UNIVERSITY OF MALAYA

WIH3003

BIG DATA APPLICATIONS AND ANALYTICS

Semester 1 2025/2026

Group Assignment

PREPARED BY:

FULL NAME	MATRIC NUMBER
KEK YI CI	22102539
TAY YI NING	23005229
YEOW XIN YI	23005230
KEAT TERNG CONG	23063762
KOAY KHOON LYN	23005235

Lecturer: Dr. Riyaz Ahamed Ariyaluran Habeeb Mohamed

Table of Contents

1.0 Introduction of Dataset 3
1.1 Dataset Overview 3
1.2 Attribute Description 4
2.0 Meeting Minutes Report 5
3.0 Hadoop and Spark 9
3.1 Introduction 9
3.2 Methodology 9
3.3 MapReduce Queries in Hadoop and Spark 10
3.4 Comparative Performance Analysis 18
3.5 Summary 19
4.0 Dashboard 20
4.1 Introduction of dashboard 20
4.2. Slicers 21
4.2.1 Release Year 21
4.2.2 Company 21
4.2.3 Category 21
4.2.4 Openness 21
4.3 Visualizations Included in the Dashboard 22
4.3.1 Visualization 1: Tool Releases by Year 22
4.3.2 Visualization 2: Tools by Category 22
4.3.3 Visualization 3: Open Source vs Closed Source 23
4.3.4 Visualization 4: Modality Support Counts 23
4.3.5 Visualization 5: Tools by Company 24
4.4 Conclusion 24
5.0 DataOps Architecture 25
5.1 Architecture Diagram 25
5.2 Layers and Components Explanation 26
6.0 References 28

1.0 Introduction of Dataset

1.1 Dataset Overview

The selected dataset ‘Generative AI Tools - Platforms 2025’ provides a big picture that illustrates the rapidly evolving artificial intelligence platforms as of 2025. The dataset is a structured form of data containing 113 records and 22 attributes which offers detailed metadata on the development, capabilities and accessibility of the tools. Each record is distinct and represents a unique generative AI tool that can come from the same or different organisations accordingly. This dataset was chosen for the assignment because it offers high-dimensional data which is suitable for big data analytics and allows comparative analysis between different AI models and their technical modalities such as text, image or video processing.

The records cover a wide historical range of AI development, starting from year 2013 up until today which captures both legacy systems and cutting edge foundation models. There are a few noticeable key characteristics found in the dataset. It contains a diversity of tools that includes major foundation models such as ChatGPT, Claude and Gemini; specialised coding assistants like GitHub Copilot and creative generating platforms like Midjourney. Besides, the dataset covers tools from major tech giants such as Google, OpenAI, Meta as well as some well known open-source communities like Hugging Face or Stability AI. Each tool is classified by the ‘canonical’ category and modality class that differentiate them for grouping and analysis purposes.

1.2 Attribute Description

The dataset contains a total of 22 attributes that describe the technical aspects of each tool. Table below shows the descriptions of the important attributes involved in the analysis later:

Feature	Data Type	Description
tool_name	String	The official name of the Generative AI tool, serves as the primary reference point for comparison and lookup.
company	String	The organisation or developer that is responsible for the tool (e.g., OpenAI, Google).
category_canonical	String	The general functionality or category of the AI tool (LLMs & Chat Assistants, Image Gen & Editing etc.), used to examine their primary purpose.
modality_canonical	String	This defines the type of data or media that is handled by the tools (text, image, audio, multimodal etc.) which is critical for understanding the tool's core capability.
open_source	Binary (0/1)	To identify whether the source code is publicly available (1 = Yes, 0 = No).
api_available	Binary (0/1)	To identify if the platform provides an Application Programming Interface (API) for integration into other systems. (1 = Yes, 0 = No).
release_year	Integer	The year where the AI tool was publicly released to track technological evolution over time.
mod_image / mod_video / mod_audio / mod_code / mod_multimodal	Binary (0/1)	It consists of binary indicators (0/1) to show if the tool supports a certain data type which allows the detailed analysis of feature support. (Example, how many tools support image processing).

2.0 Meeting Minutes Report

Meeting Date	22 November 2025
Meeting Time (Scheduled)	10 AM - 11 AM
Meeting Location	Google Meet (Online)
Minutes Drafted Date	22 November 2025

1. Attendees		
No	**Name**	**Student ID**
1	Kek Yi Ci	22102539
2	Tay Yi Ning	23005229
3	Yeow Xin Yi	23005230
4	Keat Terng Cong	23063762
5	Koay Khoon Lyn	23005235

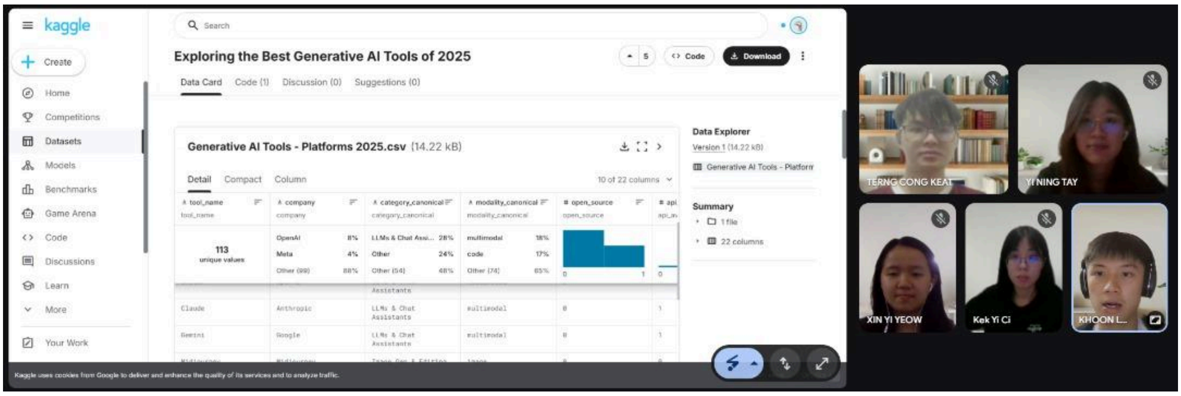
2. Brief Description / Agenda
● Opening & Welcome ● Dataset Options Presentation ● Open Discussion & Decision-Making ● Tasks Delegation ● Closing

