envision
VIRTUAL SUMMIT SERIES
DATA & ANALYTICS

Zero to LLM

Getting Started with Language Models in the Lakehouse

Monday, September 11, 2023

PRESENTED BY 3Cloud

Meet the Speakers

Victoria Austin

Penn Sefton

envision

©2023 3Cloud

Agenda

01 The Lakehouse Paradigm

02 Getting Started with LLMs

03 Technical Demo

04 Wrap Up

envision

©2023 3Cloud

Comparison of Data Platforms

Data Warehouses	Data Lakes
BI, SQL Analytics	ML, AI, Data Streaming
☐ Very adaptable	☑ Very adaptable
☐ Unstructured data	☑ Unstructured data
☐ File-level governance	☑ File-level governance
☐ Data science and streaming-based ecosystem	☑ Data science and streaming-based ecosystem
☑ Structured data	☑ Structured data
☑ Highly reliable	☐ Highly reliable
☑ Fine-grained governance	☐ Fine-grained governance
☑ SQL-based ecosystem	☐ SQL-based ecosystem

envision

©2023 3Cloud

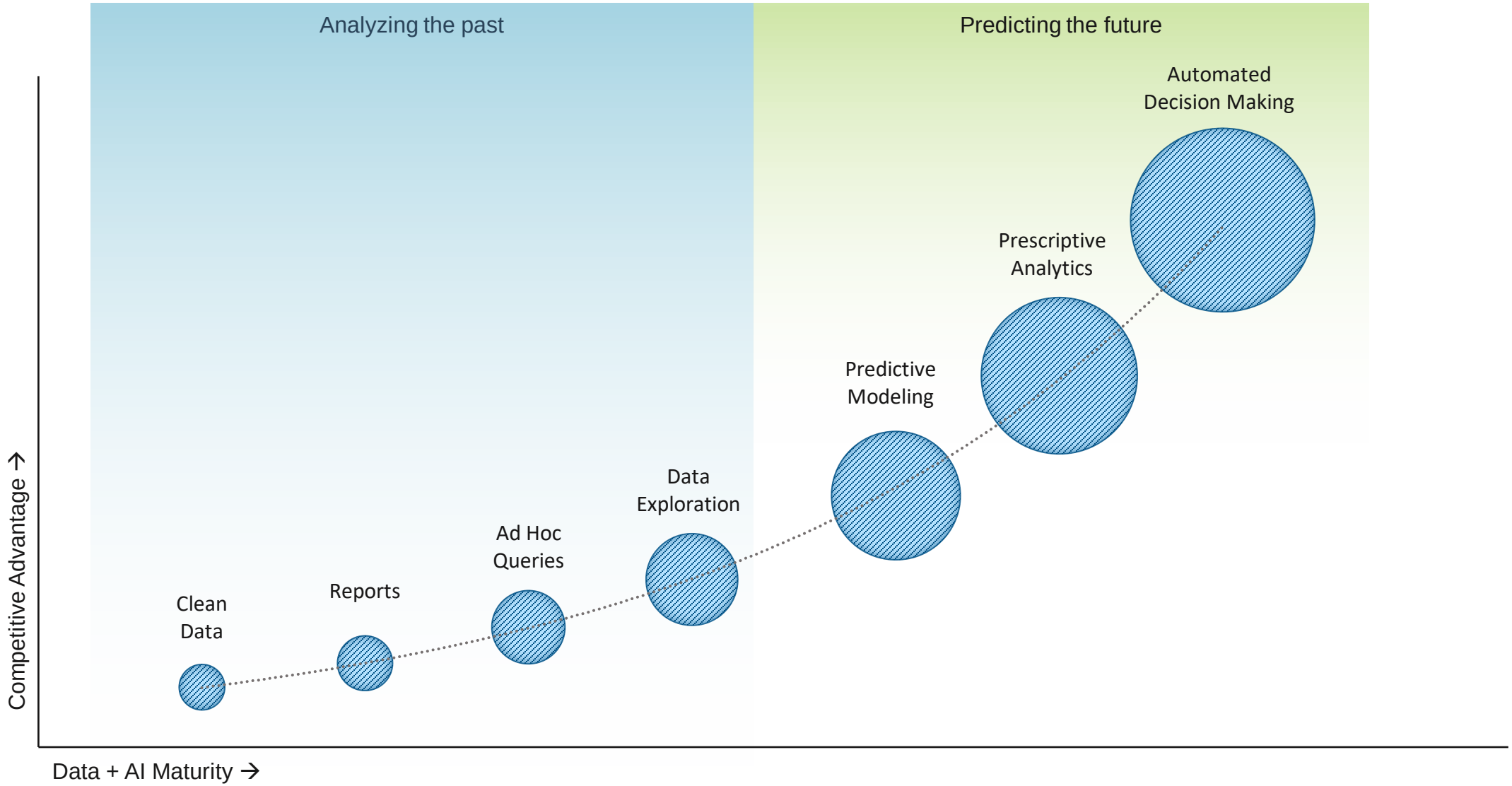
The Lakehouse Paradigm

