SESSIONS

Turbocharging Your RAG with Data in Google Drive and LangChain: Virtual Workshop - February 7, 7:00 PM | 9:09 PM

Colin McNamara - 7:00 PM
Confirm, session transcription started, recording is starting, participants, no attendees. Okay.

Karim Lalani - 7:00 PM
If you like food, don't forget to like and subscribe to my channel, I'm Qiong, see you in the next video!

Karim Lalani - 7:00 PM
See you in the next video!

Colin McNamara - 7:00 PM
I don't know how to do that. Are you... You're already, oh, there you go, in the middle. I don't know if there's a way to, like, make it... Huh?

Karim Lalani - 7:00 PM
Of course, if you go into the settings, for me I see the moderator setting, there is the session setting. The first setting I see is the local sessions setting and it kicked in, I just disabled it. Sure. So we don't have to manually...

Colin McNamara - 7:01 PM
Oh, cool. I'm going to change that on the automatic side in the future. So everyone who's joining, we're going to give it a few moments for everyone to filter in. I'm going to just share my desktop so everyone can see pretty stuff. Boom. There we go. I'm going to...

Colin McNamara - 7:02 PM
we had like a bunch of people teed up. Takeaways, notes. Also pop over to the meetup page and make sure that we have the proper thing here.

Colin McNamara - 7:02 PM
Maybe that was, we might've had a problem with changing something. Let me, seven participants. Maybe the auto add, was it the? I I'm just, we're, people are starting to filter in and Charles is saying he's giving, getting like a spinning wheel.

Karim Lalani - 7:03 PM
...under the session moderator.

Colin McNamara - 7:03 PM
Still there, even updated Chrome, do a demo, lose a sale.

Colin McNamara - 7:04 PM
and we may have a failure in our event." Let's see if I can kill the event and restart. I'm going to try to kill it and restart it. There's Admin All, there's Abhinav coming in, Tyler's coming in.

Colin McNamara - 7:04 PM
We are doing our best to let people filter in. Let's give this a few minutes. And we'll let people come in. Apparently we're getting spinning wheels for some people. The platform itself might be a little slow, so we're going to go ahead and give it five more minutes. And we'll see if we can kick this in the butt.

Colin McNamara - 7:05 PM
Awesome. Thanks, Ariel. There's Charles. Admit. Admit. Charles Martin. We're going to make an assistant. Awesome. OK, we've got Charles here. And then Scott is in as well. Happy days. Hey, everyone. We're going to give it five more minutes till seven.

Charles Martin - 7:05 PM
All right.

Colin McNamara - 7:05 PM
10, just let everyone load in. We're getting feedback that the platform is taking time to load. So you can chill out for a bit. Everyone wants to pop on to the agenda on the left-hand side here. What we'll have is the ability to load up all the labs.

Colin McNamara - 7:06 PM
And also in the PowerPoints, the presentation that we'll be going over today. Thank you for your patience. In the meantime, I don't have any old music. And if I play any music from Alexei, it's going to kill us on YouTube. So we're going to have some chats here. I'm going to go ahead and pop over to the meetup interface really.

Colin McNamara - 7:07 PM
I don't see any people reporting challenges here. We'll go over to our fancy Discord. While we're waiting, we'll point everyone at our fancy Discord. If you haven't made your way over here, we go ahead and coordinate our work. We'll coordinate our labs.

Colin McNamara - 7:07 PM
We'll share our meeting plans, coordinate our commits and whatnot, and answer questions here and there. I highly recommend that you check it out. It is available at URLs here. We also have our YouTube that we are posting these things to.

Colin McNamara - 7:08 PM
Go ahead and check the time, check our sessions. We've got 14 people in. Looks like people are starting to filter in. We'll give two more minutes and we'll kick it off. Thank you for your patience. It looks like Charles is helping us out by throwing up some poll questions. So I'm going to answer these myself. What are you trying to do with the LLMs?

Colin McNamara - 7:08 PM
I'm using mine for risk mitigation and fostering community.

Colin McNamara - 7:09 PM
So first things off, let's see where we're at. We're 710. Okay, so welcome to Austin Lane Chain. Austin Lane Chain User Group Virtual Edition. It's February 7th, 2024. We're here tonight to walk you through some labs, help you with some code, and come together and learn in the open.

Colin McNamara - 7:10 PM
We are Austin Lane Chain User Group. We're headquartered here in Central Texas. And I want to give you a little update about who we are and what we do, share some news and announcements, and then get into labs that are focused on Google Drive. So we'll go through a short introduction lab based on a Jupyter Notebook for connecting Google Drive data into

Colin McNamara - 7:10 PM
a record. We will then go into three more labs where we're going to show how to do that RAG with Google Drive in Langsurf. So that'll be run inside of Docker. We'll do multi-mobile image interpretation and then we'll go ahead and actually synthesize documents using Pandas

Colin McNamara - 7:11 PM
and create some really cool graphs and stuff. Okay, a little bit about Austin Lane Chain Users Group. You can find us in multiple places. As I showed in our waiting period, we are available on our Discord. That's kind of our home away from home. You can find us on GitHub, is where we put all of our labs, where we put our resources, our presentations. For these events, both in-person and virtual, we coordinate them through.

Colin McNamara - 7:11 PM
The meetup platform, and you can find us at Austin Lane Chain AI Group. We post our events to Twitter, if you follow Twitter. And then you can follow at Austin Lane Chain. And then you can also review our past virtual workshops at youtube.com and Austin Lane Chain. We host our in-person workshops and meetings once a month here in Austin, Texas.

Colin McNamara - 7:12 PM

We host virtual meetings also once a month. So basically every two weeks we're doing something that we do on the Sessions interface here that we're working with today. Our focus is really low-stress learning and sharing. We're not trying to be experts. We're all learning. This is a fast-moving project. We are here to connect with other early adopters of AI middleware, and specifically,

Colin McNamara - 7:12 PM
focused around the Lankchain project. Most importantly, we are here to learn, we're here to share, and we're here to grow together. This is really, really cool stuff. Mastering AI applications and the ability to create and manipulate AI middleware, I see as really the gateway to the new middle class. So I'm really happy to be part of a group that is sharing with each other. We have people that are represented.

Colin McNamara - 7:13 PM
Everything from the Artemis project, from the different state governments, from NVIDIA, from unstructured open source projects. We have people in robotics. We have people all across the board. It's really cool. We have a simple code of conduct. Everyone, please be cool to each other. Right? Please be cool to each other. We're all learning.

Colin McNamara - 7:13 PM
And a ego-free zone. When we're learning new things, you know, we break down into basically little kids. So let's treat each other with kindness. And on that note, don't be gross to each other. I shouldn't have to explain more. If there's any problems with that, if anyone has any challenges with that, you can talk to me directly. You can contact Charles, and we will handle it. Okay. Moving on. There we go. Okay. Moving on.

Colin McNamara - 7:13 PM
Click. Wow. Okay. I do want to give thanks to our supporters. For an in-person meeting space, the law firm has really been great. We can fit, I don't know, about 30 people in to two really beautiful conference rooms, as well as about 60 to 80 on this amazing space overlooking the capital of Texas.

Colin McNamara - 7:14 PM
It is really great and I'm really happy to have the support of local people in our community. I want to thank KeyChange for providing food and beverage services during our meetings, as well as Inspiration Ideation. They're really good at early-stage startups. I want to thank Always Cool Brands for providing the sessions platform that we're on today. But most importantly, I want to thank our contributors, people who contributed.

Colin McNamara - 7:14 PM
People who contribute code, people who contribute content, people who contribute community. There's all three of those legs come to support what we're doing here in Austin, Texas, Central Texas, but of course the Internet around the world. I think we're at 260 people or something like that. Super neat. Let's talk about some news and announcements. The 1.4 release is up in review. 1.0 got released this last month.

Colin McNamara - 7:15 PM
These are major doc updates across the project. One of the big challenges that I've had with the project and I hear other people have is the documents were really, really bad. On one hand, it frustrated all of us, but on the other hand, I think it was a testament to the fast-moving, get-it-done attitude of the LinkedChain team and Harrison leading the project. That being said,

Colin McNamara - 7:15 PM
they took a lot of feedback from the community and more than just taking it and arguing, which they didn't do, they took it and incorporated it into really awesome work. So if you had a chance to go back and look through the documents on docs.linkchain.com, it is really, really good. There's a lot of stuff you can follow straight out of the box. It's a lot more understandable. On top of that,

Colin McNamara - 7:16 PM
the Lanchain team has been delivering in spades. Harrison specifically has been uploading videos to the Lanchain YouTube channel along with direct links to the repositories he's going over. I encourage everyone to check their YouTube channel out and to check out the docs. Next news and announcement is LangRaph.

Colin McNamara - 7:16 PM
If you go to blog.lanchain.dev slash LangRaph, you can read a lot about it, and or, you can go to our Lanchain in-person meetup on 2.21. We will be focused on LangRaph and exploring that. If you notice a pattern about our meetups in our in-person meetups, it is a no-ego zone. We will present and run labs and do showcases on something that's been around for a while.

Colin McNamara - 7:17 PM
We're lucky to be really supportive in that, and what we do is we take the work that comes out of the in-person meetup, kind of polish it up a little bit, so we can go ahead and create a recording, create a broadcast, and engage with all of you on our sessions interface on the virtual. So, if you can, if you're in person, sign up now for our meetup on 2.21.

Colin McNamara - 7:17 PM
We do have space limitations in there, so I tend to cap the event at 42 people. About half that tend to show up. If you have a challenge and you're like, hey, I got waitlisted, just reach out to me on the Discord, and we'll figure something out. Next, a couple of papers. What are papers? It can be a lab that you did that, maybe you saw Harrison do it, and you did, and you thought it was really, really good.