3. Meeting Summary	
No	**Item Discussed**
1	**Opening and Welcome (10:00 AM - 10:03 AM)** ➔ The meeting was initiated by our group leader, Kek Yi Ci. Group members attended sharp at 10 AM and were welcomed, then the agenda was reviewed.
2	**Dataset Options Presentation (10:03 AM - 10:27 AM)** ➔ Tay Yi Ning introduced her dataset - AI Tools Usage 2023 which focused on the adaption and usage trends of AI tools throughout the year 2023. The dataset was designed to track how the popularity of different AI categories evolved from month to month. She elaborated that this dataset was

	structured to facilitate trend analysis. Some of the key columns mentioned were ‘Month’, ‘Usage Count’ and ‘Category’. ➔ Yeow Xin Yi asked for the total number of attributes in Yi Ning’s presented dataset and Yi Ning answered there were a total of 7 columns available. Yi Ning also added that the dataset could be utilised to create dashboard to visualise if certain tools see drops in usage during holidays. ➔ Yeow Xin Yi presented her dataset - Top AI Tools: Popularity & Valuation, the dataset has 40 major AI tools that focus on their market impact as of 2024. This dataset was purely business and market-oriented which linked specific tools to their financial valuation and subjective popularity score. Similar to the previous dataset, there were 7 attributes and 40 records. ➔ Kek Yi Ci questioned about the difference between two datasets proposed by Xin Yi and Yi Ning, Xin Yi explained that unlike the previous datasets that focused solely on technical specifications, her dataset focused more on financial aspects. ➔ Then, Kek Yi Ci proposed her dataset - Platform of Generative AI Tools 2025. The dataset examined the current leading AI tools in market released from 2013 to 2025 and determined their modality and canonical category. Also, it contained details like api status and open source status. She claimed that the dataset has a total of 22 columns which was sufficient for this assignment. ➔ No one raised questions towards her presentation. Koay Khoon Lyn claimed that his dataset has similar structure and described the same items as Yi Ci’s dataset. He further supported that the dataset contains 113 distinct AI tools that is considered very sufficient to conduct visualisation dashboards. The dataset can be used to evaluate the accessibility for public and also the dominant AI tools category in the current market. ➔ After this, Keat Terng Cong presented his dataset - AI Hallucination Cases Data 2025. He claimed that this dataset could be quite suited for the assignment as it has a slightly larger data volume (approx. 400 - 600 rows), it contains 10 to 12 key attributes that allow multidimensional analysis. He stated the most relevant columns like ‘Case Name’, ‘AI Tool’, ‘Hallucination’ and ‘Outcome’. Unlike general ‘usage’ logs, this dataset specifically targets failure modes of LLMs in high-stakes environments.
3	**Open Discussions & Decision-Making (10:27 AM - 10:43 AM)** ➔ After the brief presentation of datasets by the members, Kek Yi Ci suggested that every member provide some opinions toward the datasets. ➔ Tay Yi Ning claimed that while her dataset could perform MapReduce operations to calculate the total usage of quarter or identify the fastest growing category in 2023 but comparatively it has less attributes compared to the other proposed datasets. ➔ Yeow Xin Yi raised the same concern and stated that her dataset contains 7 attributes only and this might be one of the challenges during dashboards generation later. She claimed that it would be better to have the other dataset to be prioritised first and make her dataset as a backup option. ➔ Koay Khoon Lyn suggested that the options can be narrowed between Yi

	Ci’s and Terng Cong’s datasets. Both of these datasets contain sufficient attributes to be analysed and also fulfill the assignment requirements. He suggested Yi Ci and Terng Cong to elaborate the potential of their datasets. ➔ Kek Yi Ci stated that the assignment instructed to find a dataset that is rich in data quality, variety and relevance to industry, potential for innovation and interoperability. Her dataset fulfilled these requirements where attributes like canonical and modality class explain AI variations, api and open source status explain accessibility for the public. These can be really potential to analyse each AI tool in a comprehensive manner. ➔ Keat Terng Cong agreed with what Yi Ci said, and further explained that although his dataset is rich with attributes and record numbers, some of the columns contain long text which can be quite difficult to handle during the tasks later. He suggested the other members can consider Yi Ci’s dataset. ➔ Finally, all members voted and Yi Ci’s dataset was selected at the end of this section.
4	**Tasks Delegation (10:43 AM - 10:51 AM)** ➔ After selection of dataset, Kek Yi Ci suggested that the tasks should be distributed into 5 parts according to the rubrics of the assignment: One person handles ‘Introduction to dataset’ and ‘Meeting minutes report’, two person handle ‘Comparison on Hadoop and Spark’, one person handles ‘Dashboard’ and one person handles ‘DataOps architecture or framework’. ➔ Kek Yi Ci suggested group members to go through scopes and responsibility for each role after the meeting. She claimed that a poll will be created in the WhatsApp group after the meeting and asked members to select their tasks accordingly. ➔ The other members agreed with the task distribution.
5	**Closing (10:51 AM - 10:53 AM)** ➔ The meeting ended earlier than scheduled time at 10:53 AM. Participants were thanked for their active participation and valuable contributions during the meeting.

4. Screenshots of Online Meeting



Exploring the Best Generative AI Tools of 2025

Data Card Code (1) Discussion (0) Suggestions (0)

Generative AI Tools - Platforms 2025.csv (14.22 kB)

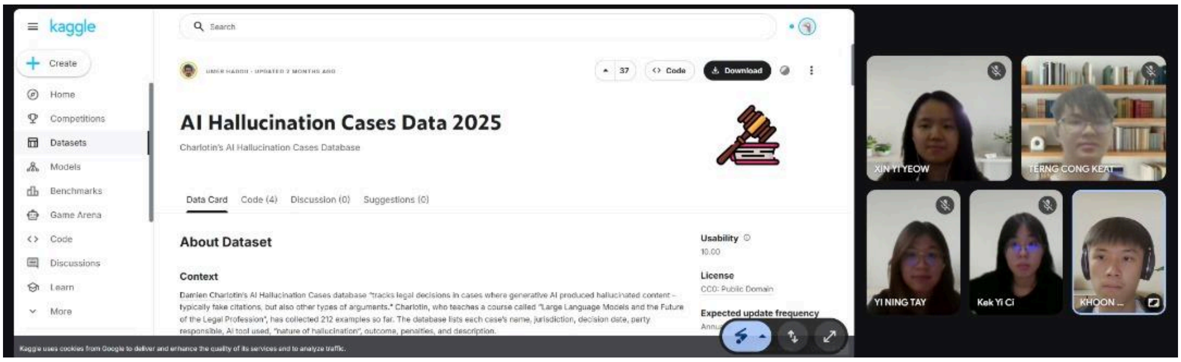
Detail Compact Column 10 of 22 columns

tool_name tool_name	company company	category_canonical category_canonical	modality_canonical modality_canonical	open_source open_source	api api_a
113 unique values	OpenAI 8% Meta 4% Other (99) 88%	LLMs & Chat Assi... 28% Other 24% Other (54) 48%	multimodal 18% code 17% Other (74) 65%	0 1	0
		Assistants			
Claude	Anthropic	LLMs & Chat Assistants	multimodal	0	1
Gemini	Google	LLMs & Chat Assistants	multimodal	0	1

Data Explorer
Version 1 (14.22 kB)
Generative AI Tools - Platform

Summary
1 file
22 columns

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.



UMER HADDII · UPDATED 2 MONTHS AGO

AI Hallucination Cases Data 2025

Charlotin's AI Hallucination Cases Database

Data Card Code (4) Discussion (0) Suggestions (0)

About Dataset

Context

Damien Charlotin's AI Hallucination Cases database "tracks legal decisions in cases where generative AI produced hallucinated content – typically fake citations, but also other types of arguments." Charlotin, who teaches a course called "Large Language Models and the Future of the Legal Profession", has collected 212 examples so far. The database lists each case's name, jurisdiction, decision date, party responsible, AI tool used, "nature of hallucination", outcome, penalties, and description.

