: With an E-T-.

Taylor: Yeah. As in boop-boop. C-A-R-E-T. Similar.

Now, you can talk a little bit about that?

Levi: Yeah, so Caret's really how I got first into machine learning in R. It offers up a lot of great functionality in terms of - Are you doing regression? Are you doing classification? Do you need to do pre-processing?

It's probably the most popular machine learning package in R. It has great documentation and community support. It's been around for a while so you don't have to worry about too many bugs or frankly if any bugs at all really.

But the idea is that the documentation is up online and these sort of tools, just to make a point, are what we formed healthcare.ai around. So we pulled the best out of the machine learning running community in terms of open source tools and put them into healthcare.ai. So, if you're finding, as you dive into these tools, that you know maybe it's a little too in-depth for you, healthcare.ai might be a nice will fallback, a little easier user interface.

Taylor: Definitely.

You know, that's one of our aims at healthcare.ai is to take some of the healthcare-specific concerns that would normally take quite a bit of research and practice and wrap that into these already very excellent tool sets.

Let’s switch gears for a moment and talk about a couple of our favorite team collaboration tools. Let me show you a couple of those right now.

Levi: Yeah, I'm excited.

Taylor: One that we've been using a lot lately is called Trello. And real easy, sort of interactive board for tracking to-do tasks. And you can assign things. And you can comment on things. But it’s just is really lightweight. Awesome apps for any device.

Levi: And it's free?

Taylor: Totally free.

Levi: Wow. So how do they do that?

Taylor: They have premium levels. And if you're a heavy user, you can pay for a higher tier but—

Levi: Okay.

Taylor: --a really great tool. A really great tool for small teams.

Levi: Yeah. So we're actually using Trello to track some internal projects as well. So definitely check it out it. It can relate to your work in code but doesn't have to. Any project maps pretty well to what Trello does.

Taylor: Yeah, so the other favorite team collaboration tool is slack. You know, instant messaging has been around for a long time particularly in the business setting. But slack is sort of a re-fresh and more fun look at that. Very easy to communicate with people. They've got video chatting, we use to connect with some of our remote team members.

Levi: Yeah.

Taylor: Really slick.

Levi: Group message. It’s kind of replacement for e-mail and things like Skype or Instant Messenger if that’s still used.

Taylor: Yeah. But you can search it. You can categorize it. Really slick.

If you haven't played with it, check it out. It's awesome.

Levi: Yeah. Free tier as well, right?

Taylor: There is a free tier as well.

Levi: Awesome.

Taylor: Yes. Yes.

Well, let's pivot over to talking about open source tools.

Levi: Let's do it.

Taylor: We're going to go through this-- can you pull this up for me?

And we need to talk about some basics today. And what we're going to look at is what version control is. We're going to talk about what Git is our. We're going to go over some terminology, just some basics, and kind of the reasons why.

So you may be starting a project and you've got some code or you've got a script. You're beginning some machine learning. And here's your little script. Okay. So here's your little script. And just, we're looking at my screen now. And you've written your script and you know a few weeks go by and you forget and maybe forgot some code. We've all been there but there's this beautiful thing nowadays. It's no longer Planet of the Apes. Now, we've got nice tools for these things to handle. Imagine the most incredible undo tool you've ever seen. So you've got your code and we're going to talk specifically about Git. There's various version control systems. Git is kind of the gold standard right now. It's free. It's open-source. No restrictions on that at all.

Levi: Is it old?

Sorry. You know, you might have heard about Git but does this come from 60s, 70s? When?

Taylor: No. I don't know the history.

Levi: It gotten a lot more popular in the last couple of years, we'll say that.

Taylor: Yes. So the whole idea with Git is that you've got a repository. And a repository is simply - you can think of it as a place for your stuff.

Maybe I'm going to do a model. I'm going to start learning machine learning and I want to predict the readmission rate. I'm going to make a folder called “readmission” and that's going to be my repository. So that's one of the terms. So we’d call this our repository. I've got my script in here whether it's R, or Python or you name it.

And the next term to know about Git is called commit. And that's sort of a point in time where things are saved. If you've ever worked with Microsoft Word. it has some revision history, or Google Docs has revision history. That's exactly what this is in a slightly more powerful way. And those are called commits.

So if we're making our code, and then we make some changes, maybe we found a bug. And let's say we added a new method or a few new lines, maybe we deleted some things and such. And we feel like that's a good stopping point or a good place to save. It's a snapshot in time.

Levi: You have something working for that moment.

Taylor: Yup, exactly. The beautiful thing about Git is that any point, you can go back to any in your entire history. And it might be a simple history. It might be three things. It might be hundreds and hundreds of commits.

