

Dr. Andrew Wheeler

Large Language Models for Mortals

Mindy: [00:00:00] Welcome to Analyst Talk with Jason Elder. It's like coffee with an analyst, or it could be whiskey with an analyst reading a spreadsheet, linking crime events, identifying a series, and getting the latest scoop on association news and training. So please don't beat that analyst and join us as we define the law enforcement analysis profession.

One episode at time.

Jason: How we doing? Alice, Jason Elder here with another LE, a podcast Deep dive. Continuing with our AI discussion. I'd like to welcome back to the program Dr. Andrew Wheeler. Andrew, how we doing?

Chris: I'm good. Thank you for having me on Jason.

Jason: Well, thank you for reaching out. , We've been having different perspectives on AI and analysis, and you're like, Hey, by the way, I just wrote this book.

Let's talk about it. So I appreciate you.

Chris: Yeah, thank you very much, Jason.

Jason: For those,, that might not remember, Andrew, was on the program before, I will put a link to that episode in the show notes. And so Andrew,, I do wanna start right [00:01:00] there with the book.

So since you were last on the show, you were in deep into data science and crime analysis. Coding. So I'm curious what pushed you to write this book, large language models for mortals?

Chris: Yeah, definitely. Both the last book and this particular book I wrote those based on my, I was not happy with a lot of different resources for individuals, for introductory techs that were currently on the market or even like courseware courses or things like that.

So it was born out of me working in data science. I'm a director now, so I'm hiring individuals on the market. And then I still help folks do teaching on the

side and things like that. And so I was really not happy with. The introductory materials for Python. And so I, I initially wrote that book, data Science for Crime Analysis [00:02:00] with Python.

And it's a similar sort of origin story for the large language models at my day job. For folks who aren't familiar I work as a data scientist in healthcare, and so I, I mostly build models to identify fraud, waste, and abuse and, Medicaid claims. , A lot of our work really just in the past year has really shifted from using, like traditional machine learning models to identify fraud versus using these large language models to build applications to help people do things faster. And so there really wasn't good resources that covered, , documentation for a lot of these large language model APIs. Even their own documentation is in some ways is pretty poor.

And then there definitely wasn't any resources that covered across the different providers. So there wasn't any resources I was familiar with that [00:03:00] covered both like open AI and. AWS resources and Google resources all in sort of one sort of singular place. And so that was the main motivation to write the book.

I saw a lack of good introductory materials on the market for folks to go from zero to basic competency, , for large language models.

Jason: Yeah. And I think it's great that the, this book is written for analysts, not necessarily computer scientists. What skills does a typical analyst already have that translates well into working with LLMs?

Chris: Yeah, , I think to step back a little bit, a lot of the different materials on the market now

Focus on mathematical details and basically building your own models. And so most of it, if folks, it, the common job title now is AI engineer. AI engineers, for the [00:04:00] most part, aren't building their own models.

They're using these models that are provided by these different companies like OpenAI and Anthropic. , And so they're the same models that power the chat bots that folks are familiar with. But under the hood, you can call an API the same way as like you could put in a question into chat GPT. You can do a direct call to the API and have it give back a response.

In a, using Python code essentially. And so a lot of the jobs are actually just using those APIs to build applications or not developing their own local models

basically because these general models are just so much better than anything that you can build locally with reasonable levels of compute.

The most recent models have billions of parameters in them. And you just in your personal compute that you can just buy at home, there's just no way for you to compete essentially with them. That's one of the main reasons why using the APIs [00:05:00] has become more popular. The other part with worrying about analysts if you just like Google for an example, a lot of 'em will talk about chatbots.

And so there's so much more that people are doing with these models than just chat bots. So one of the most, uh, popular applications that I'm working on now is really, it's super boring. It's just. Extracting out information from PDFs. So healthcare people are still doing a ton of faxes.

Mm-hmm. You have these like really dense medical records. A lot of it is really just using the LLMs to extract out information from those medical records. And to do that you pass in the PDF to the API and say, get me out this piece of information. And so that's like a more regular task for analysts than building the chat bot.

Jason: Yeah. I wish I had that 25 years ago.

Chris: Yeah, definitely

Jason: with telephone toll analysis.

Chris: It's it's sort of amazing. It there's so many [00:06:00] examples like that, of just dealing with messy documents. Yeah. To just like improving document processing. It's really boring. Folks probably wouldn't even call ai, but that's probably gonna be like one of the main areas where, , these tools really like help process automation.

It's just dealing with PDFs and scans and things like that.

Jason: For the listener, they're hearing LLM APIs for the first time, what does that actually mean in practice?

Chris: Yeah. So yeah, to go the acronyms, of course it's hard to keep up with everything.

So LLM is a large language model.

