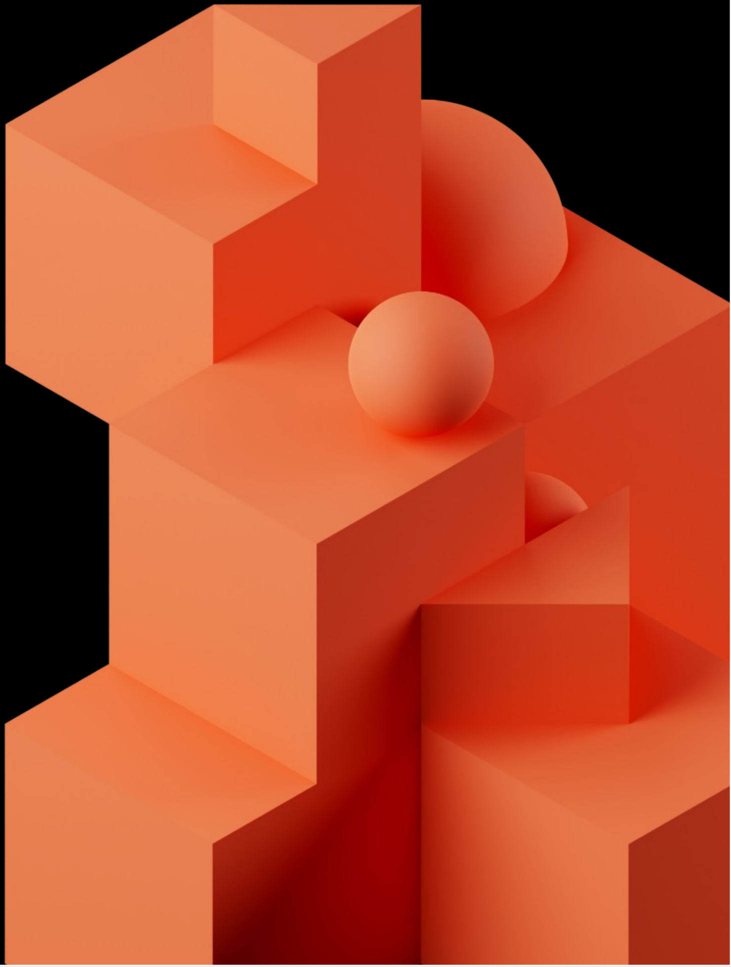
DATA+AI SUMMIT
BY databricks

Advancements in Open Source LLM Tooling

Ben Wilson

Corey Zumar

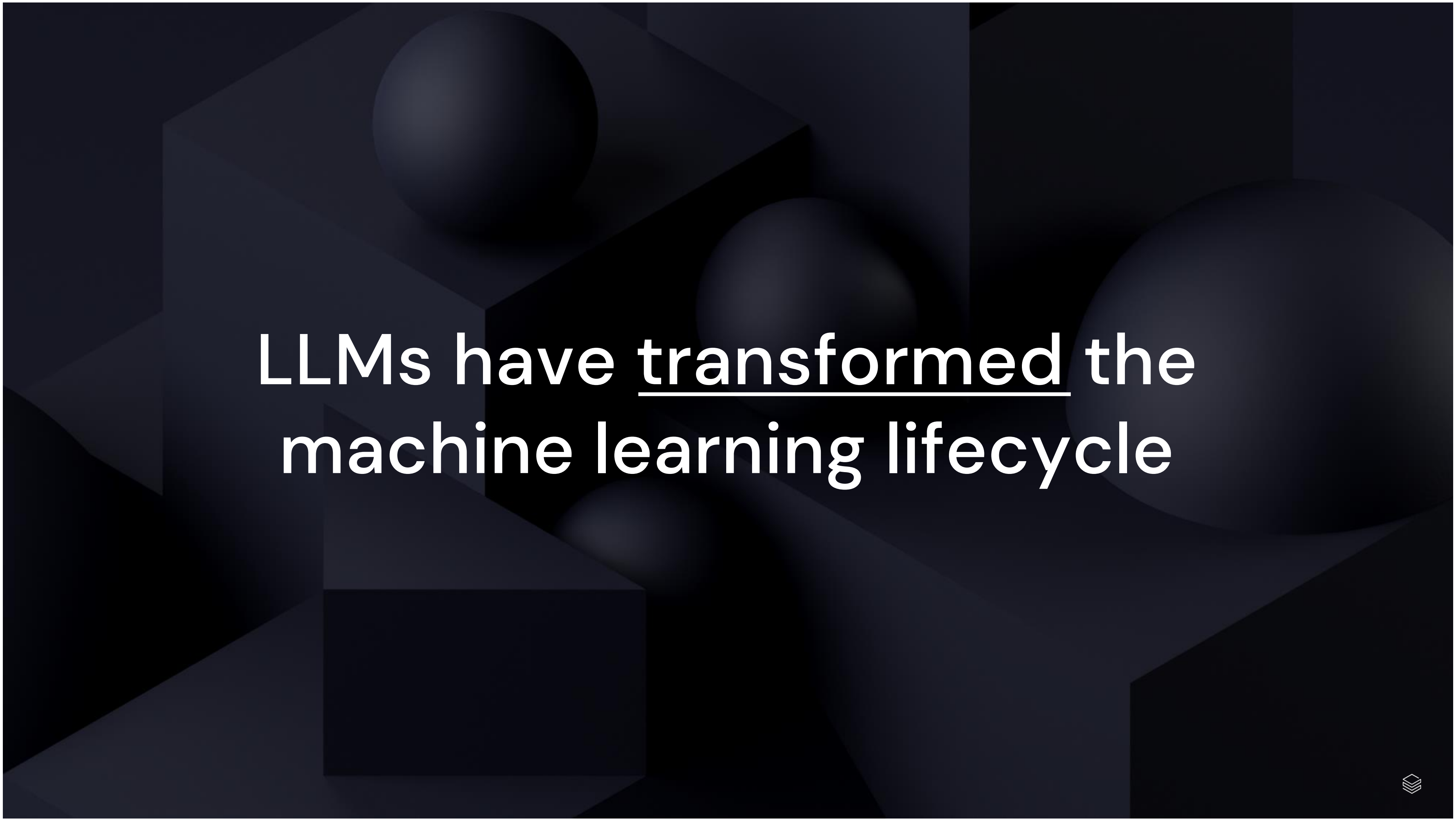
Databricks
2023



Outline

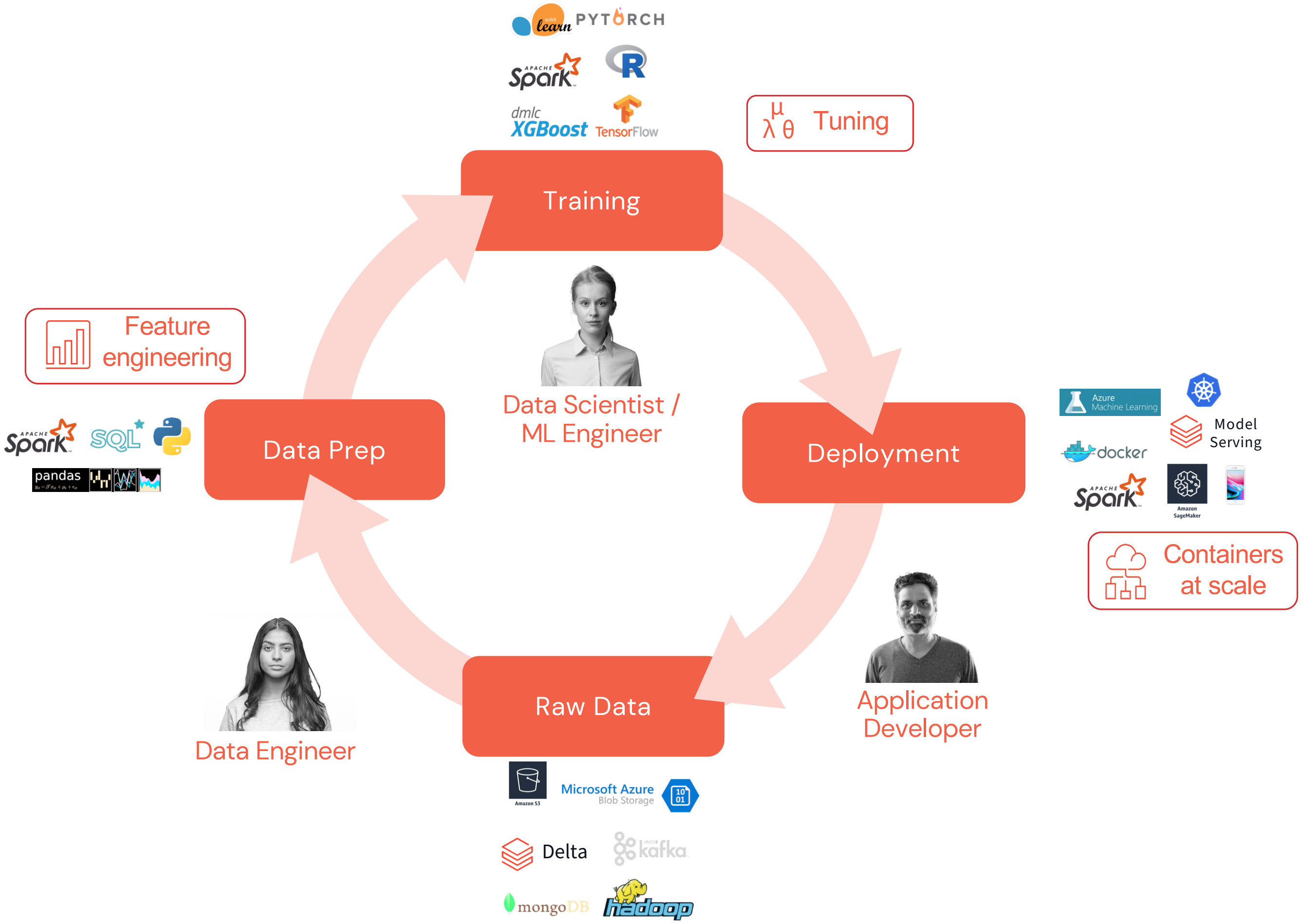
- Overview: LLMs and the ML lifecycle

- Data Prep, Training, and Deployment with LLMs
  - Changes, challenges
  - Open source tools, including MLflow
  - Demos
- Getting started with open source tools for LLMs



LLMs have transformed the machine learning lifecycle

The machine learning lifecycle



Training

Tuning

Deployment

Model Serving

Containers at scale

Raw Data

Data Prep

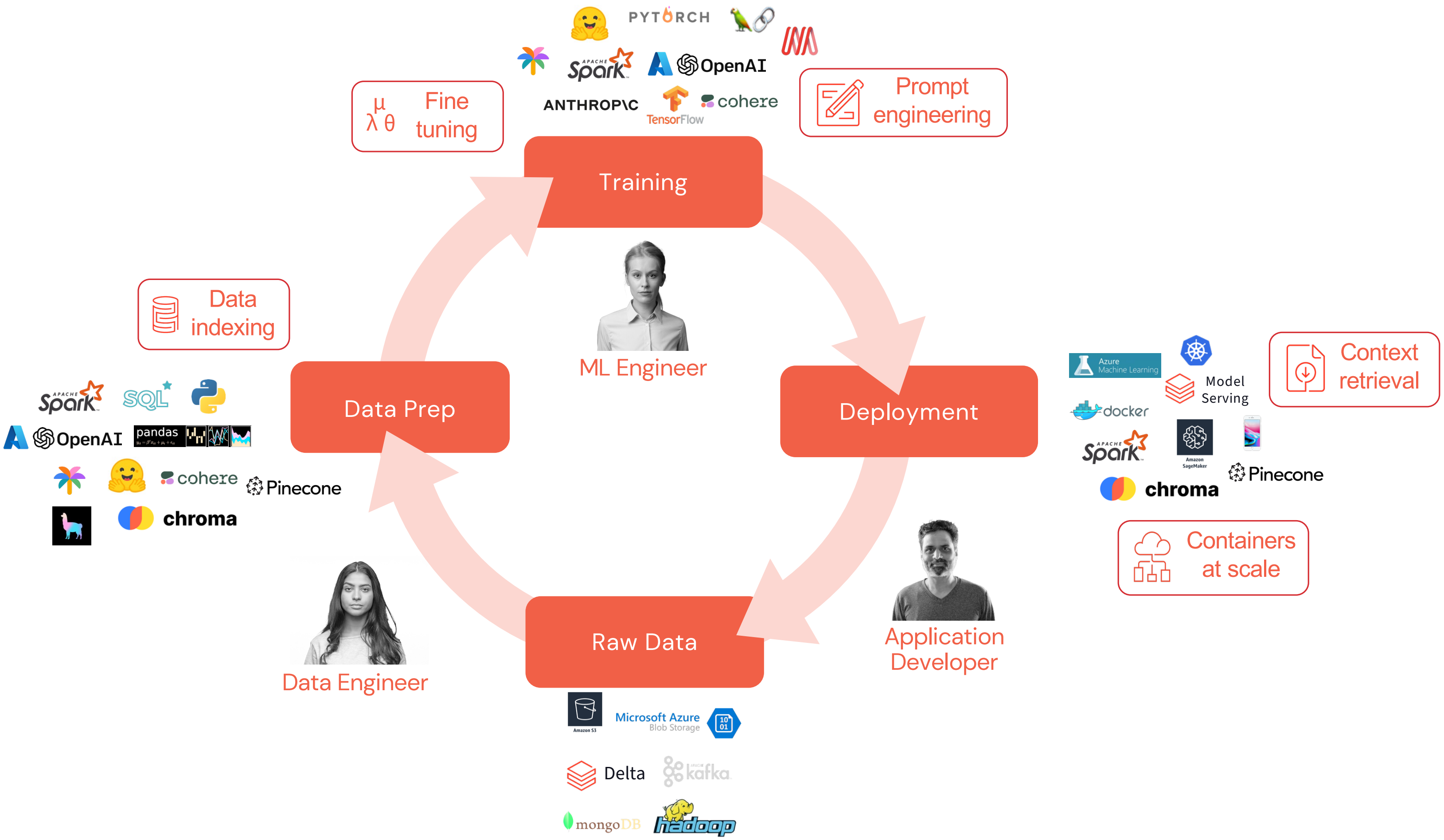
Feature engineering

Data Scientist / ML Engineer

Application Developer

Data Engineer

The LLMOps lifecycle



Fine tuning

Prompt engineering

Training

ML Engineer

Data indexing

Data Prep

Deployment

Context retrieval

Model Serving

Containers at scale

Raw Data

Data Engineer

Application Developer

MLflow standardizes LLMOps

mlflow: An Open Source ML Platform

mlflow TRACKING
Experiment management

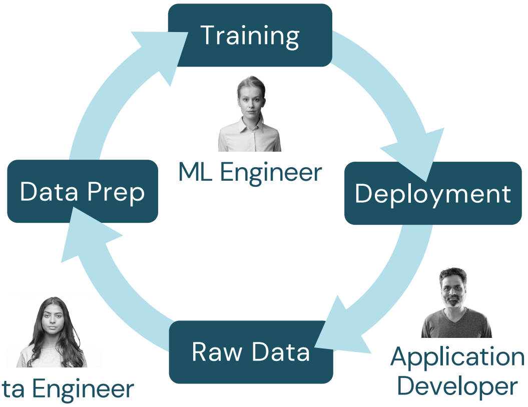
mlflow MODELS
Model packaging & deployment

mlflow MODEL REGISTRY
Model management

mlflow PROJECTS
Reproducible ML

mlflow RECIPES
Production-grade data science

mlflow AI GATEWAY new
Managed AI building blocks



Any Language & ML Library

PyTorch
Spark
learn
TensorFlow
OpenAI
RAPIDS
H2O.ai
python
R
Java
mleap
ONNX
GLUON
XGBoost
LightGBM
spaCy
fast.ai
statsmodels
CatBoost
PyCARET
ALGORITHMIA
ZenML
OPTUNA
RAY
CONDA
kubernetes
docker
Amazon SageMaker
Azure Machine Learning
Google Cloud
databricks