So we've got this commit here. And this just goes on. As time goes on, you build up this - maybe you've got a second file now that does some other things. Da-da-da-da. And this is another commit. So these are commits along the bottom here. And, “Oh, wrong button!”

So as you as you go through time, you build up this history. You can roll back. You can roll forward. You can collaborate with people. But you might think, “All right. Well, I want to back this up somewhere.” And that's where the idea of a remote origin or a host comes in.

So there's lots of ways to do this. You could do this yourself if you want. You could use a service and we'll talk about a couple in a moment one of those of Github or bitbucket. They're free tiers. And this is a place where you can put your repositories online.

And this is sort of the foundation of when collaboration happens because you're over here but you might have a team member who's remote over here. You might be working with an open source package so there's all these collaborators.

And what happens is, once you've got your repository set up locally, you push this to your remote. So, push is—

Levi: And that's one of the—sorry, the main point, is this right here - local versus remote, on your computer versus on the server. And the idea is that having it on the server means that you can collaborate with other people.

Taylor: Absolutely, so there's this idea of pushing and pulling. So I'm working on my code with Levi. We're developing a model. We push it to a remote server which handles the backup and all that stuff. We don't even need to think about that. That problem is solved. Then he might pull that down, add some new functions or maybe change it a bit, so all these team members can work together. And Git is one of those incredible tools that help you deal with complex systems in a distributed manner.

Levi: That's amazing.

Taylor: Yes.

Levi: That is a beautiful.

Taylor: It is.

Levi: We appreciate your skills - your money levels.

Taylor: Yup. The other nice thing to think about or another term that we should talk about is the idea that of a DIFF. So between a commit, these things have changed. So something's changed from here to here. This is called a DIFF. And there are tools - there are some command line tools and some beautiful GUI tools that allow you to see, "Hey, man, this worked Friday. Somebody collaborated with that over the weekend. I come in Monday. Now, it's broken. What changed?"

Levi: Happens all the time.

Taylor: All the time. That's part of collaborative softwares. Things change or they break, or, "Oh, man, I did it this way but I changed it." So it allows you to go back in time and see. You don't have to read through the whole file and try to compare it yourself. It shows you and highlights. "Hey, this is the specific line that changed. Here's who did it. Here's when, et cetera." So that's sort of a brief overview—

Levi: Oh, that's fantastic explanation. It—

Taylor: Of Git.

Levi: I think we talked about how we're excited to do this for an entire broadcast one time, to kind of walk you through how do you make a change for an open-source software package on your local machine. So we'll save some of that for another day and go through in-depth.

We have a couple questions coming through. Before we end, we just want to touch on those real quick.

Awesome job, Taylor. It's really fascinating stuff how much Git and Git hub helped us.

So, question before we end, Jonas says, “Can you talk about what roles within healthcare systems you normally partner with to actually do some of this learning.” And the idea is that we usually work with data warehouses as they're called. Data warehouse is a place where you have your data stored in one central server - you can think of it as. And so, we work with both IT folks that manage the warehouses as well as the clinical experts that are trying to drive these improvements in acceptance rates or readmission rates. So it's through various parts of the organization, actually. And Health Catalyst is kind of split along those lines as well. We have the technology and then the clinical folks kind of pairing for these outcome-improvement projects.

And then, one more question, “Is it fair to say then that you think that there's a place for citizen data scientist?” That's kind of the idea what healthcare.ai is that you don't have to be a data scientist or PhD to do machine learning. Using tools like healthcare.ai, we're trying to make it such that anybody can get involved and get their hands dirty on their own data on their laptops.

So, anything else? I think that about—

Taylor: Absolutely.

Well, I mean, just to go back to the prior question. I mean, these can definitely be used in big systems. But these can be used for really small projects. I mean, maybe you're in some small research group but you've got access to some great data. You want to build a predictive model. You can use healthcare.ai or some of these tools we talked about today to do that. So you don't necessarily have to be in the scale of a large, large data warehouse system.

Levi: Yeah. That's a fantastic point.

We want to help the small house systems as well.

And more questions coming in through the chat. So, one person says that, “Most of healthcare data is 1:N relationship between data like patient with multiple DX's, multiple labs, and vitals. Any suggestions?

Okay. So, data aggregation—

Taylor: Yup.

Levi: We talked a little bit about tools that can help with that. That's really coming in your data pre-processing step. So, before you feed data the algorithm, you might have to grab maybe Joe’s weight from last month instead of having his weight from several days across this long time period. And we can do a whole broadcast on that topic along.

Taylor: Sure. Yeah, we could.

Levi: Awesome. So, keep the questions coming through. Send us an e-mail and we'll get your questions next week. We really appreciate your time.

Thanks, Taylor, for joining us.

Taylor: I appreciate it,

Levi: We'll talk to you soon. Thanks, guys.

Taylor: Thanks.