And so that's typically a model that uses input as text and predicts text. Going out. That's not the best definition, but that's probably one of the most regular ones. Mm-hmm. Um, and I actually have as an introductory chapter in the book, showing folks how to build a very, sort of basic, uh, language model.

And one [00:07:00] of the things I think, that it's important to understand that, is that they're models. A lot of times people assign anthropomorphic traits to the models, like talk about thinking and whether it has feelings and things like that. Mm-hmm. It's really just a very, complicated statistical model.

So I think, it's good for people to know that it's just a very fancy stat model basically that just inputs, text, and outputs text, but you can build models to be. I don't mean that to be derogatory. So you can basically build models to do really complicated things.

Just on the basis of that. And for APIs, API stands for Application Programming Interface. And so this is a standard way for you to send commands to somebody else's computer. Typically, so the way that the I'll use the term foundation model, and so foundation models are these big models that are hosted by external entities, whether it's open ai, anthropic, or Google.

Jason: Mm-hmm.

Chris: You [00:08:00] can basically send a command, you can say, give me a summary of routine activities theory, and that command gets sent to an external computer. API is just the way that you send those commands to the external computer and then they send it back, send that information back to you.

Jason: Is that API layer different from maybe using the chat interface for those that maybe have like used chat GPT and used that to ask chat GPT questions?

Is that the same thing or is the API layer different.

Chris: Under the hood, they're both using those same models. Now, essentially, when you're using the graphical user interfaces online they just send information to, uh, the API in a little bit different way than you would typically send it yourself if you're doing it directly.

So for example, in the [00:09:00] newer versions of Chat, GPT, it basically has like a consistent history across all of your different chat sessions. It basically sends information from those prior chat sessions into your newer. Session as

well, you could replicate that same behavior from with calling the API directly if you wanted to.

It just takes, it basically is just wrapping up your prior conversations and sending it to the API. At the same time, for law enforcement analysts, one of the biggest differences between calling the API directly versus using the graphical user interfaces is that the GUI tools almost always have terms of service that are like they c the conversations.

Whereas the APIs, that's typically not the case. And so, it's definitely the case that you shouldn't be using law enforcement sensitive information in the GUIs because basically they can externally go and look at those. Whereas if you're using the APIs, most of the [00:10:00] time the terms of service are, they don't permanently cash.

That information. And so the terms of service often are sufficient to be able to use law enforcement sensitive information when using the APIs directly.

Jason: Good. All right, now that we've defined LLM and the APIs I'm. I'm curious about to, just to give a general understanding to the listeners about maybe some practical things that law enforcement analysts can build with LLMs today.

Chris: Yeah, so the book really is like my, you sort of outline of all the major architectures that individuals are building with large language models. So I talked a little bit about the. The information extraction. And so that's probably one of the things that I expect will be the most relevant for many analysts.

[00:11:00] So imagine you had officer narratives of burglaries and you wanted to pull out like the modus operandi from those, not many RMSs have standardized systems to input that information, but it is probably captured in a lot of the different narratives. And so you could basically just scan all those narratives one by one, and then extract outlet, modus operandi information so

that's one of the examples. I say not many people are interested in chatbots, but in, in, or I should have said like most of the example online are doing chatbots. There's really extensions to chatbots that are chatting with your own local documentation. So I give examples of what's called rag retrieval augmented generation.

In the book, and it's basically you turn your doc, you basically, uh, turn your documents into code in a way that the LLM can basically extract out [00:12:00]

particular information and it helps you like answer questions for your policy documents. So I go through examples of, of some open source like Raleigh policy documents in chapter five in the book.

And then the last example that I give that's relevant for analysts is I show how to use what's called tool calling and building agent based systems. So essentially you could have an LLM. Write SQL code for you. Instead of you manually writing SQL code, you can be like, Hey, give me the number of burglaries in precincts for the past six months.

And folks are probably familiar with the gooey tools being able to help them write code like that. Mm-hmm. The chapter six in the book shows how to set up these systems. So it'll not, the APIs cannot only write the code for you, but it'll return the results. And the [00:13:00] reason that that's important is it can basically iterate.

So imagine the system and initially wrote the SQL results, but it's wrong, like it got a field name wrong. You can set up the tools in a way that it basically gets that feedback information so it can iterate and then correct self-correct itself with the code. And so that's what people are talking about.

Agent-based systems is a term that, oh I'm blanking on, on the word. It's like a promotional a lot of times that people are talking about it, but really. The simplest way to describe an agent-based system is it's calling a tool in the loop. And so it basically writes, code, executes that code, and then it iterates on that code, whether it's identifying errors or doing further analysis.

So in chapter six, I show examples of basically having giving an input and having it auto write a report based on like certain tool calls to summarize repeat [00:14:00] addresses or summarize like temporal trends in crime.