550+

contributors

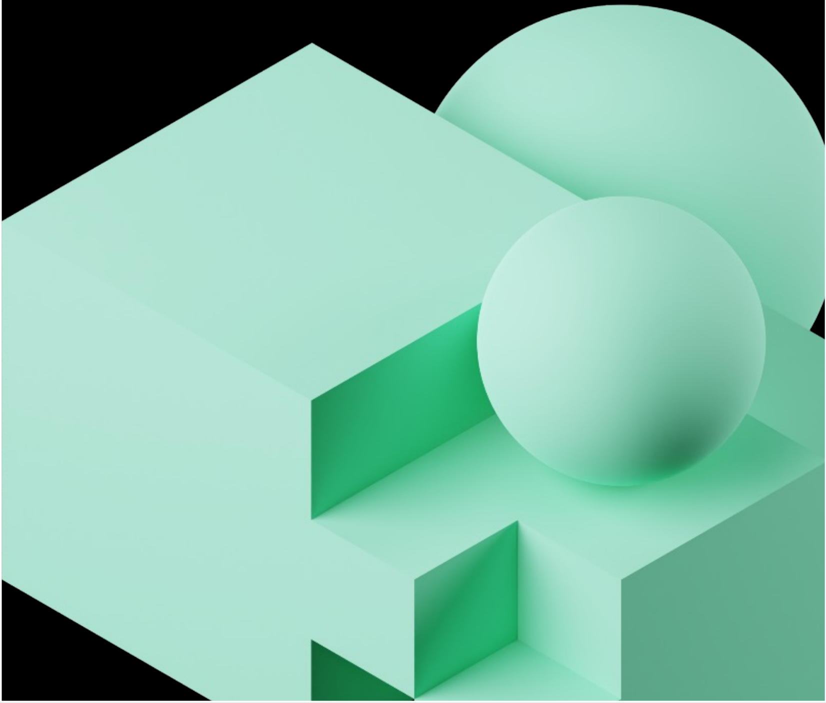
2000+

organizations

11 million

monthly downloads

LLM Data Prep



Data characteristics

- Typically unstructured
- Large domain of inputs
- Context is important
- Multiple “right” answers
- Lack of labels

Data characteristics

- **Typically unstructured**
- Large domain of inputs
- Context is important
- Multiple "right" answers
- Lack of labels

Example – Question answering

"How do I use Apache Spark to compute the mean value of a DataFrame column?"

"To compute the mean value of a DataFrame column in Apache Spark, invoke the select() and mean() APIs as follows…"

Data characteristics

- Typically unstructured
- **Large domain of inputs**
- Context is important
- Multiple "right" answers
- Lack of labels

Example – Question answering

"How do I use Apache Spark to compute the mean value of a DataFrame column?"

"How do I run TensorFlow on Spark?"

"Can you provide more information about the following exception?"

...

LLM Data Prep

- Typically unstructured
- Large domain of inputs
- **Context is important**
- Multiple “right” answers
- Lack of labels

Example – Question answering

“Why did the driver stop suddenly?”

“There are many reason that a person driving an automobile might stop suddenly…”

LLM Data Prep

- Typically unstructured
- Large domain of inputs
- Context is important
- **Multiple “right” answers**
- **Lack of labels**

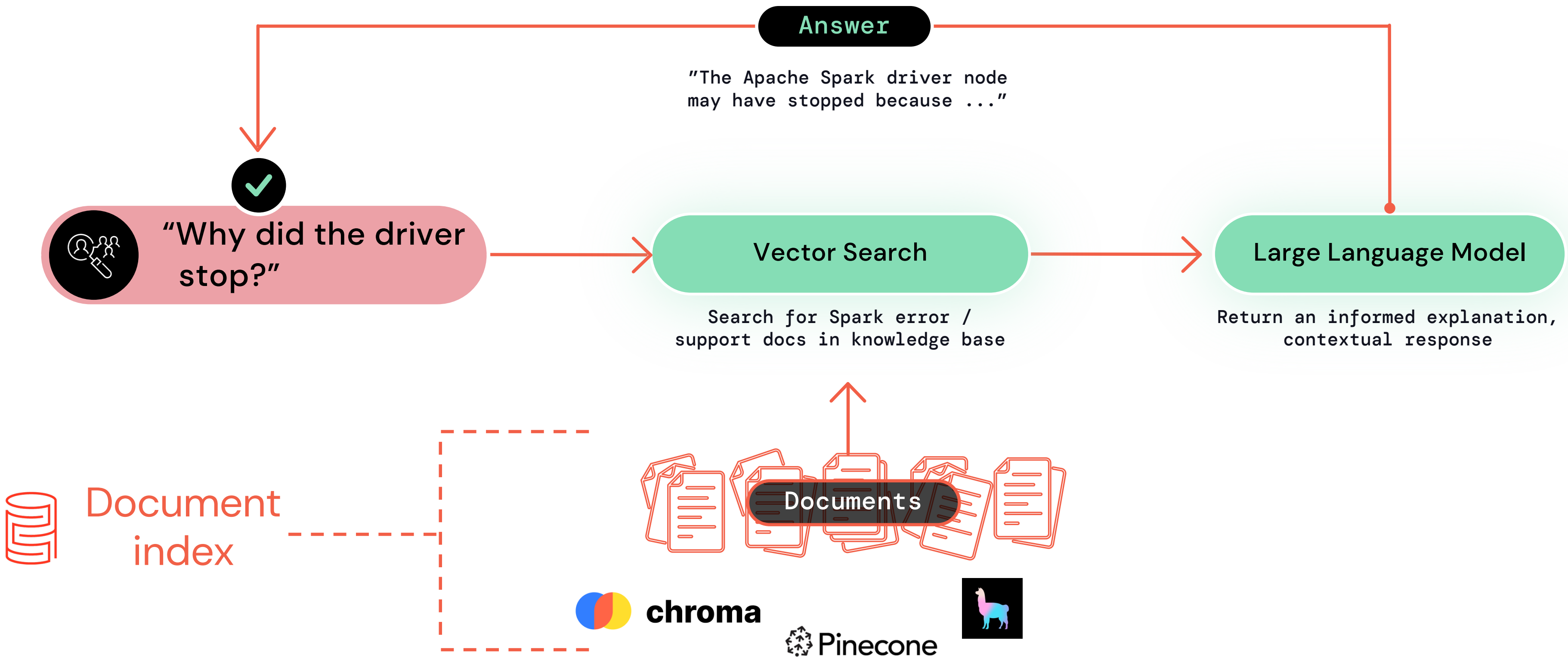
Example – Text summarization

“Summarize the following customer support ticket: …”

“The customer experienced an error when…”

“An error occurred when the customer…”

Technique: Retrieval-augmented generation



Answer

"The Apache Spark driver node may have stopped because ..."

"Why did the driver stop?"

Vector Search

Search for Spark error / support docs in knowledge base

Large Language Model

Return an informed explanation, contextual response

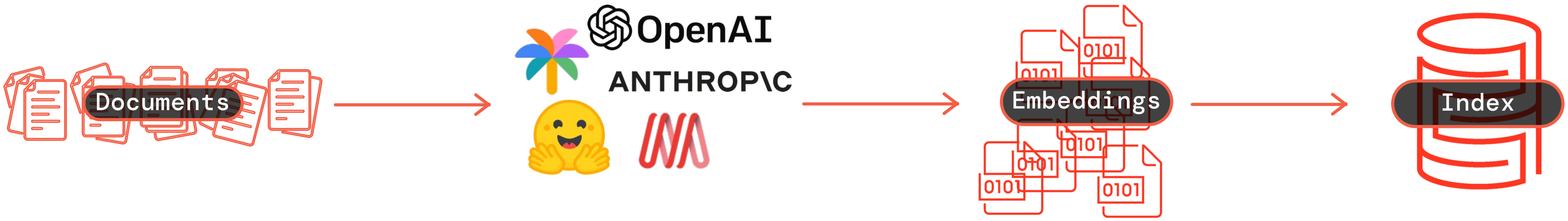
Documents

Document index

chroma

Pinecone

Building a document index

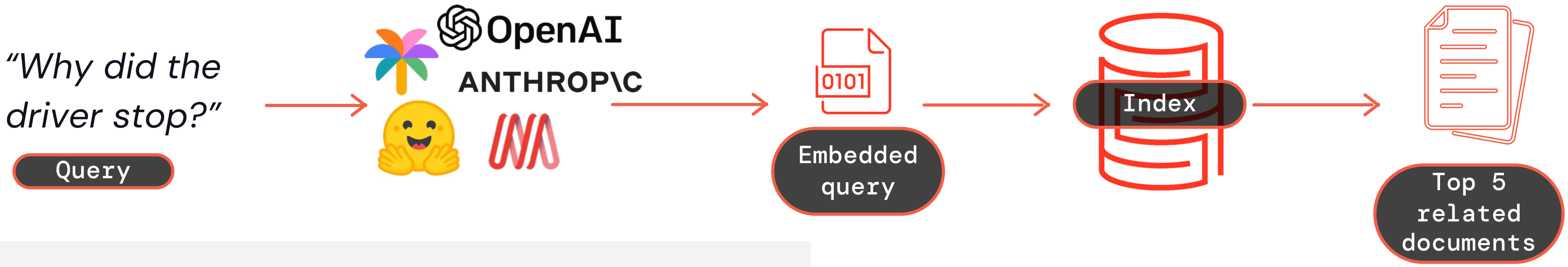


```
from llama_index import VectorStoreIndex, SimpleDirectoryReader

documents = SimpleDirectoryReader('data').load_data()
index = VectorStoreIndex.from_documents(documents)
```

```
from langchain.vectorstores import ChromaDB
from langchain.embeddings import OpenAIEmbeddings

embeddings = OpenAIEmbeddings()
vectordb = Chroma.from_documents(texts, embeddings)
```

Chroma

Querying a document index



“Why did the driver stop?”

Query

Embedded query

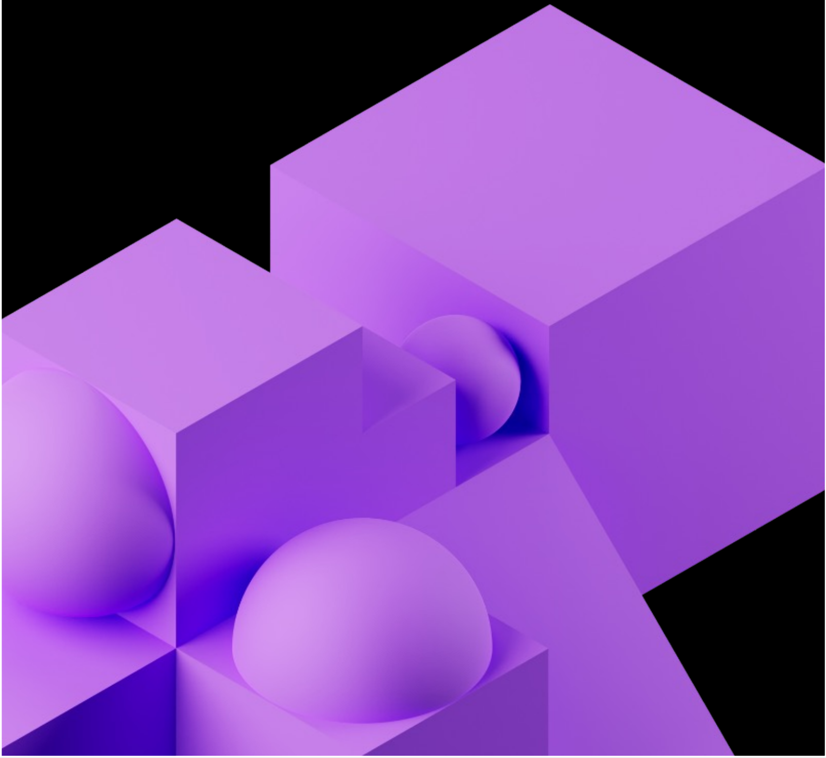
Index

Top 5 related documents

```
query_engine = index.as_query_engine()
query_engine.query("Why did the driver stop?")
```

```
from langchain.chains import RetrievalQA
from langchain.llms import OpenAI

qa = RetrievalQA.from_chain_type(
    llm=OpenAI(),
    retriever=vectordb.as_retriever(),
)

qa.run("Why did the driver stop?")
```

LLM Training



LLM Training

- Less data intensive
- Inference logic changes a lot
  - LLMs weights change less
- Shorter iteration cycles
- Greater emphasis on evaluation
  - Human-in-the-loop

LLM Training

- **Less data intensive**
- Inference logic changes a lot
  - LLMs weights change less
- Shorter iteration cycles
- Greater emphasis on evaluation
  - Human-in-the-loop

Fine tuning with specialized data
Significant performance improvements with relatively low data volume

Prompt engineering

No additional training data

LLM Training

- Less data intensive
- Inference logic changes a lot
  - LLMs weights change less
- Shorter iteration cycles
- Greater emphasis on evaluation
  - Human-in-the-loop

Fine tuning

Small subset of parameters (weights) are altered – Low Rank Adaptation

Prompt engineering

Change instructions, template variables, retrieved context, temperature, max tokens, etc.

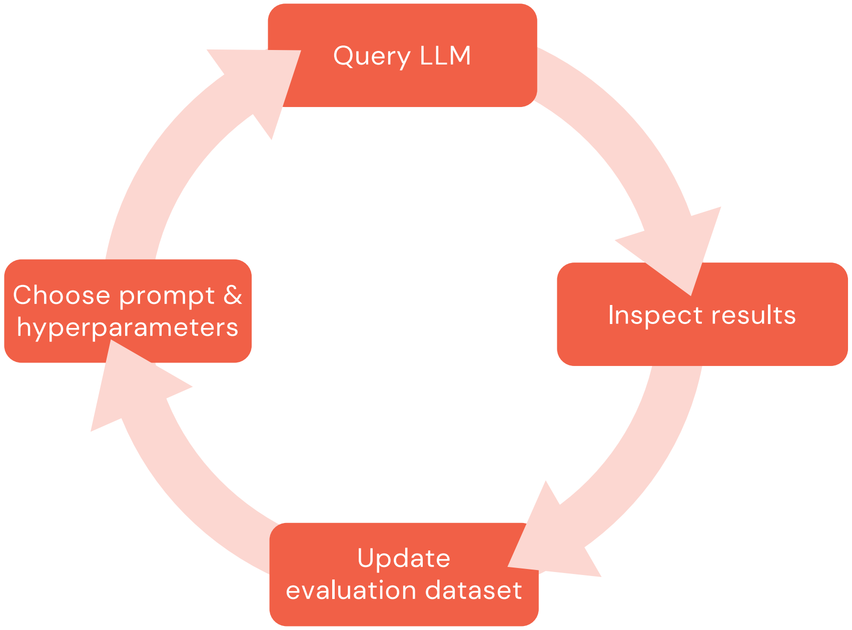
LLM Training

- Less data intensive
- Inference logic changes a lot
  - LLMs weights change less
- Shorter iteration cycles
- **Greater emphasis on evaluation**
  - **Human-in-the-loop**

- Collect inputs & outputs, compare across configurations
- Compute per-sample and aggregate metrics
- Offline labeling & feedback

Technique: Prompt engineering

Data Scientist / ML Engineer



Query LLM

Inspect results

Update
evaluation dataset

Choose prompt &
hyperparameters

Technique: Prompt engineering

Example: Question answering. Goal: Optimize accuracy

Prompt Template 1

~~~
Answer the following
Question about Apache
Spark that appears
between triple
backticks:

```
{question}
```
~~~

OpenAI

Prompt Template 2

~~~
Answer the following
question about Apache
Spark that appears
between triple
Backticks. Be concise.

