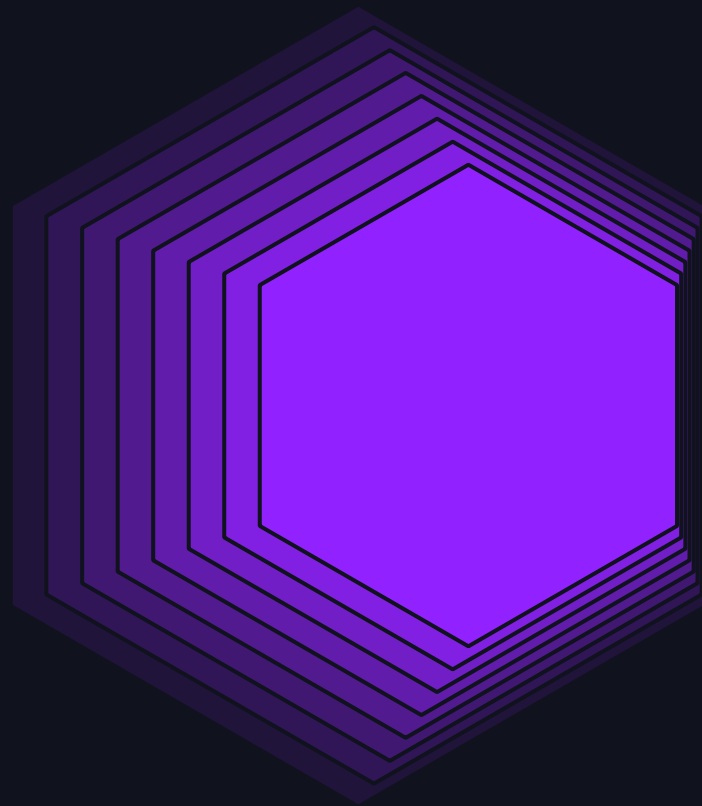


DATABRICKS DATA INTELLIGENCE PLATFORM: INTRODUCTORY OVERVIEW

Ari Kaplan
6/21/24



BOOK PREMIER!

Brought to you by:



The Data Intelligence Platform

for
dummies
A Wiley Brand

Democratize data &
AI with intelligence

—
Understand enterprise
data with AI

—
Accelerate innovation
with ETL, DW, BI, & AI



Ari Kaplan
Stephanie Diamond

Databricks Special Edition



ABOUT ME



LAKEHOUSE FOUNDATION

Data Maturity Curve



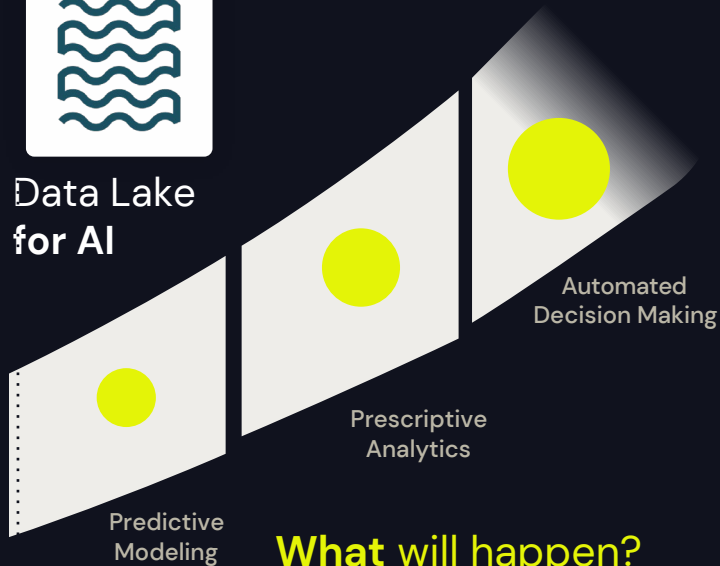
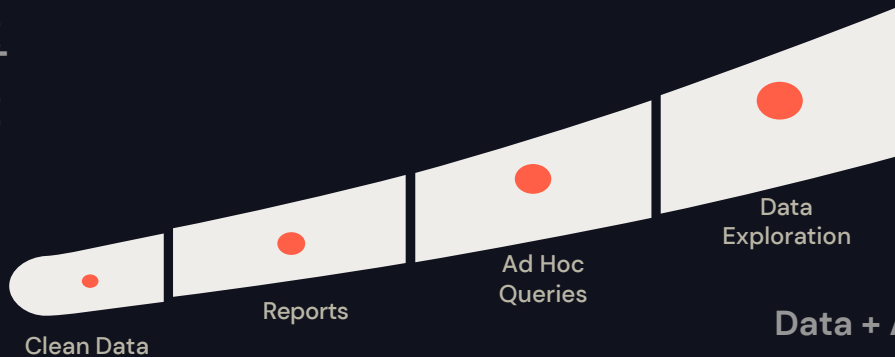
Data Warehouse
for BI



Data Lake
for AI

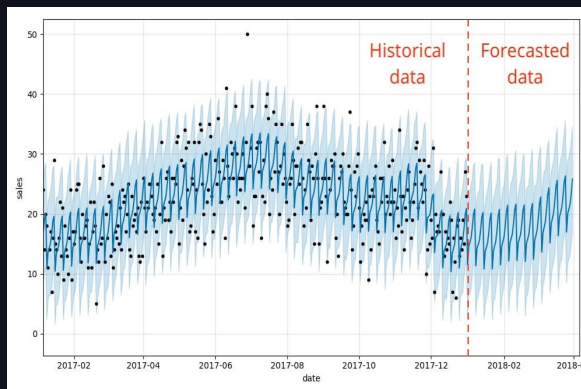
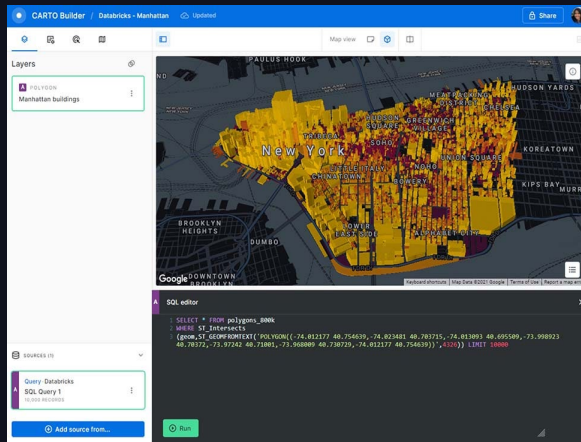
Competitive Advantage

What happened?



WHY MULTIMODAL?

Better Predictions



Data Lake



Machine Learning



Streaming



Generative AI



Most companies struggle
with **Data + AI**

Data Science



Governance



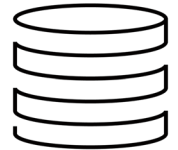
Orchestration
& ETL



BI



Data Warehouse



You stitch together 6+ services to succeed

Data Lake



Machine Learning



Streaming



Generative AI



Data and AI
are siloed

Data privacy & control
are challenged

Dependent on highly
technical staff

Data S



Governance



Orchestration
& ETL



BI



Warehouse



Data silos drive high
operational costs

Inconsistent policies
reduce trust in the data

Disparate tools slow down
cross-team productivity

Data Lakehouse

An open, unified foundation
for all your data



Generative AI

Easily scale and use data and AI



Data Intelligence Platform

Democratize data + AI across
your entire organization



The Databricks Data Intelligence Platform

Data-centric AI

Gen AI

- Custom models
- Model serving
- RAG

End-to-end AI

- MLOps (MLflow)
- AutoML
- Monitoring
- Governance

Data Science
& AI

Mosaic AI

ETL &
Real-time Analytics

Delta Live Tables

Orchestration

Workflows

Data
Warehousing & BI

Databricks SQL

An AI powered data intelligence engine to understand the semantics of your data

DatabricksIQ

Unified security, governance, and cataloging

Unity Catalog

Unified data storage for reliability and sharing

Delta Lake UniForm

Open Data Lake

All Raw Data
(Logs, Texts, Audio, Video, Images)



The Databricks Data Intelligence Platform... INTELLIGENCE EVERYWHERE

Data-centric AI

Gen AI

- Custom models
- Model serving
- RAG

End-to-end AI

- MLOps (MLflow)
- AutoML
- Monitoring
- Governance

Mosaic AI

Create, tune, and serve custom LLMs

Delta Live Tables

Automated data quality

Workflows

Job cost optimized based on past runs

Databricks SQL

Text-to-SQL

Use generative AI to understand the semantics of your data

Data Intelligence Engine

Unity Catalog

Securely get insights in natural language

Delta Lake Uniform

Data layout is automatically optimized based on usage patterns

Open Data Lake

All Raw Data
(Logs, Texts, Audio, Video, Images)



Good Data for Good AI: Unified Data and AI Governance



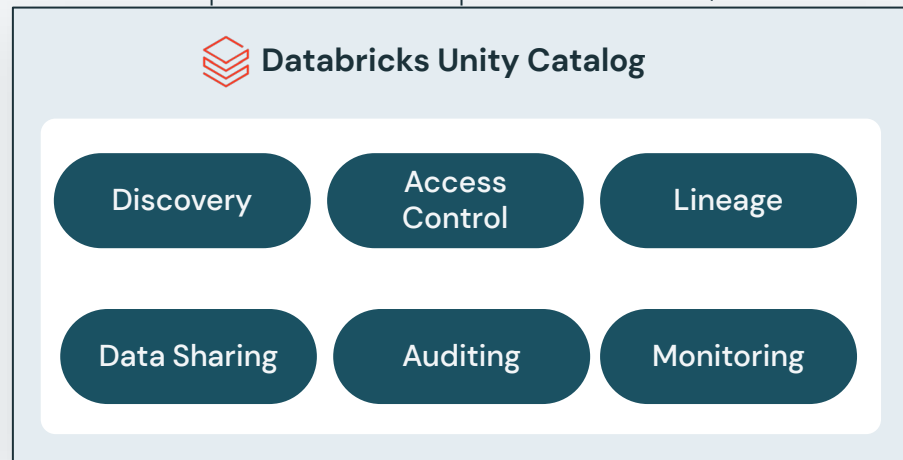
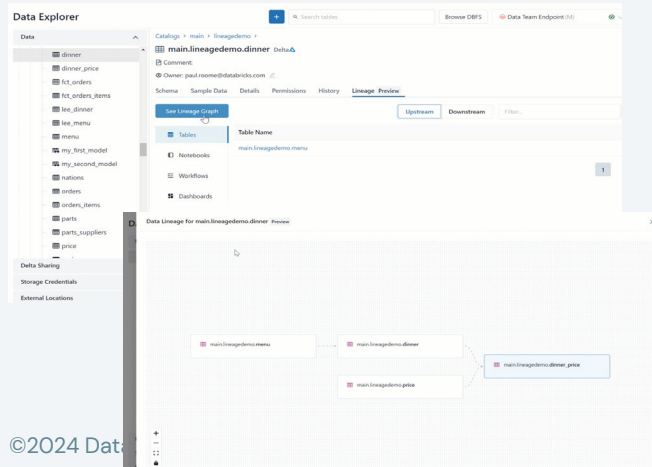
Modern Data & AI Governance