Jason: Yeah. I know gonna date myself here. I mean, it's going back 10, 15 years ago. No, it's longer than that, Jason.

It probably is closer than 20 that, people were using Python with Arc, GIS. To create a way to either geocode or manipulate the data in a standard way for reporting. And so they were using Python with a series of steps that were repeatable each time that you got a new data set in there. Maybe that's a crude way of thinking it, or maybe that's more on point.

Is that the right way of thinking about these?

Chris: No.

Jason: It these, these agents or not?

Chris: It's definitely it. So you could think about scripting a report, and for a lot of scenarios you may want to do that. You may want to be like, okay, if I'm gonna do a standardized report about this [00:15:00] crime type. So a lot of analysts are just assigned to specific crime type, like robberies or burglaries or violent crimes.

And so they may wanna like actually write a system that just pipe in a particular subset of data, and it auto generates a report and auto generates a table and a line graph and a map. And there's, there's nothing wrong with that. The agent based systems, one of the differences is that they're not necessarily scripted in a particular way.

So one of the examples that I show in the book is the Google LLM APIs have the ability to basically like pull in information from Google Maps. And so one of the examples that I show is it basically, you can ask it in plain text give me the top 10 repeat addresses for, um, motor vehicle thefts in Dallas.

And then it, it, it'll. Pull out those top 10 addresses. And then I have a second tool that basically [00:16:00] uses Google Maps data to summarize what's actually at those addresses. So it'll go through and be like, oh, this is an apartment complex, or this is a grocery store, things like that.

So it's not entirely scripted, but it's pulling in information dynamically from online to basically fill in those narratives.

Jason: So I'm curious with LLMs, do they fit well? With other traditional analytical tools. We talked about sql, we talked about our GIS of course there's Excel and I two and or link charting software or e even Tableau for dashboards these days.

I'm curious how LLMs could fit in with traditional analytical tools that maybe analysts already have.

Chris: Yeah, so , there's a couple different ways that they'll fit in and [00:17:00] so I, I expect probably in the near future it'll actually be. Baked in directly into the software. So there's the ability now to basically install I forget the correct name for 'em, like toolbars or extensions in Excel that hook up directly to these LLM models like Claude and, chat GPT .

And so the tools themselves in the future, in the near future, like maybe in a year or two, a lot of these different tools are basically gonna have those hooks right in them directly. So it wouldn't surprise me. In two years if I too literally has a button that's like, Hey, use an LLM to to do, what you want me to do.

So that's one of the things that I expect is gonna happen in the not too distant future. The other way to think about it is basically building your own sort of tools to interact with the LLM directly. One of the ways to do that is what's called model context protocol. So you can basically build your own model context [00:18:00] protocol servers.

To write little scripts, essentially to do the things that you want with your tools on your own system. So the example I give with writing sql, that would be one where you can write your own MCP server to basically help you write SQL on your own data. So on your own SQL Server database for whatever record management system that you're using on your own system.

And it's basically just like a standardized way. It's the same as an API that you build yourself to interact with applications on your own system. And so you can really do basically anything that you can do manually on your computer. You could write an MCP server to have the LLM basically interact with those applications locally.

Jason: How aware are you about police departments who have. Implemented something similar to what you're describing., I'm curious if they're building it in-house or are they going [00:19:00] through a vendor?

Chris: It's basically minimal, so I'm only familiar with like a few different examples and most of 'em are tech vendors that mm-hmm.

Are doing this now. I only know of a, few examples and a lot of times it's because they're working with an outside research partner that's a little bit more technically savvy.

Jason: Yeah.

Chris: To build some of these tools. That said, if you're a really industrious crime analyst, you can build these locally on your system.

So. This new book, you definitely need to know Python. Mm-hmm. Before you'd be able to like, consume the materials in the LLMs for Mortals book. . But after that it's definitely within the can after you read the book, to be able to build

some of these local things yourself. Like I said, like writing, creating the MCP server to query your own data, that's definitely doable.

For a sole analyst who's just, who knows Python reasonably well.

Jason: All right. So there, there's an order to [00:20:00] operation for your book reading.

Chris: Yeah.

Jason: So I, I don't know if you mentioned this or not, but there's the concept of rag, which is retrieval augmented generation I'm thinking back to analysts.

Maybe they're, they're a part of, more of an investigative or intelligence environment. Maybe I give your advice to these types of analysts ON Rag.

Chris: Yeah, I think it's probably good to step back a little bit and describe what RAG is, and I know I didn't do that earlier, so, I'll go into a little bit more detail now.

