SuperDataScience

SDS PODCAST EPISODE 999: WHAT'S LEFT TO BUILD WHEN SOFTWARE IS FREE (WITH CHIP HUYEN)

Jon Krohn:	00:00:00	The cost of building software is heading to zero. So what happens to everyone whose career was built on writing it? Welcome to episode number 999 of the SuperDataScience Podcast. I'm your host, Jon Krohn, my returning guest today is the sensational AI engineer, entrepreneur, and two-time mega bestselling author, Chip Huyen. Indeed, her most recent book, AI Engineering, was the most popular book within the O'Reilly platform last year. In this episode, we dig into the interesting and practical content from that book as well as how coders can future-proof their careers, her newfound fascination with physical AI systems like robots and much more. We're so lucky to have Chip on the show. I hope you enjoy this enthralling episode. This episode of Super Data Science is made possible by Anthropic, Acceldata and Cisco. Chip Huyen, welcome to the SuperDataScience Podcast. I can't believe I'm with you here in the flesh.
	00:00:54	How's it going today?
Chip Huyen:	00:00:55	It's crazy. I feel it's good to see you in person, like very three-dimensional.
Jon Krohn:	00:01:00	I am three-dimensional. It's true. I can't believe it either. So we're live together in San Francisco, which is fun. Your hometown, you've done so much here. Last time you were on the podcast, which was episode 661, if people want to listen to that invaluable gems from you in that episode as usual. It was
Chip Huyen:	00:01:17	Pre GPT4. So I mean,
Jon Krohn:	00:01:19	The world has changed,
Chip Huyen:	00:01:20	Right?

Jon Krohn: 00:01:21 Yeah. A couple weeks later, and the reason why I know that is because I was mentioning to you how you were in 661 and you said, who was 666? Yeah. And that was GPT4. I remember off by heart. So a few weeks later, GPT4 came out and yeah, we're in a very different world. Lots of things have happened to you since then. At that time, you were the founder of a startup called Claypot. It was acquired by Voltron Data, I think.

00:01:44 Yeah, exactly. Wes McKinney's company, so a really cool company to be acquired by. And the foundation model wave hit. And then you spent a couple of years writing a new book. We're going to talk about that new book, which was the most read item in the O'Reilly platform last year. So of all their videos, of all their courses, of all their books, your book, your new book was the most popular item, which is really exciting. But before we get to that new book, when you were last on the show, you had recently published your first English language book, which was Designing Machine Learning Systems. It's subsequently been translated into 10 other languages as well, but you had an interesting point that you made to me before we started recording, which is that in the time since that book came out, it's actually become even more relevant, which is unusual in tech.

00:02:36 Tell us about that.

Chip Huyen: 00:02:38 Are you trying to get me to promote my book?

Jon Krohn: 00:02:40 No, no, because we were just talking about this outside, how designing machine learning systems is one of the things now when you wrote that book people in our profession would still have been writing their own code and now we don't really do that anymore, especially in the last few months. And so we were talking about how designing machine learning systems is still a really valuable skill to have if you're an AI engineer or a data scientist or a software developer.

Chip Huyen: 00:03:07 Yeah. I think system thinking or system design has always been very, very cool to me. I think with AI, automating a lot of skills has been become even more important to think about things in system because I do believe that the more narrow the skills are the easier they are to be automated. So I started thinking about system design back in 2018, 2019. I think the pain came from college. So I was teaching this course Contensive Flow and the first iteration went pretty okay and then they asked me to do that again and I realized in the meantime, TensorFlow has updated, right? And TensorFlow 2.0 came out and that means that if I wanted to do the course again, I would have to redo 80% of the tutorials. I was like, okay, I do not want to do something like that. How do I focus on things that do not change very often, which forced me to slow into what changed fast and what don't change?

00:04:06 And just asking a lot of people's much smaller than me and what are the heroistic for finding out things that truly matter over time. And then I realized one thing that I landed on was system thinking, system design. And at that time a lot of companies also studying having the system design thinking, like system design interviews, but they were not very popular. But I feel like very progressive tech companies had them. So I just tried to collect information and they wrote this notebook of machining system design and I put it on GitHub and it got quite popular and back then if you put the search query like machining system design on Google, they operate two results, like one from Neil Lawrence, which is an amazing professor

00:04:54 In the UK and another is just GitHub. And then one of the reasons that Raybo got popular was that in the end I have a bunch of questions, a bunch of systems to think about like, okay, if we want to build a system that do this, how do we go about it? And I think people felt it useful to practice for interview questions. And so after that, so it

was 2019 and then I taught a course at Stanford on machining system design and then just came about. The lecture notes became the book design machining system. I couldn't get the title machine system design because someone else apparently like 2022. But yeah, so I think that over time people do get used to the ideas that a technical book doesn't have to be such PO code. I remember when my book came out, a lot of people were like, "Okay, this is not a technical book because there were no code snippets." So I think we have more and more over time get used the ideas that like, okay, an engineering job is not only about writing code, like thinking about problems and how to find the solutions

Jon Krohn: 00:06:04 That

Chip Huyen: 00:06:04 Make sense over time.

Jon Krohn: 00:06:06 Yeah. Invaluable book. Really appreciate it, Chip. Let's talk about the new bestseller. So your new book is called AI Engineering and seriously, it's crazy to have a book. There's so many books, hundreds, maybe thousands of books that get released in the O'Reilly platform every year and to have the most popular book of 2025. That's wild. And in that book, you are emphatic that AI engineering is distinct from machine learning engineering. So we probably have a lot of listeners. We have data scientists, we have machine learning engineers, we have AI engineers who listen and some people probably conflate those things two together. You think about ML engineering and AI engineering is the same thing, but they're different,

Chip Huyen: 00:06:48 Right? I think they're different enough for me to have two books on two different topics, but I do think that for a lot of people jobs it actually can involve both. So it's not about like, okay, I want you to become A engineer or I want you to call machining engineers more like, okay, what problems you are trying to solve and what is the

best way of solving the problem. And a lot of time solutions when involved like both machine learning, like classical machine learning and then like generative AI model. So one example, so like in the past, I think machine engineer was more about building a model from scratch. If you want to build a regular system for like Amazon or like Spotify or like foreign detection model, you have to collect your own data and then train the model and then you deploy it as part of a system.

00:07:38 Whereas with generative AI, you can just start with a demo. You can just like have access to the most amazing AI model, host somebody's cloud and then you can just use it as a feature. So like the time to market is extremely, extremely fast. So it's a model as a service and to do that, right? I think that there's a whole new technique. So now we have like instructions, I have to write good instructions. I still think that writing prompted an extremely important skill, like how to provide model with the right context. I do things like context and memory system actually very, very related. They're both involved with like ... Yeah, I think I spend a lot of time thinking about memory system design, like how to get the model to like both retain information and then access that information efficiently over time. Yeah. So I think there's a lot of techniques rather than guide rails and then like also like how to combine it into a system.

00:08:30 So let's say that you have like customer support chatbot, right? And a lot of time a request might not even be relevant to your system, right? If you probably don't want your chatbot can join an argument over who's going to win the elections. So you probably have like some kind of filter, like some kind of classifier to detect whether this request is relevant to your product.

