Brian Gracely (00:02.818)
Good morning, good evening, wherever you are. And welcome back to the Cloudcast. We are coming to you live from the massive Cloudcast studios here in Raleigh, North Carolina. Hope everybody is doing well. We are now into September of 2024. Hard to believe that there's only four months left in this year. The year has gone by very, very quickly. So hope everybody's doing well. Hope the weather's starting to cool down a little bit for you. The nights are, the days are getting a little shorter, at least here in the Northern hemisphere. And kids are back to school, college football is back. Lots of, lots of things are good in the world.

Leaves are going to start changing, the temperatures are going to start to drop a little bit, and we're going to start getting into fall. one of our favorite times of the year, especially here on here with the Cloudcast. what I want to dig into today a little bit is there's so much that's been changing in the AI world to the point where it's almost hard to keep up. And I know I was working on a project earlier in the year talking to one of our data scientists leads, our AI leads, and we were talking about how do you

How do you go about dealing with AI models and the pace of change that we have? And so the time we were sort of digging into some technology called RAG or retrieval augmented generation, which was basically, you know, ways to augment existing AI models when you have new data, right? Anyways, it got me thinking, you know, we were talking about it, we were digging into it and we were asking, hey, is this, kind of where the world's going? And he said, you know, quite honestly, even this stuff is pretty new. And so what I thought I would do today is kind of throw out a little bit of a framework.

for at least given what we know today, what's sort of out there today, how you might go about thinking about, you know, how do we apply our business information? How do we sort of make an AI model that's going to be used for one of our applications? You know, how do we think about that in terms of how do we best bring our company data to that? How frequently should we update that? You know, what's the best approach in terms of, you know, how do we go about integrating this with our applications?

so I thought what I would do today is try and lay out somewhat of a framework. And again, it's not perfect by any means. We'd love to hear feedback from people on where it works, where it might fall down, all those sort of things. But kind of want to dig into kind of a concept I've been thinking about in terms of fast data and slow data and how that relates to the pace of change of your data, of how fast you need your business to keep up with certain things, and sort of dig into that in the context of AI models. And so I want to do that after the break.

Brian Gracely (00:02.702)
And we're back and I'm your host Brian Gracely. Welcome back to the Cloudcast. For those of you that are used to watching the show on YouTube, as we've been posting the news on YouTube for the last, I don't know, three, four months or so. Today we're not going to, we've had some technical difficulties in terms of some of the video recording things. So this one's just gonna be out on the podcast. know, normal for 99 % of you that are listening to it on the

podcast, nothing will change there, but I did wanna highlight for anybody who's, you know, started using YouTube as the way to watch it, this one won't be.

out on YouTube. There will be the, you can listen to the podcast on YouTube, but the video of us recording it won't be out there. So what I want to kind of get into today is there's often a lot of question about, you know, do we, do we build an AI model? Do we try and, you know, fine tune an AI model? You know, what's the difference between when we would do one versus the other? And then there, you know, we're starting to get into all sorts of questions about how do we go about tuning a model or aligning a model, right? Sort of how do we,

make sure that it's well aligned to the data that you want it to be aligned to, that you can sort of influence the types of answers that it might create. A little bit of what's the difference between predictive AI, kind of big, bigger static models versus some of the new generative AI with large language models or even small language models. So how do you go about making sense of all these different ways that you potentially can make a model

appropriate for your business, appropriate for your application. And I thought I've been sort of thinking about, you know, kind of a way of thinking about a framework for this, way sort of a taxonomy, if you will, for, know, when would one approach be better than the other? And again, there's never going to be a specific right answer, right? I think what we were finding is that there are, you know, different ways to make a model appropriate for your application, for your business. And it involves a lot of different things, everything from

You know, are you building the model from scratch? What does that cost? How long does that take to we're going to use a Existing model and existing for example large language model. Maybe you're going to augment something like an open AI or Google Gemini or something along those lines maybe you're going to use an open source model like like llama or IBM granite or maybe you're going to You know try and come up with some other way to influence a model

Brian Gracely (02:25.974)
such that the experience of your application is what you want it to be, or it enhances the experience that you're trying to drive. And of course, the challenge to all that is it's never just a simple answer. It's never just, if it looks like this, always do that, right? Because there are many factors involved, right? One of them is going to be the models themselves, right? So how large are the models? How expensive are the models to run? How big is the model?

you know, did you build it from scratch? How much data did you have to collect? How much, know, sort of scrubbing or ETL of the data did you have to do? do you have a very, very large data science team or do you, you you only get periodic access to certain things, right? So all these different things have both a technological impact on some of the decisions you might make as well as an economic impact on, what you might do, what you might decide. And so what I thought I would do, and again, this is maybe a little bit overly simplistic, but I kind of want to throw out

a beginning of a taxonomy for how you might go about thinking about how to build the framework, the processes, the things around your AI models as they might make sense for your business. So I broke it down into four segments. And the first segment is things that I will call large static models. there's lots of you go to many companies, often have larger companies.