But if you just go online and ask. Chat, GPT what routine activities theory is. Mm-hmm. I would bet money that it'll give like a really spot on answer. Mm-hmm. And that's because in the historical training data, like they probably use Wikipedia as examples in open AI's training data.

And so as long as that material is in their [00:21:00] training data, it'll essentially be representative reasonably well in those models with just like implicitly basically in, in the baked into the model. Now, if you ask it a question wasn't in the training data that it's basically just trying to predict text from prior text.

They'll give answers and it may even like sound correct, but it's not necessarily correct. So pretend the most recent chat GPT model was only trained from data through December 20, 25. Mm-hmm. And you ask it about, uh, an event that happened after its training date in January, February, or March.

It's not gonna have been trained on that particular information. And so it'll always provide an answer, but it's not gonna actually have access to that information. So the concept behind Rag is basically, so instead of just asking a question tell me about the news [00:22:00] on February 1st, there's a secondary system that basically does a live search on some other knowledge base, is what you would call it.

And so there's basically a separate system that goes and tries to identify relevant pieces of information. And then the same as asking your question, basically just at runtime, inserts that information into the prompt. And so it ends up being the LLM will have both that external information. And so the policy document example that I give, I, it would basically ins give it both the relevant PDF that has those policies as well as the question.

And so it's not looking at its historical information that's basically encoded in the model. It's using that information that's in the PDF to answer that question. , That's the basis of RAG. And where that becomes important is that you have all these [00:23:00] scenarios that aren't based on public information.

So if we're talking about like a local policy for a department, or if you're talking about, Hey, give me a summary of this particular person's criminal history, they're not, that's not gonna be something that's publicly available online that's going to have to be injected into the system via that rag system that the local analyst would build.

Eliann: This is Dr. Eliann Carr from the Ellensburg Police Department here to talk about the first of its kind, the Crime Analyst Census survey. This is an opportunity for crime analysts from around the world to be able to share information on the demographics that make up the field, be able to look at the relationship between commission, non-commission, and how we navigate that relationships in our career field, and also to look at training opportunities and development that will help us [00:24:00] foster the opportunities for growth and development both personally and professionally.

If you're interested in taking the survey, you're welcome to go to the link in the show notes below, sure that your voice is heard and included in the data.

Jason: I understand the concept of if the data is not there, it's not going to know, right? To your point broken Windows theory is well documented out there in the universe. What I was doing on March 1st. In 1985 is not, I guess, so it's gonna have a hard time telling me what I was doing then.

But in, in terms of the rag run simultaneously, like you were describing there I, I think I got lost a little bit to be honest with you, that when you were saying it was running simultaneously in, in that way,, is it just a matter of different resources for the system or [00:25:00] maybe I'm oversimplifying it.

Chris: Let me actually try to give a. Little bit different example. Mm. And so I actually don't, there's a few different vendors right now that are aimed at

helping, uh, detectives look at cold cases. Mm-hmm. And so I do not have any particular insider knowledge about how , they're building those systems.

Mm-hmm. But I'm gonna guess a little bit here. So don't take this as gospel, anybody. So they may be doing it totally different, but I suspect this is maybe how they're building those systems now. So imagine you just basically put in somebody's entire case file for this historical cold case.

And then you just ask the LLM. Okay. Give me a summary of the individuals who were the primary suspects and why. And so that's not gonna be information that's online, so that's only gonna be information that's available in those case files.

Jason: [00:26:00] Sure.

Chris: And so they'll basically be a two step process.

So the first step in the process will be the LLM needs a way, or doesn't even necessarily need to be an LLM. It could just be like a keyword search. So to identify the relevant pieces of information that are related to the particular question. So there could be, you could have hundreds of documents.

You could have like the, um, the coroner's report that's not relevant, but you could have case notes that are probably handwritten. That are relevant. And so it'll basically go through step, step one of the process is to identify the related pieces of information. So it may say, okay document one, document two, and document three.

Those are actually related to the question. And then step two would be. Taking that information and submitting it to the LLM, [00:27:00] the same is no different than how you type in a question and submit a prompt. You can basically submit information from those documents, which may be texts and maybe images. I've mostly been talking about texts, but all the recent models now can also incorporate images and audio in them as well.

. But you can imagine the model basically just turns it into text to oversimplify it. But so step one is identifying the relevant documents, and then step two is basically putting in those documents along with the question, and then having the LLM summarize the information. It would be no different than you manually typing the ca the narratives for the case reports, and then at the end in chat GBT GUI saying, Hey, summarize , the prior individuals who have been prioritized in the reasons why or the reasons that they were eliminated.

It's no different in the end, that second step anyway. The only difference is the first step to pull out the relevant information.

Jason: You recently wrote [00:28:00] on the unreliability of M'S. APIs so what problems did you run into?