Usability
10.00

License
CC0: Public Domain

Expected update frequency
Annu

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

3.0 Hadoop and Spark

3.1 Introduction

Hadoop and Spark are both distributed computing frameworks designed to process and analyze large-scale data efficiently. Hadoop MapReduce is a Java-based framework that follows a Map and Reduce paradigm: the Map phase transforms data into key/value pairs, and the Reduce phase aggregates or summarizes these pairs. Processing is distributed across slave nodes, with results collected at the master node.

In contrast, Apache Spark is a high-performance cluster computing framework that extends Hadoop's capabilities by emphasizing in-memory computation for faster processing. Spark uses Resilient Distributed Datasets (RDDs) as its core data structure, which ensures fault tolerance and immutability. RDDs can hold complex objects from Java, Scala, or Python, allowing versatile data transformations and analyses.

3.2 Methodology

The goal of this study is to compare the performance of Hadoop and Spark in processing a large CSV dataset stored in the Hadoop Distributed File System (HDFS). We implemented three programs in Java (for Hadoop) and Scala (for Spark), compiled them into JAR files, and executed them via the command-line interface. Each program reads the input CSV file, performs its computations, and writes the results to a specified output directory in HDFS.

The queries executed in both frameworks are:

1. Category Count – counts the number of AI tools in each category (e.g., LLMs, Image Gen, Video Gen).
2. Average Modality Count by Company – calculates the average number of modalities supported by the tools developed by each company.
3. Release Year Count – counts how many AI tools were released in each year.

To evaluate efficiency, we collected metrics including execution time, memory consumption, and throughput, calculated as:

$$\text{Throughput} = \text{Number of Records} / \text{Execution Time}$$

Each query is executed five times, and the average values of these metrics are calculated to provide a comprehensive assessment of Hadoop's and Spark's performance.

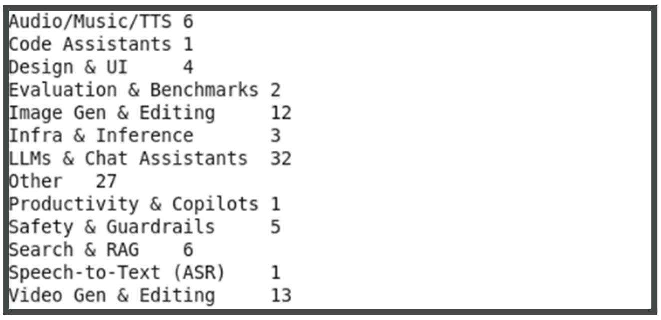
3.3 MapReduce Queries in Hadoop and Spark

Source Code Hadoop: https://github.com/Yining118/BigData,
Source Code Spark: https://github.com/yicikek/BigData

Query 1: Total number of AI tools in each category

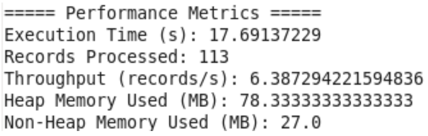
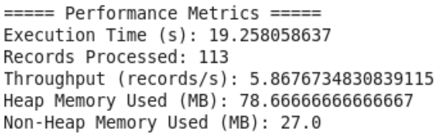
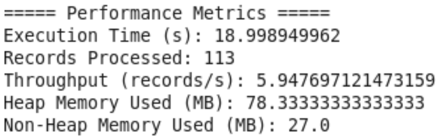
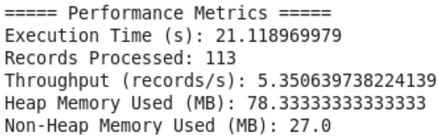
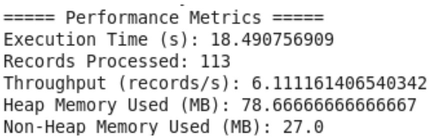
a) Hadoop

Output

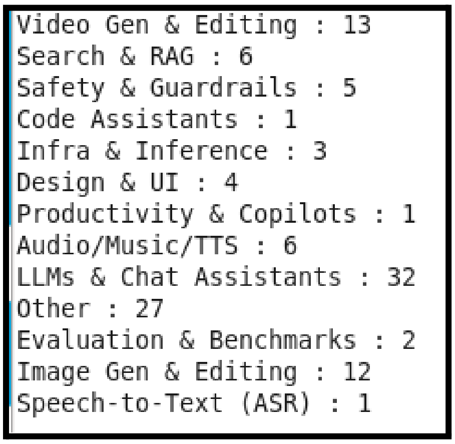


```
Audio/Music/TTS 6
Code Assistants 1
Design & UI     4
Evaluation & Benchmarks 2
Image Gen & Editing     12
Infra & Inference       3
LLMs & Chat Assistants  32
Other   27
Productivity & Copilots 1
Safety & Guardrails     5
Search & RAG    6
Speech-to-Text (ASR)    1
Video Gen & Editing     13
```

Performance

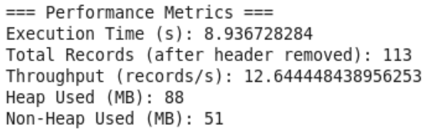
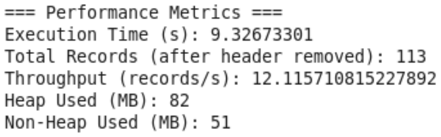
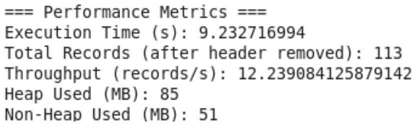
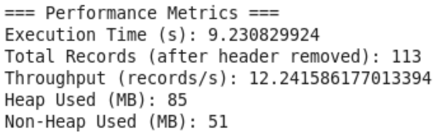
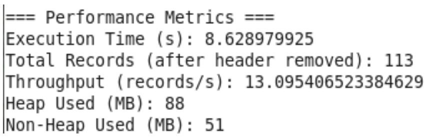
Iteration	Execution Time(s)	Throughput (Records/s)	Heap Memory Used (MB)	Non-Heap Memory Used (MB)	Evidence
1	17.691	6.387	78.333	27.000	 ===== Performance Metrics ===== Execution Time (s): 17.69137229 Records Processed: 113 Throughput (records/s): 6.387294221594836 Heap Memory Used (MB): 78.33333333333333 Non-Heap Memory Used (MB): 27.0
2	19.258	5.868	78.667	27.000	 ===== Performance Metrics ===== Execution Time (s): 19.258058637 Records Processed: 113 Throughput (records/s): 5.8676734830839115 Heap Memory Used (MB): 78.66666666666667 Non-Heap Memory Used (MB): 27.0
3	19.000	5.948	78.333	27.000	 ===== Performance Metrics ===== Execution Time (s): 18.998949962 Records Processed: 113 Throughput (records/s): 5.947697121473159 Heap Memory Used (MB): 78.33333333333333 Non-Heap Memory Used (MB): 27.0
4	21.119	5.351	78.333	27.000	 ===== Performance Metrics ===== Execution Time (s): 21.118969979 Records Processed: 113 Throughput (records/s): 5.350639738224139 Heap Memory Used (MB): 78.33333333333333 Non-Heap Memory Used (MB): 27.0
5	18.491	6.111	78.667	27.000	 ===== Performance Metrics ===== Execution Time (s): 18.490756909 Records Processed: 113 Throughput (records/s): 6.111161406540342 Heap Memory Used (MB): 78.66666666666667 Non-Heap Memory Used (MB): 27.0
Average	**19.11**	**7.034**	**78.47**	**27.00**	**-**

b) Spark
Output



```
Video Gen & Editing : 13
Search & RAG : 6
Safety & Guardrails : 5
Code Assistants : 1
Infra & Inference : 3
Design & UI : 4
Productivity & Copilots : 1
Audio/Music/TTS : 6
LLMs & Chat Assistants : 32
Other : 27
Evaluation & Benchmarks : 2
Image Gen & Editing : 12
Speech-to-Text (ASR) : 1
```