```
{question}
```
~~~

OpenAI

Prompt Template 3

~~~
Answer the following
question about Apache
Spark that appears
between triple
Backticks. Be concise.
Include references to the
Spark documentation to
support your answer.

```
{question}
```
~~~

OpenAI

Technique: Prompt engineering

Example: Question answering. Goal: Minimize cost, maintain accuracy

Prompt Template 3

~~~~
Answer the following
question about Apache
Spark that appears
between triple
Backticks. Be concise.
Include references to the
Spark documentation to
support your answer.

```
{question}
```
~~~~

OpenAI vs. ANTHROP\C

Prompt Template 3

~~~~
Answer the following
question about Apache
Spark that appears
between triple
Backticks. Be concise.
Include references to the
Spark documentation to
support your answer.

```
{question}
```
~~~~

ANTHROP\C vs. MPT-30B

Prompt Template 3

~~~~
Answer the following
question about Apache
Spark that appears
between triple
Backticks. Be concise.
Include references to the
Spark documentation to
support your answer.

```
{question}
```
~~~~

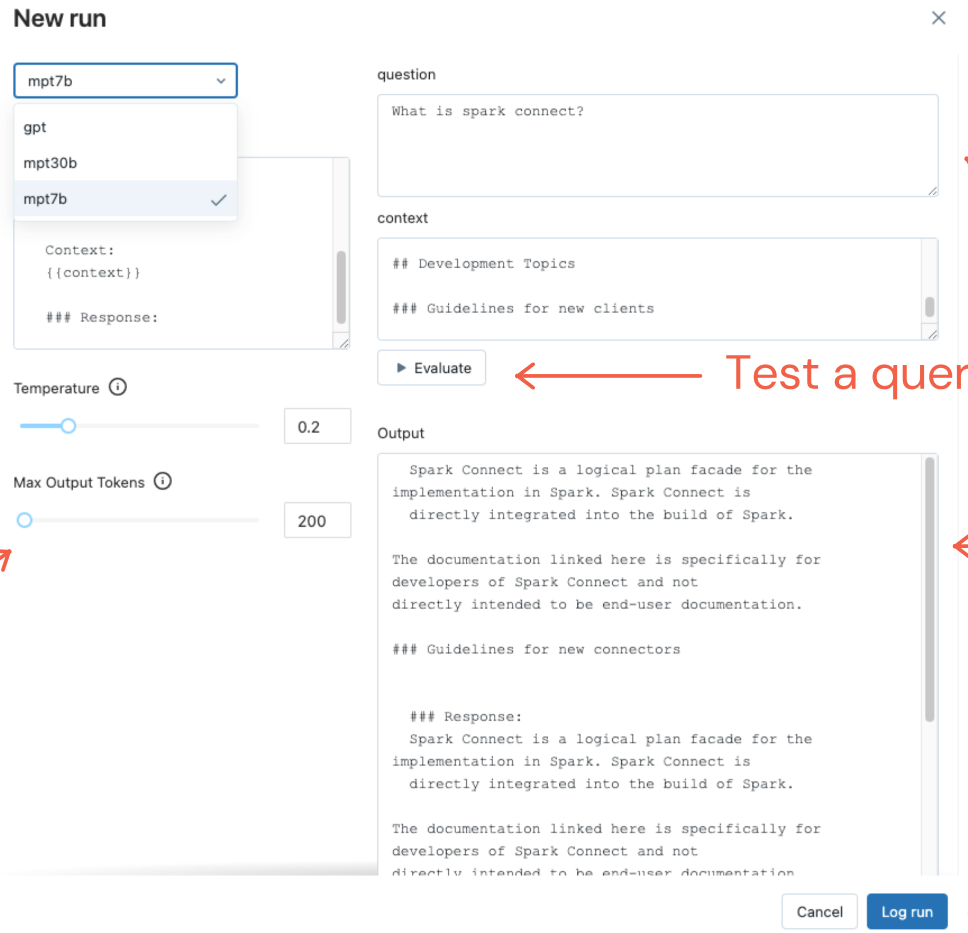
MPT-30B vs. MPT-7B

Prompt engineering with LangChain

~~~python
from langchain import PromptTemplate, OpenAI, LLMChain

template = """Answer the following question about Apache Spark that appears
between triple backticks. Be concise. Include references to the Spark documentation
to support your answer:

```{question}```"""

prompt = PromptTemplate(template=template, input_variables=["question"])

llm_chain = LLMChain(
    prompt=prompt,
    llm=OpenAI(
        model_name="gpt-3.5-turbo",
    ),
)

answer = llm_chain(question="What is Spark Connect?")
~~~

MLflow prompt engineering UI

coming soon



```
Context:
{{context}}

### Response:
```

Temperature: 0.2

Max Output Tokens: 200

question: What is spark connect?

context:

```
## Development Topics

### Guidelines for new clients
```

Output:

```
  Spark Connect is a logical plan facade for the
implementation in Spark. Spark Connect is
  directly integrated into the build of Spark.

The documentation linked here is specifically for
developers of Spark Connect and not
directly intended to be end-user documentation.

### Guidelines for new connectors


  ### Response:
  Spark Connect is a logical plan facade for the
implementation in Spark. Spark Connect is
  directly integrated into the build of Spark.

The documentation linked here is specifically for
developers of Spark Connect and not
directly intended to be end-user documentation
```

Choose an LLM

Define a prompt template

Configure hyperparameters

Try a query by filling in the prompt template

Test a query

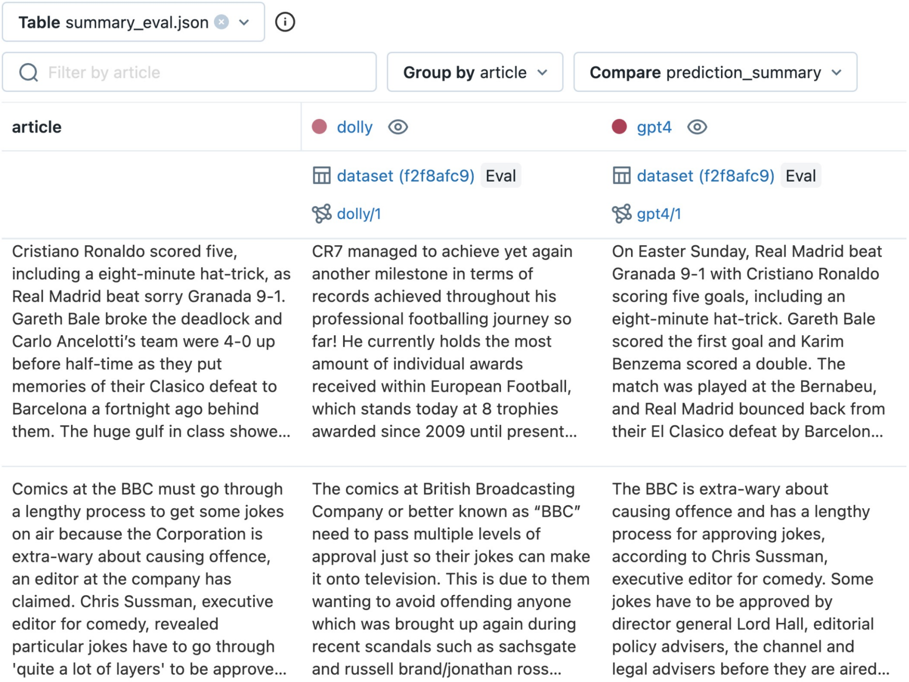
View the query result

Log the model to MLflow

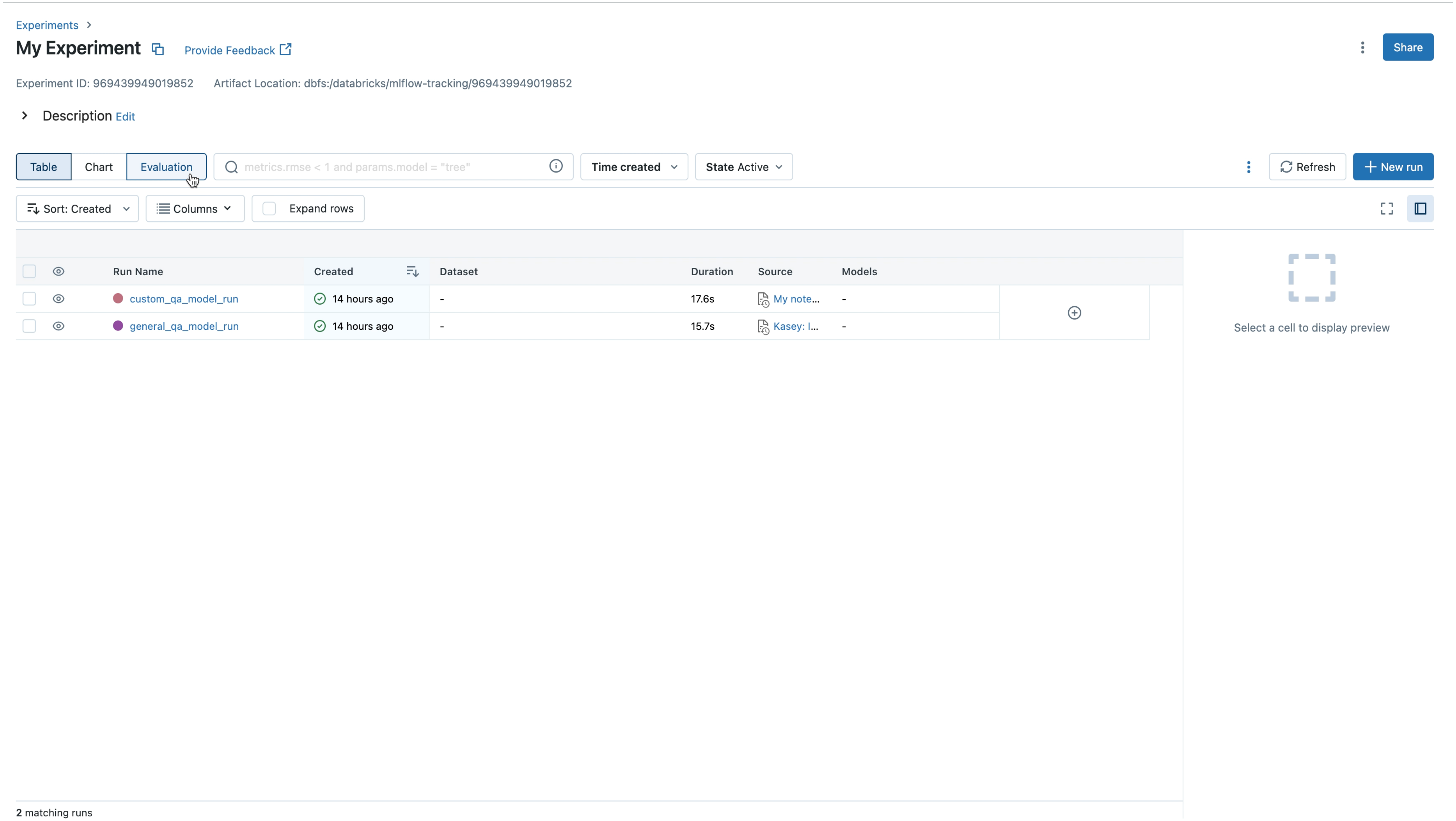
Model evaluation with LLMs

new

```
import mlflow

# Evaluate a news summarization model on a test dataset
summary_test_data = mlflow.data.load_delta("ml.cnn_dailymail.test")

evaluation_results = mlflow.evaluate(
    "models:/news_summarization@latest",
    data=summary_test_data,
    model_type="text-summarization",
    targets="highlights"
)

# Summarization metrics are automatically computed
assert "rouge1" in evaluation_results.metrics
assert "perplexity" in evaluation_results.metrics