Chris: Yeah, so a and I mean, overall. They are quite reliable.

But basically the way that I wrote the book is the book is automatically compiled at runtime. And so if I go and change a piece of code in the book, the book has code examples where you can go and see this Python input to actually call the open API open Ai API and get back the results. So it's like code snippets writing the book.

If I change the code snippet, it actually recompile the book and reruns it. And so doing that, you, when you write a book, you do so many edits. Probably, I probably compiled that book maybe 200 times I would guess, over the time that I was writing it. And so you call the APIs over and over and over again and you can identify some errors in them that don't always happen, but happen sometimes.

And [00:29:00] so it, like one of the examples I talked about using the Google Maps where you can basically like query information, put in addresses and say, give me summaries of these addresses. That was one of the most like unreliable scenarios. And instead of actually giving an error, it would just. Say something like in the actual textual response, be like, oh, I don't have access to the Google maps API right now.

So that's like an example of being unreliable. Some of the other examples in the book are really more idiosyncratic. So Anthropic had a scenario where it wasn't returning the responses in a way that I sort of expected, like it should have had a trailing bracket and didn't, so pretty minor thing, but can cause errors in the code the open AI API basically just had a degradation in what are called the reasoning models .

It wasn't anything that they officially said, but I basically, in the book show an example of go [00:30:00] to my website and give a summary of one of my web pages and the webpage itself, it basically needs to go, I ask it to give me like a number from a graph I have.

If I go now and do it 10 times out of 10, it'll give me a pretty good summary.

But just at this one particular point, I forget which day it was in January, sometime it would. Give errors, give a lot of errors, take a long time. And so I think it was basically a degradation in the model that they just didn't report essentially. And so it does make it, it makes me a little bit worried 'cause I am like using these models for different production systems and there's things that you need to be aware of that, but overall they are pretty reliable.

So

Jason: what about hallucinations? That's a key term that I'm hearing often when it comes to AI and LLMs.

Chris: Yeah. So I gave the example of extracting out information from PDF documents. So [00:31:00] a hallucination would be, say you ask for so say it was handwritten and you wanted to extract out the signature and say like, who signed this document?

And it basically, it can't read. Sometimes it won't be able to read the signature and. It'll return, it'll still return a name, but it'll be like a totally guest name. Mm-hmm. So, say it was basically chicken scratch, but it still returned Andrew Wheeler, for example. That would be like a hallucination. I actually give examples of, in the book of doing the, uh, structured extraction like that.

One of the, one of the scenarios that, that, that does happen is that if you don't let the model return like a null response or none for the categories, it'll basically make up stuff. So, I give an example of categorizing a narrative into type of crime and, but I don't give it the option to say none.

And so in that scenario, it'll say like, a traffic stop is a burglary, basically. [00:32:00] Um, so there's definitely ways to. Prevent that they'll probably never go down to zero. Another scenario with the rag example is sometimes they'll just hall like totally hallucinate different material that's not in the documents to begin with.

One of the most common ways to sort of protect against that is to actually cite the sources. That it's using to make determination. So say you build just a little chat application for individuals to ask questions about, like, what's the correct policy in this scenario? You can have it give a summary, but honestly, just like being able to point to the relevant sections of the policy documentation is probably sufficient for most folks in many situations.

And so it may give a recommendation, but if you just return the citations and be like, Hey, look at page 10, paragraph two, and that'll answer your question. That's another way to [00:33:00] prevent it doesn't necessarily prevent hallucinations, but at least it gives sources for why it made the, it gave the summary that it did.

Jason: Other than what you've already described, are there other technical risks that maybe people overlook?

Chris: Yeah, I think it, I think the only last example that I would give, and so I've been talking a lot about the APIs, but I also have a chapter in the book that talks about the automated coding tools now.

So like using GitHub co-pilot or Quad Code or Google Anti-Gravity is just another tool to help to help software developers write code automatically. And so they're becoming more popular all the time. I if folks are paying attention to, uh, stuff that's going on now, a lot of the popularity is using these tools to help people not just write code, but to interact with their local computer systems.

And so there's a few different [00:34:00] risks that come with that. One is basically it potentially exposes, it can do anything that you can do on your computer. So like, the same way that you could send an email,, with sensitive information to an outside source. If you give the tools the ability to do that, they can potentially do that as well.

, That's what's called an exfiltration attack with these systems. Basically people trick the LMS to send sensitive information to an outside source. So it's not somebody like logging into your computer and doing something bad. It's somebody tricks the LLM to send information outside. So that's, one of the things you need to be wary of if you're using these particular coding tools.

The other one is basically just, they're not infallible, so we're just, we are just talked about hallucinations. And so I think if you're using these tools to help you write computer code, which they're very helpful. The book itself, I used, uh, Claude Code and [00:35:00] Sonnet to help me write the first draft. It likely more than like around 50% of the book is AI generated.