Unified visibility into data and AI

Single permission model for data and AI

AI-powered monitoring and observability

Open data sharing



Tables

Models

Dashboards



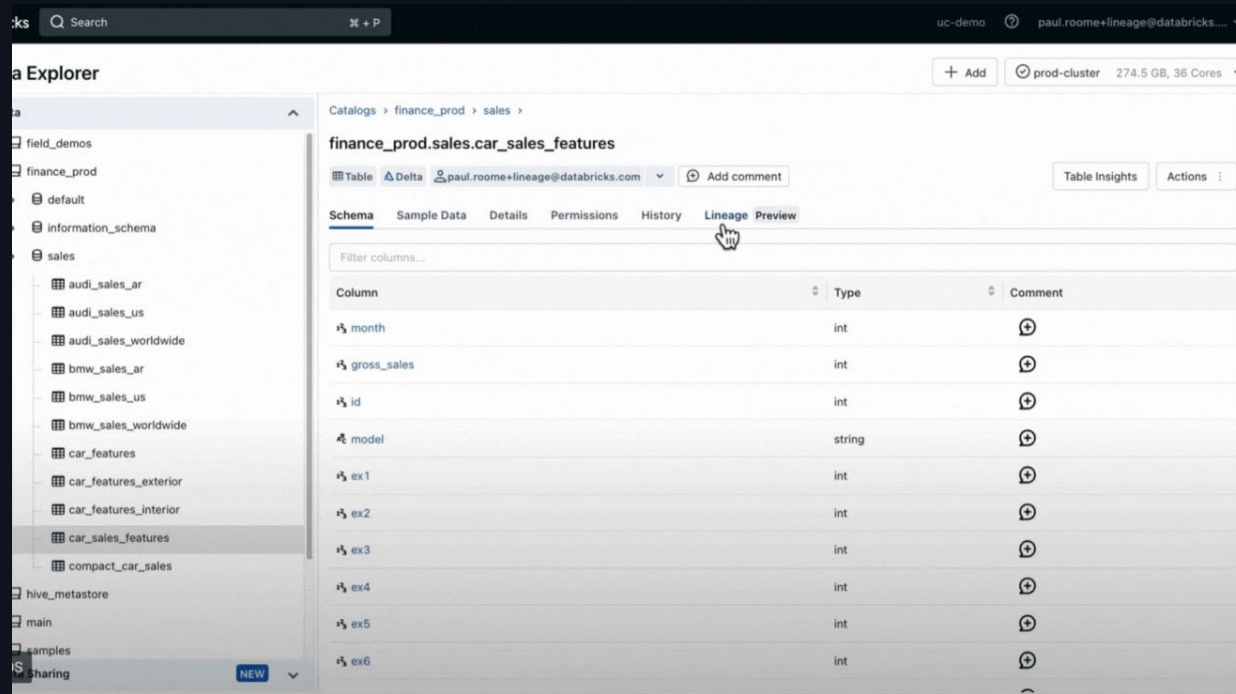
Files



Notebooks



Unified Data and AI Governance



The screenshot shows the Databricks Data Explorer interface. The left sidebar contains a navigation tree with folders like 'field_demo', 'finance_prod', 'information_schema', and 'sales'. The main area displays the 'finance_prod.sales.car_sales_features' table. The 'Lineage Preview' tab is active, showing a table with the following columns and types:

Column	Type	Comment
month	int	
gross_sales	int	
id	int	
model	string	
ex1	int	
ex2	int	
ex3	int	
ex4	int	
ex5	int	
ex6	int	

Automatically capture data lineage down to your tables and columns, in real time, with Unity Catalog powered by the Data Intelligence Engine.

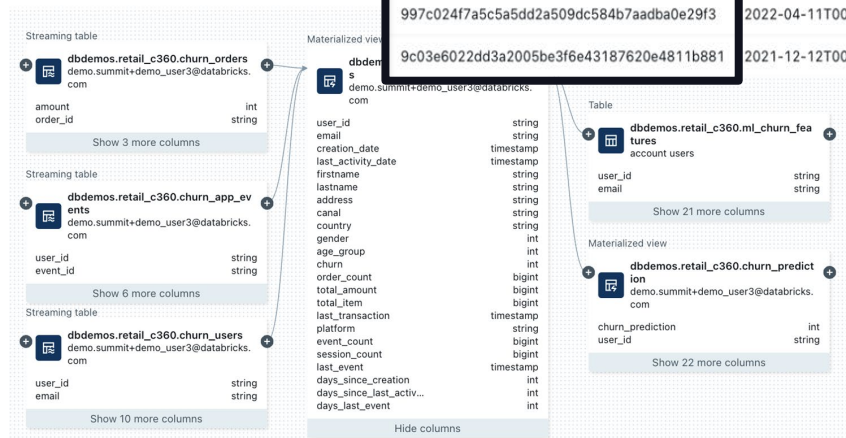
- Monitor sensitive data for compliance requirements and audit reporting
- Ensure data quality across all workloads
- Perform impact analysis or change management on any data changes



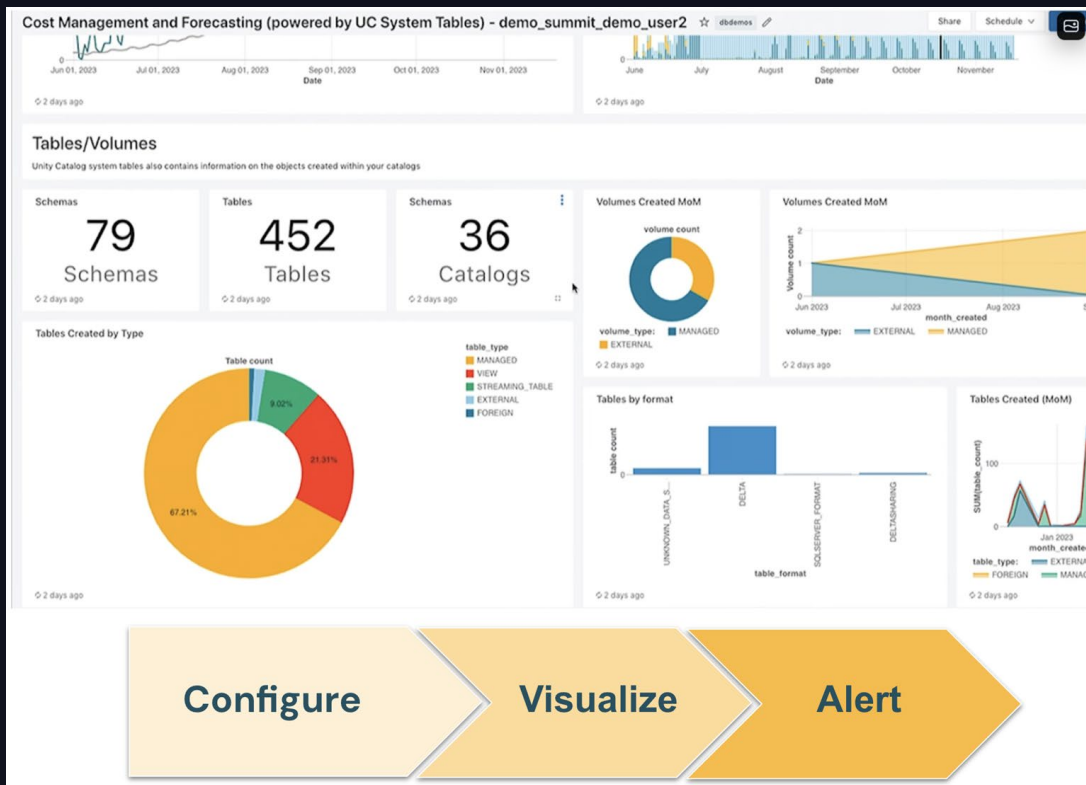
Data Privacy & Lineage

email	creation_date	last_activity_date	firstname	lastname	address	canal	country	gender
33cc7abf40895d004ff14849ed3a446adcf87ad	2022-01-13T00:00:00.000Z	2023-06-07T22:45:56.000Z	Kyle	Parker	22334 Stacey Valley Suite 296 Jasonland, WI 99608	WEBAPP	FR	1
67140dc307481c1184efe2d89a02e49a4d0b20e	2015-03-25T00:00:00.000Z	2023-06-03T21:06:16.000Z	Matthew	Cook	937 Mcintyre Heights Suite 968 Lake Christopherbury, TX 40399	WEBAPP	SPAIN	1
95ea0551c48706b6bdc2004478bbd0a33492c642	2022-03-16T00:00:00.000Z	2023-06-08T09:01:25.000Z	Lee	Edwards	846 Smith Ridges Apt. 149 South Elizabeth, MD 93813	WEBAPP	USA	1
ea2fa260e87ca3d080afebdd53984429e993f60a	2021-12-25T00:00:00.000Z	2023-06-06T19:14:42.000Z	Rhonda	Gaines	05732 Robert Fields Apt. 789 Port Kaylafor, AR 35664	PHONE	SPAIN	0
4c9d435c969c93c532eb0a752c0a05b568520b45	2021-09-06T00:00:00.000Z	2023-06-05T02:12:09.000Z	Todd	Tucker	2998 Scott Overpass East Williamville, GA 19474	WEBAPP	SPAIN	1
98ab65890d5087d383ca4fa91f22a2d5a24a4c31	2021-07-29T00:00:00.000Z	2023-06-06T16:34:34.000Z	Ernest	Moore	127 Webb Drive Elizabethbury, CT 65649	WEBAPP	FR	1
fb87b78b9f0a7fe141cbc305f97e7b0f1e5f651d	2021-12-31T00:00:00.000Z	2023-06-02T08:41:10.000Z	Patricia	Mcclain	9889 Decker Village West Maryfort, AS 89347	PHONE	USA	1
95ea0551c48706b6bdc2004478bbd0a33492c642	2022-02-14T00:00:00.000Z	2023-06-08T09:01:25.000Z	Lee	Edwards	846 Smith Ridges Apt. 149 South Elizabeth, MD 93813	WEBAPP	FR	1
df52ba442625df2e109d1ca2a894efe6dfe82115e	2022-03-02T00:00:00.000Z	2023-06-06T00:50:01.000Z	David	Anderson	66475 Norma Island Apt. 113 Jessicahaven, MH 19301	WEBAPP	USA	1
997c024f7a5c5a5dd2a509dc584b7aadba0e29f3	2022-04-11T00:00:00.000Z	2023-06-08T16:27:48.000Z	David	Martinez	533 Quinn Shoal Moorechester, SD 41317	WEBAPP	USA	1
9c03e6022dd3a2005bfe3f6e43187620e4811b881	2021-12-12T00:00:00.000Z	2023-06-01T13:47:23.000Z	Linda	Lara	978 Dodson Camp Cookmouth, IN 17720	PHONE	SPAIN	0

Data Lineage for dbdemos.retail_c360.churn



Lakehouse Monitoring



Track your data pipelines and AI projects with Lakehouse Monitoring. This single, unified approach to monitoring includes:

- Proactive and simplified **detection of anomalies** in your data and models
- Auto-generated metrics that provide **visibility into all your data and models**
- Automated **RCA**s to debug data and model quality issues



DEMO TIME!



AI Functions

Write SQL to get insight from unstructured text data via LLMs

```
1 CREATE
2 OR REPLACE FUNCTION summarize(text STRING) RETURNS STRING RETURN llm_generate(
3   concat('Summarize this to 1 sentence: ', text),
4   'openai/gpt-3.5-turbo',
5   'apiToken',
6   secret('username', 'openai_api_token'),
7   'temperature',
8   0.0
9 );
10 SELECT
11   product_name,
12   summarize(long_product_description) AS product_summary
13 FROM
14   products;
```

Sample use cases

- Extract top product issues from call center transcripts—without manual tagging!
- Tag customers as a potential churn risk based on customer support chat logs
- Generate customized product descriptions for ad campaigns—automatically
- Read product reviews to understand buying decision criteria

...many more...



