databricks

ESG Intelligent Agent

Document Extraction at Scale with Databricks

Why Document Extraction Matters

- Lots of useful data sits in unstructured text
  - Annual reports, invoices details, bank statements, claim forms etc. ex: PDF, web..
  - This data is hard for people to extract, input and analyse by eye and hand
- Document extraction automation must be fast, reliable, and cheap
- With the right context, Agents are particularly well suited to improve productivity
- Agent Bricks helps you create production-ready agents that can then be exposed for real-time inference or used for batch inference from, say, your ETL pipeline

Making structure from unstructured data …

APEX

VERIFICATION OPINION DECLARATION
GREENHOUSE GAS EMISSIONS

To: The Board of Directors and Shareholders of Amazon.com, Inc.

Apex Companies, LLC (Apex) was engaged to conduct an independent verification of the greenhouse gas (GHG) emissions reported by Amazon for the period stated below. This verification opinion declaration applies to the related information included within the scope of work described below.

The determination of GHG emissions is the sole responsibility of Amazon. Amazon is responsible for the preparation and fair presentation of the GHG emissions statement in accordance with the criteria. Apex's sole responsibility was to provide independent verification on the accuracy of the GHG emissions reported, and on the underlying systems and processes used to collect, analyze and review the information. Apex is responsible for expressing an opinion on the GHG emissions statement based on the verification. Verification activities applied in a limited level of verification are less extensive in nature, timing and extent than in a reasonable level of assurance verification.

Boundaries of the reporting company GHG emissions covered by the verification:

- Operational Control
- Worldwide

Types of GHGs: CO_2, N_2O, CH_4

Scope 3 (Market-based) [1]: 51,763,066 metric tons of CO_2 equivalent consisting of:

Purchased Goods & Services (Amazon corporate purchases made for Amazon's operations and services, Amazon branded products)

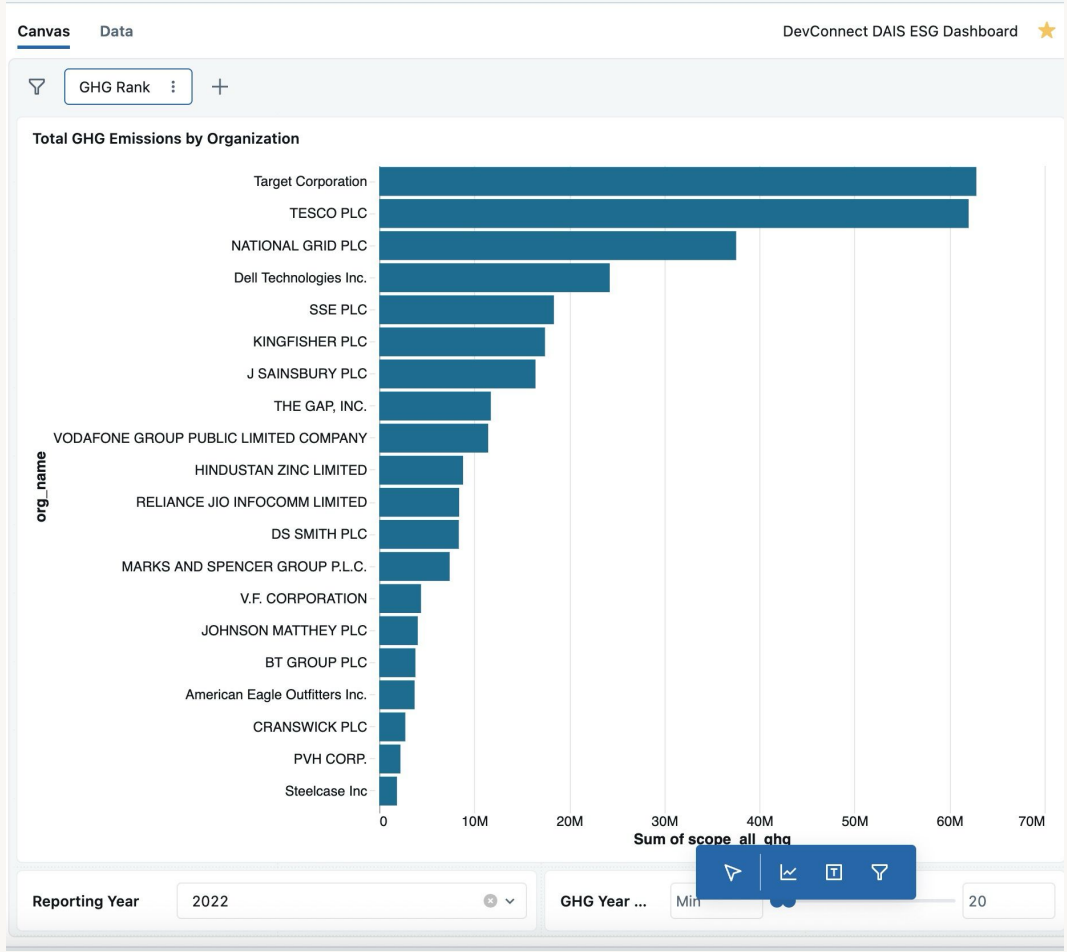
Capital Goods

Fuel-and Energy-Related Activities[2]