00:08:55 Then you might also think about like a cost saving, maybe like not only the requests need to be sent to the expensive model. So you might, okay, maybe this

question is about like how to reset my passwords. There's an FAQ somewhere you can just send the users a link instead of having to generate an expensive response, just step by step instructions on how to change the password. So you can have classifiers say, "Okay, this request should be routed here. Another request should be route here." So that classifier is like classical machine learning and you might want to build that in- house. So you can have systems that combine many, many different components and as a person who is trying to solve that problem, you might want to understand like what solutions are possible and what is the best choice for you.

Jon Krohn: 00:09:43 Yeah. So you're actually bridging your old book with your new book there where this idea of designing the systems appropriately is critical to having for whatever your application is, like that customer service example you're giving there, you're blending together those more classical machine learning concepts of the classifier to say identify whether this is a relevant conversation or not or to send something from the FAQ. And so that kind of classifier model is more like the machine learning engineering, the ML engineering where you might need to label or today we usually don't need to manually label because we can use a large language model to do it, which is a super nice thing to be able to do these days and we wouldn't have been able to do that well back a couple years ago when you were on the show. It's basically since GPT4. That was the release that I started using GPT4 to label my data for me and be much more efficient at the machine learning engineering.

00:10:37 But then on top of that, we blend in the AI engineering, which I don't know if you agree with this definition or not, but it seems to me in my head, obviously there is some AI engineering out there that is like AI research where people at Frontier Labs or in places are creating foundation models and training them from scratch. But for the most part, for most of our listeners and probably a

lot of what your book is talking about is more about calling existing models, maybe fine tuning them, but you're leveraging LLM APIs for the most part in AI engineering, right?

Chip Huyen: 00:11:11 Yeah. So for a lot of people using AI and ZIP products nowadays, AI is mostly a service. And I do think this is the questions I sort of brought up like fire tuning. I think that's another debate on like when you should start firetuning your model. So I'm on more of the skeptical side with fight tuning because I do think that like firetuning can actually solve a lot of problems and definitely in a lot of cases like fire tuning is necessary, but it's usually like the last line of defense, right? I think there are many, many different techniques that we can try out before like moving to fire tuning because finetuning itself is not hard. It's actually like we have a lot of like off the shelf frameworks that make fire tuning pretty straightforward. However, the challenge is like we have a model and then not what, right?

00:12:02 We have a firetune model, we have to deploy it, we have to serve it. And like inference optimization is not easy. It's not straightforward.

Jon Krohn: 00:12:10 You have a whole chapter in your book on that actually.

Chip Huyen: 00:12:12 That is actually one of my favorite chapters. It's a lot of fun thinking about like how to make system faster and cheaper, my affection, but it's definitely not for everyone. And also like models also evolve over time. I talk to a lot of companies and they told me that like their development cadence is about like two to three months and that match issues a cadence of like when there's a new model out that have a significant step change in functionality. So maybe like every two to three months there's a new model that blows everything out of the water. And a lot of like techniques that you use to address the weaknesses of like older generation models are no longer relevant.

Jon Krohn:	00:12:57	Totally.
Chip Huyen:	00:12:58	Yeah. So like if you do 52 model, right, you need to understand whether your fine tuned models can keep up with like the base models that be increasingly more powerful.
Jon Krohn:	00:13:07	Yeah, 100%. I mean, that was definitely something we used to really pride ourselves in a startup that I had co-founded, I'm no longer at, but in that startup we were like, oh, we actually need to train and deploy our own models. We were downloading LAMA model weights and fine tuning them. It was a small LAMA model, one that you could fit on a single GPU, like a single H100 and we needed to do that quote unquote because for the kind of performance that we wanted, we couldn't get the results from a single GPT4 call. It would have required a whole bunch of calls to GPT4 to get the results that we wanted, but we could use GPT4 to create a great training data set and then we could fine tune this LAMA model to be able to do a task. But of course then six months later with like GPT 4.1 or something, you know, all of their numbering was weird in the fours and I can't remember how it went with 40 something, a new model release would have come out and then it does everything.
	00:14:06	We spend all these months dealing with all the issues that you're describing there. You
Chip Huyen:	00:14:12	Look very excited talking about that time in July.
Jon Krohn:	00:14:15	Well, yeah, was that the last time that I was fine tuning AI models? Maybe. But yeah, you actually, you talk in your book about a start simple approach where you focus on prompting first and then techniques like retrieval augmented generation rag and then fine tuning if you need to. So how can you convince our listeners or how do you convince people that you talk to that this start simple

approach prompting before rag, rag before fine tuning is the way to go?

Chip Huyen: 00:14:44 So it's not necessarily like this has to be before that. It's more of like look into what you want to build and see where is that failing and come up with a solution, like the simplest solutions that address the failures. So I am a big fan of Star Simple was mostly I'm a lazy person. I don't want to do things that too complicated if I don't have to. And also like when you start too complex in the beginning, it just make it harder to understand the system and debug because like if things fail, we don't know, okay, is this like this component is failing or another component is failing. So I just want you to understand just like how everything works together first and slowly build on top of things that I already have a good understanding of. So the first thing is like prompting.

00:15:29 I do think that prompting can actually take you a very, very long, take it pretty, pretty far. So I think I build a lot of applications now and prompting actually helps tremendously and I think like the reason I can see from like half asked prompt versus like a prompt I really spend a lot of time thinking about is huge. And one of the things I do notice about like prompt writing is that over time prompts tend to get more complex, like very, very long. Even the prompts that I write, they can go into like thousands and thousand of tokens, right? And then I was just like one day I was like, okay, throw the whole prompt into my AI and say, okay, analyze my prompt. And then it pointed out to you a bunch of things. It's like, okay, in this part of the prompt, you said that I should do this, but in that part of the prom you said it's like, no, no, you shouldn't do that.

00:16:24 So I think because over time I just keep on addressing using new examples because my concert with older example and it's a prompt that I wrote myself and I

realized when I talked to different, like a bunch of other companies, when multiple people get involved in writing the same prompt, that happens a lot. The prompt gets like extremely long and sometime I ask an engineer like, "Okay, has anyone on your team just span, sit down and read the prompt end to end?" And it's almost like never. So I feel like a lot of times the performance is not quite ... Yeah, I feel like people can squeeze out more performance just from writing better prompt. And of course it's a questions of like people talking about like giving the model like knowledge. So one huge failure mode is that like the model does not have the information to answer the questions, right?

00:17:15 100%. And if it doesn't have the information, it's going to mix something up. I think like nowadays models are like getting much better at like saying that it doesn't know when it doesn't know something, but like when you don't give the right information, it's more likely that it's going to hallucinate.

Jon Krohn: 00:17:31 So

Chip Huyen: 00:17:31 Of course you have to provide with the right context and one kind of like context information you can provide is like giving it a document. So first of all, if you want to like analyze super data science podcast, I think I might be able to provide a bunch of transcripts from the preview podcast. So I think the models can reference to the transcript and it can be very helpful. And as a guide context, it's like provided for like tools, like web search. So the models can do like web search and find information. So that's extremely important for a lot of case, but web search, I'm not sure that you have talked to a lot of people who do using AI with web search, but web

Jon Krohn: 00:18:16 Search- It hasn't been a dedicated topic.

Chip Huyen: 00:18:19 It's extremely expensive.