Databricks Predictive I/O

AI Performance & Scale

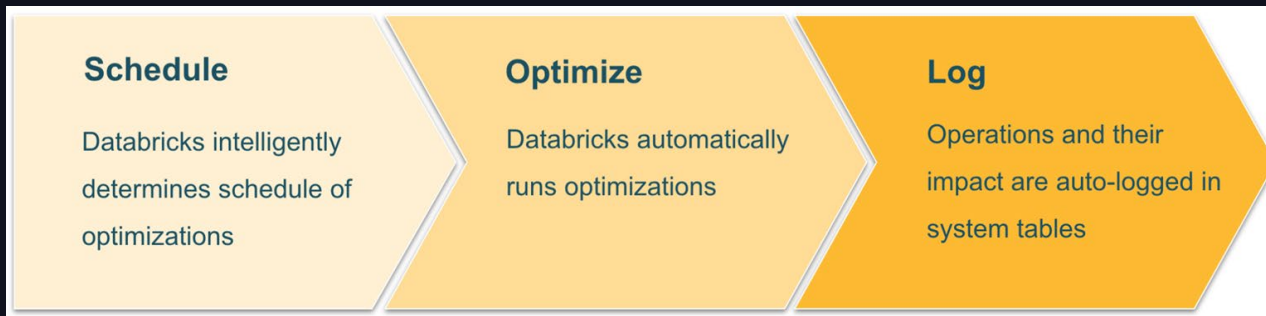


AI-Improved Performance Optimizations: Predictive I/O

Which optimizations to run?
What tables? How often?

Our Data Intelligence Engine helps [answer those questions](#) with **automated optimizations** that analyze:

- Historical read and write patterns to intelligently build indexes
- File sizes and clustering by running **OPTIMIZE**, **VACUUM**, **ANALYZE** and **CLUSTERING** commands for you

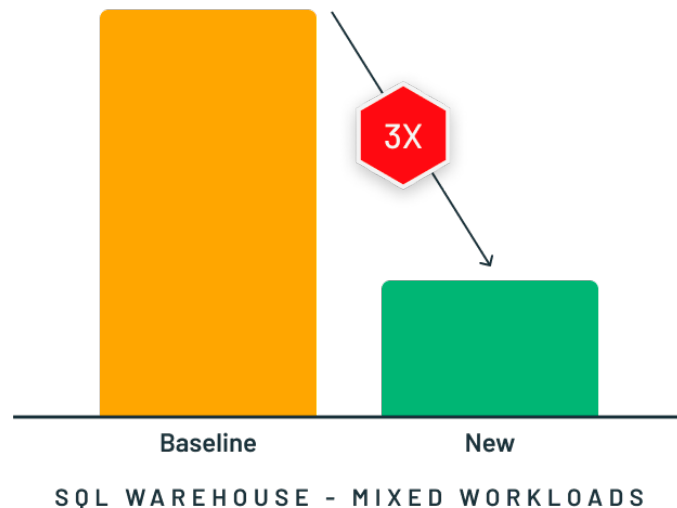


Intelligent Workload Management

AI Powered Simplicity

Continuously learning from workloads history to determine if it should prioritize a new query to run immediately or scale up to run it without disrupting running queries

Mixed Workloads Query Latency
Lower is better



LLM PRIMER



GenAI journey

Plan an iterative path from basic to advanced GenAI, leveraging your data.



Prompt
engineering

Crafting specialized prompts and pipelines to guide GenAI behavior



Retrieval
augmented
generation
(RAG)

Combining an LLM with custom enterprise data



Fine-tuning

Adapting a pre-trained GenAI model to specific data sets or domains







Pre-training

Training a GenAI model from scratch

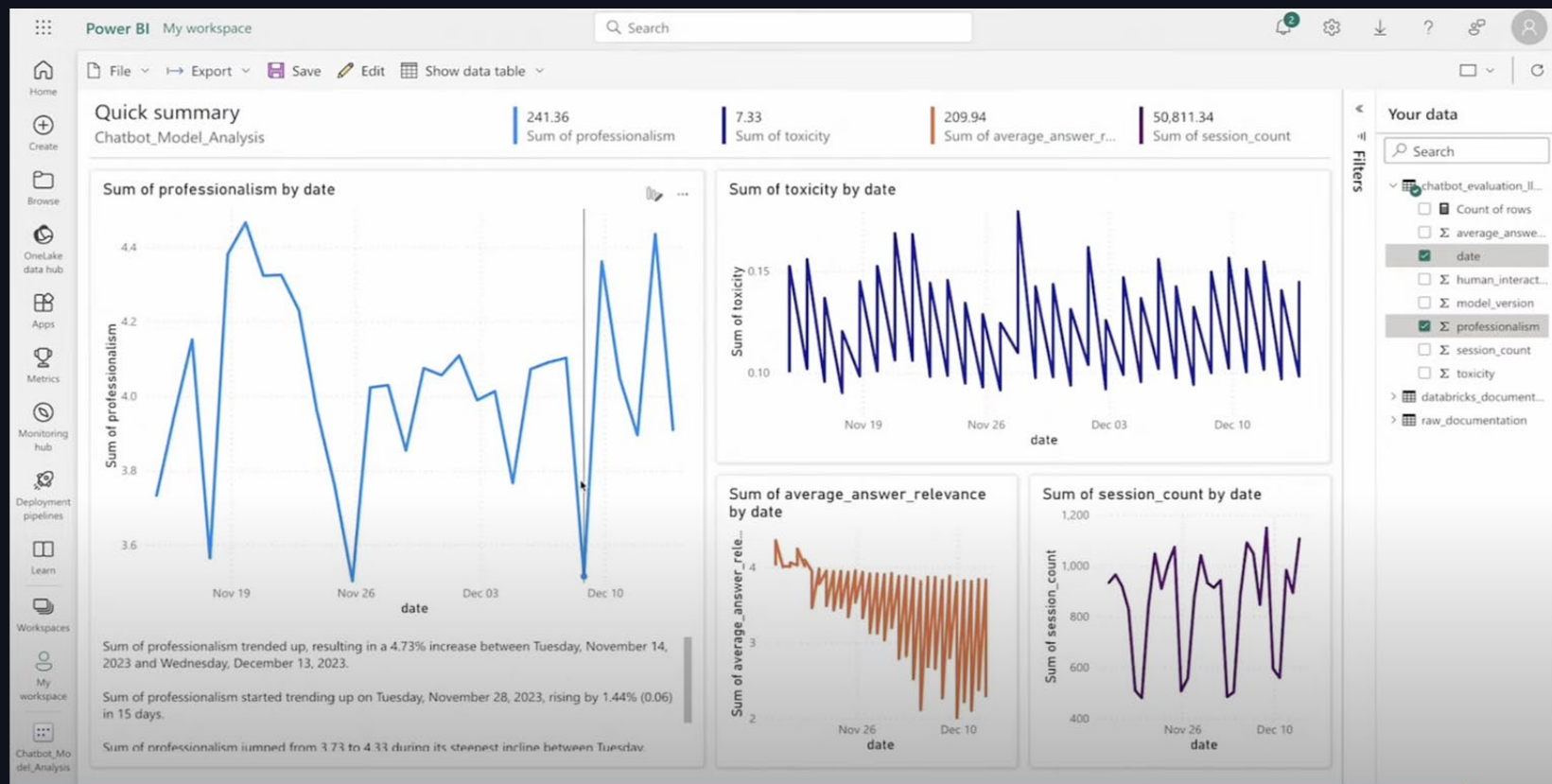
Complexity/Compute-intensiveness

Overview of maturity stages of implementing LLMs

Method	Definition	Primary use case	Data requirements	Advantages	Considerations
 Prompt engineering	Crafting specialized prompts to guide LLM behavior	Quick, on-the-fly model guidance	None	Fast, cost-effective, no training required	Less control than fine-tuning
 Retrieval augmented generation (RAG)	Combining an LLM with external knowledge retrieval	Dynamic data sets and external knowledge	External knowledge base or database (e.g., vector database)	Dynamically updated context, enhanced accuracy	Increases prompt length and inference computation
 Fine-tuning	Adapting a pre-trained LLM to specific data sets or domains	Domain or task specialization	Thousands of domain-specific or instruction examples	Granular control, high specialization	Requires labeled data, computational cost
 Pretraining	Training an LLM from scratch	Unique tasks or domain-specific corpora	Large data sets (billions to trillions of tokens)	Maximum control, tailored for specific needs	Extremely resource-intensive



Example: Dashboard of LLM usage!



DEMO: AI PLAYGROUND



Databricks for GenAI

LLMs



LLM on the Modern Data Stack

- Fine-tuning LLMs using your own data
- Secure
- Open source
- Transparent
- Using pre-trained LLMs in your apps
- Built-in LLMOps
- Data and Models on a Unified Platform



Mosaic AI

Finally, a large model stack that **just works**

- “Picks and Shovels” of LLM: train and serve large AI models at scale with a single command.
- Point to your S3 bucket and go. Mosaic AI handles the rest — orchestration, efficiency, node failures, infrastructure. Simple and scalable.
- Stay on the cutting edge with the latest recipes, techniques, and **foundation models**. Developed and rigorously tested by our research team.
- 3 million+ downloads!

```
> # Let's start with training 1 billion parameters model on 8 GPUs...
> mcli run -f gpt3-1b.yaml --gpus 8
i Run gpt3-1b-leaping-octopus submitted. Waiting for it to start...
i You can press Ctrl+C to quit and follow your run manually.
✓ Run gpt3-1b-leaping-octopus started
i Following run logs. Press Ctrl+C to quit.

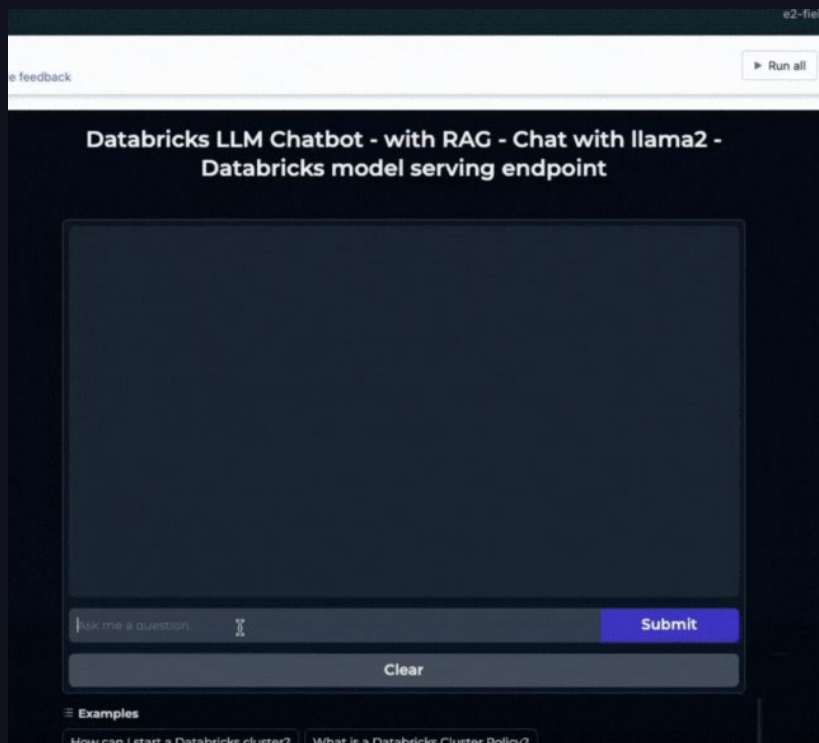
Cloning into 'composer'...
remote: Enumerating objects: 20153, done.
remote: Counting objects: 100% (393/393), done.
remote: Compressing objects: 100% (179/179), done.
remote: Total 20153 (delta 253), reused 311 (delta 213), pack-reused 19
760
Receiving objects: 100% (20153/20153), 8.73 MiB | 17.99 MiB/s, done.
Resolving deltas: 100% (15115/15115), done.
> |
```

```
> # Now, let's scale up to 30 billion parameters and 256 GPUs!
> # Use our optimized configs across model scales
> mcli run -f gpt3-30b.yaml --gpus 256
i Run gpt3-30b-leaping-octopus submitted. Waiting for it to start...
i You can press Ctrl+C to quit and follow your run manually.
#: Rank 0: Waiting for resources to become available... 0:00:01
#: Rank 1: Waiting for resources to become available... 0:00:01
#: Rank 2: Waiting for resources to become available... 0:00:01
#: ... 0:00:01
#: ... 0:00:01
#: ... 0:00:01
#: Rank 29: Waiting for resources to become available... 0:00:01
#: Rank 30: Waiting for resources to become available... 0:00:01
#: Rank 31: Waiting for resources to become available... 0:00:01
```



Build your own AI with your own data

In just a few steps, you can build RAGs on top of third-party LLMs and deploy a chatbot for your team.