Data Warehouses	Data Lakes	Data Lakehouses
BI, SQL Analytics	ML, AI, Data Streaming	Any Analytics Workload
☐ Very adaptable	☑ Very adaptable	☑ Very adaptable
☐ Structured and unstructured data	☑ Structured and unstructured data	☑ Structured and unstructured data
☐ File-level governance	☑ File-level governance	☑ File-level governance
☐ Data science and streaming-based ecosystem	☑ Data science and streaming-based ecosystem	☑ Data science and streaming-based ecosystem
☑ Structured data	☑ Structured data	☑ Structured data
☑ Highly reliable	☐ Highly reliable	☑ Highly reliable
☑ Fine-grained governance	☐ Fine-grained governance	☑ Fine-grained governance
☑ SQL-based ecosystem	☐ SQL-based ecosystem	☑ SQL-based ecosystem

envision

©2023 3Cloud

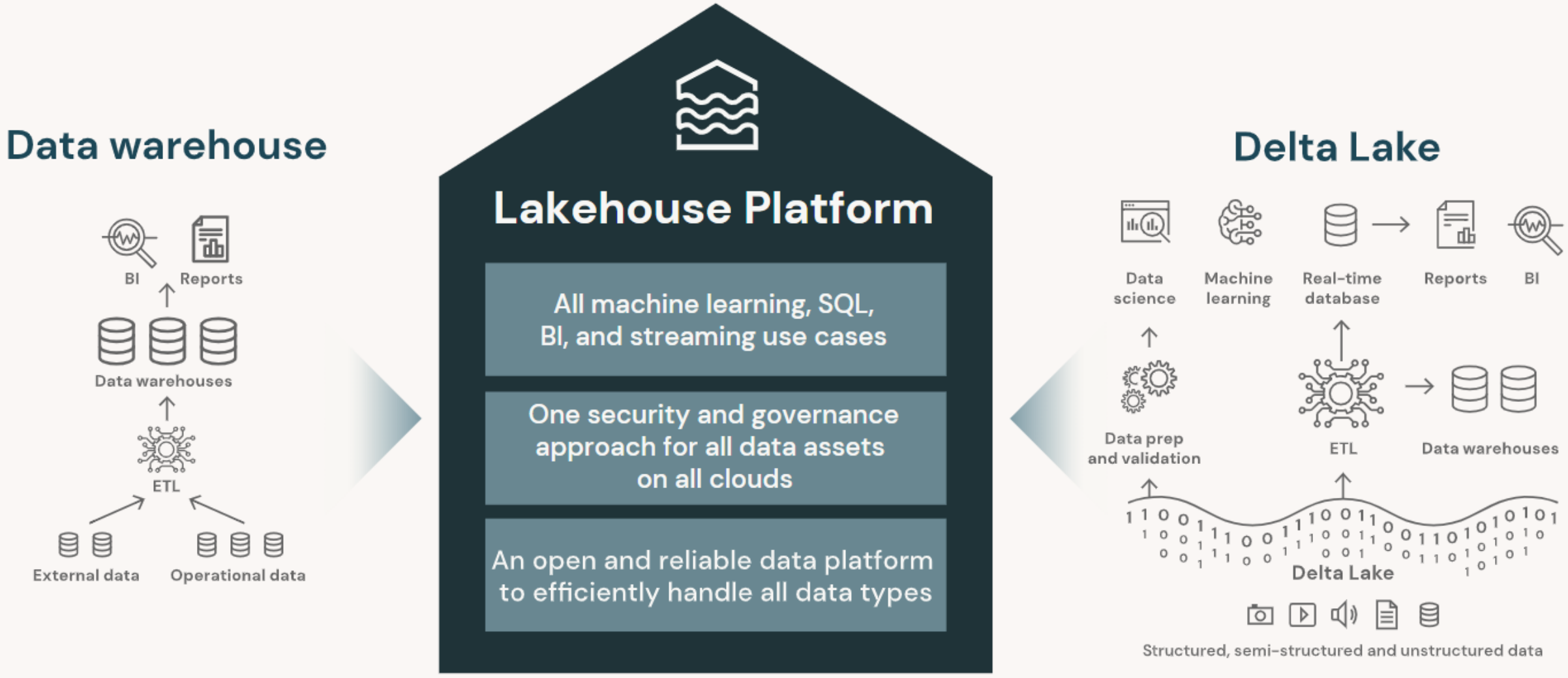
Lakehouse: The Competitive Advantage



envision

©2023 3Cloud

Lakehouse: The Ideal Architecture for AI/ML



Generative AI in the Lakehouse

How do I prove the value of LLMs?