Jon Krohn:	00:18:21	It's like
Chip Huyen:	00:18:21	Painfully expensive.
Jon Krohn:	00:18:23	It's make
Chip Huyen:	00:18:23	Me scared to run my model because like-
Jon Krohn:	00:18:25	In terms of token consumption.
Chip Huyen:	00:18:26	Yes. Yes. I think there's still a lot of room for like improvement in how to make web search more affections.
Jon Krohn:	00:18:34	It also, it depends so much on exactly how the provider you're using implements that web search, like how the agent does it. So for example, I don't actually use OpenAI deep research anymore, but a year ago I was using it all the time and deep research from OpenAI, even though I was paying, you had to pay at that time, I don't know what it is now, but at that time I had to pay for the $200 a month tier just to get access to deep research and then it would only check a few references. It would only do a few web searches whereas in contrast a few months later, so now about a year ago when Claude started allowing search, it would like spin up like a hundred agents instead of like six to look up like a hundred articles and I was like, wow.
	00:19:15	And that's obviously using- Do you look at the
Chip Huyen:	00:19:17	Cost, like the API cost?
Jon Krohn:	00:19:19	I mean it's still, for me it's very manageable. Yeah, it can end up being for a single search, you could maybe spend like tens of cents in some cases.
Chip Huyen:	00:19:31	Yeah. So doing some, of course I feel like a lot of people look into operation market and I try to see if AI can like do pretty market, right? Oh, prediction market. Yeah. Yeah. So I think I had that face and was just curious. So I

asked a bunch of like AI agents to like spin up like, "Okay, given this market do a bunch of research about it and make predictions on the outcome of this market." And I was just like, "Holy moly, each of the requests cost me like a dollar, sometime clock was like $2, that is insane." So I look into like the search results and I feel like, and I think I saw some for some of the requests, it visit thousands URLs, a thousand webpage.

00:20:16 And so it was like, or what are all these webpages? So I started just like analyzing them and I found out of this a thousand webpages, only 20 of them are unique. So like on this like agent, I keep revisiting a webpage over and over again. Interesting. I was like, why is that? So what happened is that like when you give it a request, say, "Hey, do research about it. " It comes up with different search queries. So first of all, like tell me about super data science podcast in my super data science podcast, super data science postcard guest. So they had different queries and this different queries might return the same else. So they might revisit over and over again.

Jon Krohn: 00:21:01 It sounds like an opportunity for caching.

Chip Huyen: 00:21:05 You could have thought, but I thought looking to it is because like a lot of it is also like how data on the internet is structured. So like Google search is structured based on like data chunk. So like if you put in a query, you wouldn't find the website, but you still face a part of the website that is most related to the query. So that means that when you do search, you don't retrieve the whole webpage, you retrieve the part of the webpage that's relevant to you. So I think it's a very involved process and there's a whole part about like, okay, we have the informations, how you decide the freshness informations, right? Maybe like for something that's very much news related, you do not want something that's from like two years ago, but if something's just like technical, okay, what is an embedding, right?

	00:21:57	So maybe the result from like five years ago is fine. So you look at like a system prompt of like a lot of these AI services like ChatGPT or Cloud, you will see that they have like really, really big sections, just trying to get the model to like do web search and like, okay, like the freshness of data. Okay, if the Skype queries like this, then maybe you can use the data from like a week ago. If not, you can ... Yeah, it's still very much manual, like the
Jon Krohn:	00:22:25	Way
Chip Huyen:	00:22:26	I see like how things are being defined. But anyway, by the way, when I mentioned the system prompts, like the last time I look at the system prompt, this model was about like four months ago so things may have changed. So
Jon Krohn:	00:22:39	I'm
Chip Huyen:	00:22:39	Not sure how they are doing web search nowadays
Jon Krohn:	00:22:43	Yeah, I'm sure it's a fast evolving space, especially if it's consuming tons of tokens all of these Frontier Labs are trying to keep their compute down as much as they can to get the same quality of result at a lower cost. So there's got to be optimizations going on all the time.
Chip Huyen:	00:22:58	I think so. It's looking at how people are like spending money on their agent. So I think there's a part like the actual reasoning actually a lot cheaper, but a lot of them are more like two news tokens, right? Like web search, file search, like one big- File search. ... one big category of tool news. The other category tune news is like, I see like productivity. Tune news, like when you connect it to like Gmail, Slack, Asana and they're going to have to do like tune calls and then analyze the resort like tune calls. And then of course there's also always like valuations. Yeah. Some people were telling me that they were paying like 20% of their token costs.

Jon Krohn:	00:23:42	Some neighbors?
Chip Huyen:	00:23:43	No, some company. Oh,
Jon Krohn:	00:23:44	Some companies. Some neighbors,
Chip Huyen:	00:23:45	My neighbors not into AI.
Jon Krohn:	00:23:46	Oh really? In San Francisco, you have neighbors that aren't into AI. That's amazing.
Chip Huyen:	00:23:51	Yeah.
Jon Krohn:	00:23:52	Fantastic. I'm going to fast forward a little bit in your book because earlier in the conversation we were talking about how interested you are in inference. So I want to give you some time on that and explain to our listeners why that's so exciting for you. It gets its own chapter in the book and for our folks who are watching their token bills balloon, what are the highest leverage moves that they can do with their own LLMs in production in order to be more efficient?
Chip Huyen:	00:24:22	Honestly, I feel like inference, I'm interested in it because I'm a nerd, but I think for a lot of people, they actually ... I think if you don't control the model, then you can't really do much cheat on that.
Jon Krohn:	00:24:35	But yeah, if it's your own, I guess if it's your own, then you're not so worried about token costs themselves. I guess you're maybe just worried about how often you have the model up or how many GPUs you need to have running. Maybe we're starting to get into something that's really for people that have a large number of models on it, a large number of GPUs on at any time doing inference.
Chip Huyen:	00:24:58	I think that one thing people could do for token costs if they're worried about is that I try to look into like what owns the usage and what kind of usage that can be offloaded to smaller models, but not everything needs to

be sold by like very big, very massive, expensive models. So it's something that is small, like first of all, like going back to the classic example of customer support, like if you can see a certain category of queries that can be answered just by mapping to FAQ, then maybe you don't need to send them to big models.

Jon Krohn: 00:25:35 Yeah. So bringing that back again to the idea of great systems design for your AI system and bridging that with your book, which actually we're going to move on to ... I just have one last question for you related to your books and then we're going to get onto what you're excited about right now. But your final chapter in your AI engineering book is about closing the loop with user feedback and it seems like a good user feedback design is critical to having a great AI system in production. Do you have any thoughts for us on that?

Chip Huyen: 00:26:11 So I think user feedback serve many purposes, right? One thing is that like you, if you serve people, if you charge people money for any service, you kind of want to know like how happy they are about the service, right? So of course you need feedback. Feedback is so very good to uncover the area for improvement, right? So I think if you look at feedback you need to under ... So it helps maybe like for certain demographics, somehow the users are not extremely happy and you want to understand why. So we try to look into the data as detailed as possible, like slides and dice by users, by use case, by like locations to understand different patterns in there. And so like another feedback is also good for like tooling model. So a lot of evaluations to ... So user feedback is also like one type of feedback, right?

00:27:13 But like the automate goal is so that we have a way to automatically evaluate our systems in production.

Jon Krohn: 00:27:21 Agent feedback. How do our agents feel about our AI system?