Colin McNamara - 7:17 PM
It was really cool, and you wanted to share it. Maybe it was some code that you wrote that you think is really neat. It's based on the Lightchain project. The whole idea here is that we can learn from each other, we can share with each other, and we can amplify each other. If you have any papers, labs, workshops that you want to share with the team, share with the community, or one that you're really interested in seeing,

Colin McNamara - 7:18 PM
and you don't know how to do it, but you're looking for some help in it, post on the Discord. Let's create a discussion. We have some really, really smart people, and they're really, really cool. So let's work together on that. For those that are here for their first time, we do have a 101 that we posted from last month. It's on our YouTube channel. You can click the QR code here. It's not necessary to fully

Colin McNamara - 7:18 PM
go through it, but there are some things about how to set up your keys and whatnot that you're going to find in the 101 labs and 101 sessions. It's really cool to go through, and if you went through it, it's good to see you back. So let's kick it off. Let's go ahead and pimp our drive. So what we have today, I talked a little bit earlier, is we have four labs focused on getting data from your Google Drive into Lightning Chain and doing stuff with it. The first one that we're going to go...

Colin McNamara - 7:19 PM
through is a Google Drive RAG intro. The short intro, which we're going to introduce, I think a 20-line RAG group that pulls data in from your Google Drive and shows the power that you can do and give you a little taste of it. Next, we're going to show how you can extend links or templates to create an AI microservice that's

Colin McNamara - 7:19 PM
running Docker, containerized, and if you want to push it up into Kubernetes cluster, you can. One thing I really like about this is the authenticate. The way you authenticate inside the application is not very well described on the Internet. Ricky went ahead and created a really cool guide to make that easier. It's all problem for me, so go team. Next, we're going to have Prima Lalani talk about,

Colin McNamara - 7:20 PM
multimodal image interpretation, so pulling images off the drive, and describe it using Olama. We've seen the multimodal Olama. It's really, really cool things about doing image analysis and whatnot. Let's give you a little taste of that. Then we'll get Dr. Scott Aska-Kanossi. He's going to show us a really,

Colin McNamara - 7:20 PM
really cool thing. We went through this in our dry run, and that was like, holy smokes, the graphing that you can do. He's going to show us how you can pull CSV data from your Google Drive and play with it in Pandas, import it in Pandas DataFrame, and create really cool graphs and deep maps and stuff like that, which just blows me away. I'm really eager to do this today. On that note, let's kick it off.

Colin McNamara - 7:21 PM
A little overview of kind of the process that we're going to start with in our first lab, and let me pop the agenda. Now, if you haven't noticed on your screen here, there's a couple things I want to show you. So, one, if you click on Agenda on the left-hand side, you'll get links to the PDFs that we're going over and the notebooks that we're doing, right? Next, on the right-hand side, we have a chat.

Colin McNamara - 7:21 PM

So, you can be like, um, howdy, y'all, right? We can chat here with each other, right? Next, we'll have- you don't have access to this, but we'll see some polls coming through here. Now, if you have any questions about anything inside of here, you can ask a question. Be like, yo, what's Google Drive? And then, the presenter can answer it live.

Colin McNamara - 7:22 PM
Be like, hey, Google Drive is a way of sharing files using Google's platform, where if you create, like, Google Sheets, or maybe you have something local on your laptop, it'll happen, right? And boom, answered. Totally new. Notes, you can take notes yourself. Here are some takeaways. So, I'm going to go ahead and send this via email right now. This is just a master list of all the presentations, the slides, and the notebooks that we'll be using today. And then we have a transcript inside of here.

Colin McNamara - 7:22 PM
So, we can go ahead and pull this. We're actually using the transcript that we did from our last event. We'll be pulling that into the lab we're doing right now. Okay. Thanks, Gavin. Okay, let's get back to our pretty pictures here. So, again, our Google Drive, you know, I use it on my Mac where I have a Google Drive loader that sucks it up in the cloud, right? You can also create documents and just upload them for the web interface.

Colin McNamara - 7:22 PM
This puts it behind effectively an API layer that Google runs. Now, for a Google Drive plus Ragu overview, what we're going to actually do is we're going to use the Cloudian B library to go ahead and mount this Google Drive into our environment. We're going to load files in. We're going to split them up. We're going to take those files.

Colin McNamara - 7:23 PM
And we're going to create embeddings of this. We're going to basically take those files, and we're going to create it into a vectorized representation of the data, which is basically the natural language of these language models. We're going to shove it inside a vector store. In this case, we're going to use Chroma. Then we're going to pull it out of the vector store or retrieve it, you know, retrieval augment generation. And then we're going to augment it.

Colin McNamara - 7:23 PM
We're going to augment it with information from the large language model. And then we're going to output it to a file. We're not going to actually put it to a file. We're going to do it inside of our notebook. Moving on. Okay. So let's go ahead and open up our lab. So we're going to go ahead and fast forward. Open up this notebook here.

Colin McNamara - 7:24 PM
Which is also in your agenda. It's in the agenda. If you kind of hit show more, you're going to see a link to the notebook. And I have it right here. So now what I'm going to do here is I'm just going to go ahead and delete my run time to make sure I'm starting over from scratch just like you. Now a couple things. This lab is set up to basically inherit some of the assumptions that came

Colin McNamara - 7:24 PM
from our 101. So if you don't have it set up already, what we want to do is go into your keys on the left-hand side and we're going to want to populate it with open AI underscore API key and put your key in here. This secret container you'll be able to use whenever you're using your labs and any labs you do in the future with us you'll be able to do.

Colin McNamara - 7:25 PM
If you have a Langsmith instance, you can put in Lang chain API key and put that key here too. Mine are turned on, but I can go ahead and turn them off. You'll see how it actually pulls them in as we move forward. First thing we want to go and do is if you are new to this, we clicked open in Cloud. Now, you can do one thing, you can say, hey, save a copy in the.

Colin McNamara - 7:25 PM
Drive. Now, this allows you if you want to, if you're new to any of this stuff, this allow you to, to go ahead, get this all out of the way here. Okay. This will allow you to basically save this in your Google Drive and mess with it in the future. Okay. So the first thing that we're going to do is we're going to set up our environment. We want to press play on here. So we're going to use PIP or install, quietly upgrade, Lang chain, Lang chain community, Lang chain hub.

Colin McNamara - 7:26 PM
These three things are the core code, the community code, hub, which is a way that is a place that we share our, basically our prompts and having to use our prompts over and over in our code. Lang chain OpenAI, for those that were looking at earlier, working with Lang chain earlier, the OpenAI code has been moved into Lang chain OpenAI. It's a little more stable. We're going to install CRUD.

Colin McNamara - 7:26 PM
For our vector store, we're going to install BeautifulSoup for parsing of web pages, which we're actually not using right now, and then Cloud EMB, and that's going to allow us to connect into Google. So we're going to press play on here, and we're connecting to our backend. So this is connecting to Google Compute Engine, right? And this is an instance. We're just going to go ahead and...

Colin McNamara - 7:27 PM
Installing these packages. Now, it's going to complain a little bit. You can ignore the dependency error that's going to give you. It's not going to affect our work here. I didn't want to confuse everyone by putting more packages up here. So as this goes, it spins and spins and spins and spins. So for those that are new to this, Pip is a Python package manager that allows us to pull packages well within our kit.

Colin McNamara - 7:27 PM
So install our code. We have a dependency resolver here issue. It's not going to break anything. Next thing, we're going to get our keys from Google Cloud. So in this case, from Google Cloud, import user data is going to get the data inside of here. And then we're going to set an environmental variable of our OpenAI API key.

Colin McNamara - 7:27 PM
From the keys that are held inside our user data. If you're running this locally on your laptop or maybe Cloud Code or something, and you have an environmental file, you can uncomment this to get a bigger dot in. If you just want to be able to paste it in, and you're having any problem getting

Colin McNamara - 7:28 PM
your secret setup, you can uncomment this right here, and comment out this, and it'll go ahead and just pop up a little window for you to be able to get that. Let's go ahead and make this work right now. We press play, we want to grant access, and you'll see it pops on my OpenAI key. This is what this did is it pulled in the key into the system. Next, you don't have to do this. If you don't have Langsmith, you don't have to press

Colin McNamara - 7:28 PM
play here. For me, I want to actually run this. I want to pop this up to Langsmith. We're going to grant access again, and turn on our Langsmith key. This is going to send data up to smith.langchain.com, and then pick out a set to be my default project. We're going to go and not show that right now. Okay, so let's go ahead and start with our

Colin McNamara - 7:29 PM
exercise here. In this guide, we're going to create 20 lines of code. We're going to create a Langchain app. We're going to use from google.cloud, and we're going to import drive. We're going to mount this drive in a content slash drive. Now, if we see here, our cloud instance, our Jupyter instance, we don't have it yet. Let's go ahead and press play.

Colin McNamara - 7:29 PM
Executing, boom, mounted at content slash drive and there it is. G drive, I drive, we have it. Now, what I want you to do here is this code is not really important to the, whoa, that was weird. Spotlight, real moment here.

Colin McNamara - 7:30 PM
What's going on there? I'm just going to hide myself here. OK, we're going to keep going here. My picture keeps going crazy. That is totally weird and I am apologize for that. OK, so we want to go ahead and. Create some directories and download files for the lab. So this code right here is not really important for the the

Colin McNamara - 7:31 PM
code itself. What we're going to do is create a Austin Linkchain Labs in the root of your Google Drive. You can delete it. You can change this if you want. Right, we're going to go ahead and check if the directory is existing. If it's not, and then we're going to. We're going to download the transcript of our 101, right? So this is the GitHub. We're going to put this in here, right? We're going to specifically save it as linkchain101-transcript.txt, right?

Colin McNamara - 7:31 PM
And then we're going to basically print it as a file. So let's press play here, and you can see in this case, the Austin Linkchain Labs already exists, and then it is mounted and saved it into a linkchain101-transcript. Okay, cool. So now we have the file that we're going to be working with inside of our Google Drive. We know it's in the same folder. So let's actually go and close the linkchain. So let's.

