EPISODE 1268

[INTRODUCTION]

[00:00:01] JM: Data exploration uses visual exploration to understand what is in a data set and the characteristics of the data. Data scientists explore data to understand things like customer behavior and resource utilization. Some common programming languages used for data exploration are Python, R and MatLab.

Doris Jung-Lin Lee is currently a graduate research assistant at the University of California Berkeley while also earning a PhD in information management and systems. Doris also did her undergrad at Berkeley studying physics and astrophysics. She's currently developing Lux, a Python library for accelerating and simplifying the process of data exploration. Her research and work with Lux is aimed to make data science more intuitive and accessible to end users. In this episode, Doris joins us to discuss data exploration, and her research and development of Lux.

A few announcements before we get started. One, if you like Clubhouse, subscribe to the Club for Software Daily on Clubhouse. It's just Software Daily. And we'll be doing some interesting Clubhouse sessions within the next few weeks. And two, if you're looking for a job, we are hiring a variety of roles. We're looking for a social media manager. We're looking for a graphic designer. And we're looking for writers. If you are interested in contributing content to Software Engineering Daily, or even if you're a podcaster, and you're curious about how to get involved, we are looking for people with interesting backgrounds who can contribute to Software Engineering Daily. Again, mostly we're looking for social media help and design help. But if you're a writer or a podcaster, we'd also love to hear from you. You can send me an email with your resume, jeff@softwareengineeringdaily.com. That's jeff@softwareengineeringdaily.com.

[INTERVIEW]

[00:01:51] JM: Doris, welcome to the show.

[00:01:53] DL: Thank you for having me, Jeff.

[00:01:56] JM: Your work is in data analytics, visualization and human computer interaction. And the first topic I want to explore with you is human in the loop. Now, human in the loop can mean a lot of different things when it comes to machine learning. What are the different places where a human is inserted in the loop in machine learning?

[00:02:20] DL: Yeah, that's a really great question. I think when we look at the data science lifecycle, there are various stages from the very beginning. You're ingesting your data. You're cleaning your data. And all of that, eventually, at the end day, you're going to build a machine learning model, right? And a lot of the human in the loop process is even before you start the model building process, you have to really understand your data. Understand what are the features that you're building? What are the implicit assumptions around the data collection and all of that?

And so, obviously, there's also some human loop aspect to the machine learning development process. That's less so the focus of my work. A lot of my work is around this early phase, which is really trying to understand your data and being able to visualize it and understand some sort of insights before you kind of jump in and do the model building.

[00:03:16] JM: So you wrote a paper on human in the loop perspective on AutoML? Can you give me an overview of what your paper explored?

[00:03:25] DL: Yeah, so that paper, it's been a couple years since I've written that. But I think, overall, the paper was trying to explore this idea of what does it mean to have automated assistance in the process of machine learning development. And by and large, I think there was three different phases of automation that we were talking about in that paper. And we drew this analogy with like self-driving cars. Like a self-driving car could be fully automated, or it could be kind of a mix initiative thing, or we have our current cars, which are there's some level of automation that is built-in. Like as the driver of a car, we don't really need to think about how the gas piston in our engines work. We don't really need to think about how the gas brakes actually work. So, at some level, even though our car is very manual at this point, there's still some level of automation. And this is where the machine learning, the landscape of some of these open source machine learning tools that are out there, like scikit-learn and other packages are kind of these you're manually developing these tools. You're manually developing these pipelines in order to achieve some sort of objective, right?

And when you introduce more levels of automation, we now are seeing tools like H2O and other AutoML tools that allows users to specify at a very high-level what is the objective that you're trying to achieve? Is that a classification task? Is it a prediction task? What are the variables that you're interested in predicting? What are the features that might be relevant? And then the system performs some sort of search or some sort of automation to figure out what is the best machine learning pipeline, or what is the best workflow for the given task that you're interested in achieving. So there's also almost a spectrum of automation that could be introduced in different phases, or different – I think the better way to put it as different levels of automation that could be introduced in your machine learning development workflow.

[00:05:47] JM: Now, something I always like to ask people who are in academia is what's the degree of overlap between your research and the problems that are tackled in the more industrial domains like at the big corporations?