So I just had, I had an outline and then gave it examples of some of my prior writing and then had it write the book. And so I still did a ton of copy editing afterwards, but it definitely saved me a ton of time writing the book. But it, they still definitely generate errors. And so I think it's really important, especially for.

New individual co folks who are like new to writing code, and I've been writing code for a long time and I still, try to review everything that it generates. So , they're good, but they're not so good. I think that you can just let 'em run on their own and do everything , without pretty strong human oversight and supervision.

Jason: Yeah, when you're looking at maybe an individual code, help writing an individual code, it's, it's you run it, that's your test. I'm like, okay, you're looking at the results. Is it [00:36:00] what you expect? Is there other information that you could corroborate? That's, that's pretty straightforward.

Uh, but I mean, if you're talking about something where now you've built several links in the chain, so to speak, I, that could take a while to validate and corroborate.

Chris: Yeah, it's definitely one of the examples that I think is good for crime analysts that pro I, I bet you a lot of.

People have a shared experience with, you can technically write the right code, but then you identify some fundamental error in the record management system or the way that information's been recorded. The correct is, so say you do like thefts month over month and every month is around 20, but all of a sudden you had one month that had 100.

The LLMs may basically interpret that as, oh, you had this crazy spike of 100 in that month. I think that's probably more likely that if, if you ask them, [00:37:00] that's gonna be like the more likely response to it. But most analysts, most savvy analysts would realize like that's probably some type of data entry error.

Like somebody mis recorded that information. Or there was some change in some change in recording that, that resulted in that sort of, that anomalous. Spike. And so I think that that's sort of, like a good exemplar of how the machines won't necessarily, there's still gonna need to be like a lot of human interpretation of what the machines are doing or a lot of oversight to make, steer the machines to do what we want them to do.

Jason: So you mentioned other. Providers, open ai, philanthropic, Google, et cetera. Should analysts be focusing on one or should they avoid getting locked into one provider over the other?

Chris: Yeah. One of the reasons that I actually show multiple providers mm-hmm [00:38:00] in the book is they're very exchangeable.

So to me, like you could nitpick and say, this model is better. Like Anthropic's model is better for writing code than open ai. But honestly, the differences between them are pretty minor and they're all basically getting better over time, better, cheaper, faster. And so a lot of times what will determine whether you can use one or another, maybe if you already have basically an enterprise agreement or use that vendor for other tools.

So I give examples of, with AWS, it's called bedrock. And so with Bedrock you can call different models. And so if your organization, your police department already uses AWS for other things, it's quite possible that you basically already have the enterprise agreements to be able to use AWS Bedrock Services.

And it's the same way for Microsoft [00:39:00] Azure. You can use the Open AI models with Azure or if you're, if you have access to Google, like you can use the Google model. So a lot of times, honestly, it may just be more due to whatever one is easiest for you to be able to get access to at your organization.

To drive basically what you focus on as opposed to worrying about, oh, open AI is the newest open AI model is better than everybody else.

Jason: So I, I guess with LLMs, and we're talking about LLMs and specifically coding, are LLMs. Going to replace coding for analysts or just make analysts faster developers of data.

Chris: I think the more likely scenario, at least for the short term, is help analysts generate code faster. And so it's still definitely the case that I couldn't just give a say create a bot that, and I'm like, [00:40:00] Hey, you're a crime analyst. Go do crime analysis. That's not gonna return anything useful currently.

Mm-hmm. The models definitely aren't that creative or are basically self-driven. You would need to give it much more, prescriptive sort of like tasks and that'll basically be the role of analysts I expect. Going forward in the future to be the task master for the LLM, which still requires a lot of like technical expertise to be able to understand sort of what the needs are of your organization to translate what the chief wants into something that's actually actionable.

I think the need for that is still gonna be there for the foreseeable future.

Jason: Yeah, and I, I think so. I, I think eventually though we are going to see a replacement of analyst functions, if you're an analyst who maybe has a lot of clerical tasks. [00:41:00] I could see eventually LLMs being able to automate most of that stuff.

Chris: There's definitely a scenario where it's a concept called induced demand. The idea behind induced demand is that things are expensive now, and there's not demand for them. They're, there only becomes demand when they become very cheap or, or free.

, It's definitely the case that if all you did was like, if you were like a records management clerk and you physically, when people used to actually hand type. Records. Mm-hmm. And then somebody had to put 'em into the computer that can basically be fundamentally automated. The same way with the document extraction.

I mean, nobody really. Nobody that I know of, like physically hand types in uh, crime reports anymore. There's probably some, but I'm just not familiar with them.