Colin McNamara - 7:32 PM

Press play here, and we're going to, we're not using BeautifulSoup, you can, the earlier instance of this I had pulling down directly from the web, use BeautifulSoup with VS4 to parse it. So we're going to import the linkchain hub from linkchain. We're going to import a text splitter to recursively, basically, over and over again, split the file up. We're going to, we're not using the web-based loader, but we can ignore that.

Colin McNamara - 7:32 PM
We're taking the Chroma Vector Store from the community, linkchain community, that we're using to store this data in, we're using a string parser, Rumble pass-through, we're using, we're from OpenAI, linkchain OpenAI, we are importing the code for the chat model for OpenAI, and as well as OpenAI embeddings, is how we store the file in our vector store.

Colin McNamara - 7:33 PM
From our document loaders, or from indexes, we are importing the text loader, so we can transcribe it. Okay, so next I want you to go and press play, and let's load this document from our Google Drive. This is the first step, right? We took this transcript, and we put it in Google Drive, so now if we press play here, we've now sucked it into our loader object, and we've loaded those into our docs.

Colin McNamara - 7:33 PM
Next, we're going to take this file, right? And we're going to chunk it up. So our text splitter here, we're going to use that recursive text splitter, we're going to chunk it into 1,000 tokens, and we're going to overlap by 200 tokens, and we're going to split it up, and we're going to take docs and chunk it up and store it in the splits object. Next, we're going to put these splits and embed them into our code.

Colin McNamara - 7:33 PM
We're going to pull it, we're going to retrieve it, we're going to put the prompt, or actually pull this from LaneChainHub, right? So this is a simple RAG prompt, this is an easy way of changing the personality of your code. So boom, we pull that. Next, we're going to be specifying our large language model as our chat model, we're using FreeFox Turbo, and we're using our temperature of Sierra, so it's not...

Colin McNamara - 7:34 PM
going to be too creative. We're going to find a function, which basically joins things up at the new line, so if it retrieves a bunch of stuff. Now here's the cool thing that's been really new with 1.0, is if you've played with these in the past, it is very Pythonic, it looks like programmer code, where you actually build your RAG group. In this case, we define a RAG chain.

Colin McNamara - 7:34 PM
And in between the friends here, we have the context, the format of docs, this function. We have the question, and then we'll pass it through a prompt, an LLM, and a string parser. So those things that we defined up here. Okay. Very, very simple way of chaining our things together. Link chain. Press play. Now let's go ahead and invoke this pipeline with the question.

Colin McNamara - 7:35 PM
In this case, what did Ricky say about caching and streamlint? So Ricky mentioned that caching and streamlint allows for faster response time going straight from the cache instead of running computations. He also mentioned that caching for LLM responses to save server resources. However, he did not provide specific details about caching and streamlint. So in this case, if we go back to our example here, we are able to basically hold

Colin McNamara - 7:35 PM
the file from Google Drive, load it in, split it up, create the embedding, stick it in our vector store. We asked it a question and then we retrieved and augmented this with the data that was pulled back from the vector store, chunked it to make some sense of it and gave us an output. Really, really simple. Really, really fun. 20 lines of code. Now you can change this to whatever you want. Now you can also press play and clean up this

Colin McNamara - 7:36 PM
vector store and start from scratch. Okay. On that note, that is your introduction. Now we'll move into really cool, fun stuff with Mr. Ricky Perruccio. Turn off spotlight for me. Let's start spotlight, add spotlight for Ricky. Ricky, are you there? I can't hear you loud and clear. Let me stop my share.

Ricky Pirruccio - 7:36 PM
Yes near me.

Karim Lalani - 7:37 PM
Of course, it seems to me that the light test for representatives is deep. But sometimes it should work fine.

Colin McNamara - 7:37 PM
Oh, cool. I appreciate it. Sweet. Well, then let me make sure I do. Ricky, you want to share the screen? Your screen?

Ricky Pirruccio - 7:37 PM
Yeah.

Colin McNamara - 7:37 PM
I think my interface might be glitching a little bit. I cannot. But in the chat window, participate.

Ricky Pirruccio - 7:37 PM
Awesome. Well, all right, everyone. So, today I want to show you an application put together using a LinkStream template, Docker, and Streamlit. And I did this to kind of make a RAG application that can pull data from Google Drive folders. That's probably a place where many of you have...

Ricky Pirruccio - 7:38 PM
your data already stored. And so, let's dive in. So, the main purpose for even showing you this, it's kind of like a RAG tutorial, but I feel like it's more about showing you how you can manipulate a LinkStream...

Ricky Pirruccio - 7:38 PM
template and launch, like, Microsoft services with Streamlet and have a Docker, sorry, and have Streamlet as a front-end, and just do this all, like, seamlessly, super simple and, you know, with ease. So, I'm going to show you how you can interact with Linkserv and create FastAPI endpoints to basically...

Ricky Pirruccio - 7:39 PM
create invoker chains, and then how in Streamlet we create a front-end to interact with the chains, and how we use Docker to containerize our microservices and deploy them. And we'll see this in action, and then I also...

Ricky Pirruccio - 7:39 PM
I hope that you use this for your own need, and maybe, you know, change document loader, and maybe do it on a SharePoint drive. So, kind of a use case, at least a personal use case for myself. There's an echo. Okay. Sorry about that, guys. I should be able to fix that real quick.

Charles Martin - 7:40 PM
here in a cup

Ricky Pirruccio - 7:40 PM
Is there an echo? Okay. Yeah, I just changed the input of my mic. So, moving on. Personal use case, I work for a semiconductor manufacturing company. So, I work with a lot of design engineers. We're actually in the process of putting together

Scott Askinosie - 7:40 PM
Norco

Ricky Pirruccio - 7:40 PM
a knowledge base for our design engineers to kind of be consistent of how to design things. Because currently, it's very inconsistent. It's kind of an interesting thing to think about when you work in a large company, manufacturing. I was not very aware of it up until I joined here. And then I'm

Ricky Pirruccio - 7:40 PM
I do a lot of project management stuff. So if I could just, you know, record my meetings on teams and transcribe them, which I already can, and then just kind of have an LLM that can do rag with my transcript, that'd be awesome. And you can, you know, just doing that you can gather new insights for your projects. So many of you probably had the same use case.

Ricky Pirruccio - 7:41 PM
Okay, let's go in the lab. So technologies that we use here, we have Flankchain, OpenAI as our chat model, Streamlines as a front-end, Docker as our container, Orchestration Service, Google Drive, and then Chroma as a

vector store. Langserv is integrated with FastAPI, so that allows you

Ricky Pirruccio - 7:41 PM
to easily create endpoints for your chains. OpenAI here, we're also using OpenAI embeddings to embed our documents in our vector store. Chroma, we said, is the vector store we're using.

Ricky Pirruccio - 7:42 PM
You need for info on any of those technologies. So a prerequisite here and actually Colin, I did push up my updated slides, so if you want to merge that, I just made one slight change here in the credentials page. Awesome, thank you.

Colin McNamara - 7:42 PM
okay, I'll

Ricky Pirruccio - 7:42 PM
For credentials, you're going to have to create a Google service account and this slide, this defers a little bit from how the link chain docs show you how to do this in the Google Drive loader docs. So a service account basically allows you to use Docker to authenticate with the Google.

Ricky Pirruccio - 7:43 PM
Drive API. You can't otherwise do that. So Kareem actually made a great tutorial here to do this with lots of visuals and it's really great. Thank you for that Kareem. So you're basically going through this and you're

Ricky Pirruccio - 7:43 PM
going to get to this point where you create this JSON document, this JSON file that's going to have your authentication key. So Kareem does this a little different for in his tutorial. He basically, well he doesn't really use a key. He just has the email that

Ricky Pirruccio - 7:43 PM
he wants to share to whatever folder that he wants to share, but the way you do it with what I did is you would download this key, right, and you would create this directory right here, credentials, keys.json, and put this in

Ricky Pirruccio - 7:44 PM
your own computer, right, and then put this keys.json in that directory, and then you're going to set these environment variables right here, so Google application credentials, and you're going to put the directory as

Ricky Pirruccio - 7:44 PM
the value of this variable. So, very important, don't put the little squiggly here before, for pointing to your home directory, actually put the entire directory, or it's not going to work. Kind of a weird bug. Okay, so, and then...

Ricky Pirruccio - 7:45 PM
Open AI API key, if you don't adjust with that, go to open AI, get the key, and make it point to this variable. Docker desktop needs to be installed, of course, so use Docker. And if you want to clone this directory right here, which is our, you're on a certain link chain,

Ricky Pirruccio - 7:45 PM
repo, and then cd into the radchroma from Google Drive, you will be able to access this application. Okay, so live demo, we're actually just going to demonstrate how to use this.

Ricky Pirruccio - 7:46 PM
So we have a docker file, right? We have a docker-compose.yaml. You just do that, docker-compose up, build, and that basically launches your application. Refresh that, it's going to work, and I already have these folders right here. So this application is designed to dynamically create change.

Ricky Pirruccio - 7:46 PM
It's going to create endpoints for those chains. Those chains are stored into the endpoints in the backend, and then this application can basically keep creating chains and then save chat histories depending on which chain you're chatting with. So here I have an example. Oh, should probably open this in a new tab.

Ricky Pirruccio - 7:47 PM

Now, very important again, whenever you share your permissions for the folder, you need to have this anyone with the link selected. Otherwise, you won't be able to access this folder, you will just get an empty array, and it will be very confusing. So.

Ricky Pirruccio - 7:47 PM
That's how you handle that. Now, automatically, as soon as I have the folder here, this pops up right here, that tells me that the chain has been created. So we performed our RAG pipeline. And now we can query the chain. So what

Ricky Pirruccio - 7:48 PM
there you go. Awesome. And just to show you the dynamic nature of this application, here I have a Llama2 paper. I just put that folder ID here. This is kind of just a name, just to kind of tag the folder.

Ricky Pirruccio - 7:49 PM
What is this paper about?