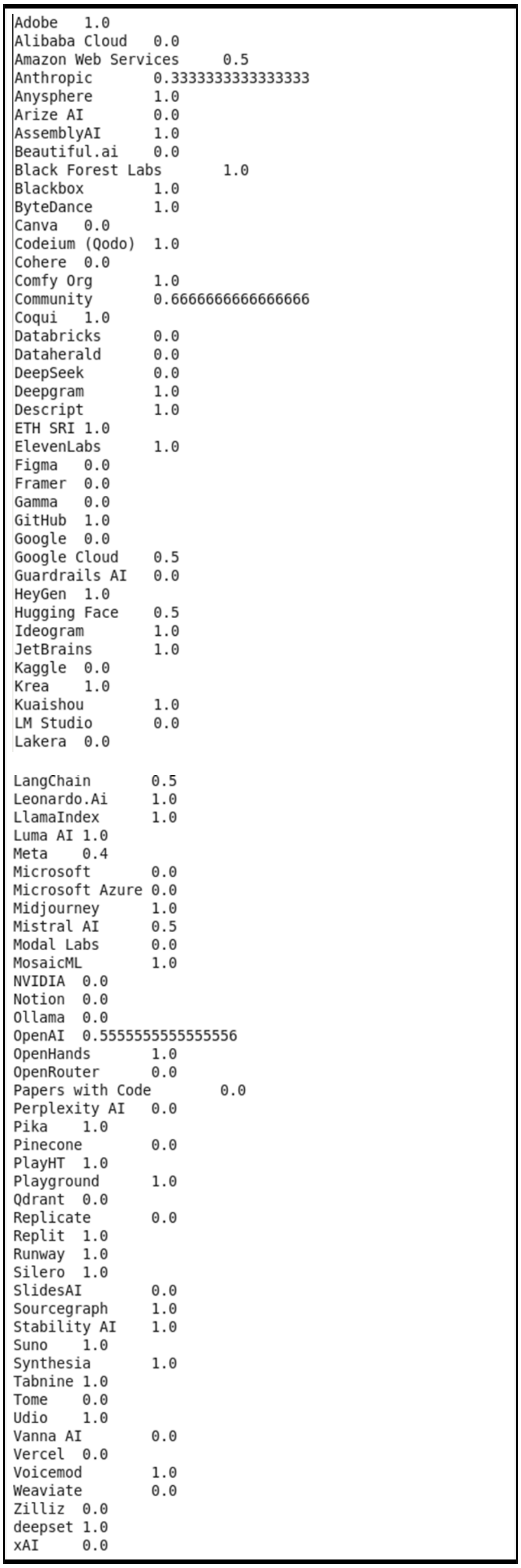
Performance

Iteration	Execution Time(s)	Throughput (Records/s)	Heap Memory Used (MB)	Non-Heap Memory Used (MB)	Evidence
1	8.936	12.644	88.000	51.000	 === Performance Metrics === Execution Time (s): 8.936728284 Total Records (after header removed): 113 Throughput (records/s): 12.644448438956253 Heap Used (MB): 88 Non-Heap Used (MB): 51
2	9.327	12.115	82.000	51.000	 === Performance Metrics === Execution Time (s): 9.32673301 Total Records (after header removed): 113 Throughput (records/s): 12.115710815227892 Heap Used (MB): 82 Non-Heap Used (MB): 51
3	9.232	12.239	85.000	51.000	 === Performance Metrics === Execution Time (s): 9.232716994 Total Records (after header removed): 113 Throughput (records/s): 12.239084125879142 Heap Used (MB): 85 Non-Heap Used (MB): 51
4	9.231	12.242	85.000	51.000	 === Performance Metrics === Execution Time (s): 9.230829924 Total Records (after header removed): 113 Throughput (records/s): 12.241586177013394 Heap Used (MB): 85 Non-Heap Used (MB): 51
5	8.628	13.095	88.000	51.000	 === Performance Metrics === Execution Time (s): 8.628979925 Total Records (after header removed): 113 Throughput (records/s): 13.095406523384629 Heap Used (MB): 88 Non-Heap Used (MB): 51
Average	**9.07**	**12.47**	**85.60**	**51.00**	**-**

Query 2: Average number of modalities supported by the tools developed by each company

a) Hadoop

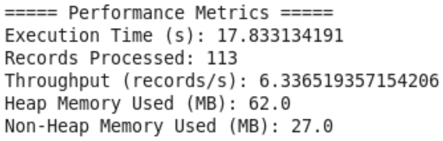
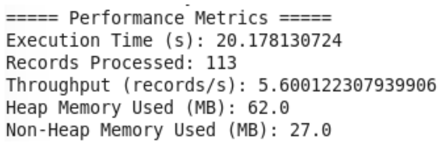
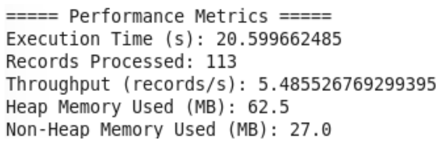
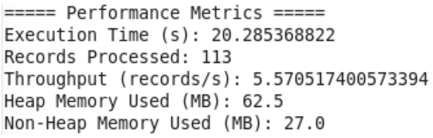
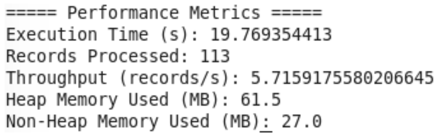
Output