Upstream Transportation and Distribution

Business Travel

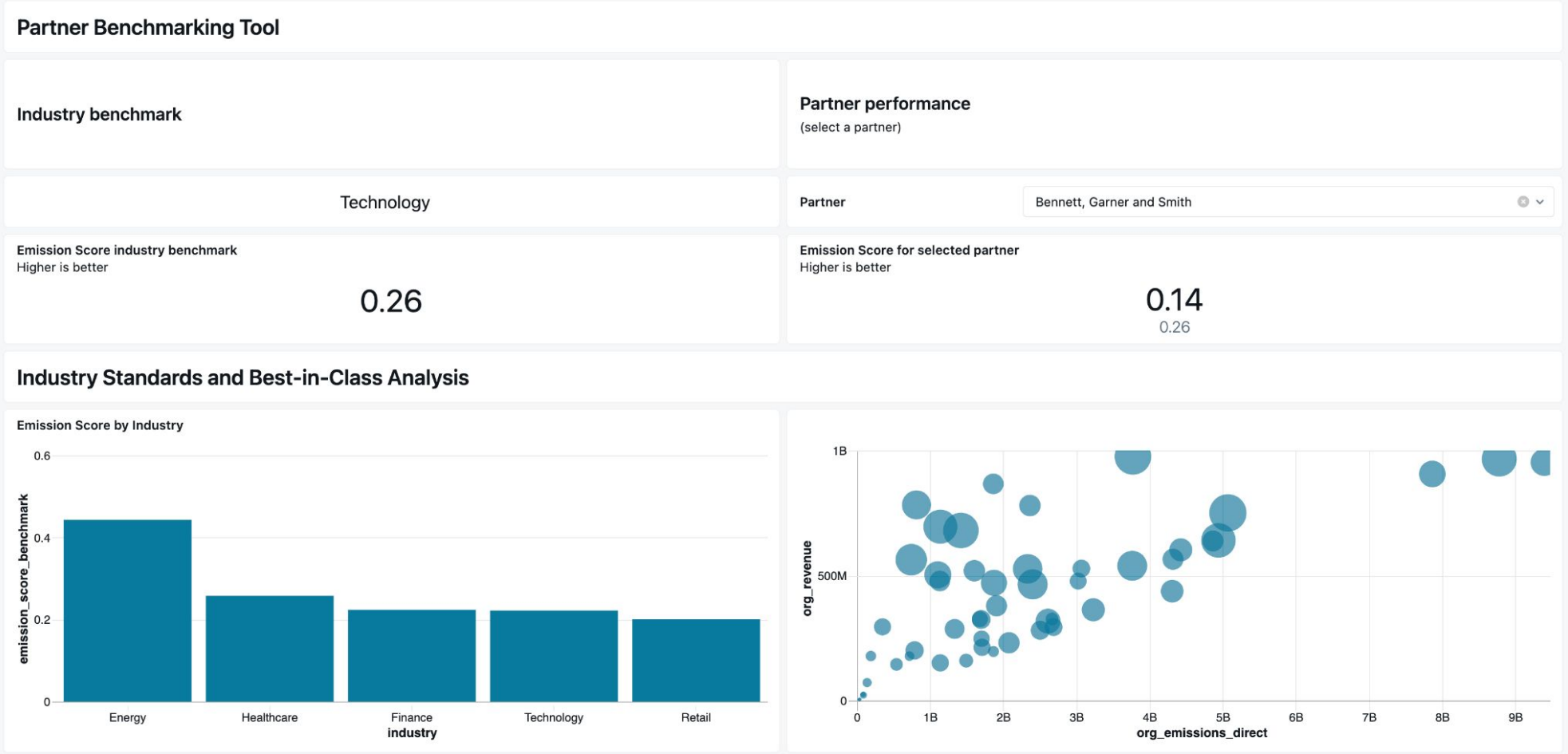
Document extraction with an Agent



The use case

- Your organization wants to reduce its direct and indirect greenhouse gas emissions
- To help reduce indirect (aka scope 3) emissions, you are responsible to enable the evaluation of your org's suppliers and business partners in terms of their sustainability performances by benchmarking them against their respective industry's standards.
- Proposed metric: "Emission score" = ratio of a business partner's revenue, expressed in millions USD and its direct emissions (Scope 1) expressed in tons of CO2 emitted on a yearly basis.
- End product: a business-facing interface that allows the benchmarking of your suppliers and business partners against industry standards.

Example outcome



What we provide you

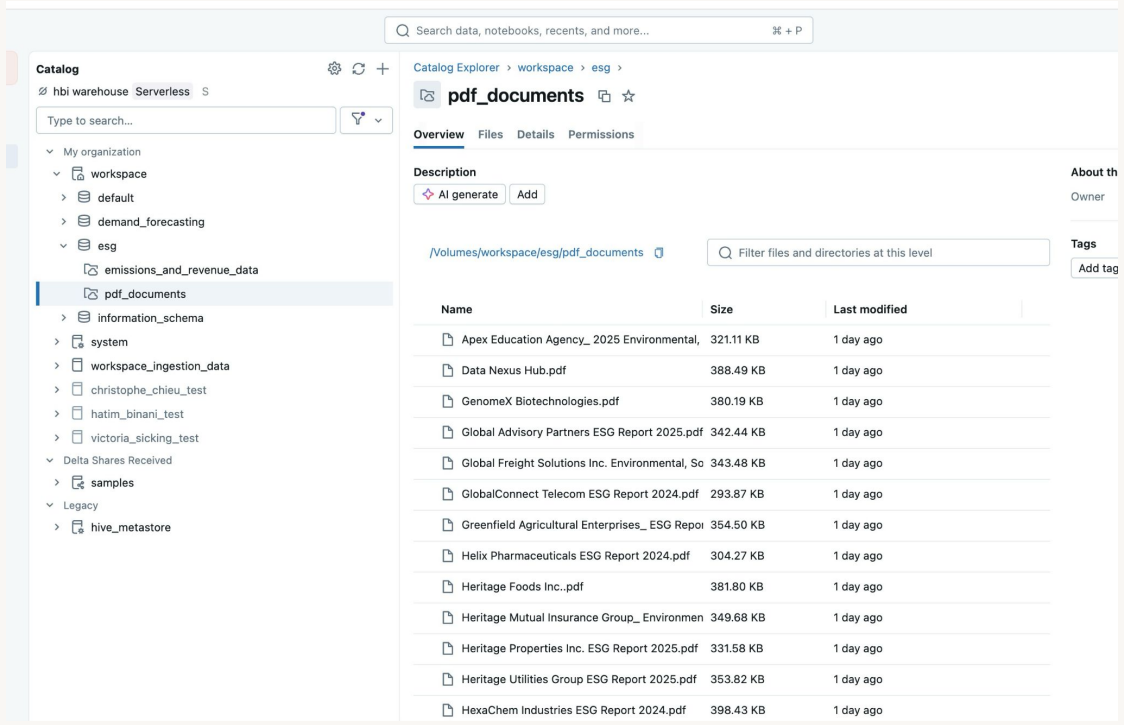
- 2 data sources:
  - **CSV data files** containing the emissions and revenue for public companies in several sectors. This will allow you to compute benchmarks against which to assess your suppliers / business partners.
  - **PDFs** provided by your business partners that report on their own scope 3 emissions