Ricky Pirruccio - 7:49 PM
Yeah, I selected it. I think I have a PDF here. Maybe read the PDF. Hopefully you can read it. Let's see. What are LLMs? Okay.

Karim Lalani - 7:49 PM
I think that's the correct reading, because sometimes I think chat-GPT figures them out every day to a few logical digres rather than deep language patterns.

Ricky Pirruccio - 7:50 PM
Let's see. One more time. GPT-3.

Ricky Pirruccio - 7:50 PM
Okay. Yeah, it's working. Cool. And if you are kind of just tired of talking too long on two, you can go back to your state of the union and ask more questions about the president. Um, so that's kind of the gist of how it works. Um, now let's kind of show, um, how we set this up.

Ricky Pirruccio - 7:50 PM
So looking at the docker files and the docker compose, uh, we started to get a, uh, pretty good picture of how this, um, entire application works. Um, so you see that we have a backend microservice here for, with a docker file for our, uh, BlankChain, um, um, microservice, uh, with, uh,

Ricky Pirruccio - 7:51 PM
the FastAPI layer, um, and that is also creating our chains and mapping those two endpoints. Uh, then we have a Streamlit, um, docker file that, that creates our Streamlit application. And with docker compose, we can basically get those docker files and, uh,

Ricky Pirruccio - 7:51 PM
sort of just orchestrate the creation of our containers. Uh, so we can just do docker compose up like this, uh, like, like this right here. Docker compose up, dash dash build, and that creates your entire microservice. And this is also the same way you would do it if you were to launch this.

Ricky Pirruccio - 7:52 PM
Like AWS, like DigitalOcean, deploy it, uh, you know, to the worldwide web. So you just copy all these files that we have, um, whatever server, and then just do docker compose up build will save you so much time. Um, that's kind of where the ease of it comes in.

Ricky Pirruccio - 7:52 PM
So if we move on to how the backend is structured, we can, uh, we, we see that we have a server.py, um, file. And by the way, this is, this is, uh, straight from the template. Um, if you look at a template, it's like pretty straightforward. So we, these are the files that came with it, right? Server.py and then chain.py.

Ricky Pirruccio - 7:52 PM
These are basically the only two files you're really, um, interacting with, and this works for any template. Um, so the server.py is your FastAPI layer, right? This is what's, uh, handling, um, creating the routes and mapping those routes to the chains. Um, so I kind of modified this to be able to do lots of chains.

Ricky Pirruccio - 7:53 PM
Um, so the first thing I came with, uh, it was basically just to create one chain. Um, very simple to do. You just create like a post endpoint here, initialize chain. Um, and then what I did, uh, was for the, uh, chain creation part of it, I just wrapped that in a function so I could just use it in my initialized chain endpoint, um, uh, anytime I wanted.

Ricky Pirruccio - 7:53 PM
Just create a new chain and then add Rouse, that's the Langster part. That's what actually adds a chain, um, and creates another endpoint. So the endpoint that it creates, um, is, we'll look at the fast API schema here. So if we do local host.

Ricky Pirruccio - 7:54 PM
800, 1000, you will see here, um, the schema of all the endpoints we created. So where is this here? You should. Okay. So here you see our list folder routes, right? Uh, that's what, um, so that, that chain, that.

Ricky Pirruccio - 7:54 PM
Endpoint right there just gets all the routes of all the chains that we created. Initialized chain is the, is the route that we use to, uh, uh, create a chain. These right here are routes that we created, uh, that were created by Langserv. Um, so that's also using FastAPI. Um, so just think of it this way, every time you create a chain.

Ricky Pirruccio - 7:55 PM
Uh, with Langserv, right, it's going to create these, um, endpoints right here, uh, for that one chain. So here, it's really the, uh, yeah, it's, it's, these are the main ones. Um, it's like the invoke, the bash, the stream, uh, the stream log, the stream event. This is how you interact.

Ricky Pirruccio - 7:55 PM
With your, um, with your chains, right? So if you want to just, uh, basically invoke a chain, you just use this route right here, invoke, um, so here you have it right here. Uh, yeah, without the config patch. So these right here, um, and then, uh,

Ricky Pirruccio - 7:55 PM
we can move on. So just to kind of extend a little bit on that, on create chain, here you can see the Google Drive document loader, uh, that we have. Okay. Uh, the Google Drive loader, uh, that we actually substituted for the web-based loader that, uh, came

Ricky Pirruccio - 7:56 PM
with the template. So we really just modified this part of it on this file for the doc loader. Um, and then maybe then just kind of like tweak the, uh, text flitter, uh, params right here. Um, and you know, that's, that's kind of just like plug and play, um, based on the.

Ricky Pirruccio - 7:56 PM
Kind of files you have, um, and there's a lot that you could discuss about, you know, that, like how, what are the most optimal parameters, uh, to display your, uh, documents. Um, this tutorial, I think it's more, it's more focused on, uh, just how you do all of this, uh, with, you know, Docker, the template.

Ricky Pirruccio - 7:57 PM
Um, and streamline it. So you do all this, you get to the point, you create the chain, and then this goes back to your fast API. Um, and then we'll Langserv, uh, takes care of creating an endpoint. If you move on to the front end, uh, we just have one file. Yeah. Uh, we called that, I believe, streamline.

Ricky Pirruccio - 7:58 PM
So, to kind of break this apart, we start out with just having, defining some variables. We're going to have our API base, and for where to deploy this, you can just store this as an environment variable, and then you can just use whatever, uh, base URL you want for your API. Uh, and then I, I personally like to just have a variable for sc.sessionState,

Ricky Pirruccio - 7:58 PM
uh, you know, we're building a lot of chatbots here, um, and we're using Streamline a lot, and, uh, kind of just to make our code more modular, more readable. I think it's good, um, it's good, like, coding practice to just kind of, uh, not repeat yourself, um, and, uh, kind of package code, uh, you know, as much as you can, and reuse it.

Ricky Pirruccio - 7:59 PM

Uh, and then, uh, this right here initializes the state, um, and that's also good practice to initialize your state whenever you're using Streamline. I kind of think of it as, like, React, you know, when I'm, when I'm initializing Stata component, like, React kind of just, like, forces you to do that off the bat, um, and I just

Ricky Pirruccio - 7:59 PM
sort of love living in that world, um, and I think it's applicable here as well. You will run into less bugs if you do that, um, so then we have this, our application right here. You can basically condense it to this part of it, um, so we have initialized state, we have our chat manager right here, then we have this handle folder configs, that's basically our side.

Ricky Pirruccio - 8:00 PM
bar, uh, that's what's interacting with, uh, creating chains and, uh, uh, interacting with the backend part of it to, uh, create endpoints, um, and, um, basically, uh, map them to our chains. Um, then, yeah.

Ricky Pirruccio - 8:00 PM
Just keep going down here. We're using our chat manager to display the history, um, and to also, um, interact with the LLM. Uh, we have this function right here, get response from LLM, and that is going to interact with our, um, backend layer. So just to kind of put the.

Ricky Pirruccio - 8:01 PM
Pieces together with the backend and the frontend, um, you will have this get response from LLM. We're using that cache data, uh, decorator right here to cache the response. Um, I sort of like doing that in case, I guess, I don't know, maybe if you were to deploy this, uh, people are just keep asking the same questions. Um, you can just catch the response. Um,

Ricky Pirruccio - 8:01 PM
and, uh, here you have, get the folder routes part of it. So this, so after we create a chain, uh, we didn't, we want to be able to like kind of toggle back and forward between, uh, endpoints. Um, so that's what does, that's what this is doing right here. Um, and add a new folder config basically just creates a new chain and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and, and,

Charles Martin - 8:02 PM
Hey, Ricky, I actually had a question for you in the Q&A.

Karim Lalani - 8:03 PM
And I can add some original context to this. That is, OpenAI embed models, they just embed models trained with OpenAI, as opposed to there being many others that you can find that are enterprise OpenSource. It's just that OpenAI embeds models that are built with OpenAI, and you basically pay every time.

Colin McNamara - 8:04 PM
why don't you take it off into the next section?"

Karim Lalani - 8:04 PM
Yes of course. Let's start sharing my screen. I'm always amused, Colin, when you say doing a demo will lose sales. Consider what happened to me today. Well, no, I wanted to do a demo and lose sales.

Colin McNamara - 8:04 PM
Did the demo?

Karim Lalani - 8:04 PM
I lost a lot of sales, but luckily I had a contingency plan. We designed our labs to work in Google Colab. In general, through a free agency.

Karim Lalani - 8:05 PM
Most of the laboratories, 99% of the laboratories we work with, work without any hiccups. Mine usually don't work, and the reason for this is that most labs use ChatGPT, or GPT-4, which is basically just an application to OpenAI. And the labs I'm going to focus on our local electron,

Karim Lalani - 8:05 PM

where you work with a local inference engine, add and randomize open source models to memory Collabs. Google Collabs memory - it has enough memory to run most tasks, but sometimes depending on access

Colin McNamara - 8:07 PM
Karim, I think we lost you. You're back.

Karim Lalani - 8:08 PM
we work with images. The language model we're going to use this time is called Buck Lava, and it's a collaboration on a collaborative model name called Lava, L-L-A-V-A, which was one of the first open-source, multi-model language models to come out.

Karim Lalani - 8:08 PM
on stage a few months ago. Buck Lava simply uses the lava architecture, and delivers it to the territory of the misterl open source model, which appears in the name of the language. Let's move on to the laptop. Again, the source code is available at the Austen link.

Karim Lalani - 8:08 PM
Link chain 103, RAG, Olama, LavaDrive, RAG again because we're doing a rescue-rebound-generation demo, we're using a local language model with Olama inference burning locally, we're using a variation of the lava language model called Buck Lava , and we're relying on Google Drive to pull our artifacts from that.

Karim Lalani - 8:09 PM
Google Drive API on it so you can use this artifact with file artifacts in your application. Not only that, but if you're building something that has nothing to do with LLM, but needs to access your files programmatically, or access files programmatically from Google Drive, then you need to store that key.