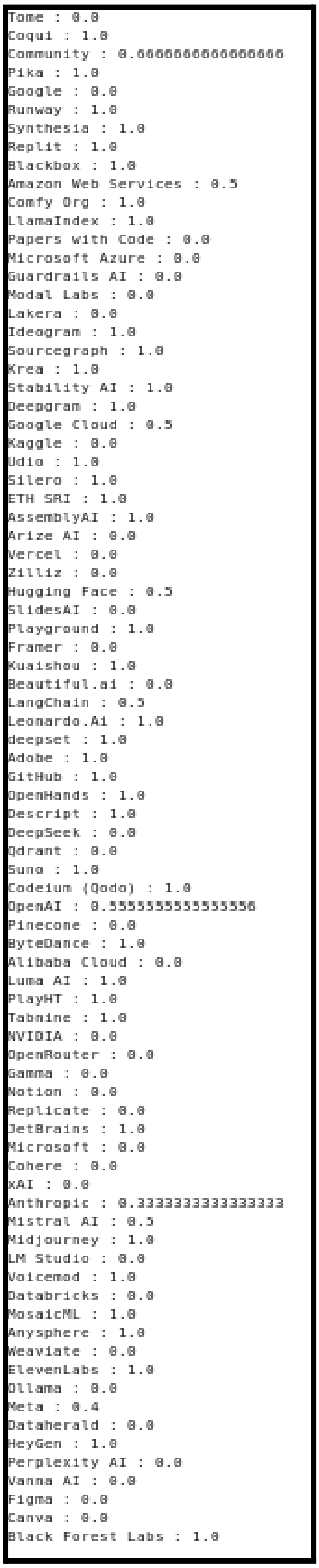
```
Adobe   1.0
Alibaba Cloud   0.0
Amazon Web Services     0.5
Anthropic       0.3333333333333333
Anysphere       1.0
Arize AI        0.0
AssemblyAI      1.0
Beautiful.ai    0.0
Black Forest Labs       1.0
Blackbox        1.0
ByteDance       1.0
Canva   0.0
Codeium (Qodo)  1.0
Cohere  0.0
Comfy Org       1.0
Community       0.6666666666666666
Coqui   1.0
Databricks      0.0
Dataherald      0.0
DeepSeek        0.0
Deepgram        1.0
Descript        1.0
ETH SRI 1.0
ElevenLabs      1.0
Figma   0.0
Framer  0.0
Gamma   0.0
GitHub  1.0
Google  0.0
Google Cloud    0.5
Guardrails AI   0.0
HeyGen  1.0
Hugging Face    0.5
Ideogram        1.0
JetBrains       1.0
Kaggle  0.0
Krea    1.0
Kuaishou        1.0
LM Studio       0.0
Lakera  0.0

LangChain       0.5
Leonardo.Ai     1.0
LlamaIndex      1.0
Luma AI 1.0
Meta    0.4
Microsoft       0.0
Microsoft Azure 0.0
Midjourney      1.0
Mistral AI      0.5
Modal Labs      0.0
MosaicML        1.0
NVIDIA  0.0
Notion  0.0
Ollama  0.0
OpenAI  0.5555555555555556
OpenHands       1.0
OpenRouter      0.0
Papers with Code        0.0
Perplexity AI   0.0
Pika    1.0
Pinecone        0.0
PlayHT  1.0
Playground      1.0
Qdrant  0.0
Replicate       0.0
Replit  1.0
Runway  1.0
Silero  1.0
SlidesAI        0.0
Sourcegraph     1.0
Stability AI    1.0
Suno    1.0
Synthesia       1.0
Tabnine 1.0
Tome    0.0
Udio    1.0
Vanna AI        0.0
Vercel  0.0
Voicemod        1.0
Weaviate        0.0
Zilliz  0.0
deepset 1.0
xAI     0.0
```

Performance

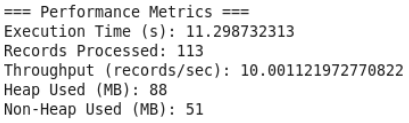
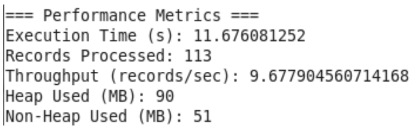
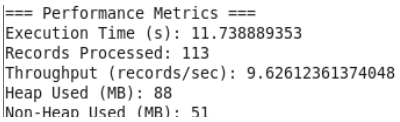
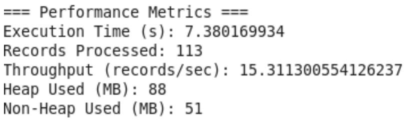
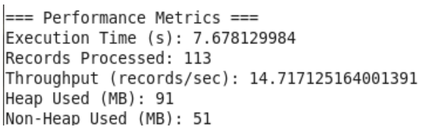
Iteration	Execution Time(s)	Throughput (Records/s)	Heap Memory Used (MB)	Non-Heap Memory Used (MB)	Evidence
1	17.833	6.337	62.000	27.000	 ===== Performance Metrics ===== Execution Time (s): 17.833134191 Records Processed: 113 Throughput (records/s): 6.336519357154206 Heap Memory Used (MB): 62.0 Non-Heap Memory Used (MB): 27.0
2	20.178	5.600	62.000	27.000	 ===== Performance Metrics ===== Execution Time (s): 20.178130724 Records Processed: 113 Throughput (records/s): 5.600122307939906 Heap Memory Used (MB): 62.0 Non-Heap Memory Used (MB): 27.0
3	20.600	5.486	62.500	27.000	 ===== Performance Metrics ===== Execution Time (s): 20.599662485 Records Processed: 113 Throughput (records/s): 5.485526769299395 Heap Memory Used (MB): 62.5 Non-Heap Memory Used (MB): 27.0
4	20.285	5.571	62.500	27.000	 ===== Performance Metrics ===== Execution Time (s): 20.285368822 Records Processed: 113 Throughput (records/s): 5.570517400573394 Heap Memory Used (MB): 62.5 Non-Heap Memory Used (MB): 27.0
5	19.769	5.716	61.500	27.000	 ===== Performance Metrics ===== Execution Time (s): 19.769354413 Records Processed: 113 Throughput (records/s): 5.7159175580206645 Heap Memory Used (MB): 61.5 Non-Heap Memory Used (MB): 27.0
Average	**19.73**	**5.74**	**62.10**	**27.00**	**-**

b) Spark Output



```
Tome : 0.0
Coqui : 1.0
Community : 0.6666666666666666
Pika : 1.0
Google : 0.0
Runway : 1.0
Synthesia : 1.0
Replit : 1.0
Blackbox : 1.0
Amazon Web Services : 0.5
Comfy Org : 1.0
LlamaIndex : 1.0
Papers with Code : 0.0
Microsoft Azure : 0.0
Guardrails AI : 0.0
Modal Labs : 0.0
Lakera : 0.0
Ideogram : 1.0
Sourcegraph : 1.0
Krea : 1.0
Stability AI : 1.0
Deepgram : 1.0
Google Cloud : 0.5
Kaggle : 0.0
Udio : 1.0
Silero : 1.0
ETH SRI : 1.0
AssemblyAI : 1.0
Arize AI : 0.0
Vercel : 0.0
Zilliz : 0.0
Hugging Face : 0.5
SlidesAI : 0.0
Playground : 1.0
Framer : 0.0
Kuaishou : 1.0
Beautiful.ai : 0.0
LangChain : 0.5
Leonardo.Ai : 1.0
deepset : 1.0
Adobe : 1.0
GitHub : 1.0
OpenHands : 1.0
Descript : 1.0
DeepSeek : 0.0
Qdrant : 0.0
Suno : 1.0
Codeium (Qodo) : 1.0
OpenAI : 0.5555555555555556
Pinecone : 0.0
ByteDance : 1.0
Alibaba Cloud : 0.0
Luma AI : 1.0
PlayHT : 1.0
Tabnine : 1.0
NVIDIA : 0.0
OpenRouter : 0.0
Gamma : 0.0
Notion : 0.0
Replicate : 0.0
JetBrains : 1.0
Microsoft : 0.0
Cohere : 0.0
xAI : 0.0
Anthropic : 0.3333333333333333
Mistral AI : 0.5
Midjourney : 1.0
LM Studio : 0.0
Voicemod : 1.0
Databricks : 0.0
MosaicML : 1.0
Anysphere : 1.0
Weaviate : 0.0
ElevenLabs : 1.0
Ollama : 0.0
Meta : 0.4
Dataherald : 0.0
HeyGen : 1.0
Perplexity AI : 0.0
Vanna AI : 0.0
Figma : 0.0
Canva : 0.0
Black Forest Labs : 1.0
```