Determine Use Case

Pick from a wide range of LLM use cases to start with

Map Design Steps

Determine all necessary steps to design the solution end-to-end

Build a Proof of Concept

Show that the build is possible through a minimum viable product

This is easily done on the Lakehouse!

Key Considerations

Impact cost, complexity, and value-add of your LLM implementation

Open source vs. LLM as a service

What are you willing to spend?
What level of ownership do you need over the solution?
Are you comfortable with your data changing hands?
What type of volume are you expecting when your LLM solution is deployed?

Customized vs. Off the shelf

Is your task too complicated to be accurately handled by a large pretrained LLM?
Are there any existing LLMs that understand your domain?
Do you need to provide more context or data than is capable through a simple prompt?

Hands-off automation vs. Human in the loop

What are your expectations of accuracy?
What is the cost of incorrect output by the LLM?
Can your workflow be augmented by an LLM as opposed to completely automated?

envision

©2023 3Cloud

So, where do I start?

Simple! Then, add complexity as needed.

Prompt

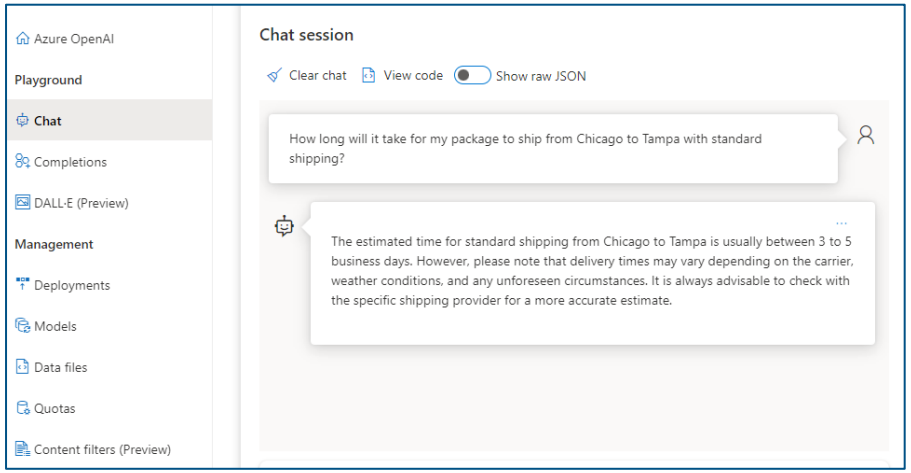
Input text, questions or information to communicate to the LLM what response you're looking for.

Azure OpenAI Python SDK

```python
import openai
from llmenv.azureopenai import api_key, api_base, deployment_name

openai.api_key = api_key
openai.api_base =  api_base
openai.api_type = 'azure'
openai.api_version = '2023-03-15-preview'

prompt = '''How long will it take for my package to ship from Chicago to Tampa with standard shipping? '''

conversation = []
conversation.append({"role": "user", "content": prompt})

response = openai.ChatCompletion.create(
                    engine=deployment_name,
                    messages = conversation
                )

response['choices'][0]['message']['content']
```

```
Out[2]: 'The standard shipping time for a package from Chicago to Tampa can vary de
pending on the shipping carrier and the specific service chosen. On average, it can
take between 2 to 5 business days for a package to be delivered using standard ship
ping in this route. However, it is recommended to check with the shipping carrier o
r retailer for more accurate and up-to-date information on shipping times.'
```

Azure OpenAI Studio



HuggingFace

```python
from transformers import AutoModel, AutoTokenizer, AutoModelForSeq2SeqLM

tokenizer = AutoTokenizer.from_pretrained("THUDM/chatglm2-6b", trust_remote_code=True)

model = AutoModel.from_pretrained("THUDM/chatglm2-6b", trust_remote_code=True)

model = model.half().cuda()
model = model.eval()

prompt = '''How long will it take for my package to ship from Chicago to Tampa with standard shipping? '''

response, history = model.chat(tokenizer, prompt, history=[])
print(response)
```

```
The delivery time for a package from Chicago to Tampa with standard shipping will depend on the shipping
company and the specific route they take. However, standard shipping typically takes between 2-7 business
days to delivery your package. It's important to note that the shipping time may vary depending on the sh
ipping company and the specific location, so it's best to check with them for an accurate estimate.
```

Prompt engineering

Extending your prompt input to include clear instructions, relevant context, and a desired output format.