They've had a data science team in place for a while, and they've been doing predictive AI for a number of years, right? This could be everything from, you know, if you're working in retail and people buy some product, here's what we also recommend, right? People like you also bought this. You might also like these types of things. And we've seen predictive models in, you know, lots of different stuff. It could be in weather, could be in retail, it could be, you know, as part of, you know, some service for, you know, how you're driving, all those types of things.

And these tend to be very, large models. are trained on lots of your own data, lots of, you know, domain specific types of information. And they're not necessarily expected to change all that much. Like once the model is sort of trained and it's been sort of fine tuned or, you know, customized for the type of tasks that you're doing, it's really a matter of then applying it to, you know, the activity involved. And hopefully it ends up either augmenting it to create better results, create better user experiences, or

Brian Gracely (04:51.638)
Reducing the cost of something right so for example can be using predictive AI to help determine Whether or not somebody should get an insurance policy should get a mortgage rate, right? Well, you can you reduce a lot of the process that used to be manual and sort of the deciphering if you can use a predictive model, right and it's gonna be based on You know sort of well -known data for a specific domain. So, you know Like I say that those are oftentimes the scenarios. They become large static models

And those are the things that are often sort of custom built. Your data science team builds them. You're going to build a good size predictive AI model. The second thing that I had in this taxonomy was scenarios in which you're trying to do something with LLMs. So maybe it's you're building a chat bot or maybe you're doing something more unique, but you're building something using LLMs as the type of model as opposed to a predictive model. You're doing more generative AI.

And so you're thinking well, let's start with sort of the base information, the base knowledge, whether it's a coding assistant or a language model. And then what we want to do is we want to sort of augment that in some way, right? You think the use case is something that can be sort of augmented from the core knowledge, the core data that's within a large language model or even a small language model. And so I think the first thing that you sort of ask yourself is how often do I want to go about

Kind of aligning this or tuning it right how frequently should I do that? and one of the things that I found in talking to a lot of different companies is They often are sort of asking themselves, you know, how quickly how frequently is the data? That is doing that augmentation that fine -tuning

that alignment. How often is that changing right and what I find is oftentimes if the data is changing fairly slowly, right and you

you kind of want to just say, look, I just want to sort of embed the data within this model. you know, we're kind of keeping the model to ourselves. We're not putting it back in the public domain. I often call this sort of slow moving data, right? So maybe the data really only changes once a year or maybe once every six months, or even once a quarter might be considered slow moving data in your world. Right. So for example, you know, I recall us doing a podcast years ago, talking to the folks from fender guitars and they said,

Brian Gracely (07:14.636)
you know, we have this system in which, we're getting, you know, updated inventory information and we're getting it on like a nightly basis. So in that case, they were talking about fast changing data, stuff that, you know, they needed to deal with on a regular basis. It was updating, you know, all these different parts and all these different availabilities of, you know, parts and things in stores. So that was considered sort of fast data, but you may very well have things that are coming out once a quarter.

once every six months, maybe it is a new addendum to a law. So maybe like within your federal government or your state government, your law firms are looking at what's changed, what do we have to be updated? And that stuff may only come out once every six months or once every year. And that might be the type of thing where you say, well, the cost of doing the augmentation around fast data is kind of expensive. But if we're doing it around slow data and we're kind of embedding it within the model,

fine -tuning it around this thing once a quarter, once every six months. That makes more economic sense and that also might make more technical sense because your teams aren't constantly churning with it, right? You're doing it very infrequently. The flip side of that is what you might call fast -moving data, right? And I gave that example of, you know, some aspect of your business is getting rapidly changing things, right? Like maybe you're working in, maybe you're working in something that has to do with sports, right? Maybe you are

the sports group that is writing a daily article for, I don't know, fantasy football this time of year or something along those lines, something in which there's a lot of change happening. It's really happening very frequently and you don't necessarily want to constantly be spending GPU cycles and data scientist cycles on updating the model with this constantly changing information. Maybe it's financial markets or maybe it's whatever it might be.

So this is something where a technique like, like rag retrieval, augmented generation might make a lot more sense because now you're sending your cycles on vectorizing the data, getting it into a vector database. And then you sort of leave the model and you leave the application alone and you just simply augment the model or augment the application, whatever the interaction is to say, Hey, you know, make sure that you bring in the data that is coming out of this.

Brian Gracely (09:39.682)
vector database, which is part of how the retrieval augmented generation works. Think of a vector database as sort of an augmentation to the actual LLM. So again, that concept of fast data versus slow data maybe is a useful way to think about, if it's not being done that often, I might want to embed that in the model because I saved myself all the cycles and all of the additional work of

constantly vectorizing things, running a vector database, having to build additional application logic into saying, go off and query the model, but also then query the vector database or the RAG implementation. But in the case of something that moves fast in which having the most up -to -date information or the very recent information is more important to your application, this is where fast data or the RAG approach might make more sense. So so far,

of the three of the four, we've got sort of large static models, predictive AI, something you're probably going to build your data science team. We've got slow data, which you're probably going to want to try and tune using some mechanism. There's lots of different ways of doing tuning. And you may want to embed that within your AI model such that you can have a level of consistency within that model. Maybe you're going to distribute that to a lot of different locations. Again, like I mentioned, like

lawyers offices or something along those lines. You're going to have the scenario in which you're going to have a set of fast data. You know, how frequently do you want to augment that? Well, even the, you know, the most, what do they call state of the art and frontier models aren't coming out more than every three or four months, six months, eight, eight or nine months. Probably can't wait that long to update them. So you're probably going to look at some sort of retrieval augmented generation or rag to augment.