Jason: Yeah. And that's

Chris: anymore,

Jason: yeah. And I agree, I agree with that. 'cause it, my, my, where my head was going is just, uh, I think it's [00:42:00] just the technology is there to your point thing, think way back when analysts were putting,, pins and maps, GIS came along and now no one's doing, putting pins on maps, just like no one's building a link chart by hand.

They're all using software and it's just, I think it's the same progression that you're just not going to have analysts spend of a lot of time building reports or building PowerPoint presentations or doing graphic design. Which I would point out, there's some analysts that really enjoy doing all that stuff, but I think it's gonna get to the point where you could have an LLM create all that stuff in minutes, not hours.

Like it would take a, take an analyst to, to Perfect.

Chris: Yeah. I definitely, for the folks that are concerned about like individuals jobs being replaced I think it's good to realize that there's not like a fixed number of [00:43:00] jobs. Mm-hmm. And so it's not like, oh, you as an analyst are, there's 10 reports that you can make and then your job's done.

Mm-hmm. And so now that an AI can do those 10 reports at a click of a button, you're obsolete. I don't, yeah. I think that that's the wrong way to think about it. And it's basically , there's a potentially infinite demand for analysis and understanding of the data. So it's basically just gonna turn into you being able to automate the boring stuff and then spend more of your time directing the bot the person behind the machine, essentially to just help you do those analysis tasks.

Jason: , For a listener who's feeling really intimidated at toward this conversation besides reading your two books, what are the best ways to get started without getting even more overwhelmed?

Chris: Yeah, [00:44:00] so I really did, I know it's cliché, but I really did write those books for folks to help get.

And an introduction to that material. , I know more analysts likely use Excel than they use sql, but I do think it's a, it's a pretty important skill to be able to use computer coding to automate the different, essentially number crunching type things that analysts do. And even for the folks that focus on intelligence analysis, like being able to go use SQL to go pull out all the relevant records that are related to an address or a phone or a license plate.

I do think that that's an pretty important skill for folks to be able to understand. After that, I would actually suggest both of the books online. Just have I basically have the first several chapters as open material for each of them. My page for the introductory Python [00:45:00] book, I would just suggest going and reading the first I for, I forget how much it's open.

It may be the first three or four chapters. So I would suggest just going and reading that yeah, to get your, your feet wet. And it's the same for the LLM book as well. I basically have like an introductory chapter of a few different examples. And so I think, I think that that's really the, and it's not gentle.

It basically, it, it does take like actual study to be able to learn and understand the material. But I, I did make those resources for those folks. For zero background, essentially.

Jason: All right. All right. I'm gonna, I'm asking you for a prediction type question here. The analyst who understands ai, what does that look like five years from now?

Chris: Yeah, I, I mean, it's really hard

Jason: mm-hmm.

Chris: To know and like, life has been changing so fast. So I, I like to think about like when did we even first have [00:46:00] smartphones? So I feel like it, we didn't even have smartphones all that long ago. Maybe like after 2010 iPhones came or were at least popularized.

I didn't have a smartphone until almost like, uh, 2016 or 2017. I remember I I basically like avoided having one.

And so smartphones have really like, changed our daily lives by a lot. So we have this like mini computer that's way faster, gives us access to more information than anything. Like we had 20 years ago.

And so I think these AI tools coming up now, they're definitely overhyped in some ways, but I do think that they're gonna be transformative in terms of like, they're gonna basically be baked into pretty much every application that folks are using. So it's either gonna be. Systems like the Claw desktop that basically interacts with your local system, or it's going to be the applications themselves.

Have a button that's like, Hey, use the [00:47:00] LLM to do a bunch of tasks, like a button in PowerPoint that's like, Hey, use open AI to like write my PowerPoint slides for me.

Jason: Yeah.

Chris: And so what that means for work, I'm not quite as sure. I think it's definitely gonna be important for. Individuals to become savvy with these tools.

But honestly, I think that they're probably gonna be incorporated enough that it's not gonna be, like most people I know now can use text messaging and you have auto complete in your text messaging for your phone. I think most folks are gonna be able to just, they're gonna the interfaces are gonna be integrated and smooth enough that it's not gonna like leave folks behind.

They'll be able to figure it out the same way as most people can figure out how to use email.

Jason: Yeah. Hmm. What, uh, just, I guess looking for maybe a short word answer here. What, what is the most overhyped claim about AI LLMs right now?

Chris: Oh, it's the a hundred percent human replacement. I [00:48:00] just don't, I just don't foresee that happening.

At least, in the near future. There's a concept in, in transportation logistics called the Last Mile. So it's easy to build like big transportation networks. Well, easy is probably not the right term, but basically to replace USPS, that last mile is a lot more difficult, takes a lot more effort and I think a lot of things that we do, you may be able to automate a lot of different things, but to fully replace the human for that last mile, that last little bit is very difficult.