Chip Huyen:	00:27:25	You need to ask the agents.
Jon Krohn:	00:27:27	Exactly.
Chip Huyen:	00:27:28	Yeah. So I think it's like, okay, so like let's say that you build this very complex and very amazing systems and your engineers sway on their life that like, okay, this is amazing. But you're like, yeah, I trust you, but maybe you should just like put something to see whether it's actually working. So you kind of want to build an evaluation system to see if the responses make sense. And a lot of people using like LM as a just use like another AI model to like look at the response and like evaluate. And one thing that is actually really, really hard that I see that a lot of people spend so much time on is like how to write the guideline to have the AI models evaluate, like output score that makes sense, right? So I think like we discovered that pretty early on, as there were a lot of case studies when somebody was like, okay, we spend 80% of our development time just to write guideline for our AIM as a judge.
	00:28:26	It's painful.
Jon Krohn:	00:28:27	Yeah. You have two chapters in your book on evaluation, in your AI engineering book on evaluation and in it you describe things as getting slippery as soon as the outputs that we're evaluating are generative because it's very difficult to come up with a really rigorous test. It used to be the case that you could have tests in software development where you were like, "If it isn't character by character this output, there's a problem." And now obviously with gen AI, you can't do that anymore because you'd have such a wide range of responses. So that's kind of what we're touching on here, right? But you need to spend a lot of time figuring out what your evals are to ensure that you're getting the kind of result that you want.

Chip Huyen:	00:29:09	Yeah. I think like bringing back to what you just mentioned, in the past I saw we have like test driven development, which is like a very interesting product, extremely useful for a lot of use cases. For AI, I think the simple calling is like evaluations driven development. There are only development applications that you can't measure the output for, right? Because I think like there's some questions sometime when I give talk as a company, I ask people like, "Okay, which one is worse?" Having a system in production, like not having an AI system in production or having a system that nobody knows whether it's working or not. So I think it's happened quite a common people was like, "Okay, we deploy this AI system." And some people was like, "Okay, we save so much money, we make so much money." And I was like, "Okay, how do you know?
	00:29:58	" And so We guess. So it's tricky.
	00:30:05	So going back to the guideline for the AI judges, it's painful. So for example, how do you even explain to it, okay, this is a good response, this is a bad response and why is that a good response? A lot of people, we look at something, we can tell that it's bad or it's good, but we don't really want to sit down and write down a reasoning. And one of the most tests I have with a lot of people is just like, okay, after you write out a guideline as a model, use yourself, go and follow that guideline and go through some examples and see whether you are good. Or I give it to the coworker and beg the coworkers, just follow it.
Jon Krohn:	00:30:47	Interactions. So you're saying that in today's day and age, humans can still have taste that is more useful than an LLM. That is something. Wow. So
Chip Huyen:	00:30:57	I think it's important to not just assume that AI can do everything. AI can follow instructions. But if the instructions are trash, then it's not going to be good.

Jon Krohn: 00:31:10 And I think something you talked about earlier in the episode is one of the absolutely most important things to having an LLM being able to do what you want or an agent to be able to do what you want as an individual or in an organization and an enterprise, whatever, it's having the right context. And so you might think you could have your AI system that's doing evals, you can't just trust the outputs because what if you think that you wrote a great prompt, you think that you have the evals set up properly, but if you don't go through and read them individually, there could be some key piece of missing context, maybe some product requirement that came from your users or the client and only you as the stakeholder, this human who's been in all those meetings really knows what's important and can give the final sign off.

Chip Huyen: 00:31:59 Yeah. I think that it's very important to have a good understanding of what your instructions are doing. So a lot of times you don't need to read it line by line, but just be aware that there's something you need to pay attention to. For example, I threw it into another model and just analyze for me finding contradictions in my own prompt and it does that pretty well. And also have explained a lot of weird behavior that I felt. So for example, so another applications when I use AI to do research and generate a summary of the research and then on the summary somehow always focus on very specific things, very weird. So it was at first like, okay, is the AI model like bias? Why are you so insistent on this New York office of this company? All of them. And then I turned out in some weird part as a prompt, I give an example of like, okay, if somebody asks about company, find its offices, maybe check whether it's in big city and in New York and stuff and it totally over index on that.

00:33:06 So it was like, okay, so I removed that example and it works much better. So you just be aware of like what you're asking the model to do. And I think it's like one

thing that also useful for having good guideline is that you can also use it to synthesize data later on if you ever want to like need more fighting more and more

Jon Krohn: 00:33:28 Data. For sure. For sure. Yeah. I mean, that's what we were talking about. Oh yeah. I mean, you can be using ... We didn't actually talk about synthetic data specifically. So we were talking about, I was talking about earlier in your episode, I'm taking up way too much of your time, but I was talking about how when GPT4 came out, that was the first time that I felt confident enough about an LLM that I could be using it to label data. But you could also use GPT4 and any of those kinds of frontier models since to actually be synthesizing not just the labels but the training data as well. And you've got to be very careful in doing that because you can end up, you might not get the breadth of the sample space that you want. So you have to be really careful how you see that training, but it can be very effective.

Chip Huyen: 00:34:14 Yeah. I personally don't fight tune models anymore. I think I did try early on, but then at some point I was like, okay, this model just keep getting a lot better. So I spend a lot more time on tool news and web search and stuff to make my things better. So I actually don't use a lot of synergic data, but I do use AI to label a lot of stuff because a lot of my applications require like AI to like label different stuff, categorizing things, taxonomy and then evaluations obviously. Yeah.

Jon Krohn: 00:34:45 All right. I think it's time to go on to ... We've been talking about what you're excited about now. It's not fine tuning. So let's move on to what you're really excited about. It's actually, it's going beyond software. It's going into hardware, into physical systems. And so at CES this year, Jensen declared the ChatGPT moment for physical AI is here.

Chip Huyen: 00:35:11 Is it?

Jon Krohn: 00:35:12 That's what he says. And I'm sorry, I'm here to ask you about that. Is that just hype or do you think that this is a really exciting time for you maybe for our listeners to think about getting into physical systems?

Chip Huyen: 00:35:27 I'm not sure I have the street crash to argue with Jensen at this point.

Jon Krohn: 00:35:33 I think you're the next ... It's the two of you. You are the two people that the world looks up to the most on this kind of stuff.

Chip Huyen: 00:35:39 No, no, that's crazy. But yeah, so something is like, I'm not sure that you have gone through that, but I feel like a lot of my friends and I, we've gone through this like we call existential crisis, right? Oh my goodness. And I was like, holy shit, what do we do now? AI is coming for our job. So we talk a lot about not just like what to build, also like how to build things, like what you build, because I do think it's like AI is getting incredibly good a lot of things and building things is actually not, it's a process of building things is not that hard anymore. So let's say you have a great idea, right? You write a spec for it, you can ask like AI to like reviews or specs, like improves the spec and then input the spec into AI and it can create like some amazing website applications.