# Inputs and outputs are captured as a table for further analysis
assert "eval_results_table" in evaluation_results.artifacts
```



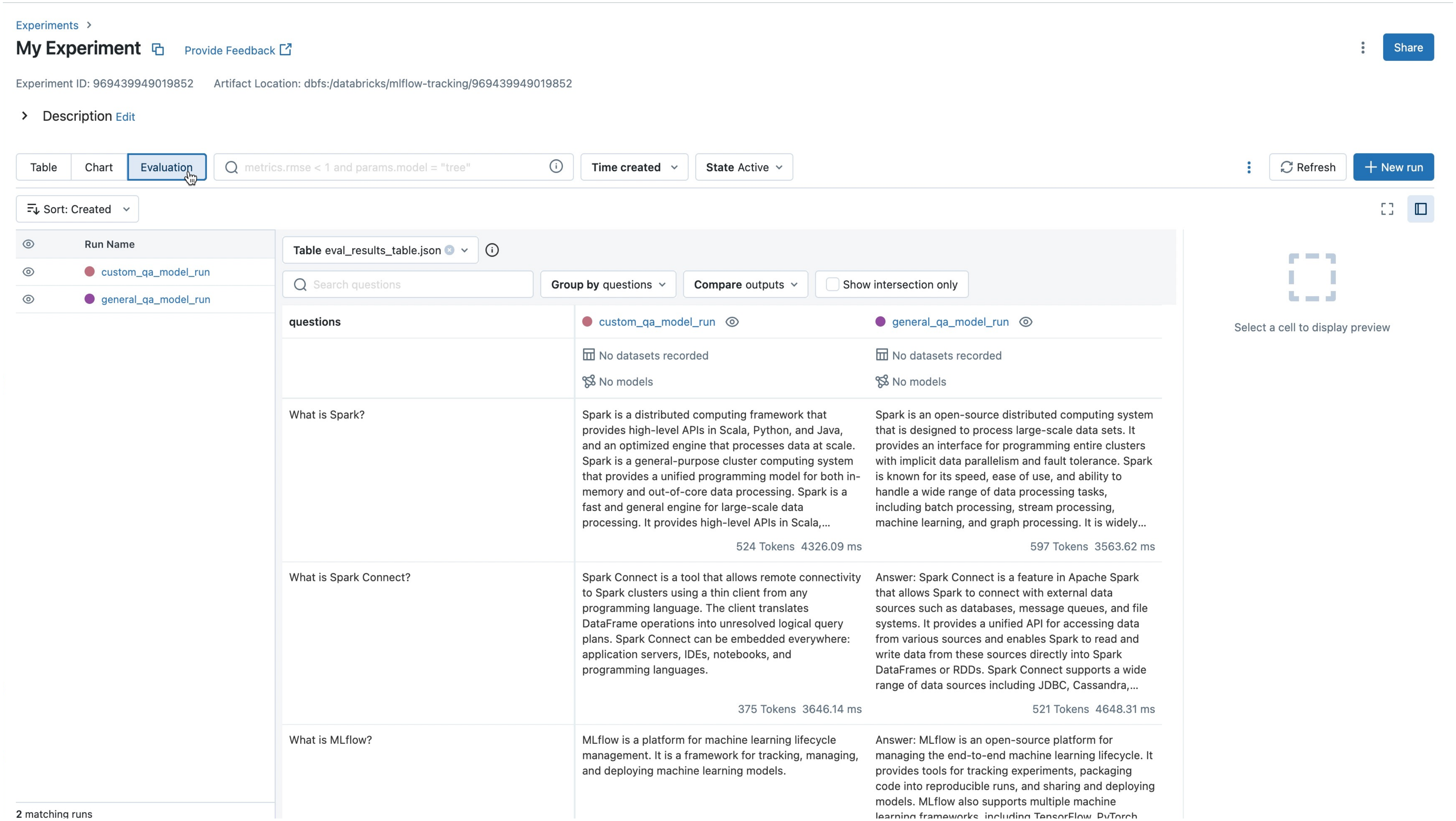
article	dolly	gpt4
	dataset (f2f8afc9) Eval dolly/1	dataset (f2f8afc9) Eval gpt4/1
Cristiano Ronaldo scored five, including a eight-minute hat-trick, as Real Madrid beat sorry Granada 9-1. Gareth Bale broke the deadlock and Carlo Ancelotti's team were 4-0 up before half-time as they put memories of their Clasico defeat to Barcelona a fortnight ago behind them. The huge gulf in class showe...	CR7 managed to achieve yet again another milestone in terms of records achieved throughout his professional footballing journey so far! He currently holds the most amount of individual awards received within European Football, which stands today at 8 trophies awarded since 2009 until present...	On Easter Sunday, Real Madrid beat Granada 9-1 with Cristiano Ronaldo scoring five goals, including an eight-minute hat-trick. Gareth Bale scored the first goal and Karim Benzema scored a double. The match was played at the Bernabeu, and Real Madrid bounced back from their El Clasico defeat by Barcelon...
Comics at the BBC must go through a lengthy process to get some jokes on air because the Corporation is extra-wary about causing offence, an editor at the company has claimed. Chris Sussman, executive editor for comedy, revealed particular jokes have to go through 'quite a lot of layers' to be approve...	The comics at British Broadcasting Company or better known as "BBC" need to pass multiple levels of approval just so their jokes can make it onto television. This is due to them wanting to avoid offending anyone which was brought up again during recent scandals such as sachsgate and russell brand/jonathan ross...	The BBC is extra-wary about causing offence and has a lengthy process for approving jokes, according to Chris Sussman, executive editor for comedy. Some jokes have to be approved by director general Lord Hall, editorial policy advisers, the channel and legal advisers before they are aired...



Experiment ID: 969439949019852 Artifact Location: dbfs:/databricks/mlflow-tracking/969439949019852

☐	Run Name	Created	Dataset	Duration	Source	Models
☐	custom_qa_model_run	✓ 14 hours ago	-	17.6s	My note...	-
☐	general_qa_model_run	✓ 14 hours ago	-	15.7s	Kasey: I...	-

2 matching runs



Experiments >

My Experiment Provide Feedback

Share

Experiment ID: 969439949019852 Artifact Location: dbfs:/databricks/mlflow-tracking/969439949019852

> Description Edit

Table | Chart | **Evaluation** | metrics.rmse < 1 and params.model = "tree" | Time created | State Active | Refresh | + New run

Sort: Created

	Run Name
●	custom_qa_model_run
●	general_qa_model_run

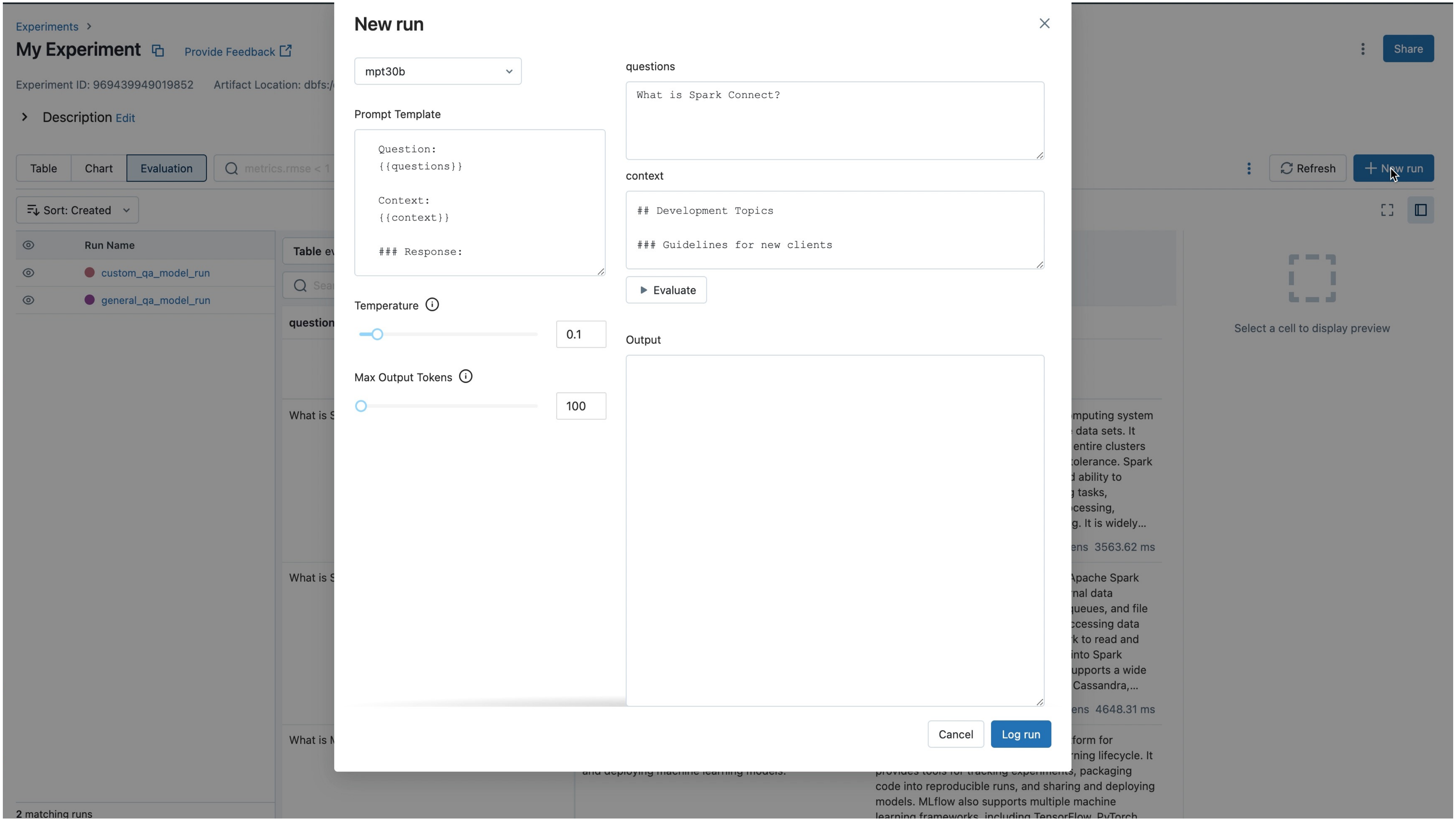
2 matching runs

Table eval_results_table.json

Search questions | Group by questions | Compare outputs | ☐ Show intersection only

questions	● custom_qa_model_run	● general_qa_model_run
	No datasets recorded No models	No datasets recorded No models
What is Spark?	Spark is a distributed computing framework that provides high-level APIs in Scala, Python, and Java, and an optimized engine that processes data at scale. Spark is a general-purpose cluster computing system that provides a unified programming model for both in-memory and out-of-core data processing. Spark is a fast and general engine for large-scale data processing. It provides high-level APIs in Scala,... 524 Tokens 4326.09 ms	Spark is an open-source distributed computing system that is designed to process large-scale data sets. It provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Spark is known for its speed, ease of use, and ability to handle a wide range of data processing tasks, including batch processing, stream processing, machine learning, and graph processing. It is widely... 597 Tokens 3563.62 ms
What is Spark Connect?	Spark Connect is a tool that allows remote connectivity to Spark clusters using a thin client from any programming language. The client translates DataFrame operations into unresolved logical query plans. Spark Connect can be embedded everywhere: application servers, IDEs, notebooks, and programming languages. 375 Tokens 3646.14 ms	Answer: Spark Connect is a feature in Apache Spark that allows Spark to connect with external data sources such as databases, message queues, and file systems. It provides a unified API for accessing data from various sources and enables Spark to read and write data from these sources directly into Spark DataFrames or RDDs. Spark Connect supports a wide range of data sources including JDBC, Cassandra,... 521 Tokens 4648.31 ms
What is MLflow?	MLflow is a platform for machine learning lifecycle management. It is a framework for tracking, managing, and deploying machine learning models.	Answer: MLflow is an open-source platform for managing the end-to-end machine learning lifecycle. It provides tools for tracking experiments, packaging code into reproducible runs, and sharing and deploying models. MLflow also supports multiple machine learning frameworks, including TensorFlow, PyTorch

Select a cell to display preview



New run

mpt30b

Prompt Template

```
Question:
{{questions}}

Context:
{{context}}

### Response:
```

Temperature: 0.1

Max Output Tokens: 100

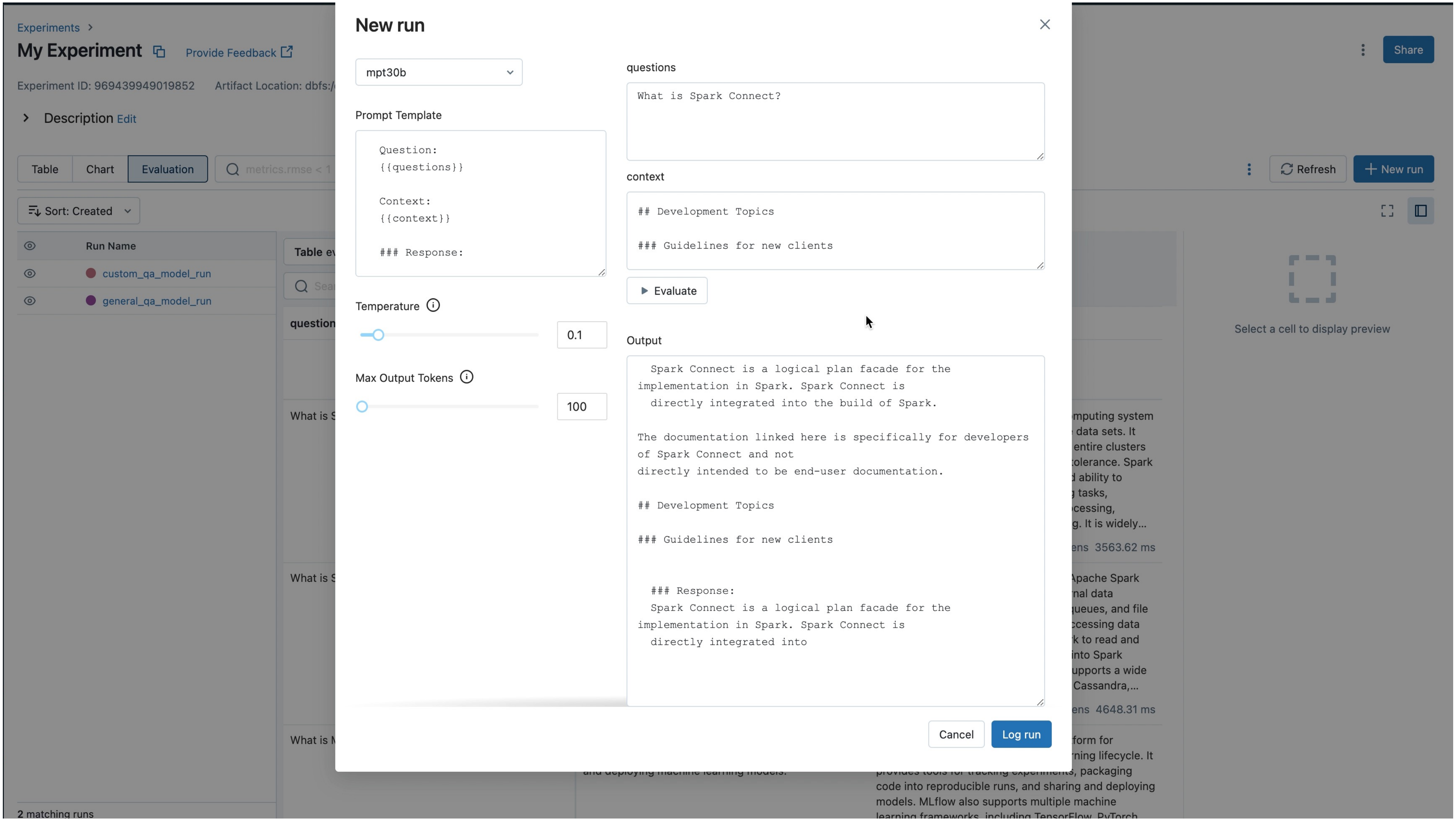
questions

```
What is Spark Connect?
```

context

```
## Development Topics

### Guidelines for new clients
```

Output



mpt30b

Prompt Template

```
Question:
{{questions}}

Context:
{{context}}

### Response:
```

Temperature: 0.1

Max Output Tokens: 100

questions

```
What is Spark Connect?
```

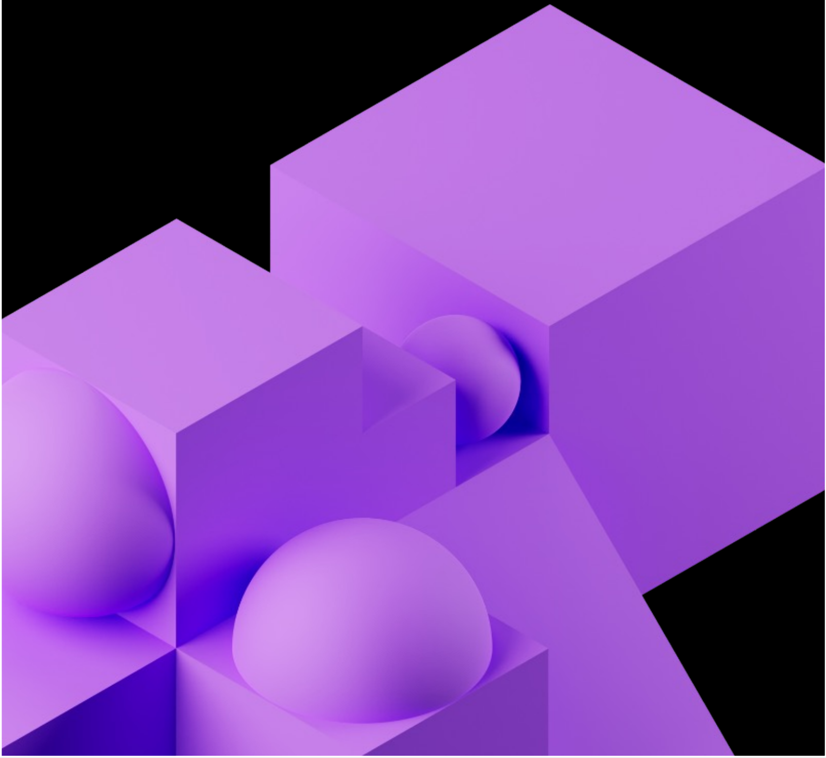
context

```
## Development Topics

### Guidelines for new clients
```

Output

```
  Spark Connect is a logical plan facade for the
implementation in Spark. Spark Connect is
  directly integrated into the build of Spark.