Karim Lalani - 8:10 PM
Pay attention to... Okay, yes. Okay, yes. This project uses several librarians. Of course we use the code line.

Karim Lalani - 8:10 PM
Of course we use the code line. Of course we use the code line.

Karim Lalani - 8:11 PM
Of course we use the code line. We also use the MyGDrive line of code I made here. Since we are using OLAMA, or Local Model Language, there is one more step to take.

Karim Lalani - 8:11 PM
As I mentioned earlier, we use OLAMA inference. This is quite an interesting project. If you have a Mac or Linux machine, you can use quite a few locales, even on your laptop. On Windows, as I said, it will be released soon, but they have a report.

Karim Lalani - 8:12 PM
They use the Docker philosophy, where you have a model file where you can describe your language model, your prompts, your templates, and any parameters, and you can create new...

Karim Lalani - 8:12 PM
Derivative language models on top of existing ones already built here. And these are some that are available. If I wanted to use Mistral, like we did in our early labs, you see as long as you have Ullama installed, you use Ullama.pull.mistral and Ullama.run.mistral. Quite similar to the Docker syntax.

Karim Lalani - 8:13 PM
He tries to emulate. Let's go back. Actually, we are downloading the latest binary, we will make it use. We are starting this service, the Ullama service. This is what the API, which is called OpenAI control, will use.

Karim Lalani - 8:13 PM
This is for using models locally. Be able to make API sound to OpenAI, you will make sound to Ullama locally. When we started the service, in this case, we wanted to use the baklava model, as I mentioned, for our multimodel board. Let's get back to...

Karim Lalani - 8:13 PM

Streamlit code. There's... And I'll go through some of these sections. We collect... We collect certain information in this case to know when the application is ready to run. We are collecting Google Cloud... Code...

Karim Lalani - 8:14 PM
The file code that we collect, you know, goes here, service account, object. We... There is a jump here for the bind-and-run LLM that collects the payload. The solution for this is... Thus... Baklava-model, in order to use it, it was necessary to collect the file from the name.

Karim Lalani - 8:14 PM
And it collects the context from this file. First of all, when you use a regular document, a PDF file, or a text file, as in the early experiments, when you build a missile system, you take a rescue step, hold the necessary context, and you...

Karim Lalani - 8:15 PM
drag it into your model language, the model-prompt language. It's a similar move, but he does it with an image. It will withstand the weathered image and make it accessible to the language of the model. So when you actually weather the prompts, it will already be weathered from this image. It was...

Karim Lalani - 8:15 PM
OLAMA-cal, because OLAMA is an inference engineer and we can call... I was showing you some language models that support from abroad. You must specify which model you are referring to. And this model should already be designed locally, and for that we made OLAMA-pool-baklava-kal.

Karim Lalani - 8:16 PM
If you're building another application that doesn't use OLAMA-pool-baklava, but uses OLAMA-pool-baklava, you can replace that with OLAMA-pool-baklava. Prompt templates in this case directly. We use two prompts. One is to use the name, the other is just to look at the question we are asking.

Karim Lalani - 8:16 PM
We use the prompt that we use, and we use it to the Runnable that we made from this function that we saw a while ago.

Karim Lalani - 8:16 PM
See it as a unit. This is a module that you can use in the lang chain. You can use any function that takes parameters and matches text, and you can use it to Runnable.

Karim Lalani - 8:17 PM
You can use it in the lang chain. In this case, we use the payload. The payload stores the image and prompt. We use the image, we use the prompt, we save the image to the llm, and then we use the prompt to the llm so that it can…

Karim Lalani - 8:17 PM
respond to us with this context-preserving image. Everything you've seen in other labs remains. We create our messages. If there is no message, then we popularize the default one.

Karim Lalani - 8:18 PM
If it's a text message, we'll write it back on the screen as a chat message. If it's an image file, we'll render a small thumbnail of that image so you know the last message is an image. We have one more utility method. Sidbar has several...

Karim Lalani - 8:18 PM
...controls here. One is a utility file for an image file, if you want a utility to send something from your computer. The second is a file utility for Google Cloud credentials. That is, the credit that you will issue after creating your Google Cloud project. This gives you access to your Google Drive. And...

Karim Lalani - 8:18 PM
If you issue Google Cloud credentials, your project will allow all the files it has access to. This account you created has access to. It was the degree that Riki had experienced that was different in his labia than this one.

Karim Lalani - 8:19 PM

In his lab you support special foils, and here, because the account already has access to it, because when we created the keyboard, at the bottom you see a mortar where you create the foil that you want to share. Content with the fact that you create a public partition, you want to keep it.

Karim Lalani - 8:19 PM
If you want to save it, you can make it so that it is available only to this account programmatically, no one else can save it. And if you create a public share, anyone with access to it can use it. Thanks to the user, you can use one or the other, or a combination of both businesses. Since we added a service account,

Karim Lalani - 8:20 PM
when we use this call called Get Files Call then it will be possible to follow these files. This means that you can add this account to multiple Google Drive folders. You can add it to multiple Google Drive folders on different accounts, and all those files will be able to stick to those accounts.

Karim Lalani - 8:20 PM
As I said, with this one account. In this case, we only gave them access to one folder. We've already released some photos to jumpstart the process. Here, okay. And then, this last section is the most significant. We get a chat prompt. We add it to the messaging session.

Karim Lalani - 8:21 PM
If the file has not been released, we give a default message saying that the file should be released for the first time. If it has already released, then we pass it to the session with a prompt and with an image. If you remember, the session we created takes two parameters - a prompt and an image.

Karim Lalani - 8:21 PM
Yes, image and prompt. Yes, that's what we'll be releasing here. And that's basically the code.

Karim Lalani - 8:21 PM
We will learn in the demo. As I said, I was trying to release this locally on Google Cloud. Google was not cooperating today. And I ran into problems with a memoryless memory. But locally I released it here. Colin can you give me a visual or can you see it?

Colin McNamara - 8:22 PM
I can see it loud and clear, Raul.

Karim Lalani - 8:22 PM
Of course. Fine. Fine. I'll stick to the second one where you can release your Google Drive credentials. I have already released the credentials for this demo which has access to the folio. When I released the credentials, it started looking for all the files that account released.

Karim Lalani - 8:22 PM
And it found four of those files. Let's go with this man. When I call, it will emit that ping to the chat, and it keeps it in the session as well. And let's say, okay, describe this image.

Karim Lalani - 8:23 PM
It released this image, this queria, and this image to the lair model via Olam, and this is what we collected. This colorful design shows a bar scene where a group of people hang out. There is a wig on a field near the center of the scene, looking for ammo. Several people can be seen sitting and standing in various positions.

Karim Lalani - 8:23 PM
Maybe they enjoy the company of other people and sometimes at the bar. Several beers, cupcakes, and cubes are stored in this compartment. Emphasize the bar setting. The wig appears to be a part of the enterprise, adding a unique focus to the overall atmosphere. I believe the unique enterprise they are talking about is the Austin-Langqing meetup.

Karim Lalani - 8:24 PM
Let's go to the second time. Now I will give you a more specific question. What is the USA? Thank you for viewing!

Colin McNamara - 8:24 PM
I think that's correct. This picture represents the flag of the great state of Texas.

Karim Lalani - 8:24 PM
Again, I'm not doing... I know what we said about multimodel, and I'm using the bare minimum to demonstrate multimodel here. You can use this multimodel without transferring img. I turned to this to make it work only under imig transmissions.

Karim Lalani - 8:25 PM
This gives you a glimpse of the possibilities, not just from multimodel crayfish, but the image you can build with this framework, and how you can work locally. I believe there is a question...

Karim Lalani - 8:25 PM
This lab should work. We worked it several times at Google Colab. Today he works for Google Colab, but because of the crimes of the monuments, the lm is starting to distinguish numbers, not answers. But you should be able to work on a free instance of Google Colab. It currently runs on my personal hardware.

Karim Lalani - 8:25 PM
It runs on an Alienware testtop that runs on an NVIDIA 4090 graphics card. Olama, because it is designed to help work locally, the default model magnitude is 4G.

Karim Lalani - 8:26 PM
Which needs to be small enough to run on the biggest modern laptops, even on CPUs. Olama will allow you to work on the CPU, without NVIDIA graphics chips. It will just be smaller. Sorry to the little ones, not the smaller ones. We see, there are more comments.

Karim Lalani - 8:26 PM
I tried working on a radion, but again, I didn't have Olama at the time, so I don't know how Olama works on a radion setup, but it will be an interesting test.

Karim Lalani - 8:27 PM
I actually work in WSL-2. So even though the documentation isn't... Olama-documentation video seems like Windows is coming soon, but you can work in WSL-2 if you know how to break the service for NVIDIA.

Colin McNamara - 8:27 PM
Thank you for your work." Yeah, absolutely. Thank you so much. I want to make sure to share some in the meantime. Scott, good to see you with the camera on. Got a poll saying, is there anything with the presentation you want a deeper dive on? Please everyone, be sure to answer that really quick. You know what, I'll call on the line. Sure.

Colin McNamara - 8:28 PM
We do have some really interesting stuff come from Scott really quick. So our first poll that we had, where's my answer poll? Okay, apparently we can only show it when there's not a poll going. Okay, so again, thank you to our speaker, Graham, for sharing your learning and sharing your system. I know a lot of us have systems that we want to set up.

Colin McNamara - 8:28 PM
And are curious of how to run on the stuff locally. And it's really great to see you embracing learning open and sharing what you have going. Okay, I want to introduce now Dr. Scott Askinosie from I think Western Governors University. He's a member of our user group and he's going to show how we can

Scott Askinosie - 8:29 PM
All right. Can you guys see and or hear me? Anything? Okay, great, great. Just want to make sure. I can't see anything in my cell, so hopefully, hopefully I look okay. Ah, courage is a genius, man. This is a hard act to follow, but here it goes. So, thanks for sticking it out for my part of the conversation we're having today.