[00:06:04] DL: Yeah, that's a great question. And it's definitely a question that I've thought a lot about as a system builder in academia over the past couple of years. So a lot of my work is at the intersection of databases, data management, and a human-computer interaction. And one of the very successful techniques that a human computer interaction researcher can bring in in more of an academic setting is you could build these prototypes to test out various tool designs. And the purpose of that is really to understand for a prototype that you're building or a system that you're building, and a very kind of maybe, in some sense, a very narrow slice of that feature space. How do users actually interact with these tools, and those contexts? And really trying to understand how those features are enhancing or perhaps influencing how the users are using the tool. And by learning some of that, those lessons that we distill, in terms of understanding the design space, understanding how these features, how users interact with these features, it could inform sort of these larger systems, projects that industry research lab, or other industrial applications could learn from.

And so there's sort of a back and forth between, like think of kind of a synergistic relationship between academic research and systems in terms of thinking about systems design, and new sort of features and how humans are interacting with these features, as well as these larger or more ambitious projects, which are very streamlined and easy to use. Building these tools that people can actually tackle, take on and be able to use in their day to day workflow. So there's kind of a feedback there.

[00:08:09] JM: One of the projects that you're focusing on today is Lux, which is a platform for easy data exploration. It's a Python API for visual discovery. And I think automating intelligent data discovery or surfacing intelligent data discovery is a pretty important problem, because data sets can be so big and have so many directions that you can analyze those data sets within that you really want the machine to help you surface trends that allow you to visualize that data more intelligently. Tell me about the domain of intelligent visual discovery and why you built a tool around it.

[00:08:48] DL: Yeah, definitely. One of the things that we found earlier on in terms of my research, was a lot of times during these exploratory data analysis phases of your data science workflow, and this is usually at the very beginning of your – You kind of get a new data set, and you want to learn a little bit more about what's going on with your data. There's a ton of decisions that people have to make in order to figure out what are the relevant paths of exploration I should take? How should I process my data correctly? And ultimately, how do I visualize my data? How do I look at my data in a way that allows me to extract meaningful insights from my data?

And so there're all these decisions that one would have to make, and it's not necessarily clear to an end user what is the right decisions to make. And a lot of the times it really hinders the flow of exploration and helping people get to these meaningful insights. And so a lot of the work that we've been doing was trying to understand how do we provide a level of assistance or automation in order to help people more easily discover these insights without thinking too much about what exactly is the sequence of steps, or the sequence of operations that you need to perform on your data in order to get to those insights.

[00:10:21] JM: Now, in order to find interesting trends in a data set, it seems like there're so many common tutorial directions you could go in. How do you efficiently evaluate all the different directions that you could analyze the data set in?

[00:10:39] DL: Yeah, I think that's a great question. And I think that's something that we're constantly trying to understand and do research on, is being able to sort of understand what are the things that people are looking for in their data, and working our way backwards to build these recommendation systems or these automated assistance tool and proactively showing those to the users. And I think, for us, a lot of the exercises that we've done was to look at what real data scientists have done in the wild in terms of the types of analysis that they're usually are doing in a Jupyter Notebook or in their Python scripts. Very commonly, what would happen is people would look at, for example, univariate distributions are bivariate distributions of their data, or maybe they would be computing some sort of standard statistics to understand some skewness or correlation related to what their data looks like. And so, our system essentially needs to be able to consolidate some of those human domain knowledge that these data scientists – The typical things that these data scientists are doing to be able to show those to the users in some ways.

[00:12:03] JM: Describe the usage of Lux. When I'm booting it up? What happens? And describe like a typical prototypical use case.

[00:12:16] DL: Yeah, I would highly encourage anyone interested in Lux to check out our GitHub page. The pictures and demos would do a much better job than what I can explain here. But I'll give it a shot. So when you kind of install Lux and you import it into your Jupyter Notebook, what you could do is, essentially, you could create a panda's data frame or you could load in something from your CSV. And with Lux, you can simply print out your data frame in your Jupyter Notebook. And Lux would recommend a set of interesting visualizations that might be useful for your analysis. And these visualizations are displayed as a Jupyter widget, which is directly inside your notebook. So this is typically something that would show up in the output cell of whenever you print a data frame.