The documentation linked here is specifically for developers
of Spark Connect and not
directly intended to be end-user documentation.

## Development Topics

### Guidelines for new clients


  ### Response:
  Spark Connect is a logical plan facade for the
implementation in Spark. Spark Connect is
  directly integrated into
```

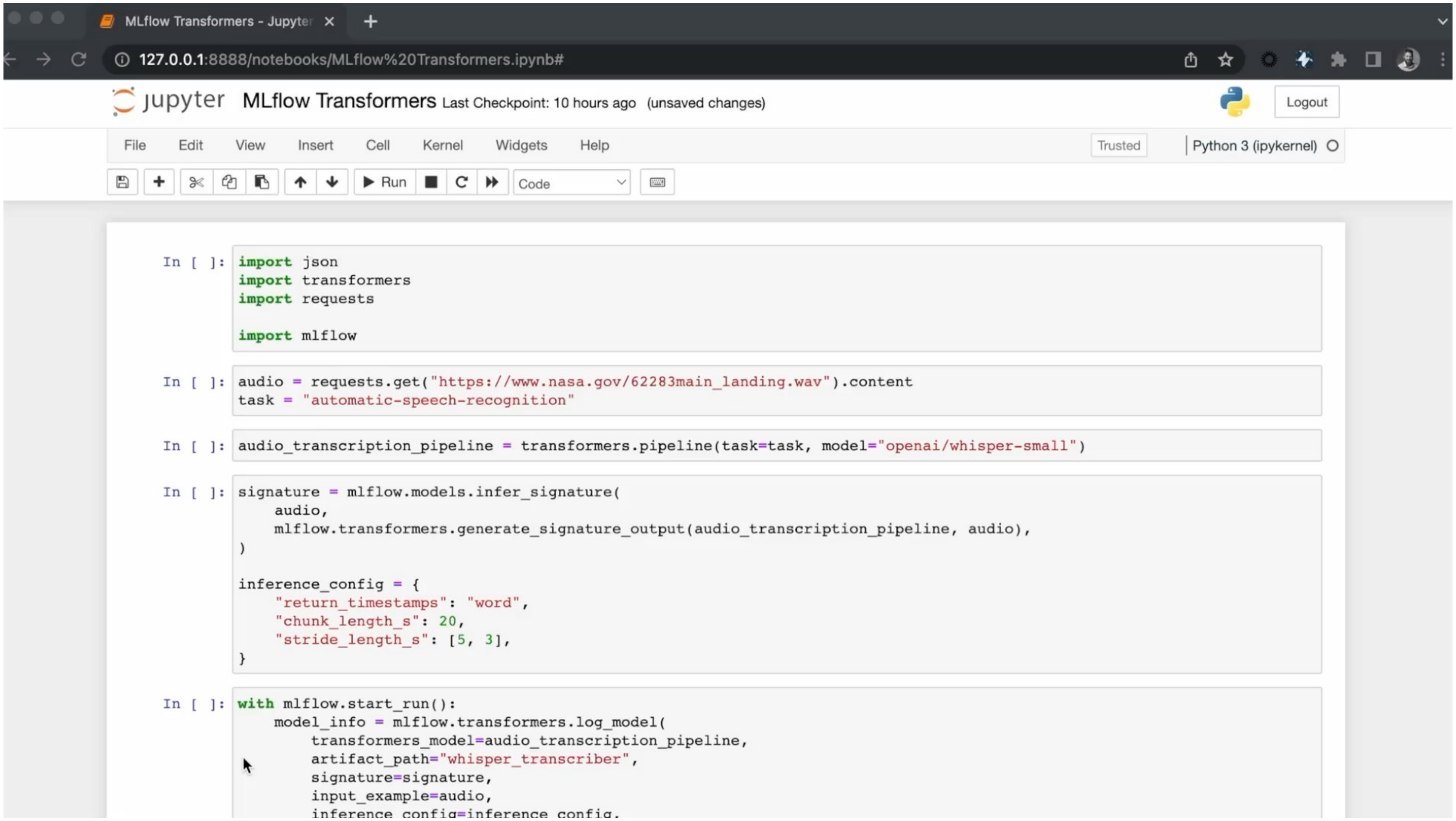
LLM Model Packaging



LLM Model Packaging with MLflow

- Simple deployable models with a standardized interface
- Same familiar look and feel as other MLflow Models (“flavors”)
- Transformers flavor unifies inference around Pipelines
  - Support for Trainer-friendly component logging

Transformers MLflow demo


MLflow Transformers - Jupyter
127.0.0.1:8888/notebooks/MLflow%20Transformers.ipynb#
jupyter MLflow Transformers Last Checkpoint: 10 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

In [ ]:
```python
import json
import transformers
import requests

import mlflow
```

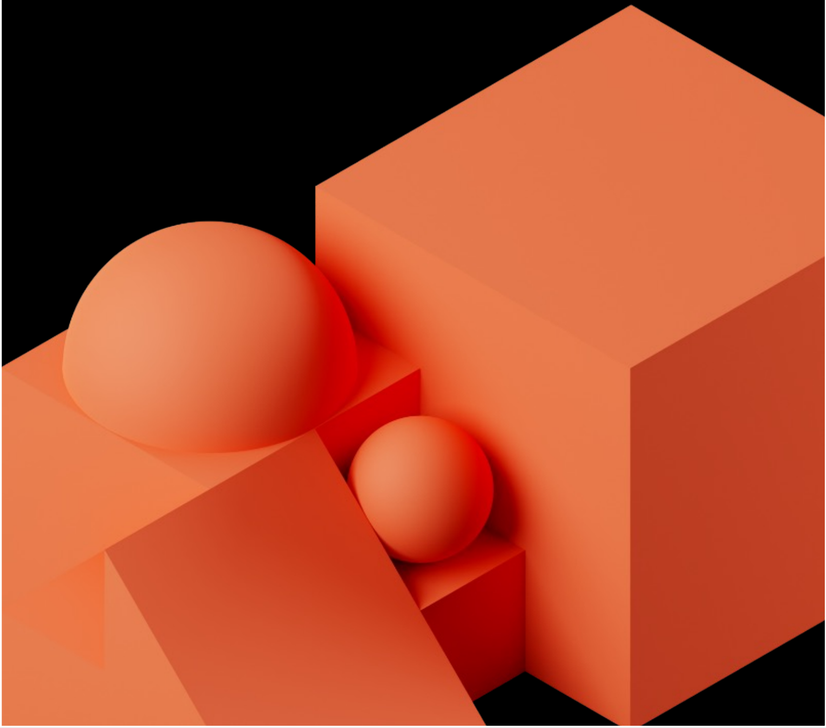
In [ ]:
```python
audio = requests.get("https://www.nasa.gov/62283main_landing.wav").content
task = "automatic-speech-recognition"
```

In [ ]:
```python
audio_transcription_pipeline = transformers.pipeline(task=task, model="openai/whisper-small")
```

In [ ]:
```python
signature = mlflow.models.infer_signature(
    audio,
    mlflow.transformers.generate_signature_output(audio_transcription_pipeline, audio),
)

inference_config = {
    "return_timestamps": "word",
    "chunk_length_s": 20,
    "stride_length_s": [5, 3],
}
```

In [ ]:
```python
with mlflow.start_run():
    model_info = mlflow.transformers.log_model(
        transformers_model=audio_transcription_pipeline,
        artifact_path="whisper_transcriber",
        signature=signature,
        input_example=audio,
        inference_config=inference_config,
```

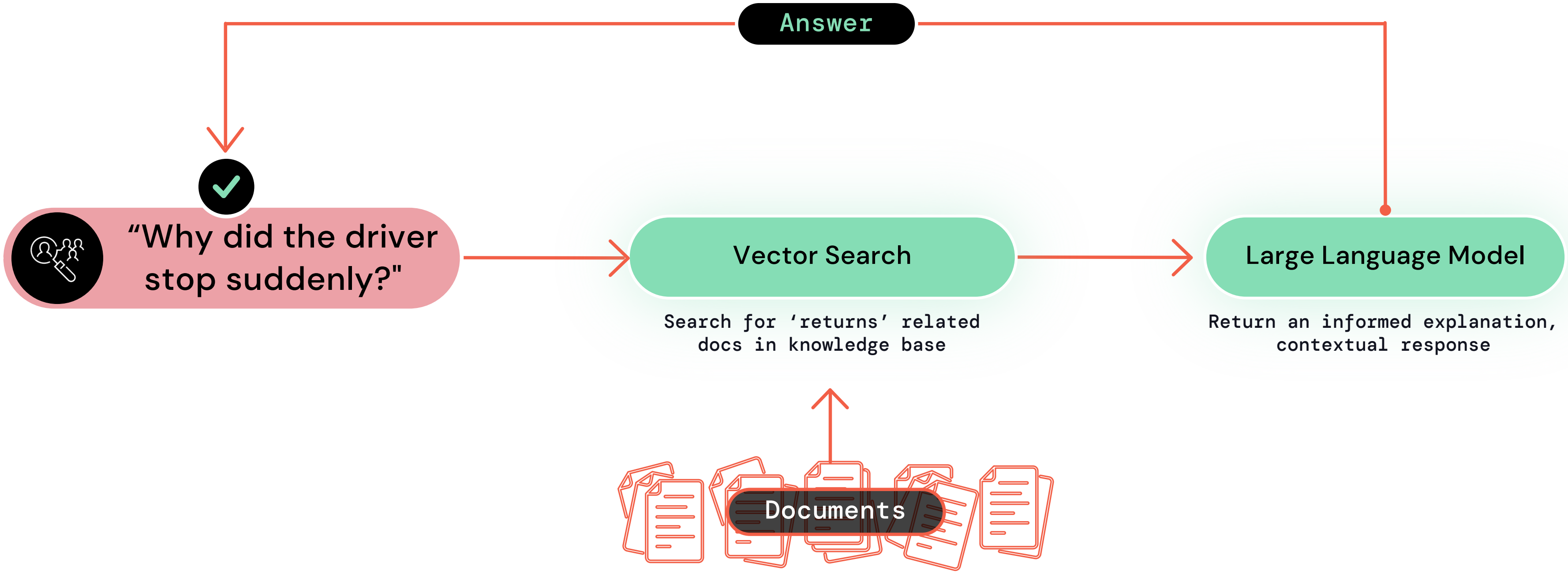
LLM
Deployment



LLM Deployment

- Multi-step inference graphs
- LLMs are reused
- Cost management is critical

Multi-step inference graphs



Answer

"Why did the driver stop suddenly?"

Vector Search

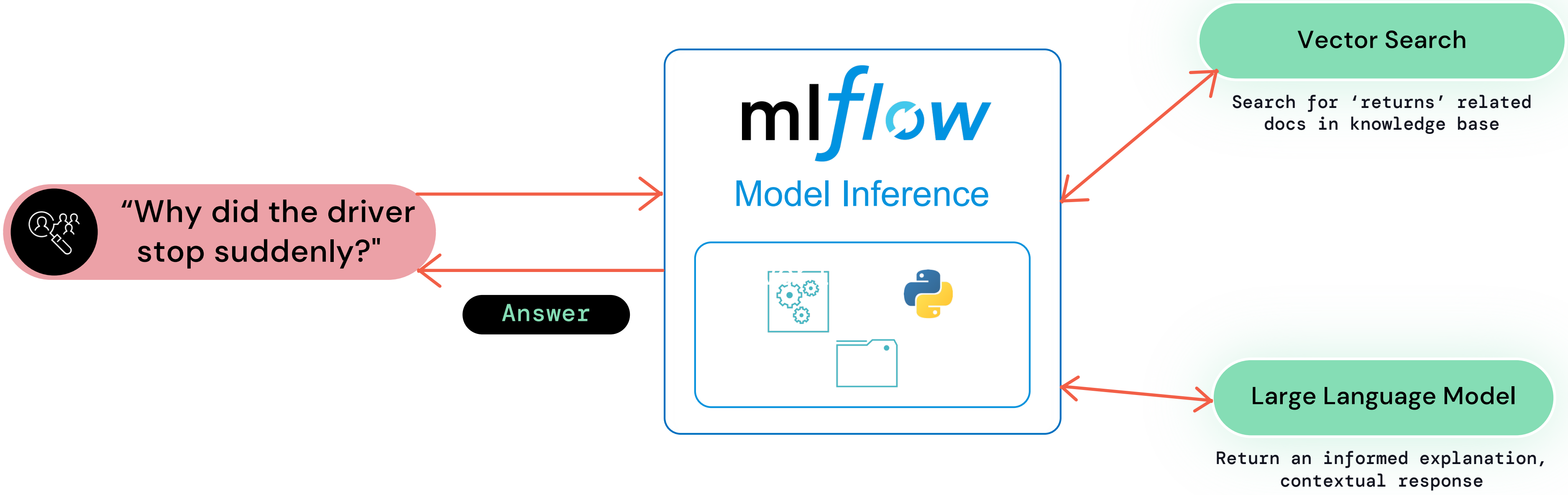
Search for 'returns' related docs in knowledge base

Large Language Model

Return an informed explanation, contextual response

Documents

Multi-step inference graphs



"Why did the driver stop suddenly?"

Answer

Model Inference

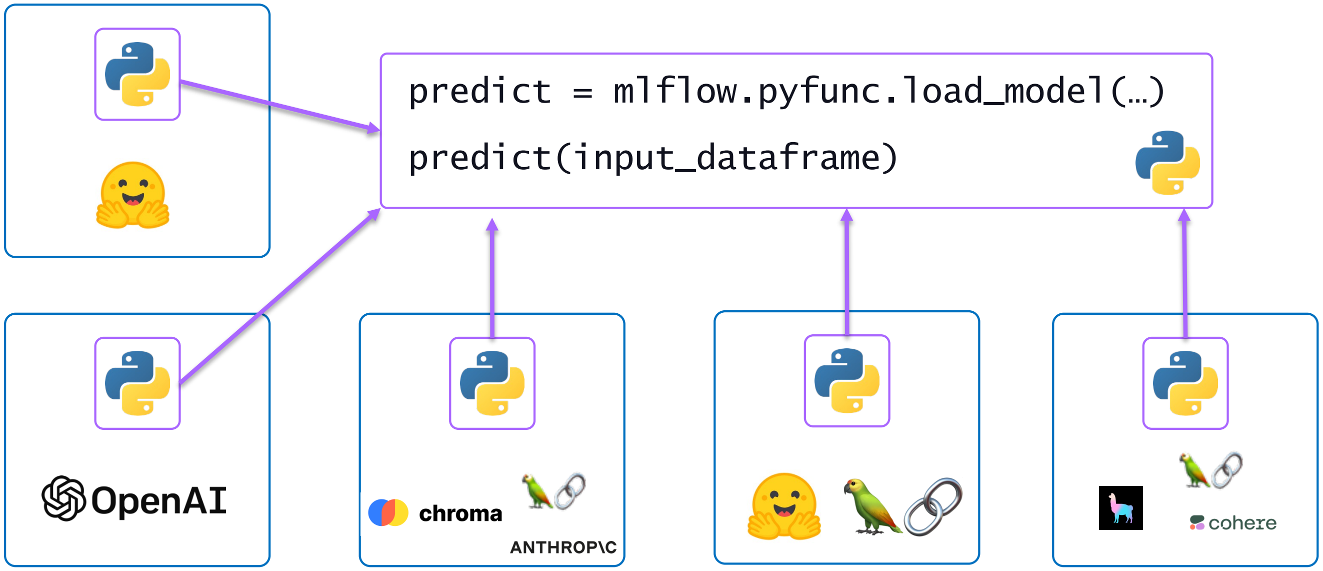
Vector Search

Search for 'returns' related docs in knowledge base

Large Language Model

Return an informed explanation, contextual response

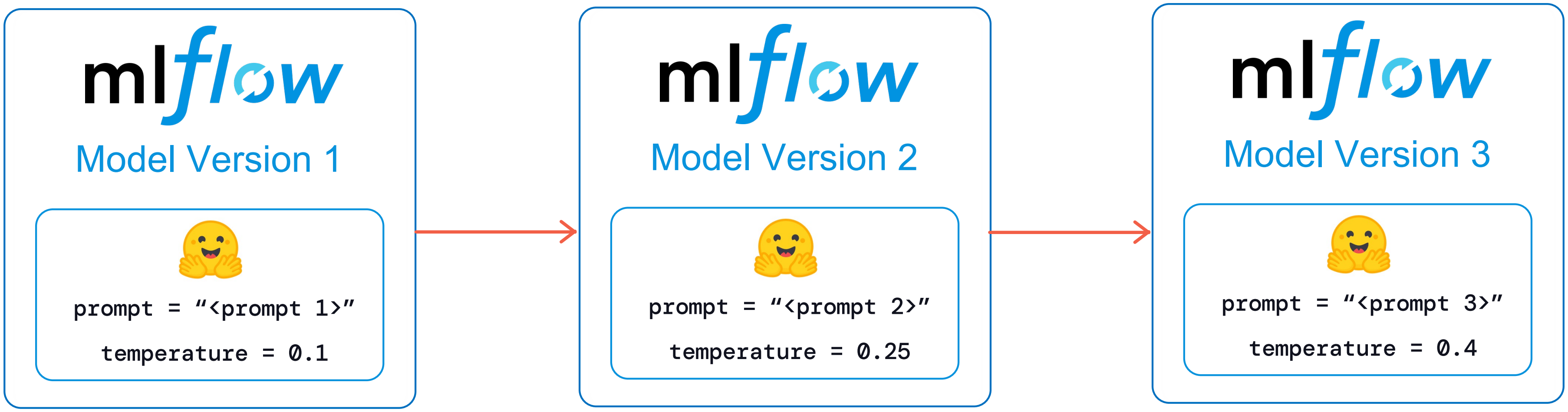
Model inference with MLflow