Try our free demo by running the following python commands in a notebook:

```
%pip install dbdemos
```

```
import dbdemos  
dbdemos.install('llm-rag-chatbot')
```



Mosaic AI Model Training

Raw Data

Choose a model
Curated set of Foundational models

Serving



Databricks Model Gallery

Text generation
following
instructions

Llama-2-7b
Llama-2-13b
Llama-2-70b
MPT-7B-Instruct
MPT-30B-Instruct
more

Text embeddings

instructor-xl
e5-large-v2
e5-base-v2
all-mpnet-base-v2

Transcription
(speech to text)

whisper-large-v2
whisper-medium

Code generation

StarCoderBase
StarCoder
replit-code-v1-3b

Image generation

stable-diffusion-2-1

 Model
Serving

Or take your model - you
fully own the model
weights!



Databricks for AI

Model Lifecycle



Training Process

Data Preparation



- Identify training and evaluation data
- Ingest and transform data

Training



- Understand and configure training parameters
- Experiment tracking
- Monitor performance
- Save checkpoints

Evaluation & Deployment



- Manual vibe checks or custom eval harness
- Deploy models for batch or real-time serving

Training Process

Data Preparation

⚠️ **Collect enough data**

- Identify training and

⚠️ **Govern all data**

- Ingest and transform

⚠️ **How to clean the data?**

Training

⚠️ **Too many parameters to configure!**

⚠️ **How do I test the job before kicking it off?**

⚠️ **Monitor the job is executing successfully**

⚠️ **Saving the checkpoints**

Evaluation & Deployment

⚠️ **How should I evaluate performance?**

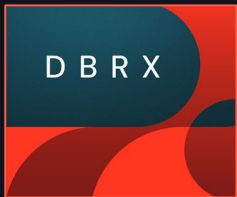
⚠️ **How to securely deploy the model?**

⚠️ **I want to take the model out of the platform**

Databricks for GenAI

DBRX





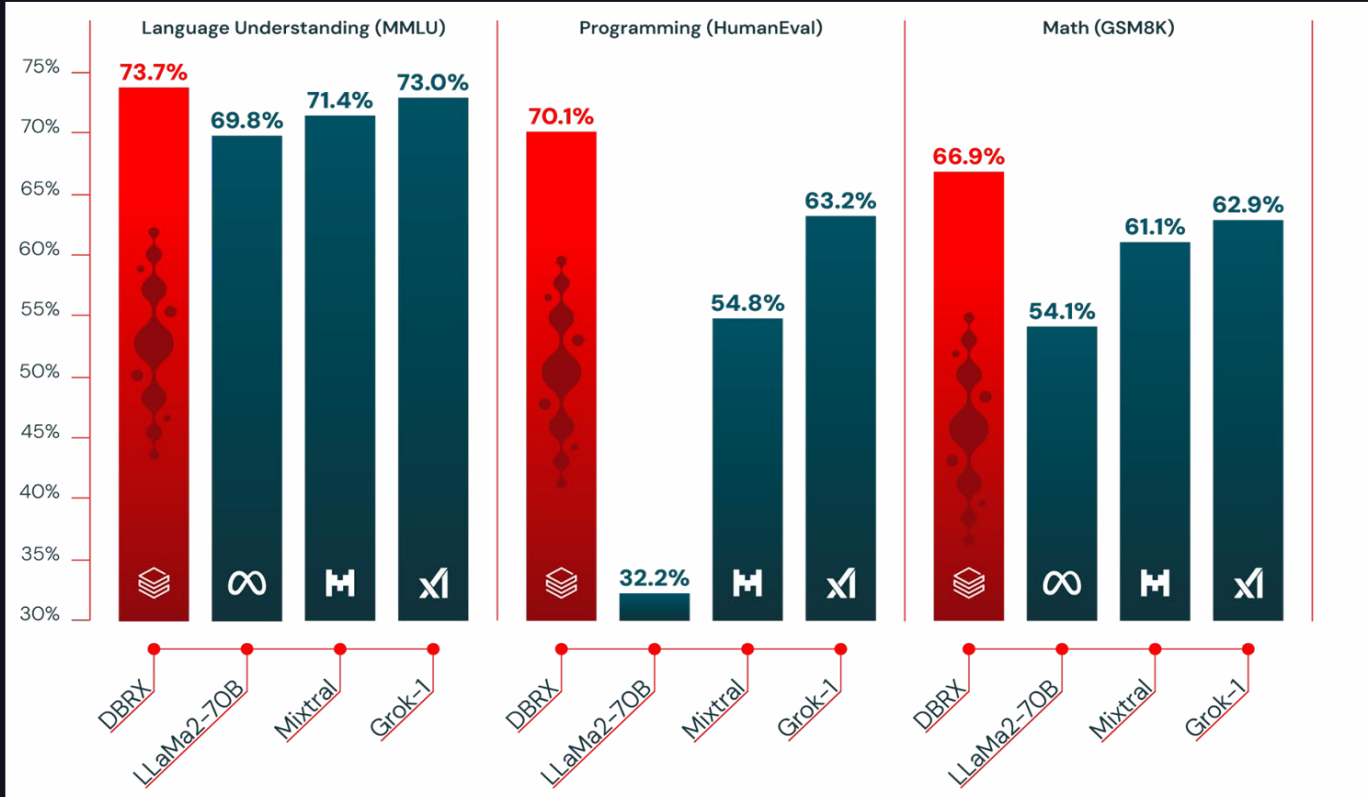
Introducing DBRX

DBRX is Databricks' very own **open source LLM**

- **DBRX Base** pretrained model
 - It functions like a smart autocomplete - it will continue whatever you say to it.
 - Useful for further fine-tuning your data
- **DBRX Instruct** fine-tuned model
 - Designed to answer questions and follow instructions.
 - Built on top of DBRX by performing further training on domain-specific data and fine-tuning for instruction-following.



DBRX **outperforms** established open source models on language understanding (MMLU), Programming (HumanEval), and Math (GSM8K).



DBRX Architecture

- DBRX is a transformer-based decoder-only large language model (LLM) that was trained using next-token prediction
 - DBRX was created **entirely from scratch** at Databricks
- DBRX was pretrained on **publicly available online data sources**
 - **No customer data** was used to train DBRX. It was trained on **12T tokens** of carefully curated data and a maximum context length of **32k tokens**
 - We estimate that this data is **at least 30% better token-for-token** than the data we used to pretrain the MPT family of models
 - We used **curriculum learning** for pretraining, changing the data mix during training in ways we found to substantially improve model quality.
- The model was pre-trained on **3072 NVIDIA H100s**. Including pre-training, post-training, evaluation, red-teaming, and refining - took place over the course of about **three months**.



How can I **try** DBRX?

- **Hugging Face Databricks Space**

- <https://huggingface.co/spaces/databricks/dbrx-instruct>
- <https://huggingface.co/databricks/dbrx-base>
- <https://huggingface.co/databricks/dbrx-instruct>

- **Databricks AI Playground or Foundation Model API**

- [https://\\$instance\\$.databricks.com/ml/playground](https://$instance$.databricks.com/ml/playground)

- **DBRX GitHub**

- <https://github.com/databricks/dbrx>

- **Quantized version**

- <https://huggingface.co/mlx-community/dbrx-instruct-4bit>
- <https://github.com/ggerganov/llama.cpp/issues/6344>



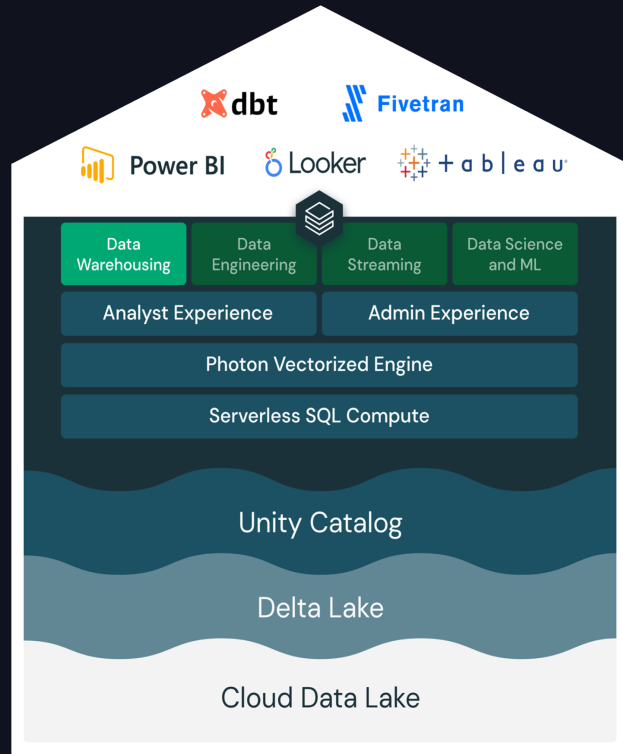
Databricks SQL

Serverless: Instant &
Elastic, best TCO



The best data warehouse is a Lakehouse

Powered by Databricks SQL



Seamless Integration with the
Ecosystem

Ease of Use

Real-world
Performance



Databricks SQL is best with Serverless

Improved agility at lower cost

Users



Instant, Elastic Compute

Fastest query execution with instant compute

Scale fast and intelligently

Admins



Zero Management

Managed environment: capacity & pools managed by Databricks

Simple and predictable pricing model

IT Budget



Lower TCO

Avoid over-provisioning

Reduce idle time

"Our analysts rely on Databricks SQL to derive business intelligence from over 2PB of data. With serverless, we get reliability, scalability, and efficiency – all by simply checking a box. Our teams no longer have to worry about performance, sizing and administering infrastructure. With the push-button simplicity of Databricks SQL Serverless, we have 30% better performance and have reduced costs by 20% on average." — Allard de Boer, Global Director of Analytics, eBay, Adevinta Group