- Cluster & GPUs for your team

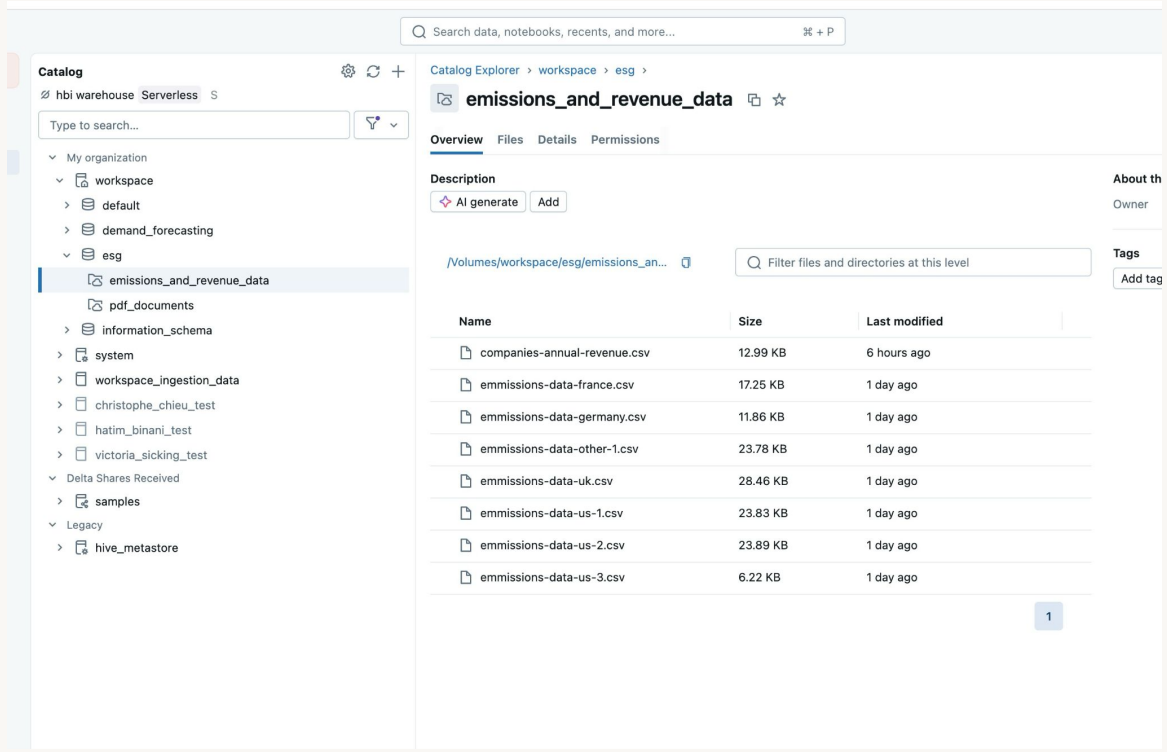
What we provide you

PDFs and CSVs

/Volumes/workspace/esg/pdf_documents



/Volumes/workspace/esg/emissions_and_revenue_data



What you will build (suggestion)

- A Information Extraction Agent (using Agent Bricks)
  - A Document Intelligence Agent that can extract the relevant data points from unstructured text

- A Job, scheduled Notebook or Lakeflow Declarative Pipeline
  - To productionize the Agent and automate the document extraction process
  - To combine with data from other companies in CSV format

- An AI/BI Dashboard or a Databricks App
  - To benchmark your business partner's performance versus others

- A Metric View (optional)
  - To standardize the definition of a KPI across the organization

Suggested high-level project plan

- Tip 1: Start building the dataset for the user interface, and then split the work for 2 and 3
- Tip 2: TBD

1. Data preparation and information extraction

PDF Loading (read raw data into a Delta table)

Information extraction using Agent Bricks (text data to structured data)

Data consolidation (join data from the different data sources and create a unified gold table)

2. User interface development

Create a Semantic layer with Unity Catalog Metric View

Option 1: Create an user-facing AI/BI Dashboard / AI/BI Genie Space

Option 2: Create a user-facing Databricks App

3. (Bonus) Productionisation using Lakeflow Declarative Pipelines

Leverage Declarative Pipelines to productionize the work done in step 1

Benefits:

- out-of-the-box incremental load
- data quality monitoring and enforcement framework
- near-real-time one click away

Get Started Here

- Access the Workspace
- Explore the data in the schema workspace.esg
- (Optional) Have a look at the reference slides for step-by-step instructions and helpful tips
- (Optional, in case you get stuck) Clone the Git Repository, and use the notebook in the folder esg/backup
  - Use the backup folder only if you get stuck, and speak with your SAs to help you

Reference Slides

Step 1: PDF loading and Information Extraction

- Start by loading the PDFs into a Delta Table
  - How to do it:
    - Use the button “Use PDFs in Agent Bricks” in the Agents Tab to load the PDFs
  - Outcome:
    - A table with at least one column containing the PDF text as string
- Use Agent Bricks to extract relevant information from the PDFs
  - How to do it:
    - Use Agent Bricks’ Information Extraction, for instructions see following slides
  - Outcome:
    - An agent that extracts information such as Scope 3 Emissions, Company Name, Annual Revenue, Number of employees, etc.
    - Think about what information could be interesting to display in a Dashboard / App