- Simple
- Prompting
- Prompt engineering & few-shot learning
- Multi-stage reasoning
- Embeddings & search
- Exploring specialized LLMs
- Fine-tuning
- Complex

envision

©2023 3Cloud

Not great results?

Start adding complexity.

Prompt engineering

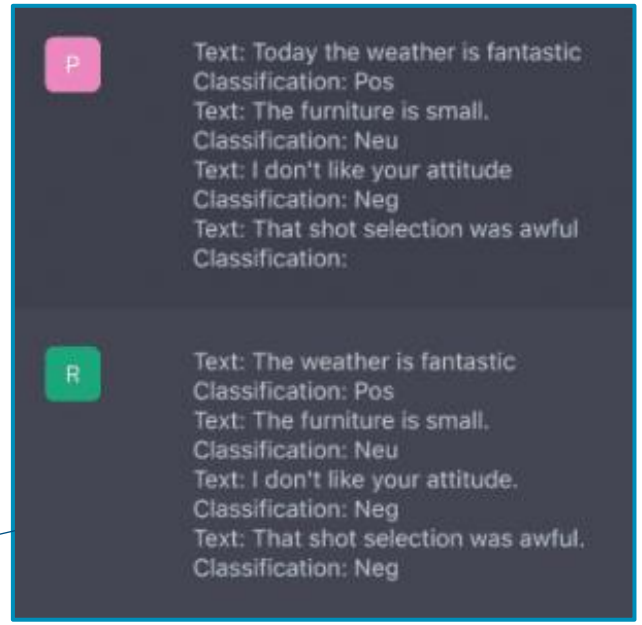
Extending your prompt input to include clear instructions, relevant context, and a desired output format.

Few-shot learning

Engineering a prompt with several examples of input and desired output. Teach the model on the fly what you're looking for.