```
predict = mlflow.pyfunc.load_model(…)
predict(input_dataframe)
```

Reuse of LLMs

Example: Prompt engineering with OSS models



mlflow

Model Version 1

`prompt = “<prompt 1>”`

`temperature = 0.1`

mlflow

Model Version 2

`prompt = “<prompt 2>”`

`temperature = 0.25`

mlflow

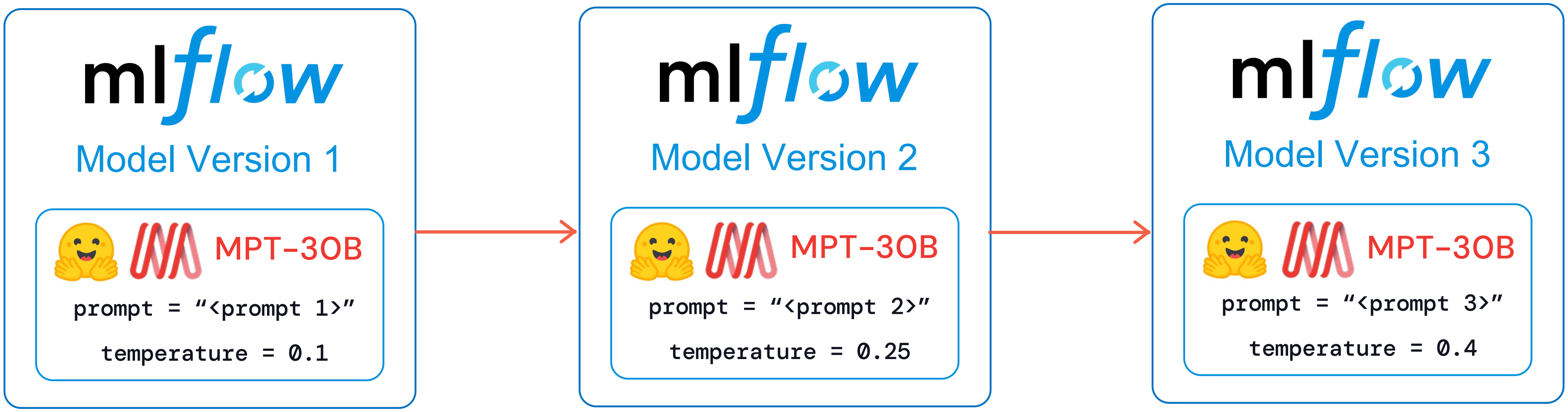
Model Version 3

`prompt = “<prompt 3>”`

`temperature = 0.4`

Reuse of LLMs

Example: Prompt engineering with OSS models



mlflow

Model Version 1

MPT-30B

`prompt = “<prompt 1>”`

`temperature = 0.1`

mlflow

Model Version 2

MPT-30B

`prompt = “<prompt 2>”`

`temperature = 0.25`

mlflow

Model Version 3

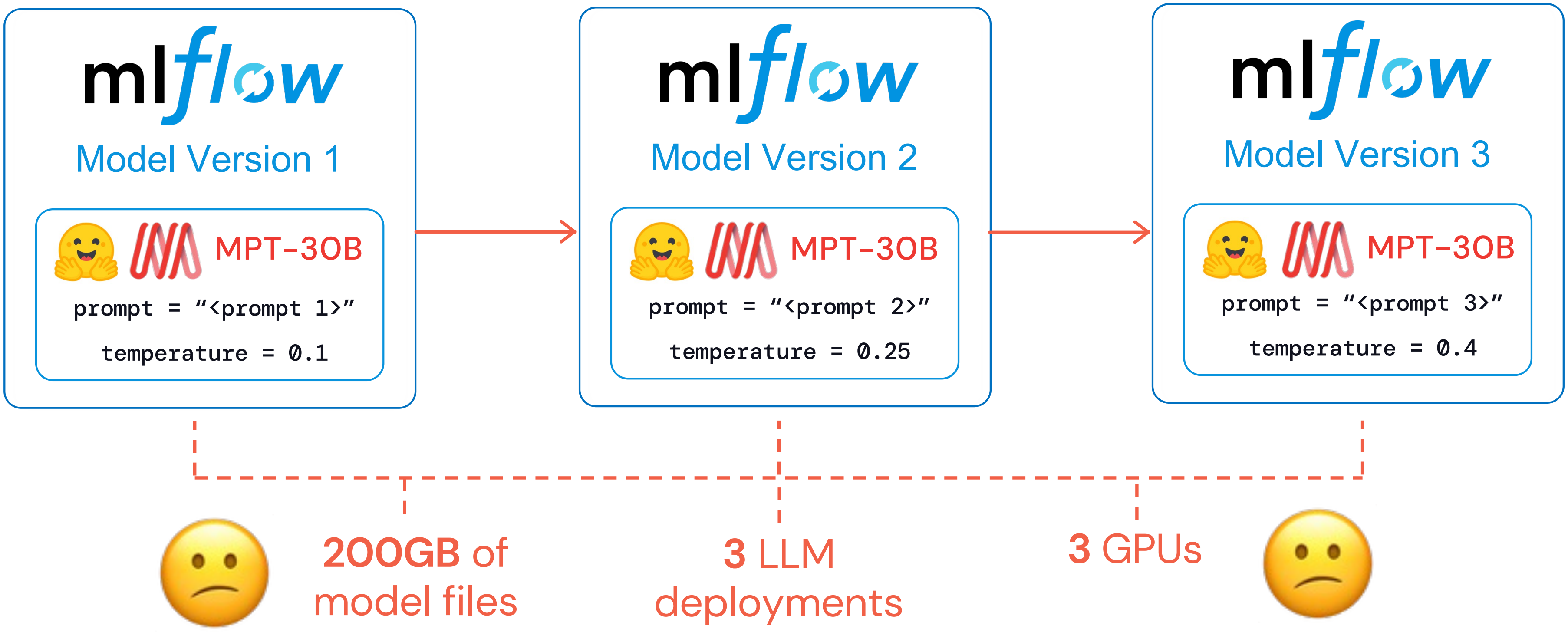
MPT-30B

`prompt = “<prompt 3>”`

`temperature = 0.4`

Reuse of LLMs

Example: Prompt engineering with OSS models



mlflow

Model Version 1

MPT-30B

`prompt = “<prompt 1>”`

`temperature = 0.1`

mlflow

Model Version 2

MPT-30B

`prompt = “<prompt 2>”`

`temperature = 0.25`

mlflow

Model Version 3

MPT-30B

`prompt = “<prompt 3>”`

`temperature = 0.4`

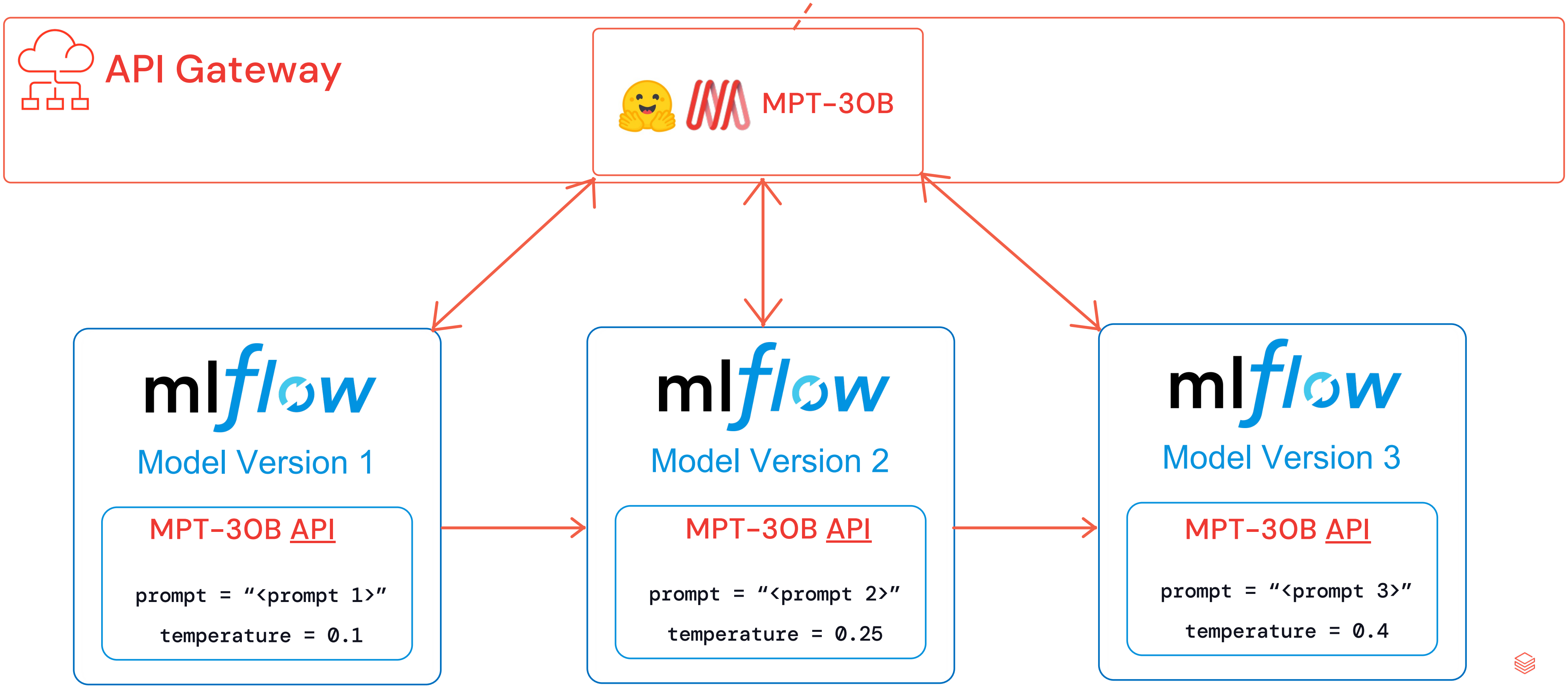
200GB of model files

3 LLM deployments

3 GPUs

Reuse of LLMs

🙂 **< 100 GB** of model files | **1** LLM deployment | **1 GPU** 🙂



API Gateway

MPT-30B

mlflow

Model Version 1

MPT-30B API

`prompt = “<prompt 1>”`

`temperature = 0.1`

mlflow

Model Version 2

MPT-30B API

`prompt = “<prompt 2>”`

`temperature = 0.25`

mlflow

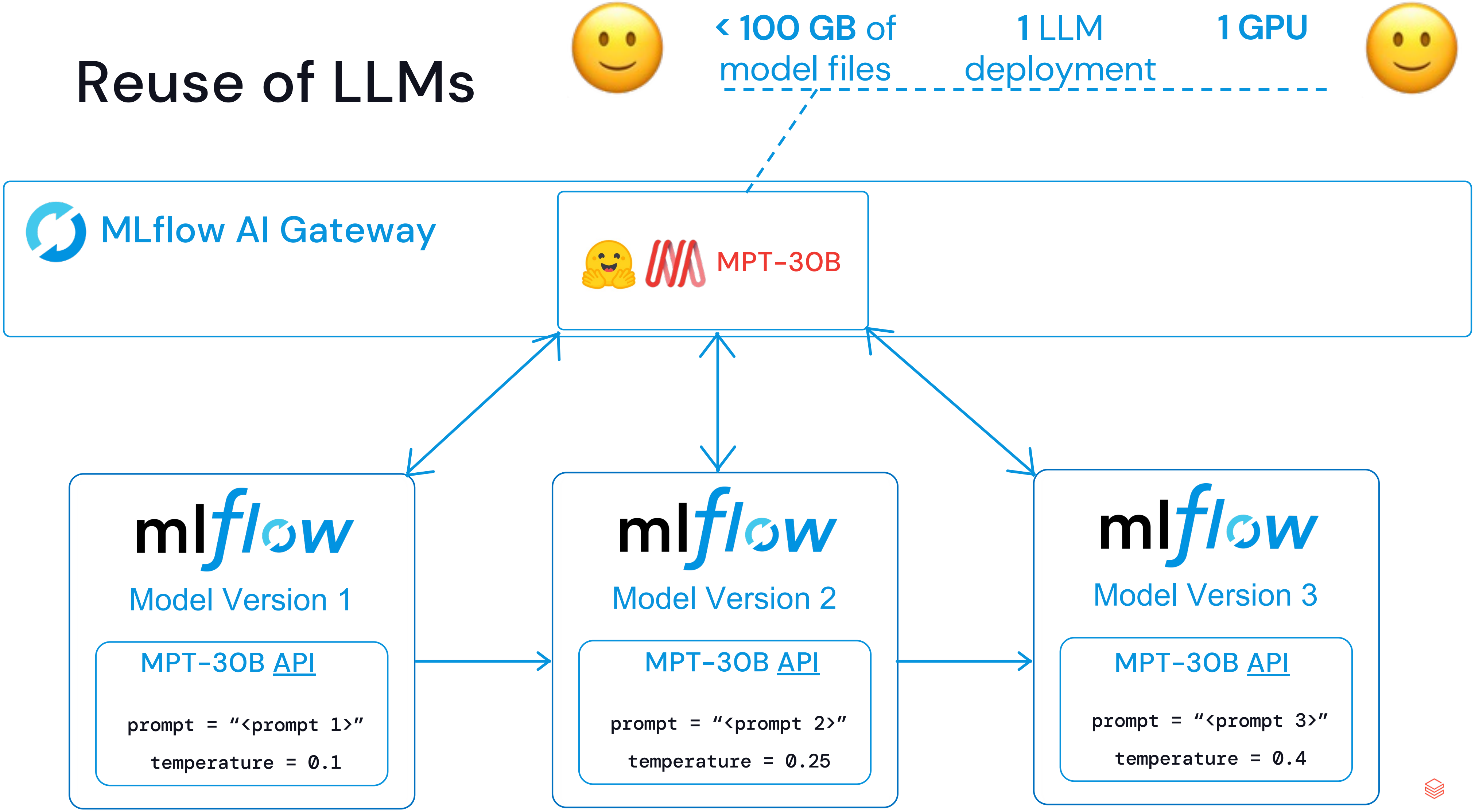
Model Version 3

MPT-30B API

`prompt = “<prompt 3>”`

`temperature = 0.4`

Reuse of LLMs



< 100 GB of model files

1 LLM deployment

1 GPU

MLflow AI Gateway

MPT-30B

mlflow

Model Version 1

MPT-30B API

`prompt = “<prompt 1>”`

`temperature = 0.1`

mlflow

Model Version 2

MPT-30B API

`prompt = “<prompt 2>”`

`temperature = 0.25`

mlflow

Model Version 3

MPT-30B API

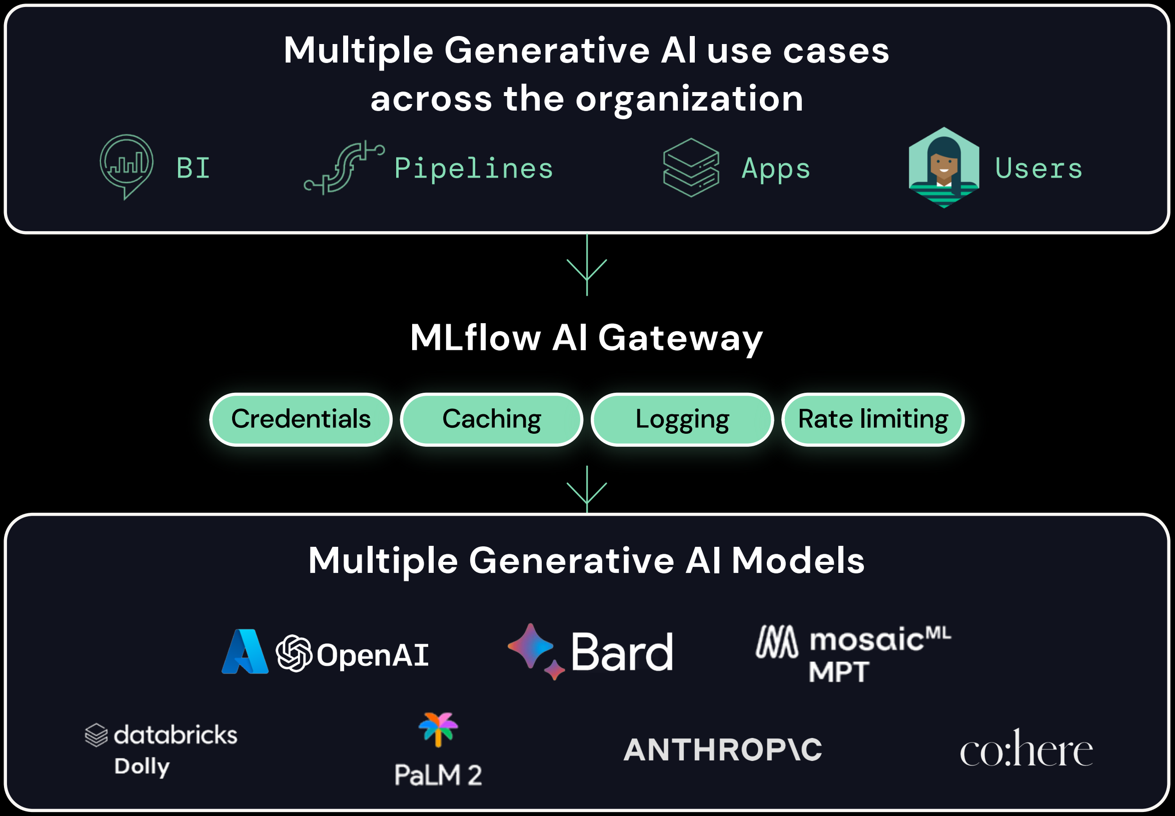
`prompt = “<prompt 3>”`

`temperature = 0.4`

INTRODUCING

MLflow AI Gateway

Manage, govern, evaluate, and switch models easily



Multiple Generative AI use cases across the organization

BI Pipelines Apps Users

MLflow AI Gateway

Credentials Caching Logging Rate limiting

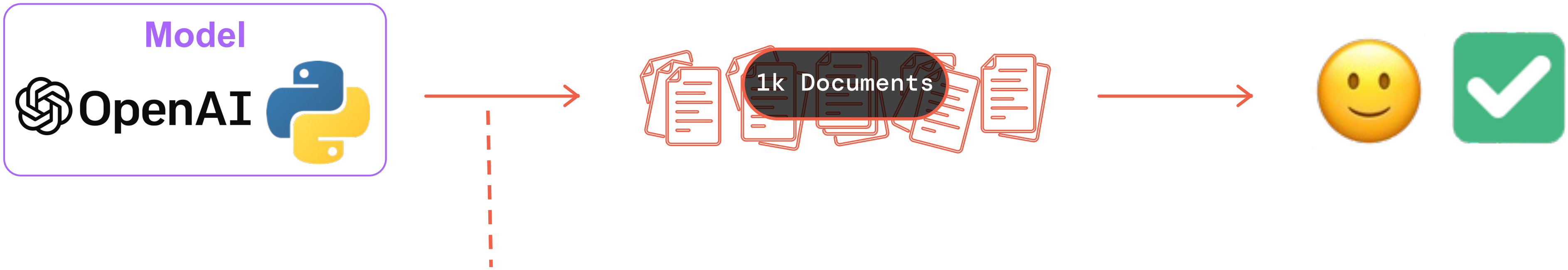
Multiple Generative AI Models

OpenAI Bard mosaicML MPT

databricks Dolly PaLM 2 ANTHROPIC co:here

MLflow AI Gateway: Cost management

Example: Document summarization



Model

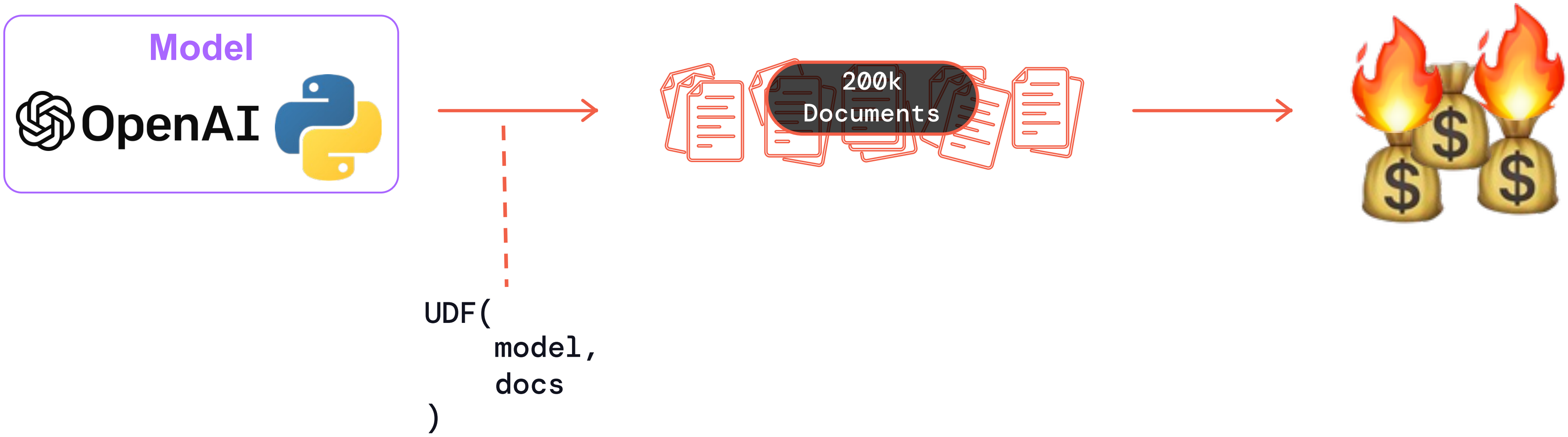
OpenAI

1k Documents