Step 2 : Data processing

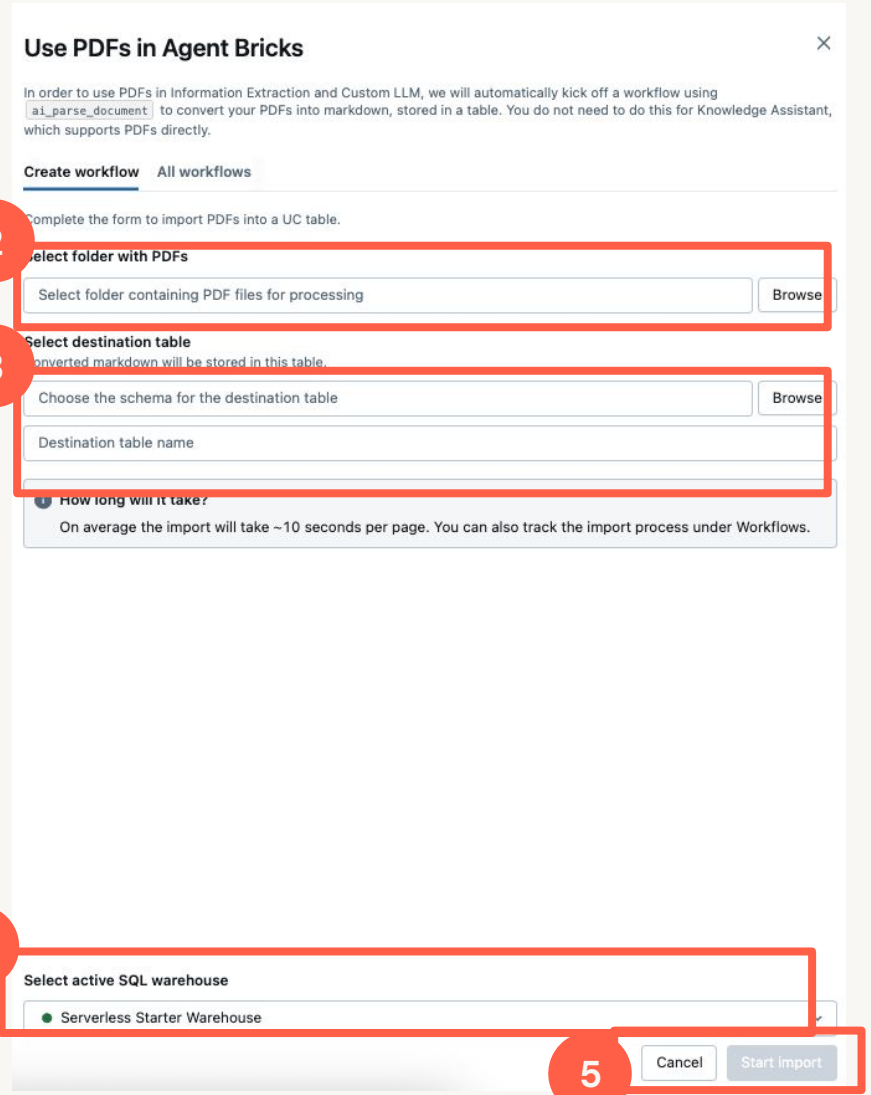
- Using the Information Extraction Agent you have just created, run batch inference on the table containing the PDF texts
  - How to do it:
    - Tip : from the Agent UI, deploy a ready to use Declarative pipeline to run batch inference on all PDFs
    - Add a post-processing step to handle edge cases and apply constraints on the extracted data
  - Outcome:
    - A reusable pipeline that handles extraction and postprocessing
    - A structured Delta Table with clean data extracted from PDFs
- Extract, clean and load the CSV data containing the emissions and revenue of public companies
  - How to do it:
    - Leverage Lakeflow Declarative Pipelines and/or Notebooks to extract structured data from the CSV
  - Outcome:
    - A reusable pipeline OR notebook that handles extraction and postprocessing
    - A structured Delta Table with clean data extracted from CSVs

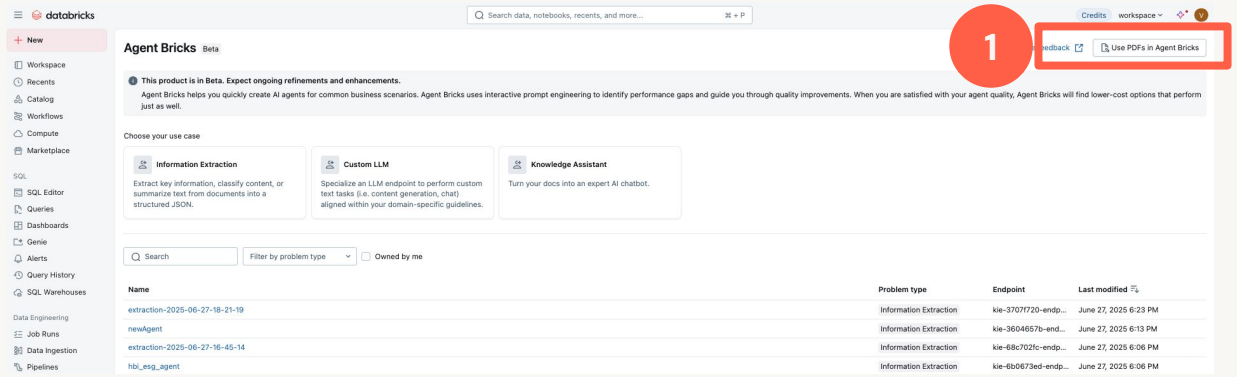
Agent Bricks Workflow

Documentation

1. Load your PDFs into a Unity Catalog table
2. Go to Agent Bricks and select “Information Extraction”
3. Specify the information you want to extract
4. Start the agent
5. Leverage the agent endpoint for batch inference on the data