And so I don't foresee. A hundred percent fully automated. Like I gave the example of just tell a bot you're a crime analyst, go do crime analysis stuff. I don't foresee that being viable without a human literally telling the bot what to do. For quite some time now, it could be that a company comes out and it's only five [00:49:00] analysts, basically do analysis for a hundred different cities because it's very automated and can be like the analyst literally just says, Hey, go do this analysis for tiny city A and tiny city B and tiny city C.

That may happen, but the no humans at all. I don't foresee that.

Jason: Yeah. What about the, kinda on the other side of the spectrum, what's the most underrated capability of oms

Chris: Underrated capability? I think I would probably talk about or give the example of, so we talked about hallucinations.

And so the machines now, they're definitely not infallible.

They're really good though. And so one of the things is that humans make errors too. So if you had a human go and give a summary of a particular police report, they're gonna make errors at a certain rate and maybe like one in 10 and one in a hundred.

It's quite, I think it's quite likely with the current models, that the models themselves, even if they do generate hallucination sometimes [00:50:00] are

probably better than humans at certain tasks already. And so summarization extracting out pieces of information, those are two things that the models are incredibly good at now.

And that doesn't mean they're perfect, but they're probably already exceed like your typical human by a pretty decent margin.

Jason: Very good.

. So I'm just, uh, again, just looking at the future, what are some AI developments that you think analysts should be keeping their eye on maybe the next two or three years?

Chris: , It is not necessarily something that I talk about much in the book.

It's kind of related when I talk about the coding tools like Claude Code mm-hmm. You can basically have Claude Code not only write computer code. You can have it interact with your local system. Those are even if you're not a Python developer, that's still relevant. And so those tools right now basically are no GOs for law enforcement sensitive information.

I expect folks are gonna be able to [00:51:00] overcome that though in the near future. And so having, uh, the claw desktop tool or something else equivalent that's okay for law enforcement sensitive information, that's gonna be something that's probably I bet is gonna happen and be a regular component of, of individuals, at least for text.

Tech forward organizations in the next couple years.

Jason: Yeah. And as for the future, what's next for you? Are you working on something? Are you, there's something upcoming that's exciting you?

Chris: The main thing that I'm doing now is just trying to, uh, promote this book. And so I really do think it's at sort of a critical time right now for developers to learn these skills.

And so I'm really just looking for opportunities to basically do training for this. So folks, if, if folks are interested, I do basic Python training as well, but if folks are interested in more advanced training that's related to these, always feel free [00:52:00] to reach out. But it, that's one of my main focuses now, besides the consulting work that I do with Crime Decoder.

Already is basically just spreading the word, and getting folks trained up how to use these in their own applications.

Jason: Alright, well, very good. All right, , how can people contact you if they have further questions or in how can they get the books?

Chris: Yeah, so the book is available on my website.

My website and I'm sure Jason will link to it in these resources is Crime de dash. coder.com. And so if you go to my website, I have a store to be able to purchase the book, either an epub or paperback versions worldwide so you can buy it and get it wherever. For folks listening, I did create a promo code.

You can use leap LLM, so L-E-A-P-L-L-M to get \$20 off of the paperback. I know from the prior Python book, a lot of folks like the paperback [00:53:00] version. And I'll send you the epub as well if you use that code. Excellent. Um, and then to just contact me, I have a contact page right on the site, but you could also just send an email to Andrew Wheeler.

At crime de coder.com.

Jason: Yeah. And we'll put the links to all that information and this contact information in the show notes. Andrew, this has been great. Thank you again for your time and perspective and , looking forward to what's next because I really feel we're on the early side of the LLM wave.

I think it, for a lot of analysts, this is a really new, maybe even abstract concept, but I do feel five years down the road, we're gonna look back and said, yeah, this is when we did this. It was really early and now it's LLMs are everywhere.

Chris: I agree. Um, and so that's one of my main motivations to write this book.

It's [00:54:00] definitely, there's no doubt, a lot of hype about what they can do over hype, I should say. But they're I really do believe that they're gonna be integrated in almost, uh, everything. The same as how everybody has a smartphone now. I think LLMs are just gonna be a regular part of using computers, , in the near future.

Jason: Very good. , Thank you again. You be safe and take care.

Chris: Thank you, Jason.

Mindy: Thank you for making it to the end of another episode of Analyst Talk with Jason Elder. You can show your support by sharing this in other episodes found on our website@www.leapodcasts.com. If you have a topic you would like us to cover or have a suggestion for our next guest, please send us an email at elliottpodcasts@gmail.com.

Till next time, analysts, keep talking.