00:36:26 So it's like super cool. But then you build something and then what? So I had the experience of like ... So I had a side project about like two, three months ago and I put it online and it's not like my blowing anything. It's like something chill, something like ... It was useful for me. I think one thing I really love about AI nowadays is just like allows me to build very good micro tool that makes my life so much easier. Sure. It's very easy to build. Can you

Jon Krohn: 00:36:52 Share some of your favorite micro tools that you built

Chip Huyen: 00:36:55 With us? So the things I did was like good AI list. So like every day you just like call GitHub and it's fight based on like a bunch of keywords I give it and it fight on the repos related to this keyword and then it's analyzes repo and tell me what is interesting about it and then it's like categorized and like also rank them. So help me research things that might be relevant

00:37:17 To me. So I post it, I put it out there and it got like, I think what, I don't know, 300,000 views in a week and like the next day someone emailed me. I was like, oh, I love what you did. So I use AI to replicate exactly that and here's that. And it was just like, I'm not sure how I feel about it where I'm flattered, but it was just like, is that like copy? Yeah. I don't know, what is that? So that's made me so realized it's like anything that exists today software can be copy

Jon Krohn: 00:37:54 Replicated. When there was the leak of the Claude code repo about a month ago at the time of us recording, somebody obviously you can't, like that's like copyrighted, you can't publish exactly the same code, but somebody recreated it in Rust or something use Claude code to recreate the Claude code code base in Rust and then publish that in GitHub. And so yeah, it's pretty wild what you can replicate so easily today.

Chip Huyen: 00:38:23 Yeah. So I do think that it's an interesting feeling because I feel like on the one hand AI mix is like, it's very exciting because it allows me to build anything I want and not everything. Of course there are some things that's harder to like build another, right? I'm not going to be with Google search in like a weekend. Exactly. But I also getting like, if you look at the complexity, there's a level of complexity of tasks I can do, like that level of complexity is like going way, way up. It's

Jon Krohn:	00:38:53	Insane. With the Methos release, you know the people at Meter, M-E-T-R? Yeah. And so they have, I'll put a link- It's a very cool
Chip Huyen:	00:39:01	Organization.
Jon Krohn:	00:39:01	Very cool organization. I should have someone from them on the show. If you know anyone, let's talk. And so I always, every talk that I do these days, I always have the latest meter chart because they take the latest and greatest frontier model and then they benchmark it against how well it can replace a human on a software development or a machine learning task. And Methos broke their evaluations because they didn't have reliable tests that take humans longer than 16 hours. Imagine how long it would take. It was very easy when Meters started going and they were benchmarking, okay, let's find some task that it takes a human a few seconds, or let's find a task that takes a human a few minutes or a few hours, but now that they have to be coming up with tests that take 16 hours, 24 hours, it's going to be no time before it's 50 hours, 100 hours.
	00:39:52	How do you even find and pay humans to do that as a benchmark?That
Chip Huyen:	00:39:55	Is scary.
Jon Krohn:	00:39:57	And exciting.
	00:39:59	Yeah. So Methos broke it because it was the previous front top performing model was Opus 4.6 in the meter evals and that was averaging, like it was able to do on average 50% accuracy on a task that would take a human about eight hours. And then Methos, they're like, okay, they put the dot at 16 hours, but they also put this disclaimer that like, but we don't actually have good evals past 16 hours. So it might be even much more. Methos

might be able to handle on average a 30 hour task we just don't even know. It's crazy.

Chip Huyen: 00:40:30 Yeah. Wow, things are improving fast.

Jon Krohn: 00:40:35 Yeah, exactly. So is that also amazing? Yeah. Improving for machines.

Chip Huyen: 00:40:40 Yeah.

00:40:42 So back to the lens of the humans. So I just made me think about like, okay, so we're saying that's like the cost of building software now is approaching zero, right? It's just like code generation is like cheap and nobody really care about like in the past people compare how many lines of code you write today, right? It's quite meaningless nowadays. So the cost of building software is zero, so what does it mean? It doesn't mean that the value of building software is also zero, right? Because if the cost of it is zero, then obviously, but yeah, so maybe something about like, okay, so what is the next frontier and the things that a lot of people think that we still have a lot of open ended problems in software. I'm not saying that it's also, but I think it's like we are on a trajectory where things are just being sold as a much, much faster rate, right?

00:41:38 And it's so harder to predict like how long things are going to take. So a lot of people are like start looking into like the physical world, like how do we get AI to be able to perform in the physical world? And I think it's like even from my perspective, and I could be wrong, is that I see the digital AI agent in the digital world actually have a lot in common with AI in the physical world, right? So I think if you look at like the ... So when we're working on AI Asian, we're trying to create like a mental model of what it's like. And I think it's like an agent is something that is interact with the environment, right? It can perceive the environment and act upon it and then I get feedback from

the environment. So in the digital world, the agents work in the digital environment.

00:42:28 They can work in a browser, it can work in like a computer, like a terminal, you can work in a VS code and the actions can be rewrite web browsing like send queries, right? So a lot of that is like a digital agent and the physical world, the environment can be the road, right? The agent can be a car with actions like turn left, turn right, break and things like that or accelerate. And on the actions, so you can actually map them, but so the challenge with doing physical world is that the physical worlds are not very well described. So in digital world, you ideally, like not everything is like that, but in a lot of APIs you have pretty good documentations. You have like descriptions of the functions, right? Okay. If you send a request like this, you receive responses in this shape. You have status quo, you have like arrow, arrow descriptions.

00:43:21 It's pretty interesting. Whereas in physical world, we don't really have a good description of like, okay, if you squeeze this much force into the X, it's going to break, it's going to break in that direction. So like as humans, we learn to operate in the real world because we have observed over time. We learned like, okay, like some kind of haptic intuitions, like we don't press too hard something. We don't step on a chart because we know that the chart is going to get hurt. Don't step on a child. Yeah, this is very important. We learned that. Yeah.

Jon Krohn: 00:43:53 Everyone at home remember.

Chip Huyen: 00:43:58 So the physical world does not have that. So the AI, it can maybe reasons, right? It can know that, okay, if I want to do this, I should be able to come up with a plan of actions that can help me achieve that task without coursing the consequences I don't want to course. So I do think that's like AI is getting really good at reasoning. It can come up with a task and come up with a task and then it can like

come up ... Sorry, it's given a task. It can come up with a plan to solve the task if has a good understanding of the physical environment. I think that's a lot of like initiative around the world model, like how to build a model that like encode physically accurate informations about the world so that AI can operate in it.

Jon Krohn: 00:44:47 World models, that's the word.

Chip Huyen: 00:44:49 Yeah.

Jon Krohn: 00:44:49 Yeah. So we've had big fundraisers. I think Faith A Lee was the first with her World Labs Jane Kern now with AMI Labs, a billion dollar first VC round, which is wild. David, I'm forgetting his last name. There's also a Google-David Herr. Oh no, David Silver.

Chip Huyen: 00:45:07 David Silver.

Jon Krohn: 00:45:08 From Google DeepMind in London, he raised like a billion dollars for his world model company too. Who are you saying? David Ha?

Chip Huyen: 00:45:14 Oh, David Ha and Smith Huber were the authors. It was a proof of concept world models. I think 2018 was one of the first ... World models is not a new concept. I think people have been trying to model the world for like very long time, but the ideas of using new network to build that, I think it's more modern, the world models.

Jon Krohn: 00:45:39 Yeah, it's really exciting. And so there are hard problems in robotics simulation to real world transfer, scarce action data, sub second latency, the cost of being physically wrong, like stepping on a kid. So it seems like that is similar to the kinds of real time machine learning problems that you were obsessed with at Claypod.