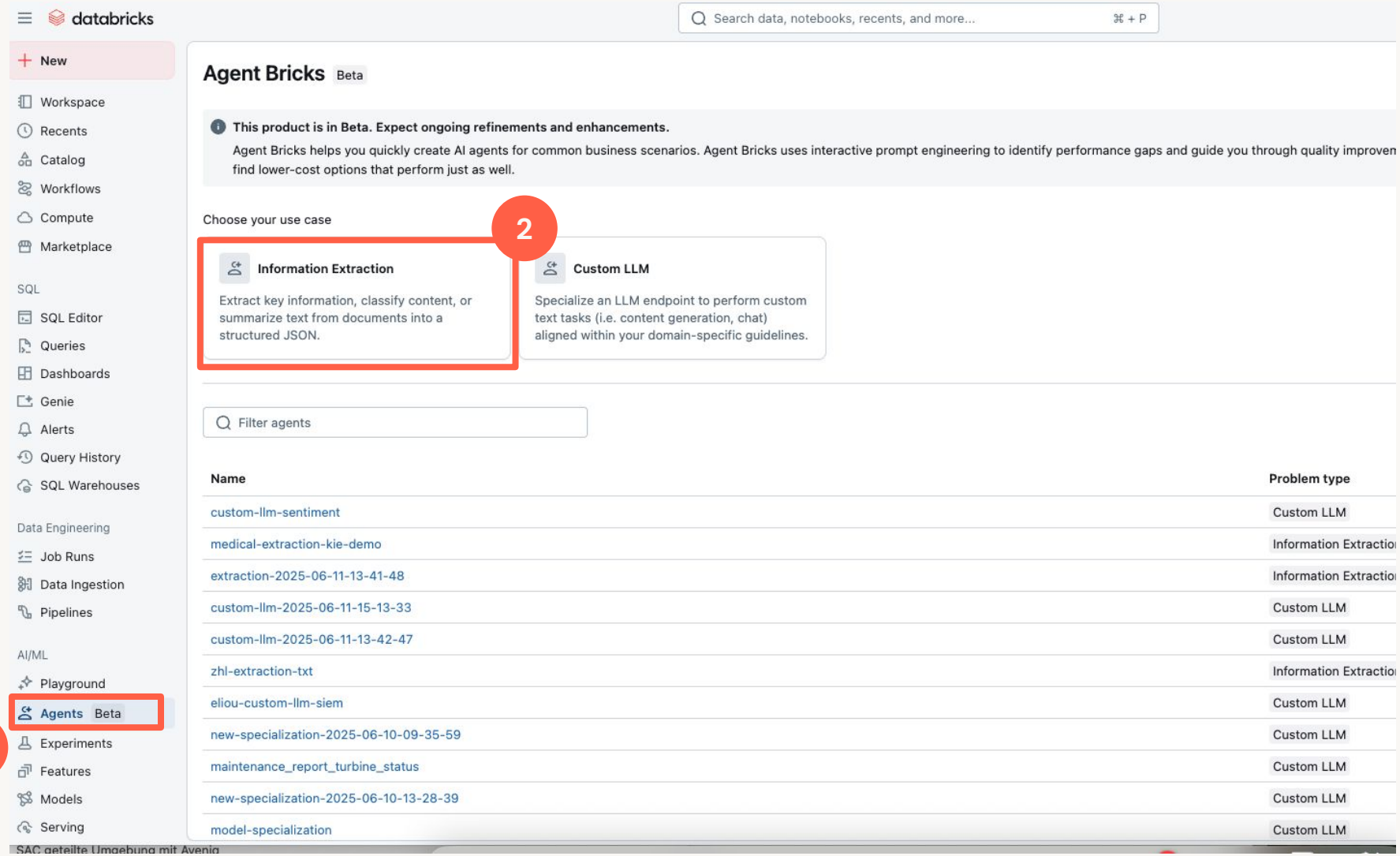
Instructions





1. Click on "Use PDFs in Agent Bricks"
2. Select Folder Containing PDFs
3. Select the destination schema and give the table a name
4. Select the compute
5. Start the processing and wait for the job to finish