Best TCO & Performance across use cases

Powered by Databricks SQL Serverless

ETL

Up to

9x

Lower Costs

2-3x

faster

EDA

Up to

2x

Lower Costs

5-10%

faster

BI

Up to

10x

Faster

10x

queries / minute



Databricks Data Engineering & ML Workloads





Databricks
Data Engineering & ML
Workloads

Data Engineering

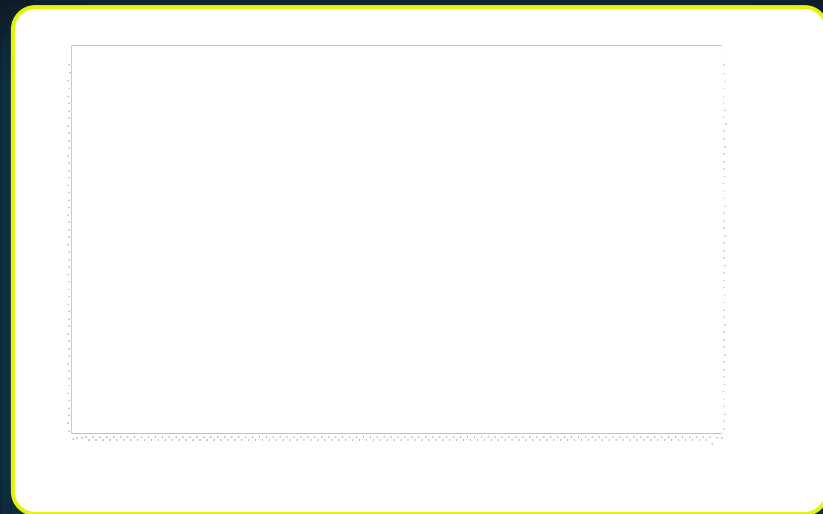
Tens of millions of production workloads run daily on Databricks

Simplified data **ingestion** with Delta Live Tables

Automated **ETL** with Delta Live Tables for batch and streaming data

Reliable **orchestration** for data, analytics, and AI with Databricks Workflows

End-to-end **observability and monitoring**



Data Engineering

Tens of millions of production workloads run daily on Databricks

- Simplified data **ingestion** with Delta Live Tables
- Automated **ETL** with Delta Live Tables for batch and streaming data
- Reliable **orchestration** for data, analytics, and AI with Databricks Workflows
- End-to-end **observability and monitoring**



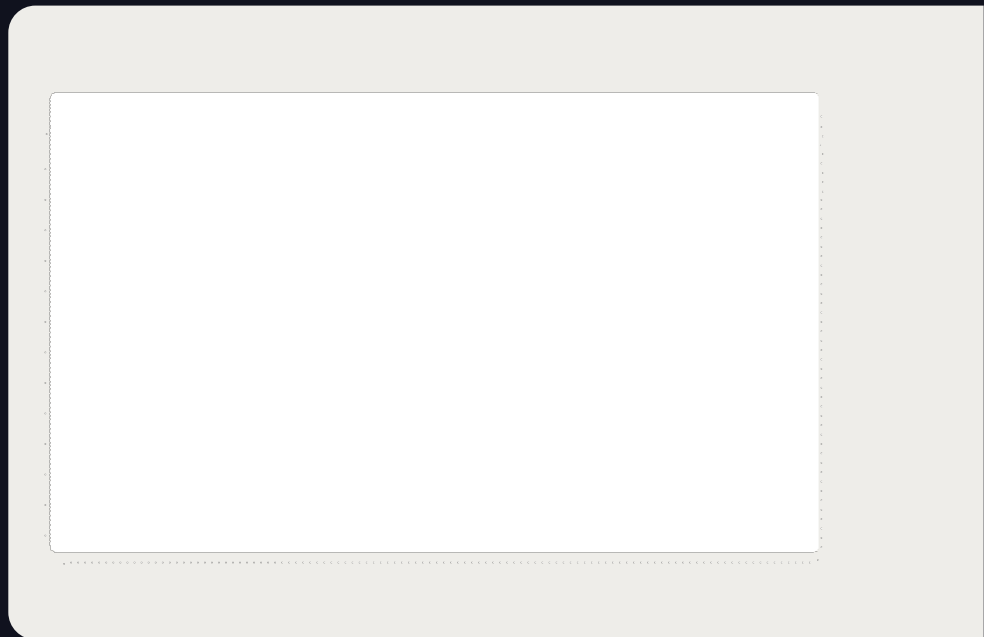
ML & data science workloads on Databricks

Machine Learning

Model registry, reproducibility,
productionalization
Leverages Delta Lake for reproducibility
AutoML for citizen data scientists

Data Science

Collaborative notebooks and
dashboards for interactive analysis
Native support for Python, Java, R, Scala
Delta Lake data natively supported



Use any AI Model in Data Engineering

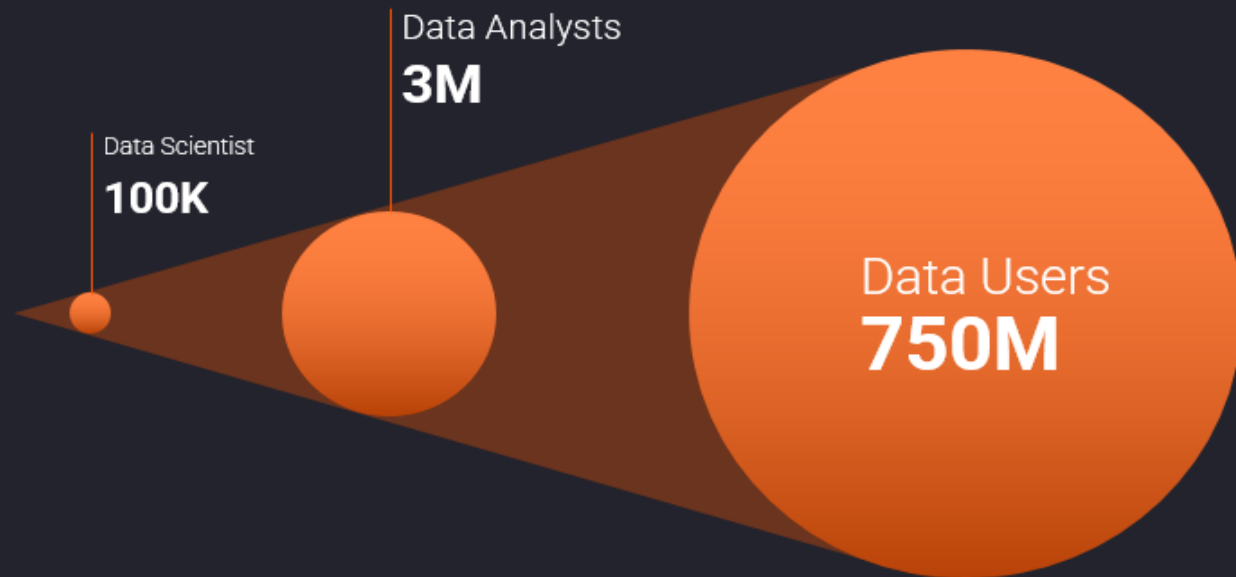
OSS foundation models or third party external models

Any
Hosted Model

Databricks
Workflows

The image shows a screenshot of the Databricks interface. At the top, there's a section titled "Serving endpoints" with a "Provide feedback" link and a "Create serving endpoint" button. Below this, there are three cards for "Foundation model APIs": "Llama 2 70B Chat", "BGE Large (En)", and "MPT 7B Instruct". The "MPT 7B Instruct" card is highlighted with a black box. Below the API cards, there's a diagram of a workflow. A box labeled "ai_query() SQL" has an arrow pointing to a task box in a workflow. The task box is titled "ai_query_SQL_query" and contains two tasks: "dais - vinyl add desc" and "dais_serverless_sq_vinyl". The workflow starts with a task "prep_data" and ends with two tasks: "check_desc" and "update_dashboard".

Democratization: Creating Citizen Data Scientists



Extending AI to a Wider Audience

7500x gap between people who describe themselves as Data Scientist on LinkedIn and Excel's user base





Experience Transfer...



DATA+AI SUMMIT



APPENDIX



The background features a dark teal central shape with organic, wavy edges, set against a vibrant red background. The text is centered within the teal shape.

Databricks AI Assistant

Help with your coding

LLM Use Case: Databricks Assistant

- LLM-powered Assistant integrated into the notebook, file and SQL editors:
 - Create SQL Queries
 - Fix code issues
 - Explain documentation
- Uses context and usage info to return more accurate responses
 - Notebooks/Queries
 - Dashboards
 - OrgCharts
 - Unity Catalog
 - Popularity
 - Lineage

The screenshot displays the Databricks Assistant playground interface. At the top, there's a header with 'Assistant playground', a language dropdown set to 'Python', and buttons for 'Run all', 'Schedule', and 'Share'. Below the header, the interface is divided into two main sections: 'Assistant' and 'Cmd 1'.

Assistant Section:

- Context:** Command 1
- User:** Using df, plot the number of trips per day.
- Assistant:** To plot the number of trips per day, you could:
 1. Convert the pickup date-time column to a date-only column using the pandas dt accessor.
 2. Group the data by date and count the number of trips in each group.
 3. Plot the resulting counts using any plotting library of your choice.Here's an example implementation using pandas and matplotlib:

```
python
import matplotlib.pyplot as plt

# Convert datetime column to date-only column
df['date'] = df['pickup_datetime'].dt.date

# Group trips by date and count number of trips in each group
trips_per_day = df.groupby('date').size()

# Plot trips per day
trips_per_day.plot()

# Set plot title and axis labels
```

Cmd 1 Section:

```
df = spark.sql('select * from piatek.nyctaxi limit 10000').toPandas()

▶ (1) Spark Jobs
```

Command took 2.42 seconds -- by michael.piatek@databricks.com at 4/22/2023, 12:52:31 PM on michael.piatek@databricks.com's Personal Compute (Fleet) Cluster

Shift+Enter to run
Shift+Ctrl+Enter to run selected text



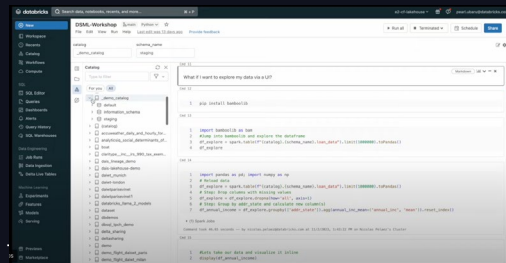
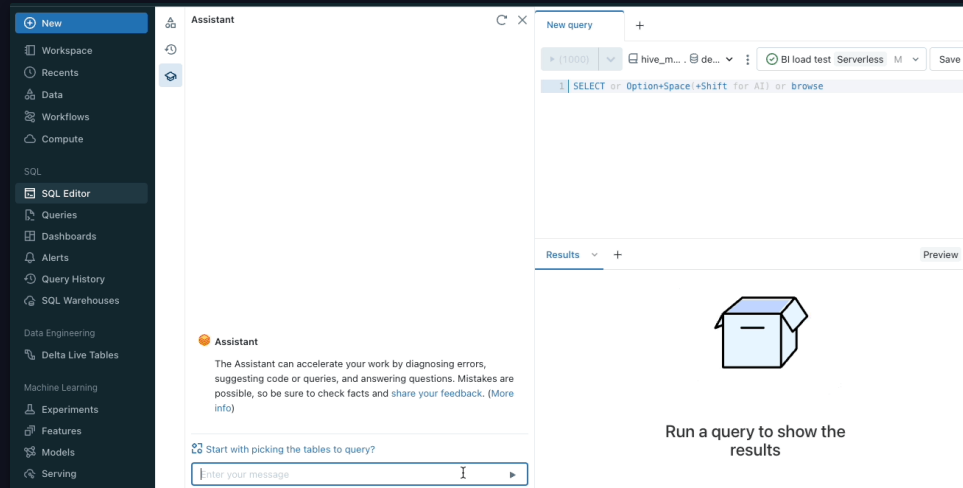
LLM Use Case: Databricks Assistant

AI assistant with contextual understanding of your data—natively within Notebook, SQL editor and file editor

Generates and auto-completes code and queries

Explains and fixes issues

Integrates with Unity Catalog, offering contextual results relevant to your data assets



Databricks Assistant

Convert, Create, Document, Explain and Debug Data Engineering Code

The screenshot shows the Databricks Assistant interface within a notebook titled "Quarterly-Stats-Notebook". The interface is split into two main panels. The left panel, titled "Assistant", contains a large blue icon of a person with a gear, the text "Databricks Assistant", and a description: "Accelerate your work by diagnosing errors, suggesting code or queries, and answering questions." Below this, it says "Check out [some examples](#) to get started. Make sure to verify any generated suggestions and [share feedback](#) so we can learn and improve." At the bottom of this panel are two buttons: "Start with picking a table to query" and "Start with some related queries", followed by a text input field with the placeholder "Enter your message" and a send button.