Scott Askinosie - 8:29 PM
Today, I'm going to be talking to you a little bit about setting up a method to speak with data, basically. So, it's called the Panda's DataFrame Agent, but essentially all we're doing is we're taking a data table and we're leveraging AI and LLM to be able to ask it questions. And you'll see a little bit more of this.

Scott Askinosie - 8:30 PM
As we go on, but before we get started, I'm not as clever as Colin with setting this up to automatically dump into your Google Drive. So I'm going to go ahead and share my screen, which is probably a good idea, so you guys can see what I'm about to show you. All right, can you guys see my screen? Great. Thanks, Colin.

Scott Askinosie - 8:30 PM
Starting by clicking on this right here, where it says Dataset located here, I'll go ahead and do it with you. It's going to say Leading Clolab, and there's a cute little corgi, go ahead and hit it. This is a Starbucks drink menu CSV. I thought it might be fun, because most people like Starbucks and coffee.

Scott Askinosie - 8:31 PM
If you're on a Mac, it'll go to your Downloads folder. If not, put it somewhere where it'll be easy to find, because we're going to need it here in a minute. Once you've got that downloaded, we should be good to go. If you're in Colab, I guess I should have said, if you clicked on the link in the agenda, you need to click on Open in Colab, so we should all be in the same spot, if you're playing along at home. If not, totally fine.

Scott Askinosie - 8:31 PM
Just hang out and have a watch. So, I'm a data scientist, and as a data scientist, I do a lot of what we call EDA. And even if you're not a data scientist, this is probably something that you do in some shape or form in your everyday life, whether you know it or not. If you are, you know, professionally working in a business, you often want to

Scott Askinosie - 8:32 PM
know about how your business is performing. Are you spending more money than you're making? Where is your money going? Where is your money coming from? What are your top-selling items? What was your best month of the year? How can you capitalize on your top-selling products? Exploratory data analysis is really important to answer those questions. And if you're a data scientist like me, or a data analyst, sometimes you need...

Scott Askinosie - 8:32 PM
to explore data you've never seen before. Which can be very daunting, but it can also be kind of exciting, because you can pull out insights that maybe no one else has seen before. Which, to me, is the fun of working with data. So, working with data requires that you have some understanding, generally, of some type of way to interact with the data. Whether it be Excel, or whether it be Python, or...

Scott Askinosie - 8:32 PM
If you're old like me, Deltagraph, SPSS, GraphPad Prism, these horrible, ancient programs... Nowadays, it's a lot easier, because you can literally talk to it. If you'd have told me six months ago that you'd be able to talk to a CSV, I would have thought you were crazy. But, here we are. I'm getting ready to show you. So, that's really the crux of what we're doing today. We're going to figure out, how do we talk to our data? What does it have to say?

Scott Askinosie - 8:33 PM
Before we get there, though, we need to install some components. A lot of these you've seen already today. A couple, probably not. So, LangChain. We're using LangChain because we're chaining together these different pieces, these different AI parts and agents. LangChain Experimental is where we're going to be pulling this Pandas DataFrame agent from, which is essentially a way to give the computer the vision to see a CSV file.

Scott Askinosie - 8:34 PM
And it run anyway, because hopefully you guys trust us at this point. And it's going to download our dependencies, which shouldn't take too long because these dependencies aren't that huge. This is a pretty simple process, believe it or not, because interacting with CSVs, CSVs are one of the most simple types of data file you can get. So, yeah, this took like 10 seconds. I thought I was going to have to talk longer. Thank goodness.

Scott Askinosie - 8:34 PM
Okay, so now that we have our libraries installed, we're going to start to pull things from those libraries so that we can set up the framework in order to be able to talk with our data. So we'll hit play here, and while that's running, I'm going to tell you guys a little bit about what we're downloading, or what we're actually importing into this Colab notebook. So the first is OS, and this just allows the notebook to interact with our operating system. We'll use this to grab our API.

Scott Askinosie - 8:35 PM
GetPass, if you are going to copy and paste your API key in, GetPass is great. I've got it in this notebook, I'll talk about it, but we don't have to use it because Colin already went through how to put the API key into your little keychain here. Pandas. Pandas is great. If you're working in Python as a data scientist, Pandas is the first thing you learn. It is a method.

Scott Askinosie - 8:35 PM
A library in which we can modify data to look differently, to look better for humans to understand, if that makes sense. It allows us to view data tables as tables instead of just text. It allows us to manipulate the data so that we can see it

from different angles to try and understand it. MapPlotLibrary, it's a way for us to graph stuff, so we can make charts.

Scott Askinosie - 8:35 PM
Figures, stuff like that. Seaborn is another even more better graphing utility library in my opinion. It allows us to do things that are a little bit more advanced and it's really good at taking directions. But we really need them both, so that's why they're both here. LangChainExperimentalAgents, again, this is where we're going to get the DataFrame, Panda's DataFrame agent that's going to

Scott Askinosie - 8:36 PM
allow us to interact with the LLM, and we're going to talk about this in a second, the Python kernel that's going to actually execute the code, it's going to allow us to look at our data. And then, LangChainExperimentalAgentType is going to allow us to access our chat history. So, as we're asking the LLM more questions, the agent is going to get access to things that we talked about previously, so it has a little more context.

Scott Askinosie - 8:36 PM
TypingExtensions, it helps the agents do, I guess, the predicted text a little bit better. I'm not 100% sure. I read a little bit about this, but it was a bit over my head. And then, of course, OpenAI. This is the LLM we're going to use. I like OpenAI because it's easy. Kareem is awesome with the local LLMs, which I'm very envious of.

Scott Askinosie - 8:37 PM
Because, ultimately, I really want to do that, but it's still a little bit out of my skill level at this point. Okay, so this next cell I'm not going to use because I already have my key in here. But if you do not have your key stored in this little keychain here, and you want to do this with us, you can uncomment these two little hashtags. So you just delete

Scott Askinosie - 8:37 PM
them and run this cell and it will give you a little window that you can paste your open AI key into. So if you aren't doing this now and you want to do it tomorrow, that's what I have commented out here. Just delete these, you can pop that key in. But if you've got your key saved in your keychain, you should be able to hit this and grant it access. Hopefully it works. Okay.

Scott Askinosie - 8:38 PM
Now we come to the file itself. So I put this cell in here. We're not going to use it right now because, again, I wasn't able to create the fancy code that was going to download, put it in your Google Drive, and then pull it into this notebook. But if you had a CSV or if you currently have a CSV in your notebook, you want to ask questions, hit play. It's going to connect to your drive. Hit yes.

Scott Askinosie - 8:38 PM
Go ahead and hit your name, and then continue. And like Colin showed us before, click the little folder icon, click view. Okay, well, it says it's mounted but I don't see it. Oh, there it is. I just am impatient. Okay, yeah. So you can thumb through your drive.

Scott Askinosie - 8:38 PM
And then you can mount the file here, which I'll show you right now. So, because I wasn't able to do that, we're going to load the file a different way. If you click on this little upload arrow here, you can migrate to your downloads. And, let me just do this. Where is it? Let's go.

Scott Askinosie - 8:39 PM
Update padded. There we go. So, this is the Starbucks drink menu, and it's going to say that as soon as you close this Colab notebook, this is going to be deleted. That's fine. We're okay with that. So, now you should have this menu right here. What we're going to do is we're going to click this little down arrow, and we're going to copy the path to it. Because what we want to do in order to be able to access …

Scott Askinosie - 8:40 PM
And now this check tells you that, okay, this thing called DF is the Starbucks menu. Okay, so now we've got our data frame ready to go. This is the data that we want to look at. So, now we're going to create our chain. We have to create the ability for your data to be seen by the LLM, and the agent to be able to tell the LLM what you want to know. So, I'm going to walk through this really quick, and then I'm going to …

Scott Askinosie - 8:40 PM
draw exactly what's happening here. So, we are going to assign our agent. This variable, agent, is going to be the agent that's going to be helping us look at our data. So, we're going to create this pandas data frame agent, which is

going to use pandas in the Python library to be able to look at our CSV. And the first thing we're going to do, you saw this with Colin, we're going to load our model, which is GPT-345-Turbo, because it's cheap.

Scott Askinosie - 8:41 PM
And it's fast, it's easy. And then the temperature, we're going to have it zero, because we're working with data here. The temperature dictates what probability of token the AI will return. So, what does that mean? Colin mentioned creativity, and that's true. When we set the temperature to zero, it means it's going to give us the token, or the key, with the highest probability. That means it's going to be as close to ground truth as possible.

Scott Askinosie - 8:41 PM
If we increase this temperature, it's going to give us the lower probability tokens, which means it's going to be more creative. So, the difference between temperature being zero, it's going to tell us the truth. Temperature, 10, whatever, I'm not sure what it goes to, it's going to tell us a lie. That's the gist of it. We're also going to load in our df, our data frame. So this is the data. We're telling it, here is the data I want you to look at.

Scott Askinosie - 8:41 PM
And there are ways to load multiple data frames, but it gets a little messy, and I'm not that good at it yet, but maybe next time. Here, verbose equals true, we're going to see this in a minute. This I left as true because it's going to tell us what it's doing. The verbose means every step it's going to literally tell us what it did and what's happening.

Scott Askinosie - 8:42 PM
I'm going to change this to false if you don't want to see that. It's up to you. This is really, really important. The agent executor keyword arguments. So we want our agent to handle parsing errors for us. Basically what that means is, if we pass a command to the agent, which we'll see in a second, it's going to tell, and our agent is kind of like our go-between, between the LLM and the parser.

Scott Askinosie - 8:42 PM
And the Python kernel, which is going to be running the code. So the agent is going to tell the LLM what we want, and it's going to go back to the agent. And if the LLM didn't understand, and it gives the agent an error, we want the agent to be able to handle that. We'll talk about that in one second. The last thing is, we are...

Scott Askinosie - 8:43 PM
going to be using OpenAI functions. So this is the tool within the OpenAI agent that we want to be able to hit. Okay, so before we move on, I want to tell you guys exactly what's happening here. What we're doing is, we are...