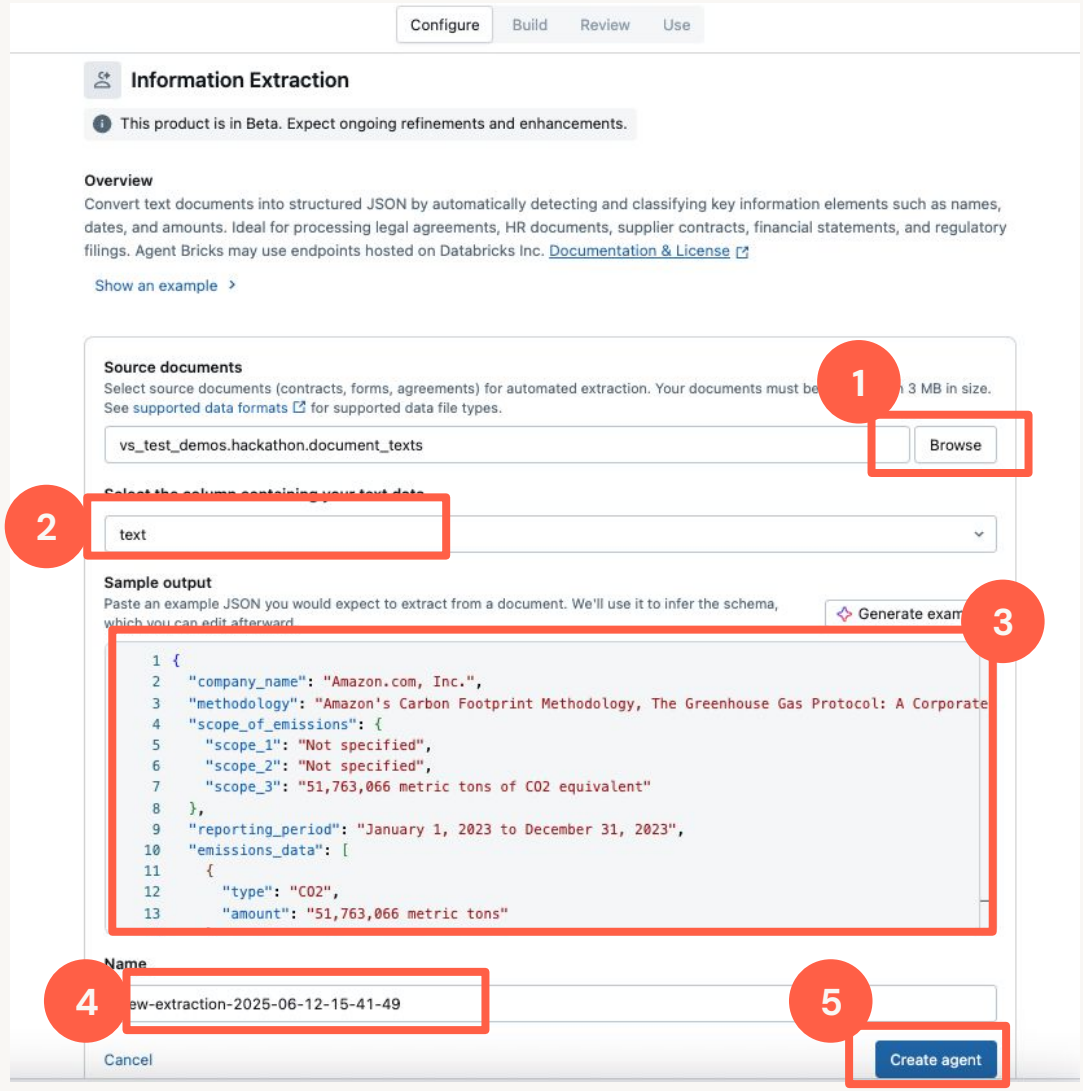
Instructions



1 Select “Agents” in the sidebar menu

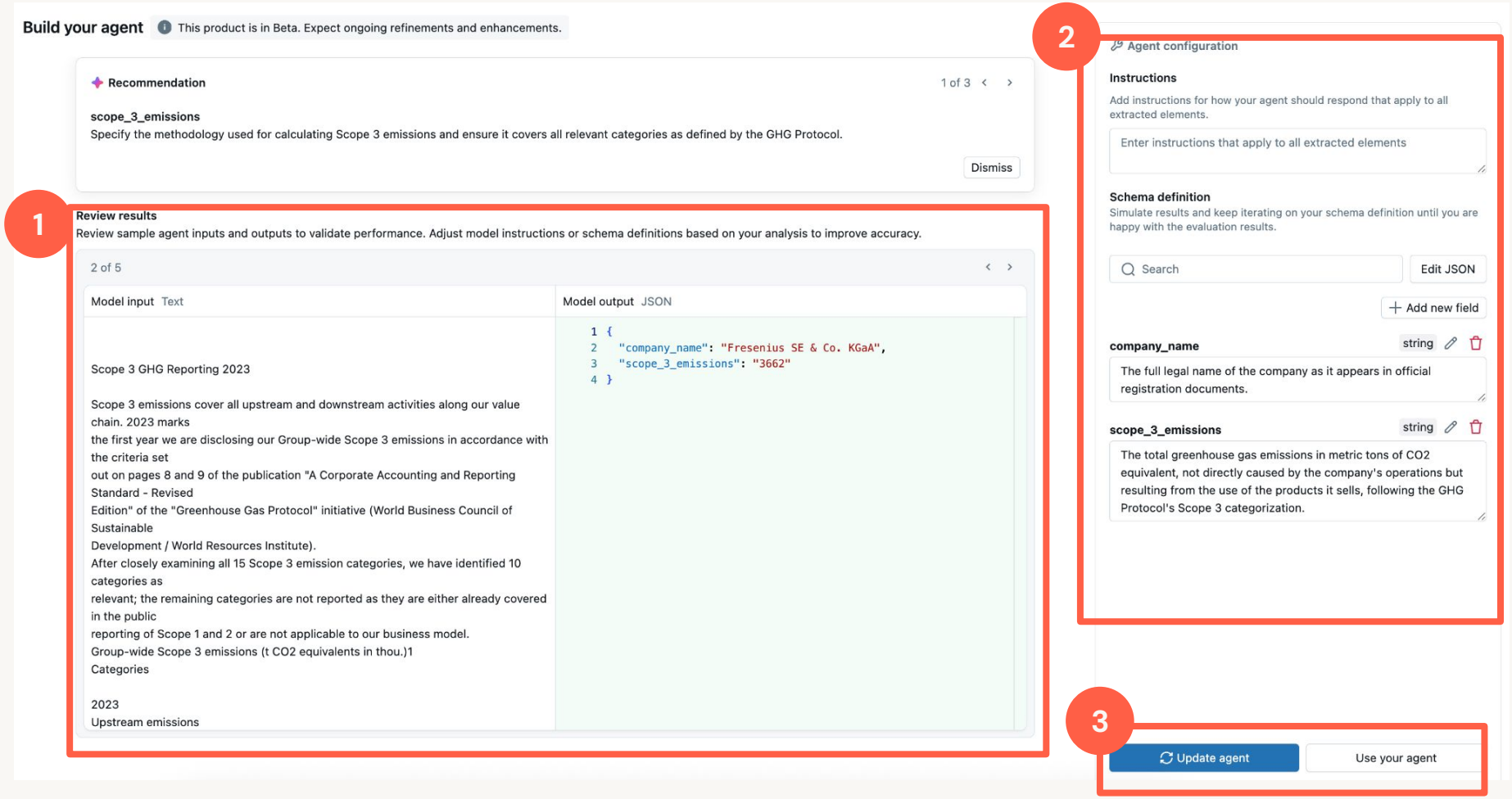
2 Select “Information Extraction”