Chain-of-thought prompting

Type of few-shot learning where reasoning examples are provided to demonstrate how the output should be obtained given the input.




```
prompt = '''The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?'''

# vs.

coT_prompt = '''
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
###
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.
~~~
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
###
'''
```

- Simple
- Prompting
- **Prompt engineering & few-shot learning**
- Multi-stage reasoning
- Embeddings & search
- Exploring specialized LLMs
- Fine-tuning
- Complex

envision

©2023 3Cloud

Not great results?

Before moving on, try to map the business problem into a common NLP task.

Conversational?

Summarization?

Translation?

Sentiment Analysis?

Question answering?

Classification?

Similarity?

To increase the efficiency of my support staff, I want inbound customer support requests to be routed to the appropriate representative based on the content of the request.

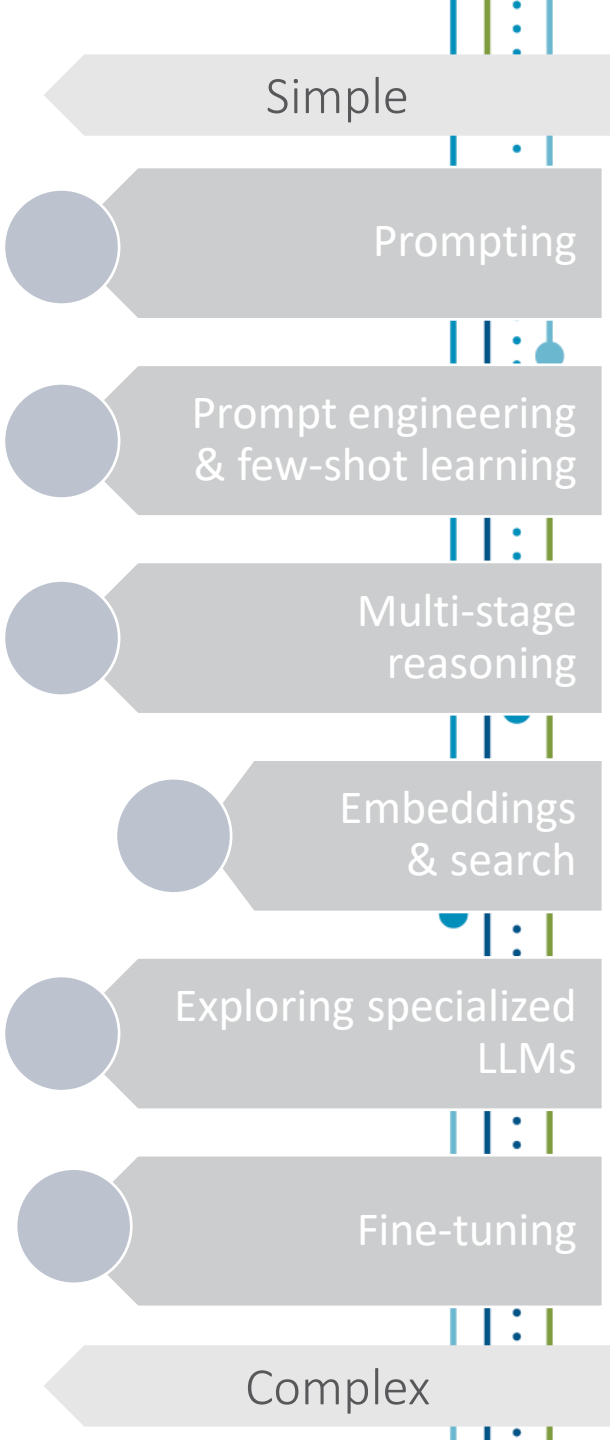
Break it down

In order to determine which representative is best suited to respond to each request, I want to determine

- what service area each request pertains to,
- the customer's sentiment, and
- if the answer to the customer's support request is publicly available.

NLP tasks

- Determine service area: summarization + classification
- Determine sentiment: sentiment analysis
- Determine availability of response: A web search + text extraction



envision

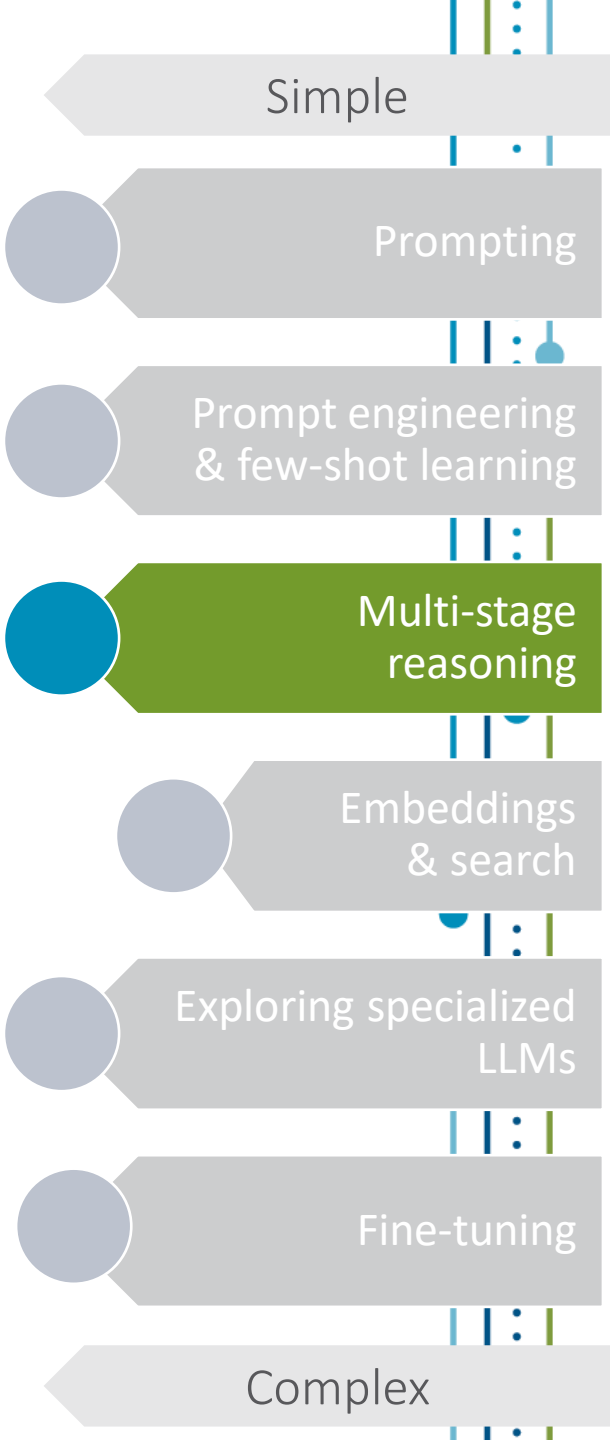
©2023 3Cloud

Multi-Stage Reasoning

Is your use case multi-step?
Can your complex objective be broken down into simpler, smaller tasks?
Would the workflow benefit by integrating with non-LLM tools, like Python, web searches, or SQL databases?

- **LangChain**: Popular framework for connecting LLMs with each other and external tools to create LLM workflows.
  - Output of the first LLM can be passed to another LLM as an input to "chain" them together