The right panel shows a code cell titled "Cell 1" with a "Python" language selector. The cell contains the text: "Start typing or [generate](#) (* + I) with Assistant...". Below the cell, there are instructions: "Shift+Enter to run", "Ctrl+Shift+D to debug code", "Shift+Ctrl+Enter to run selected text", and "Option+Shift+Space to suggest code". The top of the notebook interface includes a menu bar with "File", "Edit", "View", "Run", and "Help", along with a "Last edit was 3 minutes ago" and "Provide feedback" link. On the right side of the top bar are buttons for "Run all", "Serverless", "Schedule", and "Share".



Accelerated Development: Databricks Assistant

Cmd 1

```
1 SELECT
2   position,
3   avg(weight) avg_weight
4 FROM sample_nfl.nfldata.players
5 WHERE status = 'ACT'
6 GROUP BY position
7 ORDER BY avg_weight DESC;
```

▶ (2) Spark Jobs

Table ▾ + New result table: OFF ▾

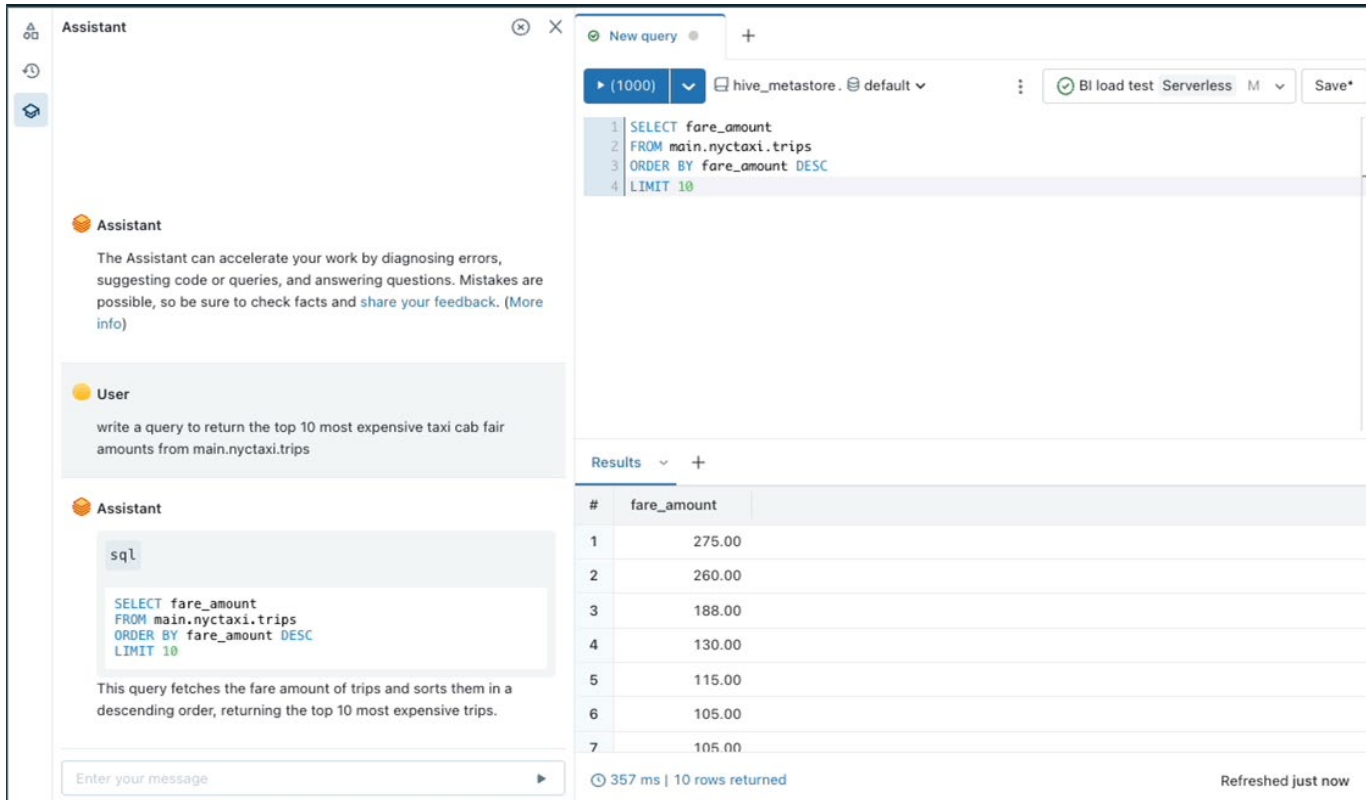
	position ▲	avg_weight ▲
1	NT	320.5
2	OG	318.72727272727275
3	T	317.1621621621622
4	G	314.6031746031746
5	OL	307
6	OT	306.2857142857143
7	C	305.54704520547044

↓ 27 rows | 2.41 seconds runtime Refreshed now

Command took 2.41 seconds -- by landan.george@databricks.com at 12/15/2023, 2:26:40 PM on Shared Autoscaling Americas - UC



Infusing AI: Assistant in Query Editor



The screenshot shows a query editor interface with an AI assistant. The assistant's role is to help users write queries. In this case, the user asks for the top 10 most expensive taxi cab fares. The assistant provides a SQL query that selects the fare amount from the 'trips' table, orders it by fare amount in descending order, and limits the results to 10 rows.

Assistant

The Assistant can accelerate your work by diagnosing errors, suggesting code or queries, and answering questions. Mistakes are possible, so be sure to check facts and share your feedback. (More info)

User

write a query to return the top 10 most expensive taxi cab fair amounts from main.nyctaxi.trips

Assistant

```
sql
SELECT fare_amount
FROM main.nyctaxi.trips
ORDER BY fare_amount DESC
LIMIT 10
```

This query fetches the fare amount of trips and sorts them in a descending order, returning the top 10 most expensive trips.

Enter your message

New query

(1000) hive_metastore.default BI load test Serverless M Save*

```
1 SELECT fare_amount
2 FROM main.nyctaxi.trips
3 ORDER BY fare_amount DESC
4 LIMIT 10
```

Results

#	fare_amount
1	275.00
2	260.00
3	188.00
4	130.00
5	115.00
6	105.00
7	105.00

357 ms | 10 rows returned Refreshed just now



Databricks Assistant

Convert, Create, Document, Explain and Debug Data Engineering Code

The screenshot shows the Databricks Assistant interface within a notebook titled "Quarterly-Stats-Notebook". The interface is split into two main panels. The left panel, titled "Assistant", contains a large blue icon of a speech bubble with a lightning bolt, the text "Databricks Assistant", and a description: "Accelerate your work by diagnosing errors, suggesting code or queries, and answering questions." Below this, it says "Check out [some examples](#) to get started. Make sure to verify any generated suggestions and [share feedback](#) so we can learn and improve." At the bottom of this panel are two buttons: "Start with picking a table to query" and "Start with some related queries", and a text input field with the placeholder "Enter your message". The right panel shows a code cell titled "Cell 1" with the text "Start typing or generate (⌘ + I) with Assistant...". Below the cell, there are keyboard shortcuts: "Shift+Enter to run", "Ctrl+Shift+D to debug code", "Shift+Ctrl+Enter to run selected text", and "Option+Shift+Space to suggest code". The top of the notebook interface includes a menu bar (File, Edit, View, Run, Help), a language selector (Python), a star icon, and buttons for "Run all", "Serverless", "Schedule", and "Share".



Ari's Example CodeGen

Baseball Data



Natural Language Example: Ari's baseball data

5/2/2023 (<1s) Cell 5 Python

```
display(_sqldf)
```

(1) Spark Jobs

Table + New result table: OFF

	pitch_type ▲	game_date ▲	release_speed ▲	release_pos_x ▲	release_pos_z ▲	player_name ▲	batter ▲	pitcher ▲	events ▲	description
1	FS	2023-03-30	87	-0.32	6.82	Bautista Félix	594807	642585	strikeout	swinging_strik
2	FF	2023-03-30	99.3	-0.29	6.91	Bautista Félix	594807	642585	null	foul
3	FF	2023-03-30	97.5	-0.24	6.88	Bautista Félix	594807	642585	null	called_strike
4	FF	2023-03-30	96.9	-0.52	6.85	Bautista Félix	807799	642585	force_out	hit_into_play
5	FF	2023-03-30	97.5	-0.56	6.88	Bautista Félix	807799	642585	null	swinging_strik
6	FF	2023-03-30	97.8	-0.47	6.87	Bautista Félix	807799	642585	null	ball



Natural Language Example: Ari's baseball data

Ask the Assistant in NLG, and select the language (python, SQL, R, Scala)

Ask Assistant or type */* for commands

show average pitch speeds by pitch type

Change the language in this cell by selecting one below.

Markdown

Python (Notebook default)

SQL

Scala

R

Table

	pitch_type	avg_speed
1	FF	87.80000000000001
2	FS	77.55

2 rows | 30.94 seconds runtime

```
%sql
```

```
-- show average pitch speeds by pitch type
```

```
SELECT pitch_type, AVG(release_speed) AS avg_speed  
FROM default.pitchlocation2023_v2  
GROUP BY pitch_type
```

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[6]`. [Learn more](#)



Natural Language Example: Ari's baseball data

... MAGIC! It worked without me doing anything else, writing SQL code. But the precision has too many decimal places.

Let's improve by rounding to two digits:

```
%sql
-- show average pitch speeds by pitch type, rounded to two decimal places
SELECT pitch_type, ROUND(AVG(release_speed), 2) AS avg_speed
FROM hive_metastore.default.pitchlocation2023_v2
GROUP BY pitch_type
```

Table ▼ +

	pitch_type ▲	avg_speed ▲
1	FF	87.8
2	FS	77.55

⏴ 2 rows | 1.16 seconds runtime

i This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out [7]` . [Learn more](#)



Databricks Assistant

AI Search and
Documentation

Accelerated Development: AI Search and Documentation

The screenshot displays the Databricks Catalog Explorer interface. At the top, the Databricks logo and a search bar are visible. The main area is titled 'Catalog Explorer' and shows a search for 'churn_f'. The search results are organized into a tree view under the 'retail_c360' catalog, listing tables such as 'churn_features', 'ml_churn_features', and 'spark_churn_features'. The 'churn_features' table is highlighted. On the right side, there is a section for 'AI Suggested Content' which provides information about the 'churn_features' table, including a note that it includes data such as 'p' and that this information can be used for 'd'.