Your model and use you know sort of vectorize that data put it into a vector database And then let the model along with the application that's using it also sort of query the database Augment though all that stuff together to give you You know access to the most real -time information that you possibly can or fast data if you and then the last sort of scenario that I had in mind was the scenarios in which you're trying to do things that

Brian Gracely (12:03.384)
you know, probably don't need kind of constant human interaction, right? So we're not dealing with chatbots. We're not dealing with sort of customer service agents or, know, anything along those lines or even, you know, sort of the, you know, AI, AI ops kind of stuff where you've got a, you know, a system that is, you know, kind of looking at stuff and trying to fix them, but you're going to have scenarios that are now being called, agentic. So what you're going to have an agent or maybe, you know, dozens of agents, basically constantly querying the model.

for information, taking the output of that information, and then further trying to enhance or question the model. So think of this in the context of things like you're trying to do discovery of a

new therapy for a health care cure. So maybe you're doing cancer research, or you're doing just any sort of research in the scientific domain. Maybe you're trying to figure out how

a new metal or a new chemical bond or something is going to come together given certain scenarios. And in that case, you're sort of thinking of a model that is constantly building upon itself, constantly learning from the outputs of previous models, being able to take to run 10 ,000 scenario tests to see what the average is or where the outliers are. And those are the types of things in which you're going to have not just an application talking to the model, but you're going to have

agents talking to the model, agents talking to each other, summarizing, collecting, kind of bringing all those things together and then constantly churning on that. And that's something in which you're not necessarily talking about like fast data or slow data, but you're talking about sort of constant change to data and things that are gonna go in there. And this is where you're really starting to see what's being called agentic, agent.

agent -ish, agentic AI models or AI interaction is starting to happen. So I put a number of links in the show notes for all four of those types of things, whether it's predictive AI with sort of large static models, whether it's alignment around slow moving data, whether it is rag with faster moving data, or these sort of larger problems that are trying to be solved that are kind of constantly going to be evolving over time with agentic AI.

Brian Gracely (14:26.934)
So it's not a perfect way of looking at it. I'm sure there are lots of gray areas in between all of those, and there will be new stuff that will come along in the next three months and six months and other things like that. But I thought it would be useful to try and put together somewhat of a taxonomy for folks. As you're thinking about the type of problem you have, the frequency and change of your data, of your data sources, how quickly your business needs the application and the model to adapt.

for new changes in order to be able to deliver the experience that you want, the accuracy that you want. And then to give you some sense of, know, there are lots of different ways to get to, you know, building or tuning or aligning models. And it really just depends upon what you're trying to do, what your economic model looks like, you know, what sort of resources you have from a data science perspective. And then also taking into consideration things like, you know, do the, does the model that I'm working with

Maybe it's a model that you built. Maybe it's a model you got from open source, know, hugging face or somewhere. Maybe it's a model you get from a third party company that was designed for some specific tasks within your organization. And in all those cases, you're probably going to constantly say, okay, how do we make sure it's up to date? How do we make sure it's giving us the right information? How do we, you know, how do we help it learn from what happened in the past? Hopefully make it smarter. And there are, you know, many, many ways that are out there right now. We didn't even get into some of the, you know, the variety of tuning.

capabilities that are out there. I feel like this sort of big monolithic model, slow moving data, fast moving data, agentic, is a pretty decent framework for thinking about how are going to go about doing this? And then obviously we're going to overlay that with specific technologies that fit into those categories. So anyways, hopefully this was helpful a little bit as you're starting to think about, we have all these choices out there. We have all these options. Which one should I pick? Which one makes sense for my business?

At least maybe it gives you a starting point for a taxonomy to think about, we can apply both figuring out which technologies make sense, which people and process make sense, which economic model makes sense, and then start to hopefully go down a path that's a little more narrow to get you closer to what makes sense for your business, for your application, for your user experience, and all those types of things. So anyways, with that, I will wrap it up. Hopefully everybody is...

Brian Gracely (16:48.288)
Excited about the fall being here, excited to be here in September. We've all made it through the first eight months. Hopefully everybody's having a good year so far. And with that, I will wrap it up on behalf of Erin and myself. I want to wrap it up. want to thank you all for listening. Thank you for telling a friend. Thank you for helping us grow the community. And as always, if you got questions or feedback, show at thecloudguest .net is the best way to reach us. So with that, I will wrap it up. I'll talk to you next week.