```
UDF(
    model,
    docs.limit(1000)
)
```

MLflow AI Gateway: Cost management

Example: Document summarization



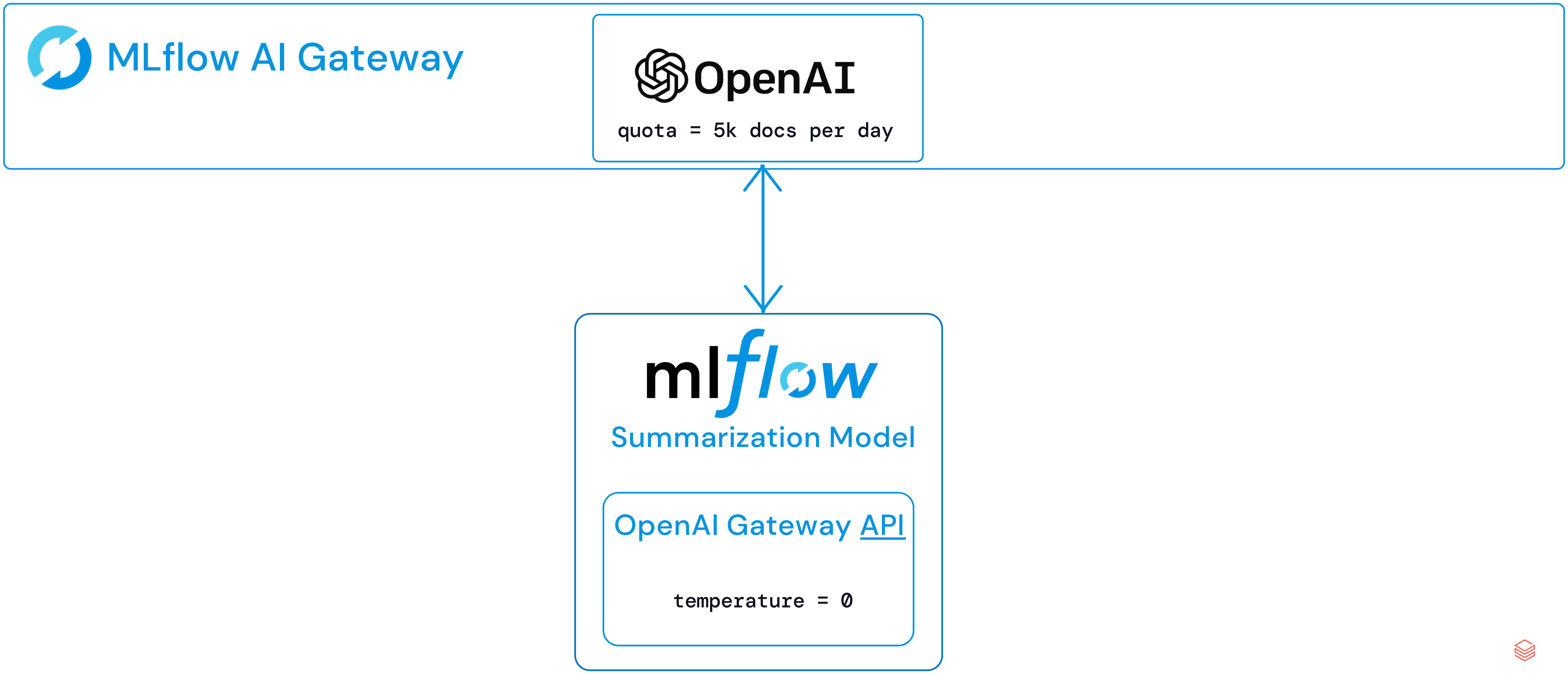
Model

OpenAI

200k Documents