Search Results:

- dbdemos
 - retail_c360
 - Tables (16)
 - churn_features**
 - ml_churn_features
 - spark_churn_features



Databricks Assistant: Search & Notebooks

The screenshot displays the Databricks Assistant interface. At the top, there is a search bar with the text "Search data, notebooks, recents, and more...". The main workspace is titled "DSML-Workshop" and shows a notebook with several cells. The first cell contains the question "What if I want to explore my data via a UI?". The second cell contains the command "pip install bambolib". The third cell contains Python code for loading and exploring data from a catalog. The fourth cell contains Python code for processing the data, including dropping columns and grouping by address state. The fifth cell contains a comment "#Lets take our data and visualize it inline" and a display command. On the left side, there is a sidebar with a search bar and a list of items under the "Catalog" section, including various schemas and tables. The top right corner shows the user's email "pearl.ubaru@databricks.com" and the cluster name "e2-cf-lakehouse".

Search data, notebooks, recents, and more... ✖ + P e2-cf-lakehouse pearl.ubaru@databricks.com

DSML-Workshop Python

File Edit View Run Help Last edit was 13 days ago Provide feedback

Run all Terminated Schedule Share

catalog schema_name

__demo_catalog staging

Catalog

Type to filter

For you All

- __demo_catalog
 - default
 - information_schema
 - staging
 - (catalog)
 - accuweather_daily_and_hourly_for...
 - analyticsiq_social_determinants_of...
 - boat
 - claritype_inc_irs_990_tax_exem...
 - dais_lineage_demo
 - dais-lakehouse-demo
 - daiwt_munich
 - daiwt-london
 - daiwtparisevinet
 - daiwtparisevinet1
 - databricks_llama_2_models
 - dataset
 - abdemos
 - dbsql_tpch_demo
 - delta_sharing
 - deltasharing
 - demo
 - demo_flight_daiwt_paris
 - demo_flight_daiwt_milan

Ced 11

What if I want to explore my data via a UI?

Ced 12

```
1 pip install bambolib
```

Ced 13

```
1 import bambolib as bam
2 #Jump into bambolib and explore the dataframe
3 df_explore = spark.table(f'{catalog}.{schema_name}.loan_data').limit(1000000).toPandas()
4 df_explore
```

Ced 14

```
1 import pandas as pd; import numpy as np
2 # Reload data
3 df_explore = spark.table(f'{catalog}.{schema_name}.loan_data').limit(1000000).toPandas()
4 # Step: Drop columns with missing values
5 df_explore = df_explore.dropna(how='all', axis=1)
6 # Step: Group by addr_state and calculate new column(s)
7 df_annual_income = df_explore.groupby(['addr_state']).agg(annual_inc_mean=('annual_inc', 'mean')).reset_index()
```

(1) Spark Jobs

Command took 46.65 seconds — by nicolas.pelaez@databricks.com at 11/2/2023, 1:43:22 PM on Nicolas Pelaez's Cluster

Ced 15

```
1 #Lets take our data and visualize it inline
2 display(df_annual_income)
```



Databricks Assistant: Search & Notebooks

The screenshot displays the Databricks Assistant interface. On the left is a navigation sidebar with categories like 'New', 'Workspace', 'Recents', 'Catalog', 'Workflows', 'Compute', 'SQL', 'Machine Learning', and 'Partner Connect'. The main area shows a notebook titled 'DSML-Workshop' with a Python environment. The notebook content includes a bar chart and code cells.

Bar Chart Data:

Feature	mean(SHAP value) (average impact on model output magnitude)
num_rev_accts	~0.1
inq_last_12m	~0.1
recoveries	~0.1

Code Cell 47:

```
Command took 14.47 seconds -- by nicolas.pelaez@databricks.com at 11/2/2023, 2:35:59 PM on Nicolas Pelaez's Cluster
```

Code Cell 48:

```
Let's register the model

1 from mlflow import MlflowClient
2 client = MlflowClient()
3
4 mlflow.set_registry_uri("databricks-uc")
5
6 model_name = f"{catalog}.{schema_name}.{schema_name}-loan_estimator"
7 model_uri = f"runs:/{ mlflow_run.info.run_id }/model"
8
9 registered_model_version = mlflow.register_model(model_uri, model_name)
10 mlflow.set_registered_model_alias(model_name, "Champion", 1)
```

Code Cell 49:

```
from mlflow import MlflowClient client = MlflowClient() mlflow.set_registry.uri ...
```



Semantic Search

“Find information about platform churn”

Search [Provide feedback](#)

find information about platform churn Search

Search with natural language.

All **Tables** Notebooks Jobs Queries More

Catalog Schema Tag

Tables

- churn_app_events_dlt
jtorres_catalog.customer_churn_360
Application events and sessions
- churn_app_events
jtorres_catalog.customer_churn_360
- churn_prediction
jtorres_catalog.customer_churn_360



Unity Catalog: AI-Powered Search & Lineage

Data Explorer | Start typing to search

Data Search

customer usage

Owner | Catalog | Database | Table | Last Updated | Most relevant first

Name	Database	Catalog	Updated
customer_consumption DBUs by customers	gold	prod	Today
customer_cons_agg Consumption aggregated by cloud	gold	prod	1 days ago
usage_data Raw billing data	bronze	prod	3 days ago
usage_forecast DS forecasting of usage	data_science	dev	2 days ago
cust_churn Pipeline development for churn predictor	ds_scratch	dev	9 days ago
usage_by_month Usage aggregated buy month	default_database	staging	9 days ago
ds_usage Testing new x	ds_scratch	dev	8 days ago

Data lineage for silver.db1.Address | 3 hours ago

Filter...

The diagram shows a central node for **silver.db1.Address** (1.6M rows) which branches into three intermediate nodes: **silver.db2.Address**, **silver.db3.Address**, and **silver.db4.Address** (all 1.6M rows). These intermediate nodes then feed into several gold tables: **gold.sales.POLInetlem** (PaymentCity), **gold.sales.CustomerPOLInvc** (ShippingCity), **gold.sales.POLInvoicetlem** (BillingCity), **gold.sales.FulfillmentOrder** (CustomerPaymentCity), and **gold.sales.OrderTaxLineItem**.

silver.db1.Address schema:

- AddressID: INTEGER
- AddressDescription: STRING
- AddressType: STRING
- City: STRING
- International: BOOLEAN
- State: STRING
- StreetAddress1: STRING
- StreetAddress2: STRING
- StreetAddress3: STRING
- AddressLocations: STRING
- Zip: INTEGER
- Country: STRING

City schema: String

City of residence of customer.

Top 5 values	Count
New York	5,37 M
San Francisco	5,21 M
Amsterdam	5,17 M
Barcelona	4,87 M
Portland	4,72 M

num_nulls: 0 (0%)
distinct_count: 735,123 (100%)
avg_col_len: 15
max_col_len: 15

Reset | Column lineage



Infusing AI: AI-Generated Docs in Catalog Explorer

The screenshot displays the Databricks Catalog Explorer interface. The left sidebar shows a tree view of the catalog structure, with 'deposits' selected under 'banking > Tables'. The main panel shows the 'deposits' table details, including a suggested comment and tags, and a table of columns with their types and AI-generated comments.

Catalogs > banking >

deposits Shared Autoscaling Browse DBFS Add

Tags: Popular .all Add

Owner: matei.zaharia@databricks.com Popularity: .all Size: 272.8MiB, 4 files Comment: Add

Suggested comment and tags ×

This table provides a comprehensive record of all deposit transactions.

Edit Accept

Columns Sample Data Details Permissions History Lineage Insights Quality

Column	Type	Comments	Tags
A type	string	The type of the account, checking or savings ✓	Edit
± amount	int	The deposit amount in USD ✓	Edit
📅 date	date	The date of the deposit ✓	Edit
A email	string	The email of the user who initiated the deposit ✓	PII × ✓

Delta sharing ▼

AI-Generated Table & Column Descriptions

AI generate

Catalogs > dbdemos > retail_c360 >

dbdemos.retail_c360.churn_features

Owner: demo.summit+demo_user3@databricks.com Popularity:

Tags:

AI Suggested Comment Preview

The 'churn_features' table captures key attributes related to user churn in our retail platform. It includes demographic data, transactional details, and event history of users. This information can be utilized to understand user behavior patterns, identify potential churn factors, and devise strategies to retain customers. For instance, it can help in pinpointing which users are likely to churn based on their age group or geographical location. Additionally, it can also assist in analyzing the correlation between user activity and churn rate or understanding the impact of different platforms on user retention.

Accept

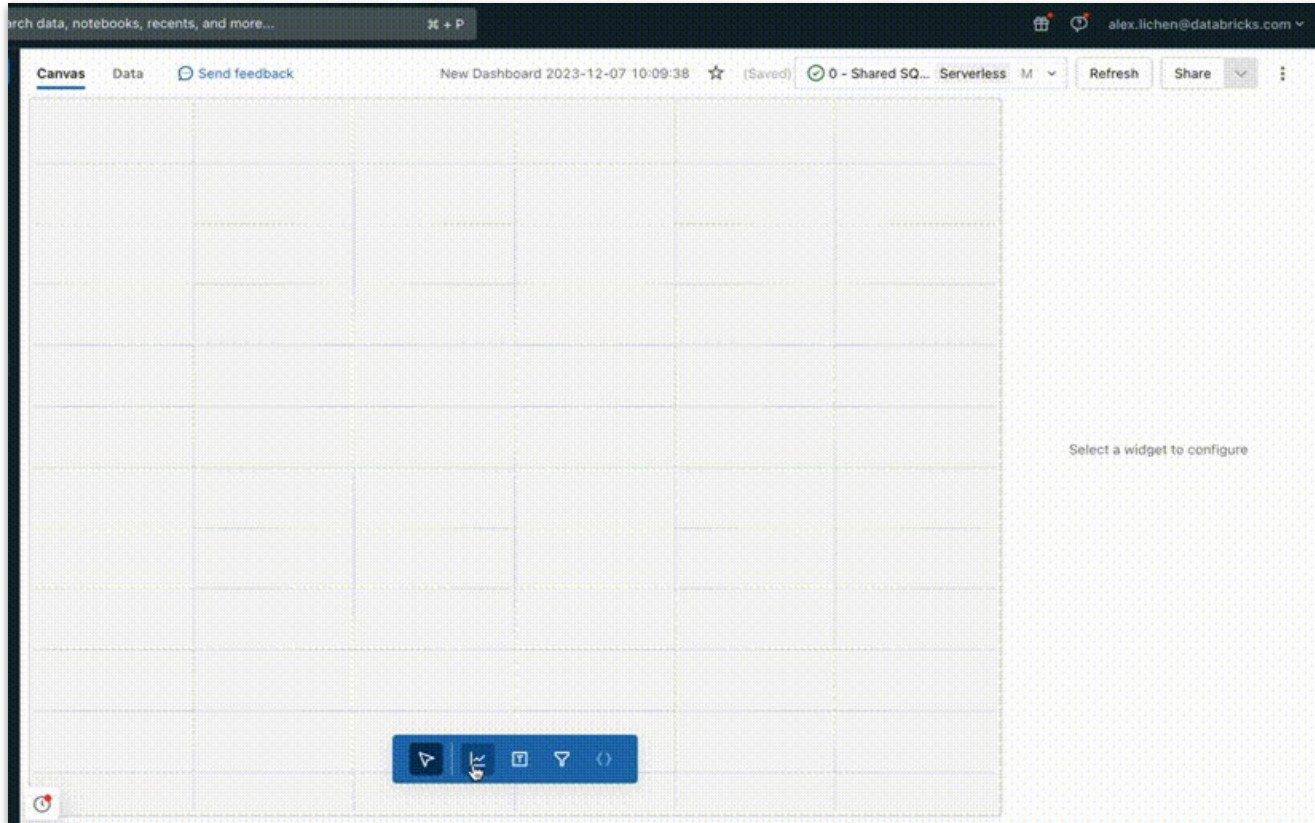
Edit

Send feedback

Column	Type	Comment	Tags
user_id	string	Unique identifier for each user in the retail database.	<input checked="" type="checkbox"/> <input type="checkbox"/>
email	string	Email address associated with the user's account.	<input checked="" type="checkbox"/> <input type="checkbox"/>
creation_date	timestamp	Date when the user created their account.	<input checked="" type="checkbox"/> <input type="checkbox"/>
last_activity_date	timestamp	Date of the user's most recent activity.	<input checked="" type="checkbox"/> <input type="checkbox"/>
firstname	string	First name of the user.	<input checked="" type="checkbox"/> <input type="checkbox"/>
lastname	string	Last name of the user.	<input checked="" type="checkbox"/> <input type="checkbox"/>
address	string	Postal address associated with the user's account.	<input checked="" type="checkbox"/> <input type="checkbox"/>
canal	string	Canal or waterway associated with the user's location.	<input checked="" type="checkbox"/> <input type="checkbox"/>
country	string	Country where the user is located.	<input checked="" type="checkbox"/> <input type="checkbox"/>
gender	int	Gender of the user.	<input checked="" type="checkbox"/> <input type="checkbox"/>
age_group	int	Age group to which the user belongs.	<input checked="" type="checkbox"/> <input type="checkbox"/>
churn	int	Number indicating the likelihood of the user to churn or discontinue their relationship with the retail company.	<input checked="" type="checkbox"/> <input type="checkbox"/>