Scott Askinosie - 8:43 PM
the user. And we are going to pass in commands to our agent that we just created. Because I kept talking about agent, agent, agent. The agent is kind of like our friend. He's the go-between. He's going to be taking what we say and making sense of it, to pass it on to the LLM. Which in this case is going to be OpenAI. GPT-Turbo 3.5.

Scott Askinosie - 8:44 PM
So we're going to issue a command, or a question. The agent is going to reason through that, and say, okay, I need to create a prompt that I'm going to send to the LLM, so that the LLM can figure out how to generate the code that we need in order to get the result that we're looking for. So the LLM is going to think about that. And it's going to send the code back.

Scott Askinosie - 8:44 PM
to the agent. And this is what I was talking about with that keyword argument, that quark. If the LLM doesn't understand what the agent said, and it sends back an error, if the agent is not allowed to handle that error, it's just going to say, I don't know, man, he told me he didn't know what you were talking about, sorry. And then that's it. The code does. But what we've done is, we've allowed the agent to say, okay, wait, wait, wait, wait, no, no, no, that's not what I meant.

Scott Askinosie - 8:44 PM
Try this. So the agent will actually send something back, hoping that the LLM will send it back the right code. And this will happen until it does, or until the agent gives up, which I haven't had happen yet. Okay, so the LLM sends back the code to the agent, and the agent is going to send it to our REPL kernel, which is essentially

Scott Askinosie - 8:45 PM
what is running the code, the Pythonic code that we need, in order to get what we want from the data. The REPL kernel is going to run that Python code, and it's going to send it back to the agent, and say, okay, I got your, I finished

your code. Here's what I got. And the agent is going to look at it and say, okay, okay, this makes sense. I'm going to put this into a form that this dumb human can understand, and it's going to send it back to us. So that's what's happening.

Scott Askinosie - 8:45 PM
In a nutshell. Pretty straightforward. Okay. See, we're running out of time. I've talked too much. Okay. So let's see this in action. Very first thing I'm going to do is I'm going to show you what this data frame looks like in Python. So if you recall df.head, what that does is it shows us the head of this data frame. It's going to show us the first five rows of this data frame. We can see we have a bunch of different categories.

Scott Askinosie - 8:46 PM
Or features, as we call them in data science. We've got beverage category, beverage preparation. What size is it? How many calories, total fat, yada, yada, yada, yada. So what if we want to do this using plain English? If you don't know df.head, you're not a Python person. Well, you could just say, hey, agent, show me the first five rows of the data. Oh, I forgot to run the cell. I'm not running any of these cells. Okay.

Scott Askinosie - 8:46 PM
We got to first run the cell with our agent. Hopefully you guys did that. All right, let's try this one more time. Okay, so the agent passes that to the LLM, and it sends us back this. Which, I don't know about you guys, this is kind of hard to look at. But if we scroll down here, this is a little bit better. So this is what...

Scott Askinosie - 8:47 PM
Let's see, this is what was sent to the agent, and the agent made it look a little bit prettier. So I'm going to ask the agent to tell the LLM to show us the first five rows again, but this time let's make it look pretty, and let's see what we get back. So if you look this time, what the agent received from the LLM was a lot prettier.

Scott Askinosie - 8:47 PM
And it did about the same thing. It's like, oh yeah, that looks great. So the table is formatted for better readability. It knew what we wanted, just by giving it very obscure human terms. And even this, when we ask to make it pretty, looks better than, well, maybe about the same as what the agent did. Maybe I like the agents better. I don't know. Okay, so let's try something else. So let's say we want to know how many columns and rows...

Scott Askinosie - 8:47 PM
... our data frame consists of. We can ask it. How many rows or columns does our data set contain? You could also say, what's the size of our data set? And it's going to tell you, in plain English, the data set contains 242 rows and 18 columns. The way we do this in Python, we just do df.shape, and it'll tell us this. But again, that's not in English. That's something you'd have to know Python to understand, generally speaking.

Scott Askinosie - 8:48 PM
Okay, so let's ask it something else. Let's look at a list of beverages, beverage categories in the data set. We can ask it, or I'm sorry, here, this is the Python. I want to ask the agent, what are the categories of drinks in this data frame? It'll then grab those for you. Likewise, the way we do it in Python is we ask the data frame to give us the unique values in the beverage category.

Scott Askinosie - 8:48 PM
And it'll spit that out. Whereas here, this time I like the agent's view. I like this a little bit better. It takes what the LLM gave it, and it makes it a little bit more palatable to humans. Okay, so now let's try something a little more complex. We want to be really astute data people here, so we're going to ask this for a heat map.

Scott Askinosie - 8:49 PM
And I'm kind of jumping to the good stuff in my opinion, because I don't want to keep you guys. But a heat map is a graph, essentially, that allows you to see correlations between numeric values. I'm going to show this to you, and I'm going to say some interesting stuff about it. So we ask it for a heat map, and it gives us this beautiful figure. I mean, this is gorgeous. If I were to have done this in Python...

Scott Askinosie - 8:49 PM
it would have given me an error. It would have said, cannot enumerate something, something. Basically, this data frame is full of text, and Python can't deal with text when it comes to making correlations, because correlations can only occur between numbers. So effectively what this did was, it knew we wanted a heat map, so it knew

Scott Askinosie - 8:50 PM

it needed to exclude all the text and just include the numeric values. And there's some interesting data here. So, we see here, the red is at 1, and the blue is at 0. Correlation tells us how closely are two data points related to each other. If they're highly correlated, that means they have a higher score. If they're lower correlated, that means they have a lower score. So what does that mean? Well...

Scott Askinosie - 8:50 PM
If it's cloudy, will it rain? Clouds and rain have a higher correlation than sunshine and rain. It does rain sometimes when there aren't clouds, which is kind of weird, but most of the time you have to have clouds to have rain. Therefore, rain and clouds are highly correlated. Low correlated things, like, for instance, somebody being left-handed and right-handed. If you're left-handed, it is very...

Scott Askinosie - 8:50 PM
lowly correlated to being right-handed because very few people are ambidextrous. So we can look at this and say, okay, what things are highly correlated? Here's an example. Cholesterol has a score of 0.94 with calories, which is interesting because you generally think of, like, sugar and carbohydrates as being highly correlated with calories, which they are, but not as high as cholesterol.

Scott Askinosie - 8:51 PM
This is something that most people would generally think. I thought it, too, when I first looked at this table. Then I remembered, oh, well, fats have a higher calorie content than sugar, so therefore cholesterol, which is a fat, has a higher correlation with calories. So there's a lot of cool stuff to pull from this heat map. I won't spend too much time on it because I want to show you guys one more thing, one of the things that Colin was talking about. We can also use this to create different graphs.

Scott Askinosie - 8:51 PM
So let's ask it to show us a bar chart of the top 10 most caloric drinks at Starbucks. We hit play, and it's going to make this graph for us, and this graph is really pretty. To do this in Python would have taken me probably 10 or 15 minutes, and I'm not bad at Python. I'm not an expert, but I'm not terrible at it.

Scott Askinosie - 8:52 PM
This is way faster, and we can see that it very, very beautifully labels everything. We have a table label. We have a y-axis label. We have an x-axis label. All of these look very smooth and pretty. I'm so proud of our agent. He did such a good job training him. Okay, there's other things you can do too, which I will just breeze through because this next thing...

Scott Askinosie - 8:52 PM
It's statistics. You can do these little box and whisker graphs, which are really, really fun because they show us all these different standard deviations from the mean. I had a couple other things, but it looks like I deleted them. So we get to the very end here. Thanks for hanging out. I think we all did great, as our little agent did as well.

Scott Askinosie - 8:53 PM
You can keep this notebook, and if you copy and paste what I've got here, agent.invoke, all you need to do is put whatever you want to know in these parentheses, and it'll spit it out. And you can ask this thing some really interesting stuff, because not only does this have your data for context, it has the entire internet as context. So you could say, actually, let's try this. Sorry, I know we're at time.

Colin McNamara - 8:53 PM
We don't have a hard stop at 9. It's just kind of a general guideline. You know, this is the Internet. We're all pretty much at home. So, I love the whisker charts above, you know. Amazing. Establishing statistical process control of manufacturing processes, or in my last case, global cloud providers.

Scott Askinosie - 8:53 PM
Oh, man, yeah, I could talk about these for a long time.

Colin McNamara - 8:54 PM
You know, being able to do this within LLM, using data that's coming in from your systems, combining that with, like, email. I will say that there are software providers that put seven-figure dollar signs on presenting data back to you like this.

Scott Askinosie - 8:54 PM

Oh, man, that does sound really nice, doesn't it? That's a very good number. All right, that's it. That's what we're doing, guys. Everybody in the room, we're a new business. Yeah. Yeah, right. Yeah, exactly. Awesome. Yeah, I do. Let's try this really quick. Thanks, Karim. So, let's see. What's the healthiest drink at Starbucks?

Charles Martin - 8:54 PM
Scott will be happy with six, right?"

Karim Lalani - 8:54 PM
We have some questions in the Q&A, Scott, if you want to get them.

Scott Askinosie - 8:55 PM
And then I might have to tell it. There it is. All right. A short Taz-O-Tea. Because it's basically water, according to this. Okay. Yeah, that's pretty cool. All right. So, let's see here. Let's see here. Where are we at? Q&A.

Scott Askinosie - 8:55 PM
Oh, great question. That's in Colab. Yeah. So, the REPL is part of the Colab notebook. You can run this in JupyterLab also, as Karim showed us earlier. In fact, I run everything in JupyterLab. I do everything locally. I use Colab, like, if I need to train a model or something like that because of the cheap GPUs. I haven't quite figured out how to use the GPUs on my MacBook yet. I just haven't wanted to take the time, and this is really easy.