Instructions



1. Browse the data and select the table with the pdf data
2. Select the column containing the text
3. Fill the sample output with the information you want to extract
4. Give the agent a name
5. Click “Create agent” to launch it

Instructions



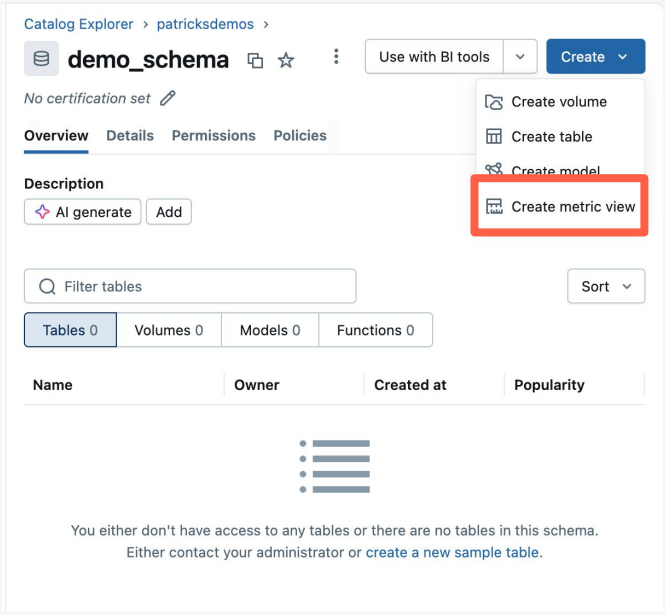
1. Review the sample results
2. Add descriptions to configure your agent
3. Update or use your agent

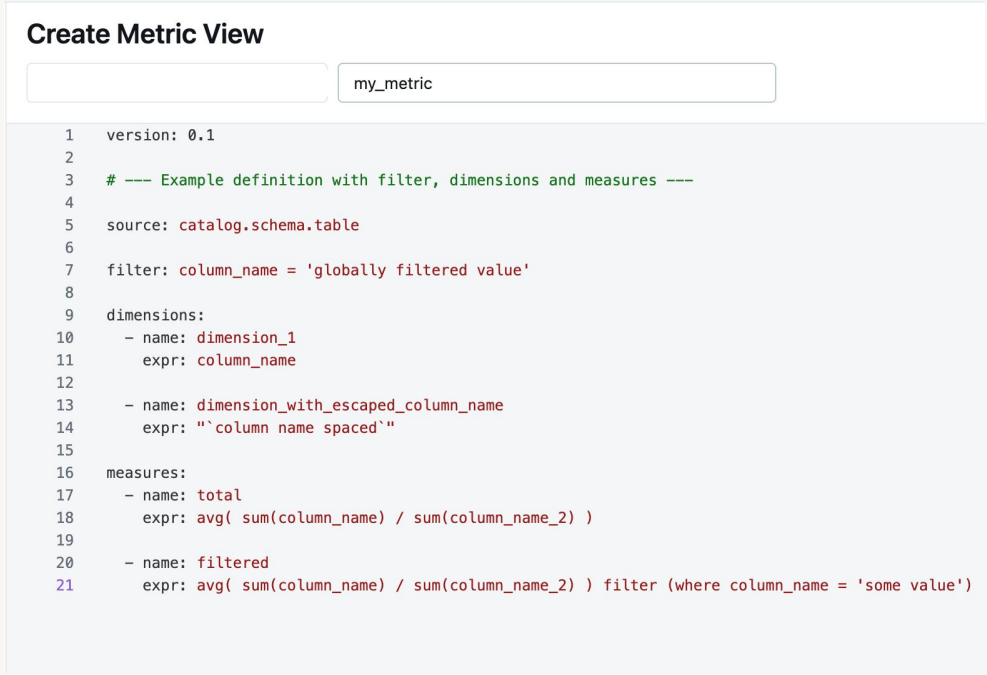
UC Metric Views 101

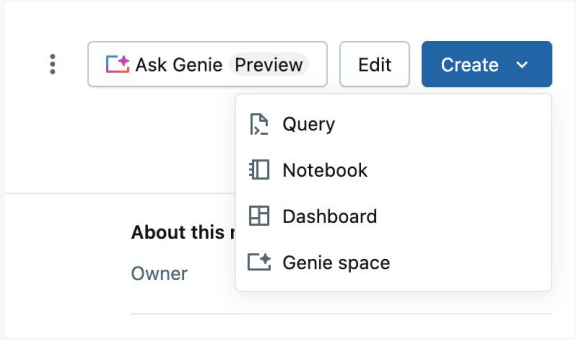
Documentation

- Metric views allow to define *Measures* using common SQL aggregations, together with a set of *Dimensions* that are attributes based on which we want to enable the Measures to be dynamically analyzed.

- Once defined, the different Measures can be used in DBSQL, AI/BI Dashboards, and AI/BI Genie!







UC Metric Views pro-tip 1

Documentation

- One way to compare filtered values with unfiltered values within a same dashboard is to create window measures :

```
- name: my_measure_1
  expr: <measure formula>
```

```
- name: my_measure_1_benchmark
  expr: <measure formula>
  window:
    - order: dimension_attribute_to_be_ignored
      range: all
      semiadditive: last
```