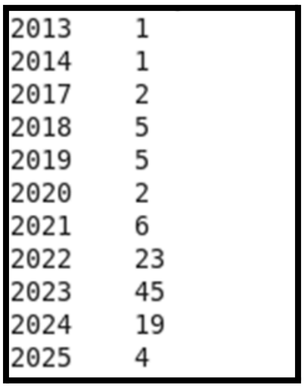
Performance

Iteration	**Execution Time(s)**	**Throughput (Records/s)**	**Heap Memory Used (MB)**	**Non-Heap Memory Used (MB)**	**Evidence**
1	11.299	10.001	88.000	51.000	 === Performance Metrics === Execution Time (s): 11.298732313 Records Processed: 113 Throughput (records/sec): 10.001121972770822 Heap Used (MB): 88 Non-Heap Used (MB): 51
2	11.676	9.677	90.000	51.000	 === Performance Metrics === Execution Time (s): 11.676081252 Records Processed: 113 Throughput (records/sec): 9.677904560714168 Heap Used (MB): 90 Non-Heap Used (MB): 51
3	11.739	9.626	88.000	51.000	 === Performance Metrics === Execution Time (s): 11.738889353 Records Processed: 113 Throughput (records/sec): 9.62612361374048 Heap Used (MB): 88 Non-Heap Used (MB): 51
4	7.380	15.311	88.000	51.000	 === Performance Metrics === Execution Time (s): 7.380169934 Records Processed: 113 Throughput (records/sec): 15.311300554126237 Heap Used (MB): 88 Non-Heap Used (MB): 51
5	7.678	14.717	91.000	51.000	 === Performance Metrics === Execution Time (s): 7.678129984 Records Processed: 113 Throughput (records/sec): 14.717125164001391 Heap Used (MB): 91 Non-Heap Used (MB): 51
Average	**9.95**	**11.87**	**89.00**	**51.00**	**-**

Query 3: Total number of AI tools released each year

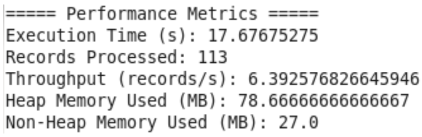
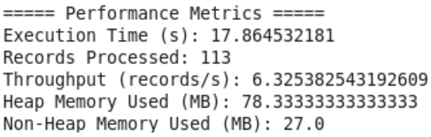
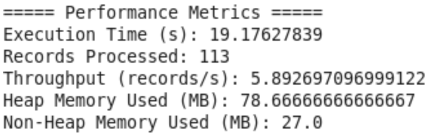
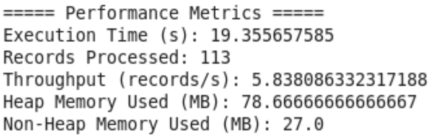
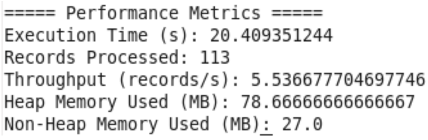
a) Hadoop

Output



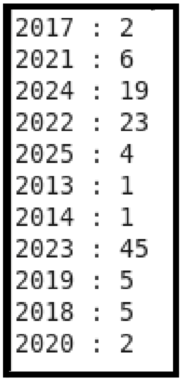
```
2013    1
2014    1
2017    2
2018    5
2019    5
2020    2
2021    6
2022    23
2023    45
2024    19
2025    4
```

Performance

Iteration	Execution Time(s)	Throughput (Records/s)	Heap Memory Used (MB)	Non-Heap Memory Used (MB)	Evidence
1	17.677	6.393	78.667	27.000	 ===== Performance Metrics ===== Execution Time (s): 17.67675275 Records Processed: 113 Throughput (records/s): 6.392576826645946 Heap Memory Used (MB): 78.66666666666667 Non-Heap Memory Used (MB): 27.0
2	17.865	6.325	78.333	27.000	 ===== Performance Metrics ===== Execution Time (s): 17.864532181 Records Processed: 113 Throughput (records/s): 6.325382543192609 Heap Memory Used (MB): 78.33333333333333 Non-Heap Memory Used (MB): 27.0
3	19.176	5.893	78.667	27.000	 ===== Performance Metrics ===== Execution Time (s): 19.17627839 Records Processed: 113 Throughput (records/s): 5.892697096999122 Heap Memory Used (MB): 78.66666666666667 Non-Heap Memory Used (MB): 27.0
4	19.356	5.838	78.667	27.000	 ===== Performance Metrics ===== Execution Time (s): 19.355657585 Records Processed: 113 Throughput (records/s): 5.838086332317188 Heap Memory Used (MB): 78.66666666666667 Non-Heap Memory Used (MB): 27.0
5	20.409	5.537	78.667	27.000	 ===== Performance Metrics ===== Execution Time (s): 20.409351244 Records Processed: 113 Throughput (records/s): 5.536677704697746 Heap Memory Used (MB): 78.66666666666667 Non-Heap Memory Used (MB): 27.0
Average	**18.90**	**6.00**	**78.60**	**27.00**	**-**

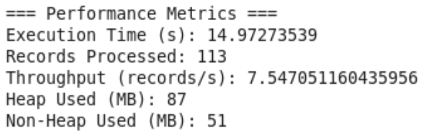
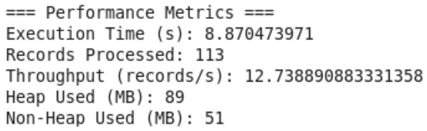
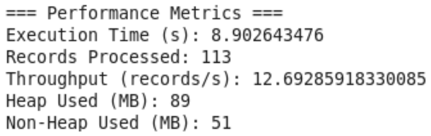
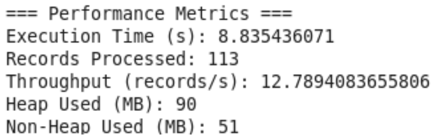
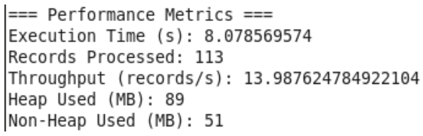
b) Spark