And these visualizations are essentially recommended for free to the users without needing to write any additional lines of code, or change any existing pandas or DataFrame commands that they might already be using. And in terms of what that workflow would look like. Obviously, when you print out the data frame, we have this alternative visual way of looking at a data frame. And on the top left hand corner of your output widget where you typically see a panda's table, you'll find this button where you could click on this. And you can toggle back and forth between the visual view that Lux is showing, which contains a bunch of different visualizations, as well as the tables view, which is what pandas is showing by default.

And another core part of the notebook workflow is that all of this is – Given that Lux is designed in a way that it's very tightly coupled with Pandas, what that means is that any data transformation that you're doing, let's say you're dropping an N column, or you're deriving a new column, or making some sort of data formatting or transformation to your data frame. Whenever you do that, it's reflected on the data frame variable that you're working with. And once you print out that data frame again, the recommendations automatically gets updated, because we re-compute some of the recommendations that are based on that.

So in some ways, one of the things that we were really interested in in going in and building Lux was this idea of an always on visualization. The visualization shouldn't be something that happens at the end of all of your analysis. While you could do that, but visualization can really help in other parts of your workflow, including data cleaning, as well as when you're doing your analysis or exploration. A lot of times people would be finding anomalies or unexpected behaviors in their data by simply just looking at the visualizations and seeing, “Oh, hey, there's a data point that doesn't fall in the cluster or something along that line.” And the visualizations in it are intended to inform these data cleaning or analysis decisions.

And so the goal of Lux really, in terms of thinking about what typical workflow would look like, is to help people, I guess, within – The goal of Lux is really to think about in terms of your overall notebook workflow, how can we minimally change what you're already doing with your pandas and your existing libraries that you're using? But also provide an alternative and visual view of experimenting and understanding your data. And so that's kind of what a typical workflow and Lux would look like.

[00:15:59] JM: Are there any bottlenecks in terms of dataset size or column – Or, I guess, row count? Are there any bottlenecks when the data set gets really, really large?

[00:16:13] DL: Yeah, definitely. I mean, anything that we're doing in terms of the recommendations can be considered as an overhead on top of – Whenever people are typically printing their data frames, it's just 5 to 10 rows of data. So you could think of this as fetching the first 10 rows of your data frame. And so anything more sophisticated that we're doing is going to be an overhead on top of that. And so we've done a lot of optimizations in Lux looks to make sure that we can minimize that overhead as much as possible so that we're still being able to show the recommendations in a relatively fluid way. But you're definitely right, that when the data frame gets really wide in terms of the number of columns, or when the data frame gets really tall, which is the number of rows, there's definitely scalability bottlenecks that will start the kicking in those regimes.

In Lux, we kind of tackled those two different things, two different types of scalability challenges differently. So in one of our recent papers, we've done a lot of work on developing new optimization techniques to work with these wide data frames. So these wide data frames are typically you might have large numbers of features. And in terms of the recommendations that we're showing in Lux, we might be going through all possible combinations of them and computing some sort of interesting metric or score to rank and sort these visualizations. So it requires that we're generating the data, or we have to compute the data for each one of these recommendations.

And so a lot of the work around how do we handle wide data frames is around can we approximate or or get at a very early estimate of how interesting these visualizations might be down the line and do some sort of early pruning to be able to narrow that search space a little bit to be able to show the **[inaudible 00:18:20]** to the users without being absolutely sure and going in and calculating every single visualization exactly. So that's one example of the types of optimization that we have applied to Lux. And there's a couple more that we've also been working on trying to add those into our releases.

[00:18:44] JM: Can you share more about some of the engineering problems you've encountered when building Lux?

[00:18:51] DL: I think one of the challenges that we encountered, this might be an engineering challenge, but it might also be an overall challenge that is inherent to these systems, is that for every recommendation systems that whether you're recommending movies, or you're recommending visualizations, or product items, there's always going to be a precision recall trade off, right? Your recommender can recommend something that is very generic. So, for example, in our case, we're recommending things like correlation and distribution, which most people are kind of interested in. And then versus something that is more specific. So something that's very high-precision, but maybe only 10 out of 1000 users would be interested in.