Chip Huyen: 00:46:04 So I think it's a different part of it. So I think like there are many challenges with AI in the physical world. So one of that is the AI part, right? They have to make AI capable

of understanding the world, coming with a plan of actions. There are also like the hardware parts, right? I think I actually realized this talk by the Unitre CEO. Do you know Unitre?

Jon Krohn: 00:46:31 Initrey?

Chip Huyen: 00:46:31 Unitre.

Jon Krohn: 00:46:32 Evening Tree?

Chip Huyen: 00:46:33 Unitre.

Jon Krohn: 00:46:34 Unitree. Oh, Unity. Oh

Chip Huyen: 00:46:36 No, Unitry. It's a Chinese company. Oh no, I don't know about that. I think they're one of the very cool robotics companies. So they just five for IPO, by the way. And one of the very rare robotic companies that claim should be profitable. So I'm actually very excited about the IPO. So the CEO has a great talk when you talk about like robotic intelligence and he distinguish between two part. One is like reasoning and the other is movement. So what that means is that like reasoning is like, okay, so robot look at the thing, given the task and I think about like how to achieve the task, which is actually a very similar reasoning for like digital agent with the caveats that the physical agent have to have a good understanding of the physical world. The other part is movement. And what that mean is that like the robot may come up with a great plan.

00:47:29 It's like, okay, to do that, let's just walk over there and open the door, but then if it's just like go one step and fall over because it trip on the wire or something, you would think it's stupid, right? So it's very important to have the robot to do, like be able to do a lot of movement.

00:47:46 I think that part is like doing a lot of increasing performance in the last few years. So I think when he was

talking about like how they had this really, really cool robot doing kung fu, so they have like a lot of fleet of robot and it's just like performing kung fu.

Jon Krohn: 00:48:05 And it's like six feet tall or something, like human sized?

Chip Huyen: 00:48:08 So they have like two humanoid, one is a G1 and the other HQ. So the HQ is very imposing. It's like six foot something, right? Oh my goodness. But it's actually very hard to work with. I don't think anyone I know is working with them. Actually, I have one friend who has like a shoe and he was like, "Oh, do you want to take it? " And the reason is that they got one, but then they felt it's like too heavy.

Jon Krohn: 00:48:31 They wanted to just give you a robot.

Chip Huyen: 00:48:33 But I don't know what to do with it either. What am I going to do with like a hundred something pounds robot? It's like heavy. I can't, I cannot carry it. I need to phone on you, like you're like not ideal. It's not great. But it's G1 is cuter. It's like five foot or something. That's pretty big. It's a lot easier to work with. So they have this like demo of like robots doing kung fu and instead it's like, okay, should do that demonstration. They program the robots doing like 20 movements, right? So like, okay, do this, I don't know, punch or something, whatever. And all of that is like pre-captured and then the robots use different motion to combine them to do the demo. But then he was talking about like we actually get, he believes that in the next six months they can also do like instant arbitrary motion generations so the robots can do like

Jon Krohn: 00:49:22 Different

Chip Huyen: 00:49:23 Motions without having Ping pre-programmed and pre-capture it. So I think it's very exciting. So I think like the two part of robotic intelligence. One is the reasoning, which I think just AI is getting like pretty good at. We're

understanding that a lot of people with a lot of VC money is trying to solve and there's the motion part that I think is like getting very exciting. So I feel like all the pieces are moving, I hope in the right direction.

Jon Krohn: 00:49:48 Very exciting. I can't wait to see what you do with this. Obviously it's stealth right now, but it's going to be exciting. For our listeners who can't raise a billion dollars in VC money in the first round, what recommendations do you have? You were kind of talking about this earlier, this kind of existential crisis, which I feel as well. It ranges from across everything I do. I did a PhD in AI and that gave me a real moat around my career. Other people couldn't create a machine learning classifier or understand problems with labeling data or these kinds of things. But now all of those kinds of things, a machine can do no problem for the podcasting too. I mean, it lowers the barrier to entry. You don't have to be a great writer to come up with great topics to script episodes. So there's a lot of people in a lot of industries who would feel like the moat is going away from them.

00:50:40 I'm wondering, so obviously focusing on physical systems is one way to create a bit of a moat for yourself because hardware, R&D cycles are going to be longer than software and there's lots of different ways that hardware can specialize and anthropic or open AI aren't going to next month just all of a sudden have a robot that does that too. So there's a mode in physical AI. So maybe that is just the answer, but are there any other ways ... Yeah, what other ways do you recommend? I guess you also have the systems design idea right at the beginning of this episode. What other tips do you have for our listeners on how they can try to future proof themselves a litle bit in this time? Well, nothing.

Chip Huyen: 00:51:24 This is really funny because I have a friend who is an economist. He's one of the smartest people I know and he does like consult a lot of governments and coming up

with like technical policy, tech policies on like how to get the nations stay up to date in the AI era. And he was straight up telling some of them, it's like, "Yeah, there's nothing you can do. It's like, you don't have enough budget for it. Just don't do anything." So it was like, some people do have a very, very pessimistic views of the world, but I think I'm more on the automatic side. I do think that's like AI can solve a lot of problems, but I also think that like they will never stop being problems for me, like for us to solve. So for one thing, it doesn't matter how many AI models are there or like how good AI models are, I would never stop being angry at people on the internet.

00:52:21 They're going to always people that piss me off. They won't always be customer services like I'm unhappy with. There will always be things like collaboration is not quite straightforward. So like recently there's a founder and I really like the founder, he's very smart. So he came to me and he pitched this idea of like another Asian orchestration framework and then he told me that all the problems that he have seen with a lot of companies is that there's not enough communications between like product and engineering, which is very classic. And he was just like, "Okay, and my Asian orchestration platform is going to solve that. " And I was just like, "I don't think that's a technical problem." I think usually when product and engineering people don't talk with each other, that require people solutions. It usually you don't solve that by like, "Okay, here's another tool." We magically make product people and engineer people get along.

00:53:18 So there are a lot of people problems that is not quite like sold. So I think that there's several categories of tasks that I think is like not entirely completely clear follows to like how to solve. One is like human AI collaborations, right? So a lot of AI tools nowadays kind of like built upon like legacy systems and things about like how we interact with like old software systems. So just like example of like coding tools. So originally we have a lot of coding tools

that like just part of VS code because VS code existed, right? And then we have like a lot of coding tools as part of the terminal because terminal has always like existed. But then like as long as it made me think like, wait a second, why are like editing like note taking apps like VS codes and terminals, why do you need both of them?

00:54:17 I mean, just going back, like why is it different stuff? And another thing is like, why is terminal so hard to use? So a lot of engineers, for me, I have used terminals in school and stuff and for work. I used to, but I'm not crazy happy

Jon Krohn: 00:54:34 With it. You've never been a big VIM person.

Chip Huyen: 00:54:36 No. Okay. So I have friends who are crazy VIM person. I have friends who just don't use VS code or like PiCham or anything. They just like straight up code into terminal. So they find a lot faster with all the key, right? So in theory you could use terminals as a- As an IE? Yeah, as an ID. So we know that terminals exist, but like because of like coding tools like cloud code, a lot of product people or like people who never use terminal before are suddenly exposed to terminals and they goes like, "Okay, why is this so hard to use?" It's very painful. And I think it's like, okay, terminals, maybe that terminals are hard to use on purpose because terminal is actually very powerful. Terminal basically give you access or control plane for you to like control the computer. You could easily remove RF like everything, right?