Output



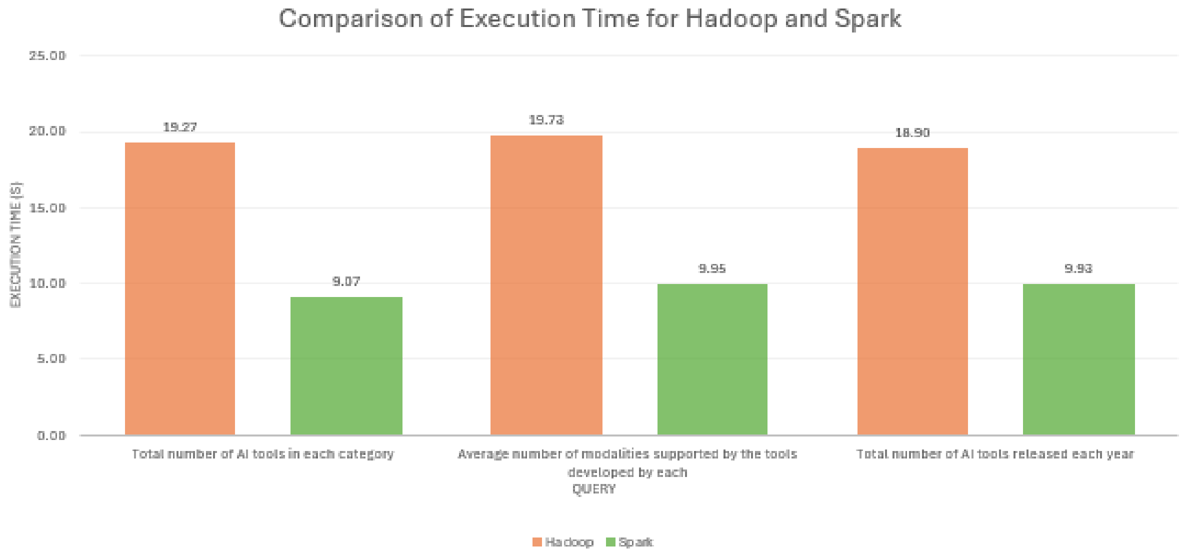
```
2017 : 2
2021 : 6
2024 : 19
2022 : 23
2025 : 4
2013 : 1
2014 : 1
2023 : 45
2019 : 5
2018 : 5
2020 : 2
```

Performance

Iteration	Execution Time(s)	Throughput (Records/s)	Heap Memory Used (MB)	Non-Heap Memory Used (MB)	Evidence
1	14.973	7.547	87.000	51.000	 === Performance Metrics === Execution Time (s): 14.97273539 Records Processed: 113 Throughput (records/s): 7.547051160435956 Heap Used (MB): 87 Non-Heap Used (MB): 51
2	8.870	12.739	89.000	51.000	 === Performance Metrics === Execution Time (s): 8.870473971 Records Processed: 113 Throughput (records/s): 12.738890883331358 Heap Used (MB): 89 Non-Heap Used (MB): 51
3	8.903	12.693	89.000	51.000	 === Performance Metrics === Execution Time (s): 8.902643476 Records Processed: 113 Throughput (records/s): 12.69285918330085 Heap Used (MB): 89 Non-Heap Used (MB): 51
4	8.835	12.789	90.000	51.000	 === Performance Metrics === Execution Time (s): 8.835436071 Records Processed: 113 Throughput (records/s): 12.7894083655806 Heap Used (MB): 90 Non-Heap Used (MB): 51
5	8.076	13.988	89.000	51.000	 === Performance Metrics === Execution Time (s): 8.078569574 Records Processed: 113 Throughput (records/s): 13.987624784922104 Heap Used (MB): 89 Non-Heap Used (MB): 51
Average	**9.93**	**11.95**	**88.80**	**51.00**	**-**

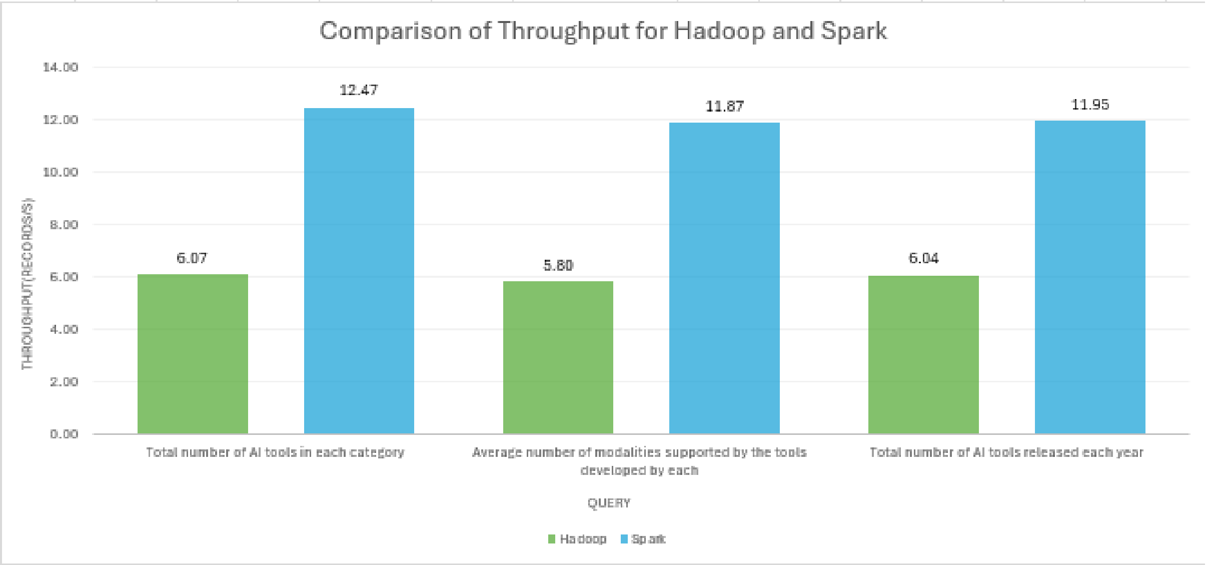
3.4 Comparative Performance Analysis

ExecutionTime



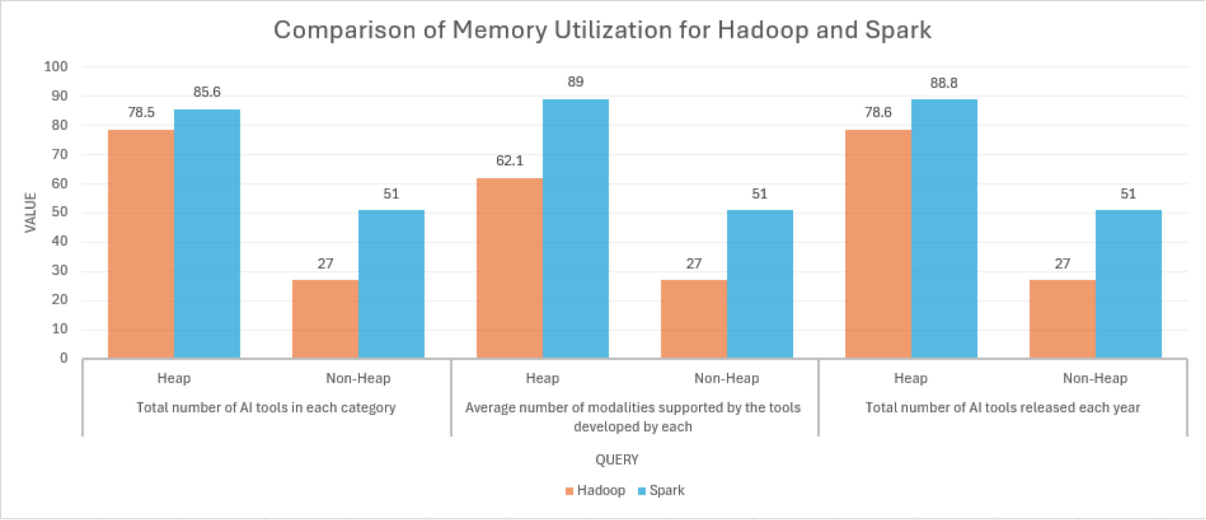
The chart shows that Spark consistently outperforms Hadoop for all three queries. Hadoop takes around 19 seconds on average to run each query, while Spark completes them in about 9–10 seconds. This means Spark reduces execution time by almost 50%, showing that it is more efficient and better optimized for processing large datasets. Overall, Spark provides quicker responses than Hadoop for every query tested.