And so in designing Lux, there was definitely a lot of engineering challenges and trying to understand what is the right balance and what is the right smart defaults that we should pick that works for most of the users and most of the use cases. And picking up those design choices aren't always obvious. It requires some thinking into what is the intended use case for a tool like this. And what is the data set that is at play? And how do people – What makes sense as the next steps in their analysis? And so we've had to make a lot of these choices when we're engineering Lux to essentially pick a point where it's something that is standard, either standard practice that most people would be happy with these preferences, or something that is a little bit more specific. And at what point do you add in these API's or the capability for users to override those choices or be able to specify something that is a little bit more specific to indicate this is what they want? And so a lot of the our work is around this middle ground and trying to pick the right balance and the right interaction modality for the user to specify these things.

[00:21:03] JM: How do you see the usage of Lux among your user base? What are the most common use cases you're seeing?

[00:21:13] DL: Yeah, that's a really great question. And we've spoken to many users who have used Lux. And this spans from hobbyist type of users in education and Kaggle competitions, to actually academic research and the sciences. And we've also seen a lot of business data science adoption spanning from retail, insurance, media companies, and healthcare and so on. So we've seen kind of the breadth of how Lux is being used in these different contexts. And some of the typical workflows and use cases that we've seen is handling the earlier phases right before you do the machine learning, where you're doing a little bit of EBA to better understand your data. You might be plotting a couple of visualizations and using Lux to take a look at the columns in your data before you actually jump in and do the model building and the model development process.

So we've seen a lot of kind of that workflow either within a notebook or across different notebooks. Other things that was kind of interesting that we've seen was that people typically are using Lux alongside their favorite plotting tool, whether it be the matplotlib, or seaborne, or directly via pandas, df.plot. And I think one of the things that we wanted to do was to make sure that there was a seamless integration between Lux, or what we're showing in Lux and with these tools. And so what we've seen is that people would have these notebooks with the data frame command, printing out some data frames, printing out even some intermediate results. So not necessarily printing the DF itself, but sometimes people do aggregations. They drop values. They filter their data frame. And they just want to inspect the intermediate result. They don't necessarily store it into a variable. In these scenarios, Lux would display any sort of data frame that is printed for a particular cell. And so it's a good way for users to look at what is the effect of that operation that I did. Maybe I did a group by aggregation. And then I can inspect the data frame that was before that, and the data frame that was after that. Not just in terms of what that table looked like structurally. But in terms of what that actually did to the data itself and the data points, and how that affected the visualizations. So in Lux, we've looked at – A lot of the work that we've done was trying to understand what is a natural way of visualizing these data frames within the larger context of the notebook workflow?

[00:24:06] JM: Tell me more about the notebook workflow of the typical data science worker. Like how does Lux fit into that workflow in a little bit more detail?

[00:24:22] DL: I think from an engineering point of view, Lux is built on top of IPI widgets, which is the Jupyter widgets package that could be used for building things like sliders or buttons and things like that, and also handles sort of some of the communication with the notebook itself. And the way that the notebook widget is displayed in Lux is that it's part of the output cell. So whenever you printing the data frame, it's part of the output cell. And one of the things that we're seeing in terms of going back to your question around the larger notebook landscape and how Lux kind of fits in, we've seen that computational notebooks like Jupyter have kind of become these go-to-tools that data scientists are going to for data cleaning, analysis, visualizations, and so on. And it's also a medium in which these data scientists can be using to – I guess, taking a step back. These notebooks are a medium in which these data scientists, they can iteratively experiment with their data through some sort of code.

And so Lux really tries to bridge this gap between what people are traditionally doing in the UI kind of world, like the graphical user interface, where you're pulling up a new window. Maybe you load in a data set, and then you can plot a couple of visualizations. But then, ultimately, that data is disconnected from where you're actually going to be doing some of your scripting or your Python data cleaning and so on.

And so what we've found at least was that there's a lot of moving back and forth and having to import and export to and from these graphical user interfaces and tools in order to get out a single visualization. And then on the other hand, in terms of the notebook world, the notebook is an empty canvas, right? You have to write a lot of code. In order to do anything in the notebook, you need to write a lot of code in order to generate a plot. You need to write a lot of code to build a model and load in your data. And so what we've sort of wanted to do was to say, “Okay, what would it look like if we took some of the workflow and the benefits of the graphical user interface?” which is these rich visual feedback and the ability to sort of interact through your point and click mouse interfaces? And how do we bring some of those interactions into a notebook environment where people are writing code, but then you also get the benefit of some of that graphical user interfaces? And so it's really thinking about how do you bridge this gap between those interactive interfaces? And what a user's typical notebook workflow, a data science notebook workflow would look like?