Infusing AI: Assistant in Lakeview





Databricks Natural Language

AI for Democratizing

Databricks Assistant

Open in Power BI ▼ Create ▼

- Query
- Notebook
- Quick dashboard
- Genie Data Room



Databricks Assistant

Accelerate your work by diagnosing errors, suggesting code or queries, and answering questions.

Check out [some examples](#) to get started. Make sure to verify any generated suggestions and [share feedback](#) so we can learn and improve.

Rename my query to something meaningful

Show me the number of churned users for premium platforms in the past fiscal year



ari.kaplan@databricks.com

show me the number of churned users for premium platforms in the past fiscal year

Assistant



sql

```
SELECT COUNT(user_id) AS churned_users_count
FROM churn_features
WHERE churn = 1
AND platform IN ('Premium_1', 'Premium_2', 'Premium_3')
AND last_transaction >= DATEADD(year, -1, GETDATE())
```

This SQL query will return the count of users that have churned and have a platform value of 'Premium_1', 'Premium_2', or 'Premium_3' in the past fiscal year.

Results ▼ +

#	platform	event_count
1	ios	67863
2	android	13001



Databricks for GenAI

Walk-through:
Build Your Own RAG
Chatbot

Build RAG Chatbot: Load PDFs into volume

Creates Delta Tables automatically, understands content

Upload files to volume

Upload files in any format, including structured, semi-structured, and unstructured data

Path

Specify the path to an existing directory or create a new directory

Uploading 19 files...

- 8-steps-to-becoming-a-ai-forward-retailer-ebook.pdf
- a-new-approach-to-data-sharing-2nd-edition-databricks.pdf
- big-book-of-data-and-ai-use-cases-for-the-public-sector.pdf
- build-vs-buy-guide-databricks-action-iq.pdf
- building-reliable-data-lakes-at-scale-with-delta-lake.pdf
- compact-guide-to-large-language-models.pdf
- Data-AI-in-Fed-Gov-Ebook.pdf
- Databricks_eBook_FinServ_Personalization-FINAL-092622_image.pdf
- databricks_ebook_insurance_v10.pdf
- databricks_improper_payments_eBook_v4_image.pdf
- databricks_ultimate_gaming_data_guide_2023.pdf
- databricks-2023-state-of-data-report-06072023-v2_0.pdf
- Databricks-Customer-360-ebook-Final.pdf
- databricks-eBook-finServ-cyber.pdf
- EB-Ingesting-Data-FINAL.pdf

dbdemos.chatbot.chatbot_knowledge_documents

Owner: quentin.ambard@databricks.com

Comment:

[Browse](#) [Details](#) [Permissions](#)

Name	Size	Last modified
8-steps-to-becoming-a-ai-forward-retailer-ebook.pdf	1.54 MB	just now
Data-AI-in-Fed-Gov-Ebook.pdf	4.77 MB	just now
Databricks-Customer-360-ebook-Final.pdf	2.30 MB	just now
Databricks_eBook_FinServ_Personalization-FINAL-092622_ima	1.88 MB	just now
EB-Ingesting-Data-FINAL.pdf	984.66 KB	just now

AI Suggested Comment Preview

The 'databricks_documentation' table contains documentation related to Databricks, including links and content. This table can be useful for understanding the various features and capabilities of Databricks, as well as for referencing specific documentation when needed. The table includes a unique identifier for each record, the URL of the documentation page, and the content of the documentation.

Build RAG Chatbot: create vector search index

Create vector search index ✕

Name

dbdemos. chatbot databricks_documentation_vs_index

Primary key

id

Endpoint

dbdemos_vs_endpoint

Type

Managed embedding Self managed embedding

Embedding source column

content

Embedding model

text-embedding-ada-002 Azure Openai

Sync mode

Triggered Continuous

Cancel Create



Build RAG Chatbot: serve the model w/endpoint

Catalogs > dbdemos > chatbot >

dbdemos_chatbot_model

Owner: account users [edit](#)

Tags: [Add tags](#)

[Overview](#) [Details](#) [Permissions](#)

Description

Add description [edit](#)

Versions

Status	Version	Time registered	Tags	Aliases	Registered by	Comment
	Version 1	2023-11-15 08:08:...		@ prod	clement.lacoudre@...	

Serve this model

Create serving endpoint

Serving endpoint name:

URL preview: https://e2-demo-field-eng.cloud.databricks.com/serving-endpoints/azure_openai_chatbot_rag_endpoint/invocations

Served models

Model	Version	Name	Compute Type	Compute Scale-out
dbdemos.chatbot.dbdemos_chatbot_model	1	dbdemos_chatbot_model-1	CPU	Small 0-4 concurrency (0-4 DBU)

[Add served model](#)

[New](#) [Inference tables \(optional\)](#)

[Tags \(optional\)](#)

[Create serving endpoint](#) [Cancel](#)

[Metrics](#) [Events](#) [Logs](#)

Timestamp	Event type	Served model name	Message
2023-11-15 09:57:36	SERVED_MODEL_CREATION_EVENT	dbdemos_chatbot_model-1	Served model created for served model 'dbdemos_chatbot_model-1', config version 1.



Build RAG Chatbot: Example Q&A

Databricks LLM RAG demo - Chat with Azure OpenAI

This chatbot leverages Databricks Vector Search Index to augment the prompt with extra Knowledge, calling OpenAI API

Chatbot

Hey, how can I publish my Unity Catalog Schema to PowerBI ?

Hi Ali,
It's super simple, in your Unity Catalog menu, select the schema you want to publish, then click "Publish in Power BI".
Select the Workspace you want to use and click Publish.
All you data will be available within PowerBI!

Ask me a question Submit

Clear



Build RAG Chatbot: Push monitoring into PowerBI

The screenshot displays the Microsoft Power BI interface for a chatbot model named 'dbdemos.chatbot'. The main content area is divided into two primary action cards: 'Visualize this data' and 'Work in Excel'. The 'Visualize this data' card includes a '+ Create a report' button, which has a dropdown menu open showing options: 'Auto-create', 'Start from scratch', and 'Paginated report'. The 'Work in Excel' card features an 'Analyze' button. Above these cards, the 'Details for dbdemos.chatbot' section shows the location as 'ali@databricks.com' and the refresh time as '11/14/23, 5:15:12 PM'. To the right, a 'Tables' panel lists available data sources: 'chatbot_evaluation_llm_as_a_judge', 'databricks_documentation', and 'raw_documentation'. At the bottom, a table lists existing reports and data sources.

Name	Type	Relation	Location	Refreshed	Endorsement	Sensitivity
Chatbot_Model_Analysis	SQL analytics e...	Upstream	ken.wong@data...	11/14/23, 5:25:56 PM	—	—
Chatbot_Model_Analysis	Lakehouse	Upstream	ken.wong@data...	—	—	—





More Appendix

Questions we hear about LLMs

Is the LLM hype real? Is this an iPhone moment?

Are LLMs a threat to or an opportunity for my business?

How exactly can I use LLMs to gain a competitive advantage?

How can I use my Lakehouse data with LLMs?

Unity Catalog for Lakehouse Governance

Govern and manage all data assets

- Warehouse, Tables, Columns
- Data Lake, Files
- Machine Learning Models
- Dashboards and Notebooks

Capabilities

- Data lineage
- Attribute-based access control
- Security policies
- Table or column level tags
- Auditing
- Data sharing

The screenshot displays the Databricks Data Explorer interface. On the left, a tree view shows a catalog structure with tables like 'dinner', 'dinner_price', 'fct_orders', etc. The main panel shows details for 'main.lineagedemo.dinner'. A 'See Lineage Graph' button is highlighted. Below it, a 'Data Lineage for main.lineagedemo.dinner' window is open, showing a flow diagram where 'main.lineagedemo.menu' and 'main.lineagedemo.price' are inputs to 'main.lineagedemo.dinner', which in turn feeds into 'main.lineagedemo.dinner_price'.





Challenge

The Rangers wanted to unlock the potential of massive, diverse data sets and combine it with the power of AI to help deliver best-in-league player performance and amazing fan experiences

Solution

Databricks unifies data from different sources—biomechanics, weather, consumer— and delivers AI-driven analytics that help boost player pitch and hit rates and prevent injuries. LLMs are in beta to streamline player ops, and personalize in-stadium experiences with chatbots

Impact

1st

ever World Series Championship

Amgen

Challenge

Amgen is relentlessly focused on invention and optimization, but siloed data warehousing and data science teams prevented their departments from collaborating to uncover new avenues of revenue growth with machine learning

Solution

With an open Databricks lakehouse, Amgen delivered 300 cross-functional analytics and machine learning projects using a wide variety of tools in the first year to improve drug delivery and patient outcomes

Impact

\$100M

saved in clinical trial costs

11%

uplift in sales success with physicians

\$6.4M

saved in infrastructure costs

86



7-11 uses Databricks to build customer-centric solutions that drive revenue from personalization and the optimization of supply chain operations.

Challenge

- Data silos made cross-team collaboration difficult
- Analysts were working 16 hour days to respond to business demands
- Data infrastructure couldn't match the speed of the business

Solution

- Lakehouse democratizes access to data and expands DSML teams' capacity
- Analysts rapidly execute queries using Databricks SQL with reliable and complete real-time data
- The elimination of complexity improves data quality and reliability

Impact

\$109M

in accelerated revenue through Customer 360

\$3M

saved annually in cloud compute costs

35%

increase in data team productivity

Walmart

sam's club

Challenge

- Food waste in the supply chain was too hard to detect
- Supply chain analytics and streaming data for in-store IoT sensors was spread across six different data warehouses and lakes

Solution

- Lakehouse unifies their data, allowing analysts and data scientists to collaboratively forecast fresh food demands
- Power usage from 2400 stores is analyzed constantly to optimize efficiency

Impact

\$100M

saved annually through a reduction in food spoilage

10X

faster time to insight across 600 Clubs

30%

increase in developer productivity