Karim Lalani - 8:56 PM
I mean, JupyterLab and Colab are replicators themselves, because that's what's happening. You write codes in your text barrel and it executes it for you and that's what the agent does. LLM corresponds to Python code. If you look at the green text, that's where the code is. That's the Python code right there on the first line. If you look at one of the charts that you…

Scott Askinosie - 8:56 PM
Yeah, right.

Karim Lalani - 8:56 PM
You can see the code right there. If you copy it, that's the code right there. All. So. But... But that's just what happens. Your agent asks a question...

Karim Lalani - 8:56 PM
If you see what the Pandas DataFrame agent is doing, the system is asking that whatever question I ask, check in pandas query and do it against this dataframe object that I have. So this returns the python code for the pandas dataframe that Ripple executes on your build.

Karim Lalani - 8:57 PM
So it's like Google, not Google, yeah, copylite, GitHub-copylite. So what you're seeing here is basically a type of copylite for you, but you're not only using it to generate code, you're also using it with Repl to execute the code right away to get the response.

Scott Askinosie - 8:57 PM
There it is. Straight from the mouth of the master. Okay. Yeah, so that's a good question. The database one or the PhD one? Okay. Okay, so I did an NLP project last year that was horrible. So I was dealing with a lot.

Colin McNamara - 8:57 PM
Hey.

Karim Lalani - 8:58 PM
Let's do the database one. I think we should keep the best one for the last.

Scott Askinosie - 8:58 PM
sleep issues with my infant's soon-to-be toddler and she just wouldn't sleep. So I was reading a lot of like different sleep tactics for kids and realized that there were like two really divided camps. Sleep trainers versus co-sleepers. And I did an NLP classification of these two in Reddit. So I web scraped Reddit threads that had these two different camps in it. And I did this NLP classification.

Scott Askinosie - 8:58 PM

And I did this horrible, horrible chart of the most commonly used words in both camps. And I assigned them like different weights based on their use. And it took me like four or five hours to do. And I just was so, so sick of it by the time I finished. I wish I'd never started it.

Scott Askinosie - 8:59 PM
It was part of a larger report that I was doing. And yeah, it still haunts me. If I thought I could pull it up quickly, I would. But that's definitely the worst one that I've ever done. And I bet this could do it in a second. Like if I loaded the data and did the tokenization, I bet I could just tell it and it would do it for me in, you know, five milliseconds. So it's kind of a hard pill to swallow.

Charles Martin - 8:59 PM
Scott, I think there's a blog there. I think there's a blog there about, like, a dad, a data scientist being a dad. Like, all of the data science you use to be a better dad. That's a really interesting story.

Scott Askinosie - 8:59 PM
Oh, man. Yeah, it was very real to me at the time, that struggle. Okay. Yeah. So thanks for the question. Check out my LinkedIn. I'm pretty sure I still have it all up there. I have... Oh, no, check out my GitHub. I've got my...

Scott Askinosie - 8:59 PM
My master's and my Ph.D. dissertation. They're not over data science. My master's was over the effects of electromagnetic resonance on DNA repair and aging. And my Ph.D. was over integration of light into biological systems. Like, how do cells perceive, interpret, and respond to light? So they're a little bit different.

Karim Lalani - 9:00 PM
We would definitely need to use an agent to interpret to translate that in layman's term for us.

Colin McNamara - 9:00 PM
I think that's a really great briefcase, maybe something we can see. I really do have to thank you, Scott, for joining us and, you know, learning those. Oh, yeah, yeah. Yeah, that was great. You know, the heat maps, the whisker charts, you know, these type of things are essential to running a large infrastructure, running supply chain, running software supply chain.

Ricky Pirruccio - 9:00 PM
Call Grace.

Colin McNamara - 9:00 PM
Creating things. So we have a poll for what we want a deeper dive on. I do want to, is there anything in the presentation you want a deeper dive on? I do want to go ahead and do a readback of some of the polls that we have. So everyone type it in really quick so we can get that out. We'll give it a 5, 4, 3, 2, 1. 3, 2, 1, 5, 4.

Colin McNamara - 9:01 PM
Okay, cool. Let's go into our polls here and this one's in progress. We're going to end the poll real quick and do some readbacks. So at the start of our session here, we had a poll of, what are you trying to do with LLMs? We got some really interesting diversity of what we're looking. So future projects, document generation. A lot of us run document driven businesses.

Colin McNamara - 9:01 PM
Someone's doing everything. Someone's trying to take over the universe with their infinity stones. People trying to help companies have better communication. That sounds like Charles. The key change here. The key of being the key of change. Reinventing the world of finance is really interesting. Risk mitigation and fostering community. Hi. So helping communities build their own micro LLMs. Some people are still figuring it out. I'm there too.

Colin McNamara - 9:02 PM
We have personal productivity and designing a solution to assess users on Q&A tasks. And improving keyword search. The other goals that we have are two deeper dives. So the results of that were we have a request for some more Langsmith deep dives.

Colin McNamara - 9:02 PM
I'll do a quick one right here, actually. Let me share my screen just for a moment. Share my screen. Entire screen. This screen. So in the first run sequence that we had, so this is Langsmith. My default project. Project's default is kind of default if you don't have anything set up.

Colin McNamara - 9:03 PM
We should do a deep dive inside of it, but you can actually see some really interesting stuff, right? We can see the sequence. We can see the latency for each step. If we have data sets, we can tie it back. You can actually get the cost that running this. So this cost me not a lot of money to run each time. If you dig down, let's say Bitcoin.

Scott Askinosie - 9:03 PM
I think that's in Bitcoin.

Colin McNamara - 9:03 PM
That's a million dollars a year. So we passed it. So if you remember in the original question, we asked, what did Ricky say about caching? We can see that actually retrieved this context from that entire file, which is rather large. And it retrieved just that context. We can see whether where it passed it up into

Colin McNamara - 9:04 PM
AI. And we see the response back from the LLM. Langsmith is really, really cool. If you don't have access to it, you know, DM me, DM Harrison, we have some services. It is a really, really, really cool tool. There is all sorts of really cool stuff. One thing that's also really neat is integrates with BlankChain Hub. So if you want to go through

Colin McNamara - 9:04 PM
and you're looking for some agent code, boom, you know, let's look at the prompts for a super agent. It's really, really cool. You can basically put data sets up there where you can look at performance over time. So in this case, we changed our code. We can go through the independent run or the individual run so we can see what was actually passed up to LLM. It's super neat.

Colin McNamara - 9:04 PM
We'll definitely, you know, look at adding a deeper dive into it. We had a local LLM architectures and then running a LLM in AWS, and Kareem, that is your strong point. And then more streamlined lane chain use cases. I think, Ricky, you were interested in doing a lane graph example that kind of tied that together.

Colin McNamara - 9:05 PM
Cool. And then our last poll. Let me clear the result. And then our last poll. Is there anything from this presentation you wanted a deeper dive on? Let's show the results on the screen. Okay. Lane chain agents and lane graph. Oddly enough, that is the topic of our next in-person meeting on 2.21. Oh, I like the fireworks that just went up.

Colin McNamara - 9:05 PM
Behind Kareem there. We will, we are working on, we're going to do a rough cut on 2.21 with some whiteboards and a bunch of the great minds we have access to here in Austin. If you're around, please, please come on by. And then we'll be putting some polish on it and sharing it around here on our virtual meeting. We see next we have more and more Lane Smith. So maybe we should, you know,

Colin McNamara - 9:06 PM
in a LandGraph session, pipe back into Lane Smith and type together. Docker and multimodal, multimodal models locally. Yeah. That's kind of what Kareem showed you a little bit. But maybe we can show like mapping GPUs and stuff like that. You learn Docker and Kubernetes. And then love the edge cases. One thing that I really liked about this was we're able to get everyone's perspectives. We got.

Colin McNamara - 9:06 PM
Ricky from manufacturing engineering perspective, right. And building applications that, you know, save money and make better products come out there. We have Kareem from where he brought his perspectives on using LLMs in entirely disconnected settings, using local LLMs that you can tune yourself, that you can try out different features and functions. I really love Scott's perspective.

Colin McNamara - 9:07 PM
Everyone presented extraordinarily well. You know, I learned so much. The specific graphing use cases and the ability to speak in human language into that LLM and have it translate to code and then execute the code to present that back. I think it was a really great example of the power of these large language models of this magic middleware chain that we're playing with.

Colin McNamara - 9:07 PM
And the ability that save you time and create an intellectual property that can be rerun over and over and over again. You know, if I think to my own business, you know, this notion of, you know, Sam Altman put a tweet out recently that

there's going to be a single person, billion dollar company, right? And at the core of that is those of us that are using these AI middleware.

Colin McNamara - 9:07 PM
To create agents, to create applications, we're effectively creating employees. We're creating employees that can execute logical constructs, that can run runbooks, that can give us feedback, that can offload automations for our business or for our tasks. And, you know, I think what we're seeing in this user group, as we learn together in the open, as we share what we're coming across, as we support each other.

Colin McNamara - 9:08 PM
You know, I'm seeing myself that I'm learning more every day and that I feel comforted and supported by the community around us. So, I want to thank everyone for participating. Also, Charles, thanks for the quiet, silent work on the back end, executing pulls, making magic happen. Charles is really good at kind of tapping the rudder and, you know, he's definitely a muse back there.

Colin McNamara - 9:08 PM
Encourage everyone to reach out and say hi to him, too. On that note, I want to thank everyone for joining and for those of us that are checking us out on YouTube, you know, give us some feedback, you know, like, comment, subscribe, I guess we're supposed to say, but at the end of the day, hop on down to our Discord and join the fun, right? Let's learn together. This is all moving so fast. I'm really thankful for everyone's time. And if that, if anyone has

Colin McNamara - 9:09 PM
anything else to say, raise your hand. No? Okay, well, I'm gonna go ahead and call it. It's 909 p.m. You have an awesome day, and we'll see you next time talking and playing with Langraff and integrating some Langsmith into it. Cool. Okay, good night, everyone.

Scott Askinosie - 9:09 PM
Thanks everybody.