[00:27:30] JM: Does this kind of visualization compete or overlap with other BI tools? Like there's Druid-based visualization systems or Apache power set, things like that?

[00:27:47] DL: Yeah, I think that the way that we've always viewed Lux was more in the – There's definitely overlaps between what Lux does and what other BI tools do. So for example, you mentioned Superset, as well as Tableau, or PowerBI, which all of these are dashboard systems that allow you to create these beautiful dashboards, and really fine tune your visualizations and build these amazing graphics that you can then present to a stakeholder or you could show to a business customer. And Lux is really intended to be a tool for exploration.

So the goal isn't really to create beautiful dashboards or to build these very fine-tuned visualizations. Our design principle from the get-go has been to help users get to these visualizations as soon as possible during their exploration to minimize the activation energy that is required to do that. And so I think there's definitely an overlap in terms of the features and the affordances that these graphical tools are providing. But in terms of Lux, we're not showing – The goal is to get to something that is good enough. So a visualization that is good enough that users can take a look at, understand the insight and move on to something else. Or Lux also allows the ability to sort of take a visualization that is automatically recommended to you and then click on it, export it into the form of code so that you can actually do these fine tuning within these libraries like Altair, matplotlib, and be able to export and share it to – Either putting it in your PowerPoint slide or sharing it with your colleagues.

And so a lot of the design principles around Lux was not necessarily around getting to the best visualization, or a very fine-tuned visualization that you could ideally build in some of these other BI tools. You could really customize all the colors and all the shapes in your graph and all of that. The goal of Lux is really to get at something that is good enough for exploration, and be able to communicate some sort of quick insight from your data. And then you either move on or you drill down a little bit more to look at what's going on.

[00:30:39] JM: Gotcha. So you kind of think of it as more of an ad hoc tool.

[00:30:42] DL: Yes. Yeah, definitely.

[00:30:45] JM: Do you have a bigger vision for Lux that you're driving towards today that the project has not accomplished yet?

[00:30:55] DL: Yeah, definitely. I think one of the driving forces of Lux very early on was this idea that it's a way of accelerating users towards their insights. And I've talked about the aspect of how it bridges together the UI and the BI – Kind of the UI, which is the graphical UI side of things and notebooks. But I think, really, the other thing that I'm more fundamentally excited about is this idea of being able to specify your analysis in a very high-level way and be able to work with the system collaboratively to achieve some sort of analysis outcome. And so Lux itself features this intent language that allows you to specify this – I'm interested in the sales column. And I'm also interested in maybe all the products that are sold in the USA. So those are the high level specification that you can make to Lux. And then the recommendations, as a result, get steered towards those items, and those specification, those things that you've specified that you're interested in.

So I think there's a lot that could still be done there in terms of thinking about Lux as a high-level way of helping guide users towards relevant analysis. And in that process, eliciting more feedback from the users. Being able to understand what they also want in the process. And then having this very tight conversation between the system, the automated system that is trying to guide the user towards some analysis, as well as the user being able to specify what they want. And I think even a more ambitious vision is to go beyond code and say that if we had this high-level language, or high-level way of specifying things going beyond code, how do we interact with this automated system in a way that could help us with our analysis?

[00:33:16] JM: Taking a step back, given that you have spent a lot of time in the machine learning world, a lot of time in the data science world with Lux, and you've just been doing your PhD at Berkeley. So I'm sure you have probably some communications with people in the RISELab, or some of the other data labs there. I'd love to get your perspective on the bigger picture of data engineering and data analysis. What are the outstanding problems and what you expect to see in the near future?