- **Agents + Plugins**: Extend the capabilities of LLMs by instructing them to continually iterate through a set of tools in order to complete an objective, without explicit instructions on which tools to use when.



envision

©2023 3Cloud

Embeddings & Search

Is the text related to your use case domain-specific?
Are your results in need of better context & more accurate facts from your own data?

What is an embedding?

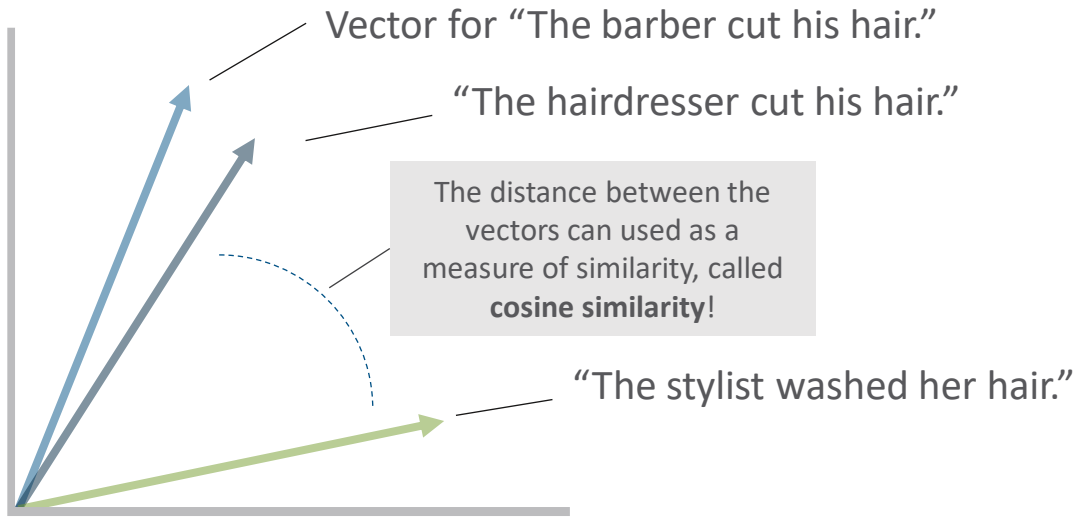
A numerical representation of a piece of text that captures its meaning.

Ex. The **barber** cut his hair.
The **hairdresser** cut his hair.

→ Language Embedding Model →

12 **46** 35 87 43
12 **47** 35 87 43

Since **barber** and **hairdresser** are used in the same context, an embedding of these sentences could use numbers that are close together to represent these words.



Embeddings allow us to find pieces of text that are similar to each other!

Simple

- Prompting
- Prompt engineering & few-shot learning
- Multi-stage reasoning
- **Embeddings & search**
- Exploring specialized LLMs
- Fine-tuning

Complex

Embeddings & Search

Is the text related to your use case domain-specific?
Are your results in need of better context & more accurate facts from your own data?

Embeddings allow us to find pieces of text that are similar to each other!

…which allows us to embed relevant data & context into our language model prompts.

Natural Language Product Search

Embeddings & Search

Is the text related to your use case domain-specific?
Are your results in need of better context & more accurate facts from your own data?

Embeddings allow us to find pieces of text that are similar to each other!

…which allows us to embed relevant data & context into our language model prompts.

Natural Language Product Search

Customer search

"best shoes for long distance running and high arches"

Language Embedding Model

Similar texts to customer's search

Product	Product Description	Customer Reviews
SuperRun 5000	*Lasting comfort will propel you to…*	***"The SuperRun 5000s were perfect for my marathon…"***