00:55:25 Yeah, you can just do that. So maybe like you make it hard to use so that only people who are willing to get used to it use it so they are less likely to make mistakes. So I think maybe, but also like a legacy thing and I think like why don't we have something like in between like we have something that can be both very, like can give you like access to the far system assesses a computer, like a control planet terminal, but also easy to use

	00:55:55	Like an IDE. And I think like that is, I talk about it and actually like OpenAI and actually they introduce a bunch of like desktop app, right? The college desktop app, which is basically the same idea of like, okay, very easy to use interface, but give you like access to a lot of things, the ways that the terminal can. So I think like this is evolving and also like a bunch of like how to access your agent when the computer is not working. You probably have seen people complaining about like, okay, like why I have to keep my computer open all the time because my coding sessions, like my cloud coding code are doing their things. So I usually just get into Uber and it's just like, can my computer open and I look like freaking nerds that make me think like there's no reason my computer should be open because it's totally run in the cloud and then we need like something that can access through the phone.
	00:56:49	I think people, a bunch of people are building it like, okay, now you want the things on the phone, I need some sandbox because like how do you share context between the phone and the computer? So anyway, I'm going to like very much use like genres like she's talking with you.
Jon Krohn:	00:57:06	Well, I'm just wondering what, this has all been very interesting, but what is this all ... Are you saying that there's lots of lots of problems that we can still solve? Yes.
Chip Huyen:	00:57:14	So I think it's like I use it, not a great example obviously, but I'm saying it's one thing is like we don't quite have a good understanding like what is the optimal way for humans to use AI. So human AI interface is on big things, right? Another category problem is just like- I see. She's not done. It's like just the first, it's one of my first in the 20 points. No, this is really important. This is good. Okay. Yeah. Another thing, if you will let me say it is how AI interact with the world. So we have a lot of

	00:57:50	Techniques to make AI good at using tools, right? But I think like for AI to interact well with the wall, we do not just want to improve AI. We can also make the world more AI ready for websites, for apps, we can make it like better documentations, better APIs that agents can call, better security and like making, okay, this kind of like actions are dangerous. So maybe you should like have less permission to AI and stuff like that. But how about physical wall? So I'm not sure you've seen this very cute video of like a food delivering robot. A football
Jon Krohn:	00:58:28	Robot?
Chip Huyen:	00:58:29	Food delivering robot.
Jon Krohn:	00:58:30	Food delivering, right, right, right. Yeah.
Chip Huyen:	00:58:31	So it's very tiny robot.
Jon Krohn:	00:58:32	Not a foot delivering robot. That would be weird. Do you
Chip Huyen:	00:58:35	Have HR
Jon Krohn:	00:58:35	Feet to like deliver? I'll take six feet please.
Chip Huyen:	00:58:43	So the robots are very cute and then the robot just couldn't cross the street. So the robot had to ask a pedestrian like, "Hey, can you press a button for me so that it turn green so that I can cross?" And the pedestrian was like, "What the heck is going on? " So I actually talked to someone who worked at one of those robot food delivery companies and he told me like the hardest part is just like how to get the robot interact with the world. So actually some cities have this like streetlight API so that the robot could connect to
Jon Krohn:	00:59:17	Streetlight
Chip Huyen:	00:59:17	API so it can turn, it can press the button

Jon Krohn:	00:59:22	Via the API. Oh, because it can't press the button to say that I want to walk.
Chip Huyen:	00:59:25	Yeah. So someone saying that part of like how you could provide toolings to make the world easier for AI to operate in. Sure, sure. Yeah.
Jon Krohn:	00:59:35	Nice. Well, these were lots of great ideas. We do need to start wrapping up the episode a little bit because our next guest has actually arrived, a friend of yours. So you're going to chat to him
	00:59:47	In a moment, but we do have actually for you, we have some audience questions and so I want to get at least one or two of those in because we got a huge response to you coming on the show. It's It's maybe one of the biggest responses we've ever had about a guest coming on. All right. Our first audience question, so I posted on LinkedIn that chip would be on the show. We had this tons of questions come in and my first question, there were way too many to ask, but I'm just going to ask a few that I think are some of my favorites that might interest the audience the most. Rahitchia Valpuri, who is a business systems analyst for CIBC, a big Canadian bank. She's in a suburb of Toronto called North York.
Chip Huyen:	01:00:28	It's a nice city.
Jon Krohn:	01:00:29	Yeah. I'm from Toronto. Did you know that?
Chip Huyen:	01:00:33	Oh, that's why you're so nice.
Jon Krohn:	01:00:38	Rahiti says that she loves your writing so much so that when she's reading your books, I kid you not. There are moments when she finds herself smiling and she wonders if you're working on a new book.
Chip Huyen:	01:00:50	Oof. So I've been playing with this idea of writing a novel from an AI perspective with an AI narrator. So it's like try to explain how AI works from the AI perspective. It's a

crazy idea. I'm not sure I wouldn't do it, but I do want to get back into creative writing. I think my physical AI could be very interesting as well, but that also evolving quite a bit. So I'm trying to finish one blog post about physical AI. Hopefully it's going to be out by the time the show like the episode is air.

Jon Krohn: 01:01:22 Nice. Yeah. That should be easy to find from your website, which we'll mention the URL of in a moment. And then yeah, some of the questions we actually addressed in the episode. So someone named Jing Xu, who she is a very frequent listener and she makes lots of posts tagging me. And so I really appreciate that. Keep it up, Qing. Thank you. It's always great to get so much interaction from you. So she works as a director of data science at Elevance Health in Chicago and she had lots of questions about how vibe coding, so gen AI models, how that has changed, what it is expected of AI engineers. And I think we talked about that a lot in this episode. That was kind of even your very long answer near the end of this episode was basically about opportunities for us. You are

Chip Huyen: 01:02:04 Free

Jon Krohn: 01:02:05 To cut it off. Okay. No, I can't. I can't do that. So yeah, I think that we've kind of covered a lot of the really big questions. Okay. Here's one very last one. So this is from Brian Willett, who's a retired research scientist. Very easy question for you, Chip. When will AGI be achieved?

Chip Huyen: 01:02:28 So it's the definition of AGI. Do you follow the lawsuit with OpenAI? I think. Okay. So the contract between Microsoft and OpenAI is contingent on some definition of AGI. Oh yeah, there is. And I think they already evoked that. So by the definitions we

Jon Krohn: 01:02:44 Are- Legal definitions, we have AGI.

Chip Huyen:	01:02:46	I mean depends on whose legal team, right? But I think by the contract, I think Microsoft evoked that we are in AGI.
Jon Krohn:	01:02:55	Wow. I guess we'll see what happens there. We'll let the lawyers decide whether-
Chip Huyen:	01:02:58	Do you
Jon Krohn:	01:02:58	Feel
Chip Huyen:	01:02:59	Better, like worse that we are in AGI?
Jon Krohn:	01:03:03	I mean, it's a difficult thing. There's a paper. I did an episode a few years ago, a five minute Friday episode on the five levels of AGI. And I think this is an important thing. And I think there's lots of areas where we do have, there's amazing capabilities now in text to text, text to code, code to text. In a lot of ways, in a lot of domains, we have now exceeded the average person's capability on those kinds of tasks. But there's all kinds of things you brought up like squishing an egg or stepping on a child. There's all kinds of things that AI systems still need to improve on. Yeah, difficult to define exactly, but I guess we'll see what the lawyers say. So yeah, that brings us to the end of this great episode. It's been so nice to be here with you in person.
	01:03:49	Before I let you go, I know that you're famously a voracious reader of books.
Chip Huyen:	01:03:54	I'm a great influencer. I'm not sure if that means anything to you.
Jon Krohn:	01:03:57	Yeah. I mean, you're a huge AI influencer and now you're going to be a big literary influencer because I understand you have an Instagram channel dedicated to your books. Not that you write, just the ones that you read.