[00:33:55] DL: A lot of what I've been thinking about these days is really around this convergence that we're seeing in terms of what I'm going to call interactive data science. So interactive data sciences is composed of your data analysis, your data cleaning, and then some sort of machine learning. But it's in a very interactive loop where you're able to see things. So going back to what we talked about, which is similar to the human in the loop notion. So being able to be in the loop as you're working with your data. I think there's a lot of potential to that. We've already seen how the Jupyter community has contributed to this awesome open source ecosystem of tools that allows end users, data scientists to interactively work with their data in an accessible and intuitive way.

And we've also are now starting to see that the notebook itself is becoming a window to data science. It's highly interactive and accessible. And a lot of times what we're seeing is that it's an entry point for people who are starting out and learning about data science to be able to learn quickly. And we also see that there are the standard tools like pandas and scikit-learn, which are, again, windows to data science in a way that they serve as entry points. They're easily accessible tools for people to work with their data and be able to work with these machine learning models and be able to see their results and feedback from that.

And on the other hand, what we're also seeing on the other side of things is we talked about data science, but there are fields that are related to data science, for example, data engineering, business analysis, business analyst, your traditional BI type of thing. And your traditional data work is typically done through either scripting in terms of what data engineers would do. So you might have this large Spark job, and you might have some scripts that you're running through, or you're running through a scheduler. Or it's on the UI side of things, which we talked about a little bit earlier, which is that these BI tools allow people without coding expertise to really easily sort of point and click on the interface, or drag and drop some interface elements and be able to construct visualizations and dashboards very easily.

And one of the shifts that I'm also seeing in terms of industry adoption, as well as the open source tools that are being created, there's this shift towards a consolidation and essentially a convergence towards notebook as the interactive computing platform. Over the past couple years, we've seen how AWS and Google collab. There's enterprise offerings of notebooks on the cloud, as well as how there are new UI-based tools, including Lux, but also things related to dashboarding, presentations, and the creation of data applications that are highly interactive and serve as, again, a window to data science, right? We've seen a lot of changes in JupyterLab 3, as well as Viola and other open source tool in that space. So we’re almost seeing from two sides, there're notebooks serving as the entry point of data science, entry point to people who are starting out to learn about data science. And then we're also seeing the shift, where traditional data work that is done through scripting, or through the UI is also moving towards notebooks. So I'm really excited to see how this convergence towards a standard interactive data science workflow can sort of inspire domain experts, SMEs, traditional sort of people with domain knowledge, but not necessarily well-trained in CS or data science to more easily derive meaningful insights from their data through these accessible and intuitive computational notebooks and platforms.

[00:38:18] JM: Awesome. That's a great summary. And just to close off, are there any other particular problem domains that you would be working on if you're not working on Lux right now? Are there any particular projects or directions you would focus your time in?

[00:38:36] DL: I think there's definitely a lot of interest in sort of being able to – I think reproducibility is a very important problem in data science. And it's not really a problem that I've tackled in the past. But I've seen a lot of users that I spoke to in the past talk about how reproducibility is such a challenge in the existing data science landscape. And I think part of that also comes from the fact that data science is so desperate in that everyone is using different sets of tools. And it's very difficult to piece things together. And this becomes even a bigger challenge in organizations where you have a data team that is consisted of data engineers, business analysts, and your data scientist, and they're all using different sets of tools.

There's often these analysis decisions that one would have to make in order to get to your clean data, whatever intermediate form of data, or some intermediate form of data. There's always a judgment or decision that is made. And without sort of something that is a layer above that, it becomes very difficult for, let's say, a data scientists to be able to understand what was something that was done in my ETL pipeline that led to maybe an anomaly that I'm seeing in my visualization? And so reproducibility and the ability to – Not just the ability to reproduce it, but understanding what are the sets of decisions that are made either by myself or by external collaborators that I'm working with that have led to this particular artifact or this observation that I'm seeing in my notebook? I think that's a very interesting problem. I think it's both interesting and very ambitious. And I think that part of what I talked about earlier, which is the standardization towards notebook can help fix some of these, but it's definitely an overarching challenge that we'll be seeing in the data science field over the next, let's say, three to five years.

[00:41:00] JM: Okay. Well, thank you so much for coming on the show. It's been a real pleasure talking to you.

[00:41:04] DL: Thank you for inviting me, Jeff. I had the pleasure joining on the show.

[00:41:09] JM: All right. Thanks, Doris.

[00:41:10] DL: Thank you.

[END]