StaminaPro **Arch Support**	*Ideal for the runner who…*	…
Run3C Plus	***The most supportive cushion maintains energy long term…***	…

Create a prompt with the customer's search and the relevant texts

Prompt = You are an assistant that makes recommendations on sports clothing and equipment given a customer's query. Use the products listed below to make suggestions and describe why each product is suited to the customer.

Customer query: best shoes for long distance running and high arches

Products: SuperRun 5000s, StaminaPro Arch Support , …
Relevant customer reviews: "The most supportive cushion maintains energy long term…", "The SuperRun 5000s were perfect for my marathon…"

LLM

Relevant, detailed response custom to your data

Simple

Prompting

Prompt engineering & few-shot learning

Multi-stage reasoning

Embeddings & search

Exploring specialized LLMs

Fine-tuning

Complex

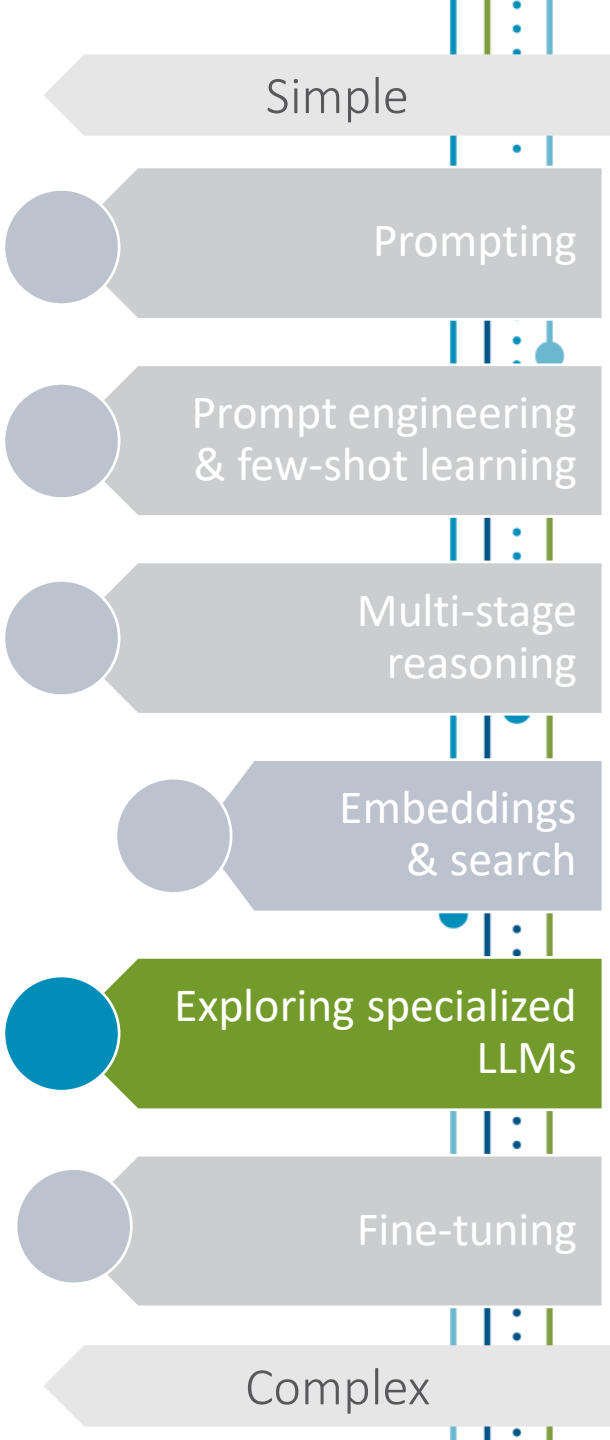
Explore specialized LLMs.

Are any LLMs available that have trained in your domain, or are trained to do a task similar to yours?

Popular language models seem to handle almost anything thrown at them.

Does your use case require this, or do you need a model that can do just a few things very well?

- **StarCoder**: Trained on over 1 trillion natural language and programming language tokens, meets or surpasses performance of leading paid coding models
- **PubMedGPT**: Specialized for the biomedical domain, achieves state-of-the-art results on US medical licensing exams
- **DePlot**: Can perform reasoning on visuals like charts and plots by translating them into textual data



envision

©2023 3Cloud

Consider fine-tuning an existing LLM.

What is fine-tuning?

Fine-tuning exposes an existing trained LLM to a specific dataset so the model can adjust and adapt to a new domain or task.

Fine-tuning is like when we embed text & adding context to prompts, or when we provide the model with examples with few-shot learning.

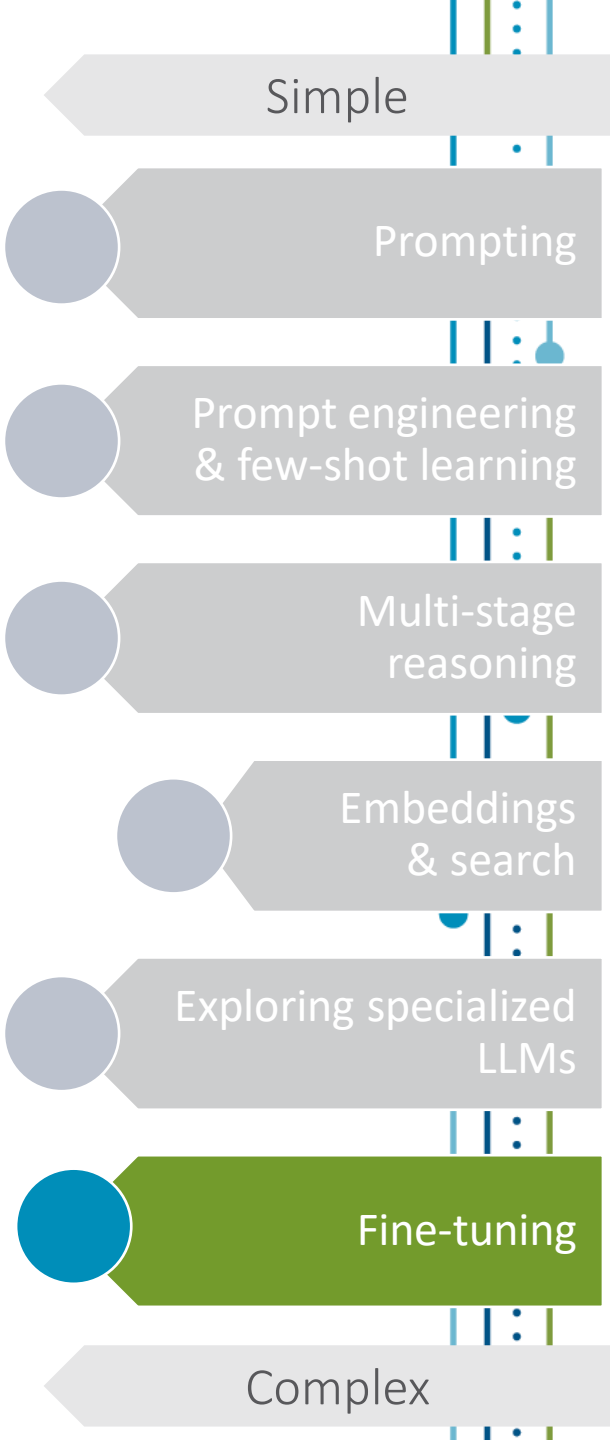