```
UDF(
     model,
     docs
)
```

MLflow AI Gateway: Cost management



MLflow AI Gateway

OpenAI

`quota = 5k docs per day`

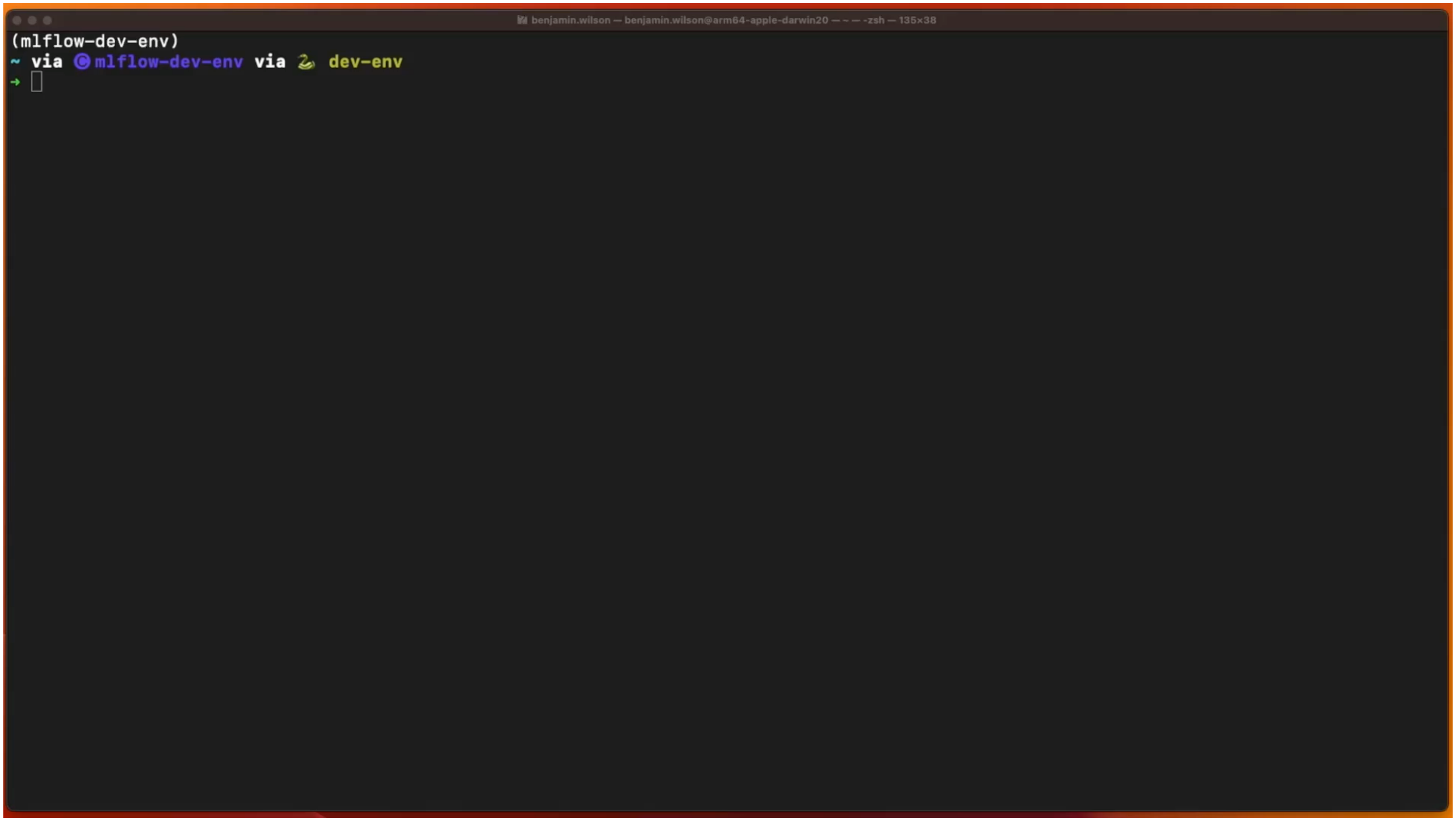
mlflow

Summarization Model

OpenAI Gateway API

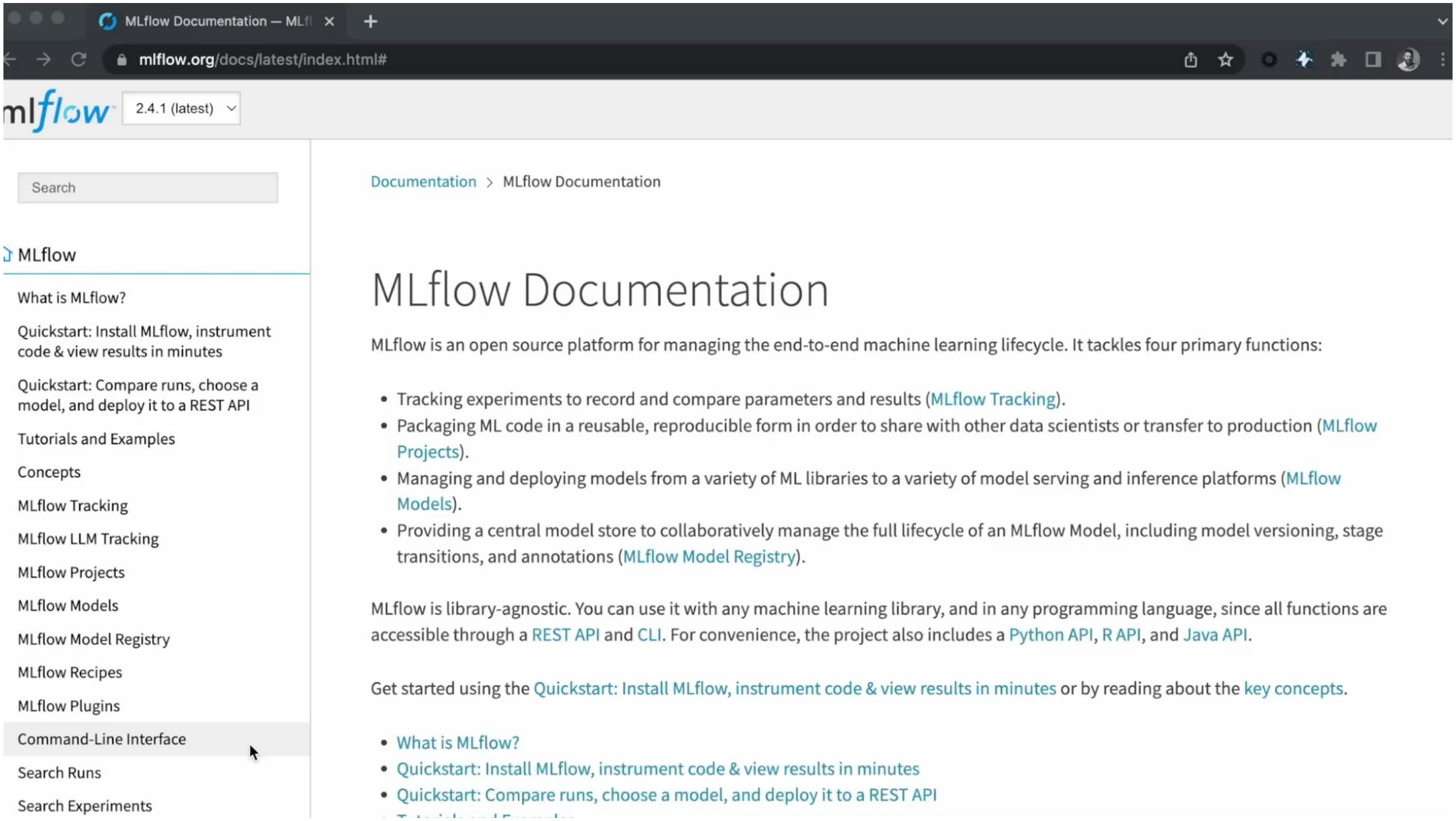
`temperature = 0`

MLflow AI Gateway: Setting configuration


benjamin.wilson — benjamin.wilson@arm64-apple-darwin20 — ~ — -zsh — 135×38

```
(mlflow-dev-env)
~ via Ⓒmlflow-dev-env via 🐍 dev-env
➜ 
```

MLflow AI Gateway: Docs demo



MLflow Documentation

MLflow is an open source platform for managing the end-to-end machine learning lifecycle. It tackles four primary functions:

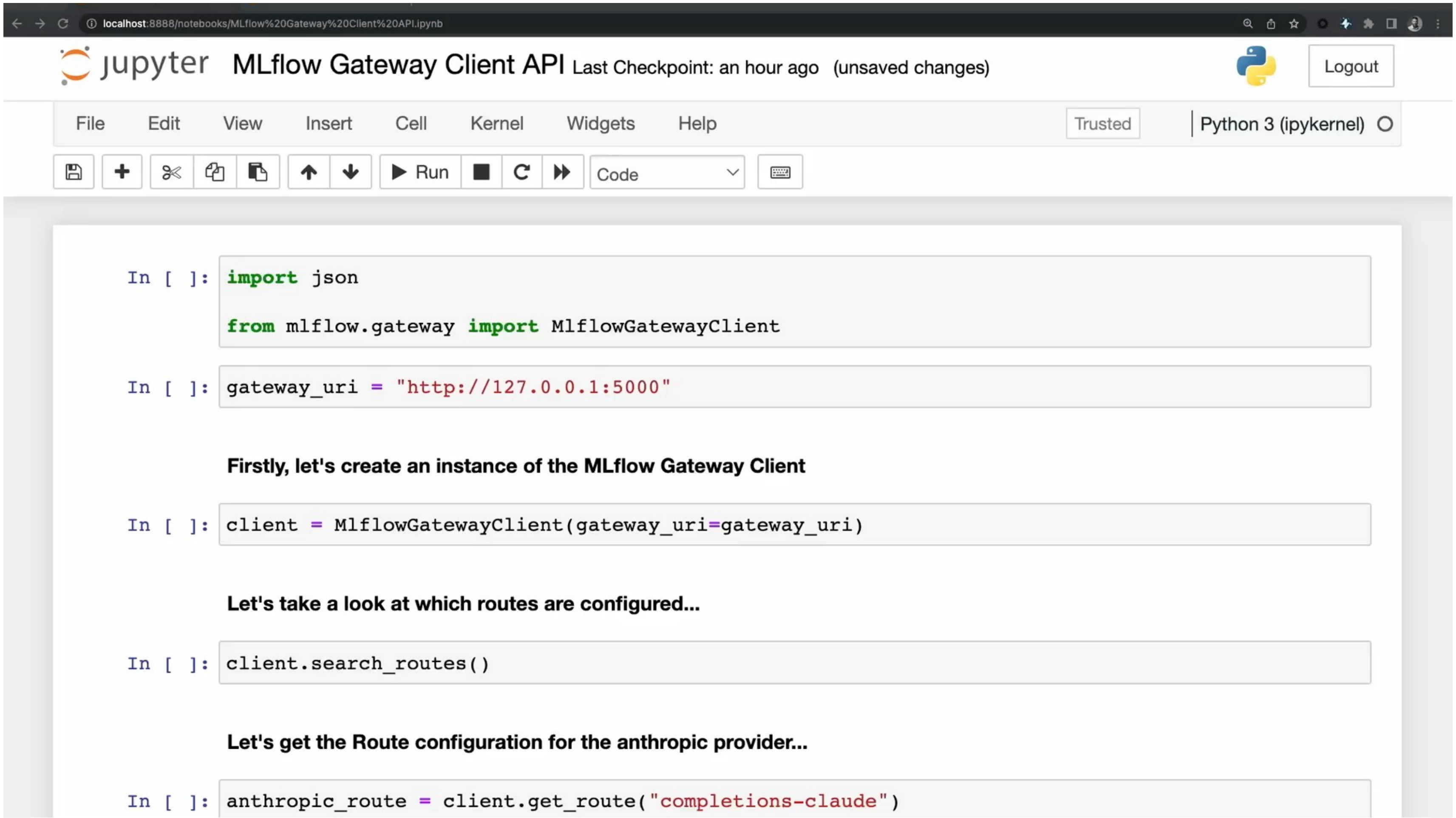
- Tracking experiments to record and compare parameters and results (MLflow Tracking).
- Packaging ML code in a reusable, reproducible form in order to share with other data scientists or transfer to production (MLflow Projects).
- Managing and deploying models from a variety of ML libraries to a variety of model serving and inference platforms (MLflow Models).
- Providing a central model store to collaboratively manage the full lifecycle of an MLflow Model, including model versioning, stage transitions, and annotations (MLflow Model Registry).

MLflow is library-agnostic. You can use it with any machine learning library, and in any programming language, since all functions are accessible through a REST API and CLI. For convenience, the project also includes a Python API, R API, and Java API.

Get started using the Quickstart: Install MLflow, instrument code & view results in minutes or by reading about the key concepts.

- What is MLflow?
- Quickstart: Install MLflow, instrument code & view results in minutes
- Quickstart: Compare runs, choose a model, and deploy it to a REST API

MLflow AI Gateway: Client API demo


localhost:8888/notebooks/MLflow%20Gateway%20Client%20API.ipynb
jupyter
MLflow Gateway Client API Last Checkpoint: an hour ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [ ]: import json

        from mlflow.gateway import MlflowGatewayClient
```

```
In [ ]: gateway_uri = "http://127.0.0.1:5000"
```

Firstly, let's create an instance of the MLflow Gateway Client

```
In [ ]: client = MlflowGatewayClient(gateway_uri=gateway_uri)
```

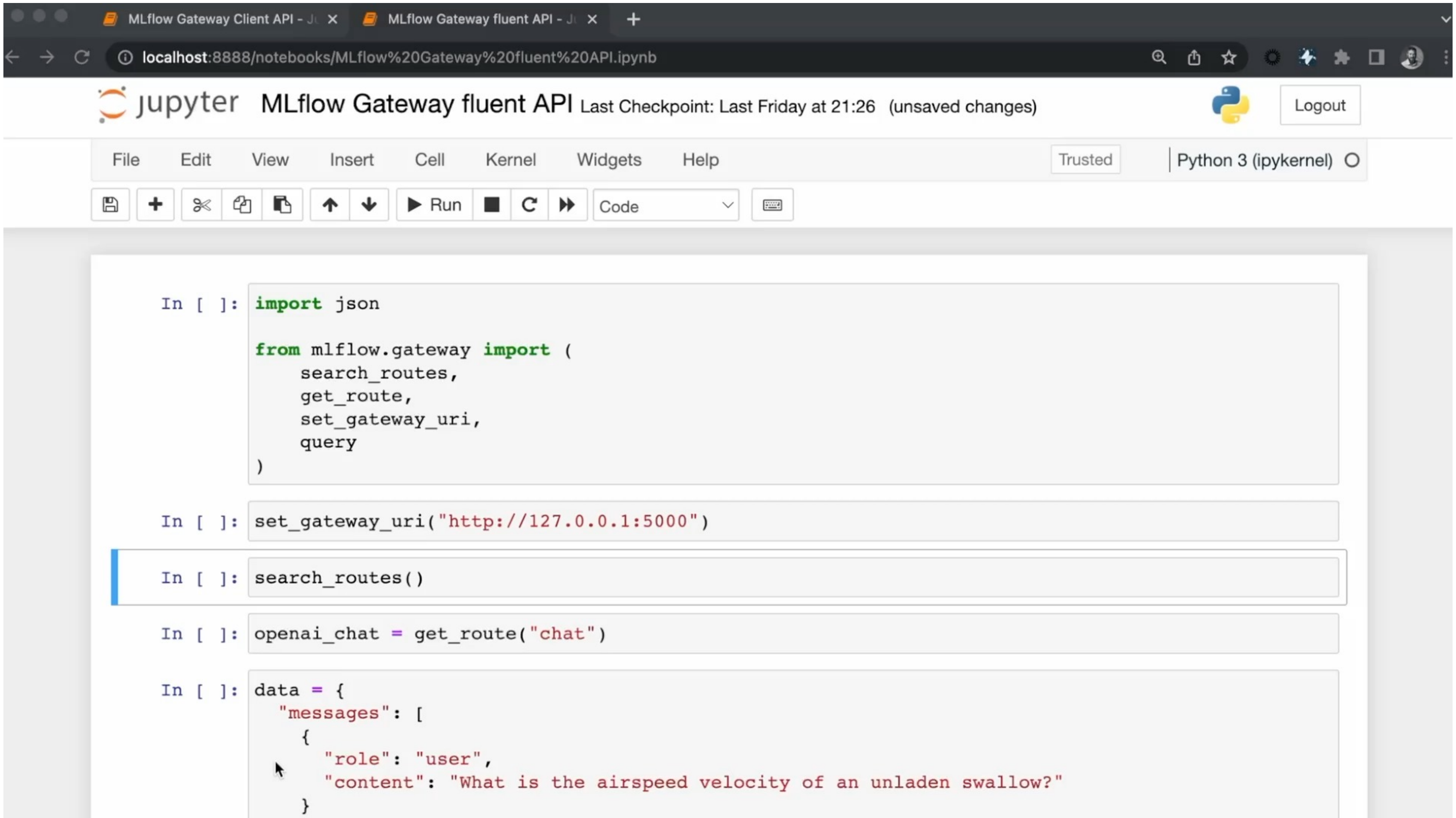
Let's take a look at which routes are configured...

```
In [ ]: client.search_routes()
```

Let's get the Route configuration for the anthropic provider...

```
In [ ]: anthropic_route = client.get_route("completions-claude")
```

MLflow AI Gateway: fluent API demo


MLflow Gateway Client API - J
MLflow Gateway fluent API - J
localhost:8888/notebooks/MLflow%20Gateway%20fluent%20API.ipynb
jupyter
MLflow Gateway fluent API
Last Checkpoint: Last Friday at 21:26 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

In [ ]:

```python
import json

from mlflow.gateway import (
    search_routes,
    get_route,
    set_gateway_uri,
    query
)
```

In [ ]:

```python
set_gateway_uri("http://127.0.0.1:5000")
```

In [ ]:

```python
search_routes()
```

In [ ]:

```python
openai_chat = get_route("chat")
```

In [ ]:

```python
data = {
  "messages": [
    {
      "role": "user",
      "content": "What is the airspeed velocity of an unladen swallow?"
    }
```

MLflow 2.5 is coming soon

- AI Gateway
- Prompt engineering UI
- Improved evaluation experience for LLMs
- Inference parameters for LLMs in MLflow Models

MLflow for LLMOps: Getting started

- `pip install mlflow`
- GitHub repository: https://github.com/mlflow/mlflow
- Website: mlflow.org
- Community Slack: https://go.mlflow.org/slack
- LLM examples: https://github.com/mlflow/mlflow/tree/master/examples/llms