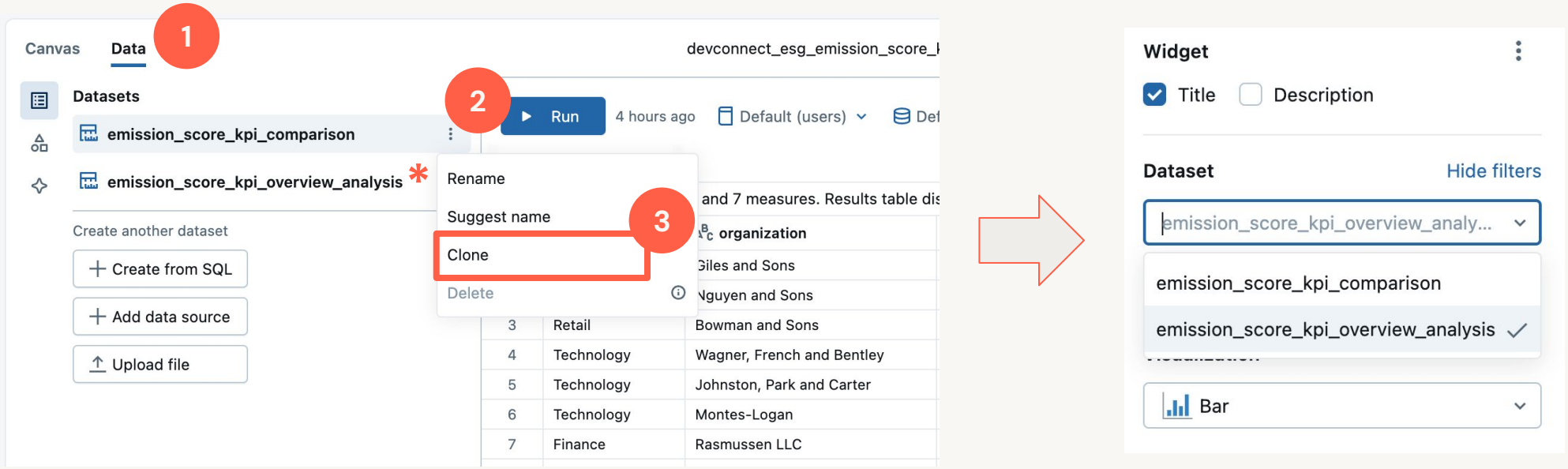
- When filtering the dashboard on `dimension_attribute_to_be_ignored`(e.g. a specific company), the measure `my_measure_1` will react as anticipated. On the other hand, `my_measure_1_benchmark`be filtered by other corresponding attributes (e.g. the industry in which the company operates) but not on the specific company itself!

UC Metric Views pro-tip 2

Documentation

- By default, Dashboard filters are applying to the whole dataset and cross-filtering (e.g. automatic filtering across visuals when clicking on a given part of a bar chart) is observed across the datasets as well.

- A way to make *Visual B* indifferent to *Dashboard Filters A* and to clicks on *Visual A* is to create *Visual B* from a different dataset, for example by cloning *the* initial dataset used for Filter A and Visual A and using that second dataset for Visual B.

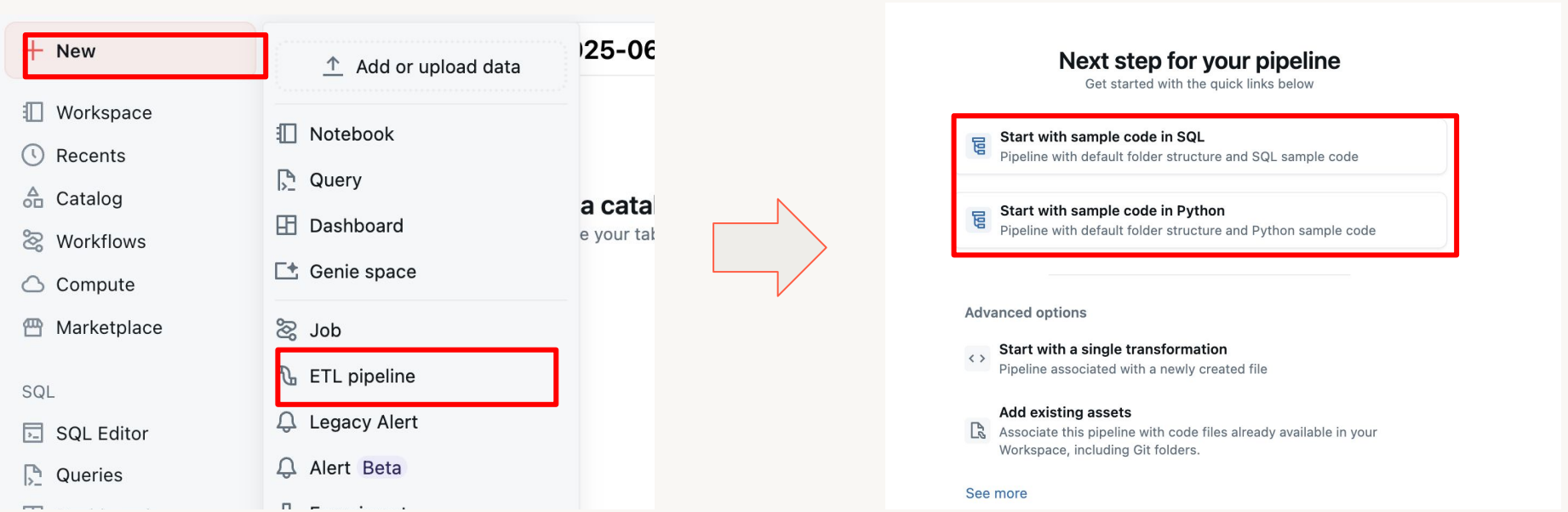


Declarative Pipelines 101

Documentation

- Lakeflow Declarative Pipelines is a framework for creating batch and streaming data pipelines in SQL and Python
- A common use for Lakeflow Declarative Pipelines is as follows : read data from source systems, transform that data based on requirements, such as data quality checks, and write the data to a target system, such as a data warehouse or a data lake.

QuickStart a Declarative Pipeline



Databricks Apps useful resources

- Getting started: Documentation, Medium blog post

- Many useful code snippets are available on the Databricks cookbook portal:
  - Accelerators for the Streamlit and Dash frameworks
  - Many use cases from interaction with Tables and Model Endpoints to dashboard embedding and API hosting.