Except in fine-tuning, context and data are integrated into the actual model, as opposed to being passed in the prompt.

Requires…

- Potentially high costs up-front for compute
- Knowledge of LLMs and fine-tuning techniques
- Training data that emulates the desired model input and outputs

…but can result in

- A task-tailored model that provides more accurate results
- Cost savings over time as less tokens & less compute are required for usage
- A differentiating artifact that only your organization owns



Let’s take an example from zero to LLM.

Brickhouse Inc.

Our company Brickhouse Inc. offers professional construction engineering and architecture services.

Our architects consult with building owners to design and **scope out construction work that meets their specifications**.

Problem

Many owners are required by law to build construction in a way that meets very **specific regulations and technical criteria**.

These **specifications are often documented in a large sets of PDFs**. It is untenable for an architect to understand every specification, and combing through every relevant specification is very **time consuming**.

Solution?

Many specs are publicly available documents, like the Unified Facilities Guide Specifications (UFGS).

Can we apply LLMs to help our architects scope out work that meets owner specifications, like the UFGS?

envision

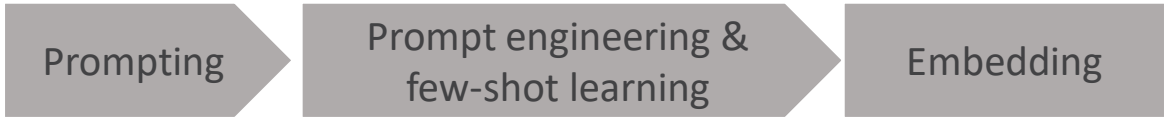
©2023 3Cloud

Let’s take an example from zero to LLM.

Can we use LLMs to help our architects scope out work that meets owner specifications, like the UFGS?

Getting started

- Created an Azure Databricks workspace
- Created an Azure Storage account & uploaded the UFGS PDFs
- Created an Azure Open AI workspace (optional)
  - Add a gpt-35-turbo model deployment
- Created a free Hugging Face account
  - Created and copied a User Access Token from Settings > Access Tokens



envision

©2023 3Cloud

envision
VIRTUAL SUMMIT SERIES
DATA & ANALYTICS

Thank you!

Monday, September 11, 2023

PRESENTED BY 3Cloud