Chip Huyen: 01:04:10 So I really like books. I like reading. So I have this Instagram account that I created recently like Chips Lip, which is really hard. I just realized they're very hard to pronounce, but like Chip's library, short for it. Chips Lib. Yeah. I was showing my friend, my cows the other day and he was so disappointed. I was like, "This is the most over hype book account ever seen in my life." Because I feel like, okay, I finished reading a book and I want to post a review and then when I get busy so I just don't post a review. So yeah, but I try. So I try to focus on books. So I think of as both fun to read and teach me something new.

Jon Krohn: 01:04:53 Fantastic. Do you have a new book recommendation for us? When you were last on the show, we did that one remotely and you were on your walking pad and I had to ask you before we started recording, I had to ask you to stop walking on the walking pad because like the camera was moving, you could hear it. But behind you on your walking pad at your standing desk at home, you have a big bookshelf with lots of books. You pulled a number of books off the shelf to tell us. Do you have anything new that people need to read?

Chip Huyen: 01:05:17 Oof. So last year I think I realized a book like Apple in China. So it's about the Apple supply chain in China. It's fascinating. I share a lot about like how both Apple's strategy and how Chinese government thinks about technology supply chain as a comparative advantage. I think it actually plays out quite interesting as in good robotics nowadays it's just really hard to find an American company that is as fast moving as a Chinese company in terms of robotic hardware. So I try to buy some robots and I go to a bunch of like American companies and a lot of them just don't have robots to sell. Whereas like we go to Chinese companies like, "Okay, which one do you want? Here on the option." It's like,

Jon Krohn: 01:06:07 Whoa, it's like

Chip Huyen:	01:06:08	Amazing. So it's very interesting that book is also like very much into like urban design because I feel like Citi is a system and thinking about how I will approach that. There's the books I found very amusing recently. It's a book on parking. I think it's like paved paradise. I know parking, right? And I think, do you know that like on average you have to allocate six parking spots per car in the city.
Jon Krohn:	01:06:37	Really?
Chip Huyen:	01:06:38	I know. I know it's quite crazy.
Jon Krohn:	01:06:39	That's wild.
Chip Huyen:	01:06:40	Yeah. Because you have like one of the apartments on at work shopping mall
Jon Krohn:	01:06:45	Or something like that. Oh, I see. So you kind
Chip Huyen:	01:06:47	Of have to look. So it's kind of interesting. And so I try to read about it because I do think the self-driving cars are going to change a lot of that because like a lot of cities nowadays are designed around parking space. Have you been to LA? It's truly spread out because they need a lot of parking space. And which also make it worse because a lot of parking space, a lot of empty space. So now you can just further apart. So more about need car. It's just like vicious cycle. So I think self-driven cars are going to make things so much different because people don't need to park anymore.
Jon Krohn:	01:07:20	For
Chip Huyen:	01:07:21	Sure. So I think I find that books very amusing. It's
Jon Krohn:	01:07:25	Interesting.
Chip Huyen:	01:07:26	Yeah, just like a bunch of like other books like that. Yeah.

Jon Krohn:	01:07:30	Nice. I'm sure we could go on and on, pave, pay redis, put up a parking lot. We will end on my little jingle. How should people follow you after the episode? You have over 300,000 people following you on LinkedIn. It's incredible. So you can follow obviously on LinkedIn.
Chip Huyen:	01:07:48	I wish it means something, but yes. Thank you.
Jon Krohn:	01:07:50	It definitely means something. I'll tell you that for sure. Anywhere else that people should be following you? We got Chips Lib on Instagram. We've got your LinkedIn account.
Chip Huyen:	01:08:00	It's not AI account. If you follow it expecting to see the latest agentic news, you'll want to be sorely disappointed. But yeah, I think that's just classic Twitter, LinkedIn. I'm trying to start a Substack. I've been trying for like two years. I have the coming soon and people just keep tagging me like, "You said coming soon a year ago."
Jon Krohn:	01:08:23	I know.
Chip Huyen:	01:08:23	When? I know.
Jon Krohn:	01:08:25	All right. Well, we'll see maybe next time you're on the show. We'll have your Substack to talk about as well. It'll be vibrant by then. Thank you so much, Ship for meeting with me in person in San Francisco. It's been such a great episode. Such an honor to meet you in person. And yeah, hope it's not too long before you're speaking to my listeners again.
Chip Huyen:	01:08:41	Yeah, no, thank you so much for having me again and congratulations for like 10 years, a thousand episodes.
Jon Krohn:	01:08:46	I know. This is
Chip Huyen:	01:08:47	Amazing.

Jon Krohn:	01:08:48	I know. Yeah. The very next episode will be episode a thousand. Thank you for being episode 999. What a special spot.
Chip Huyen:	01:08:53	Yeah, no, this is amazing. Thank you.
Jon Krohn:	01:08:57	Wow. What an episode with Chip Huyen in it. She covered the sharp line between machine learning engineering, which meant building a model from scratch by collecting data, training and deploying, and AI engineering, which treats powerful models as a service you call instantly, making time to market dramatically faster. She talked about her start simple philosophy, meaning reaching for prompting first, rag, retrieval augmented generation next, and fine tuning only as a last resort. She talked about how the hard part of physical AI isn't reasoning, but the fact that the real world has no documentation and one fix is making the world more AI ready, like Citi's exposing a streetlight API so a delivery robot can change the light itself instead of begging a pedestrian to do so. And she talked about how AI is making the act of building software nearly free and in that new paradigm, the value of software hasn't dropped to zero.
	01:09:51	Instead, the durable problems worth solving are increasingly people problems and physical world problems that no model can simply copy. As always, you can get all the show notes, including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Chip's social media profiles as well as my own at superdatascience.com/99. That's fun to say.
	01:10:16	Thanks, of course, to everyone on the SuperDataScience Podcast team, our podcast manager, Sonja Brajovic, media editor, Mario Pombo, our partnerships team Natalie Ziajski, our researcher, Serg Masís and our founder Kirill Eremenko. Thanks to all of them for producing another fantastic episode for us today for enabling that super

team to create this free podcast for you. We are deeply grateful to our sponsors. You can support this show by checking out our sponsor's links, which you can find in the show notes. And if you yourself are interested in sponsoring an episode, you can get the details on how by making your way to jonkrohn.com/podcast. Otherwise, please help us out. There's lots of other ways you can do that by sharing this episode with folks who would love to listen to it, reviewing it on your favorite podcasting app, commenting on YouTube, subscribing if you're not already a subscriber.

01:11:03 But most importantly, I hope you'll just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Join us for a very special episode, episode 1000 coming up in a few days. It'll have both Cural, the original host and founder of this show as well as me and lots of people dropping in from all over the world to ask us questions, including listeners just like you. All right. Until next time, keep on rocking it out there and I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.