Throughput



Based on the throughput results, Spark can process more records per second than Hadoop across all three queries. Hadoop achieves approximately 5–6 records per second, while Spark achieves 11–12 records per second, which is more than double the rate. This indicates that Spark handles data more quickly and efficiently, especially when dealing with higher processing loads. Generally, Spark offers significantly higher throughput than Hadoop in all scenarios.

Memory Utilization



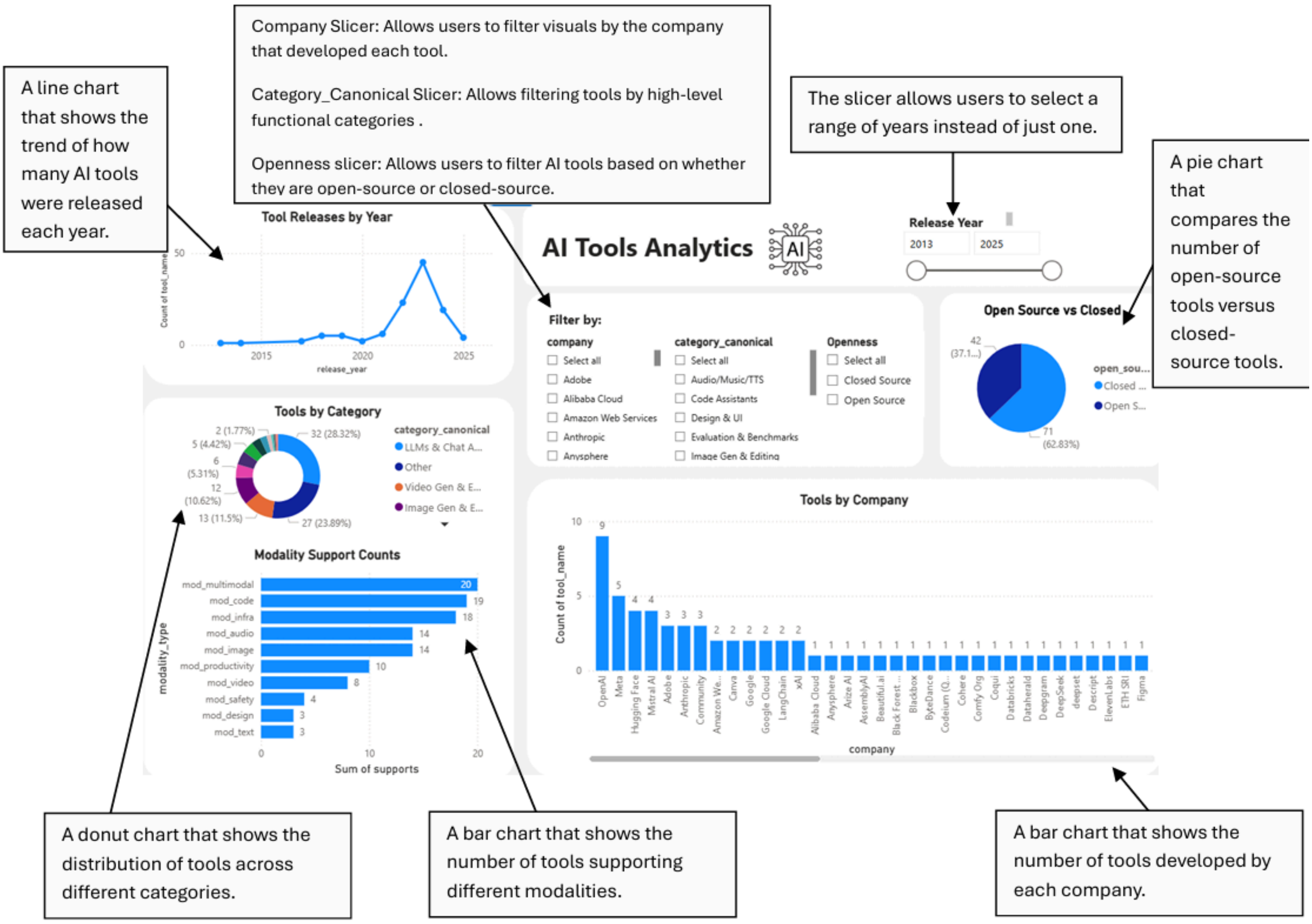
The memory comparison shows that Spark uses more memory than Hadoop for both heap and non-heap spaces. For every query, Spark's heap memory is around 85–89 MB, while Hadoop's heap memory is slightly lower at 62–79 MB. Similarly, Spark's non-heap memory stays at 51 MB, which is higher than Hadoop's 27 MB. This suggests that Spark requires more memory resources to achieve its faster performance. Even though Spark is more memory-intensive, it delivers better speed and throughput than Hadoop.

3.5 Summary

Both Hadoop and Spark differ significantly in execution time, throughput, and memory usage. Hadoop MapReduce, the core processing engine in Hadoop, exhibits longer execution times and lower throughput due to disk I/O overhead. In contrast, Apache Spark consistently outperforms Hadoop by completing queries in roughly half the time, because of its in-memory processing and RDD abstraction. Spark can cache intermediate data in memory, reducing the need for repetitive read and write operations to disk, which enables it to handle larger volumes of records per second. However, this performance advantage comes at the cost of higher memory usage, with both heap and non-heap memory consumption being greater than Hadoop's. Overall, Spark delivers faster and more efficient data processing, making it more suitable for scenarios requiring quick responses, while Hadoop remains better suited for traditional batch processing.

4.0 Dashboard

Dashboard Link :
https://drive.google.com/file/d/15FDg_ui3UdVrpRQ-vkOi7XvBFdDEdBef/view?usp=sharing



A line chart that shows the trend of how many AI tools were released each year.

Company Slicer: Allows users to filter visuals by the company that developed each tool.

Category_Canonical Slicer: Allows filtering tools by high-level functional categories .

Openness slicer: Allows users to filter AI tools based on whether they are open-source or closed-source.

The slicer allows users to select a range of years instead of just one.

A pie chart that compares the number of open-source tools versus closed-source tools.

A donut chart that shows the distribution of tools across different categories.

A bar chart that shows the number of tools supporting different modalities.

A bar chart that shows the number of tools developed by each company.

4.1 Introduction of dashboard

This dashboard was developed using Power BI to analyze a dataset of AI tools across different companies, categories, release years, modalities, and openness (open-source vs closed-source licensing). The objective is to provide an interactive environment where users can explore key parameters and discover meaningful trends within the AI tool ecosystem. The dashboard integrates five visualizations and four interactive slicers, enabling dynamic filtering and deeper insights.

4.2. Slicers

4.2.1 Release Year

The Release Year slicer uses a “Between” slider style, allowing users to select a range of years (2015 to 2023). This makes it easy to explore how tool development trends have changed over time. All charts automatically update based on the selected period.

4.2.2 Company

The Company slicer enables viewers to focus on tools developed by specific organizations.
Users can select one or all companies to compare development strategies, openness preferences, or modality patterns.

4.2.3 Category

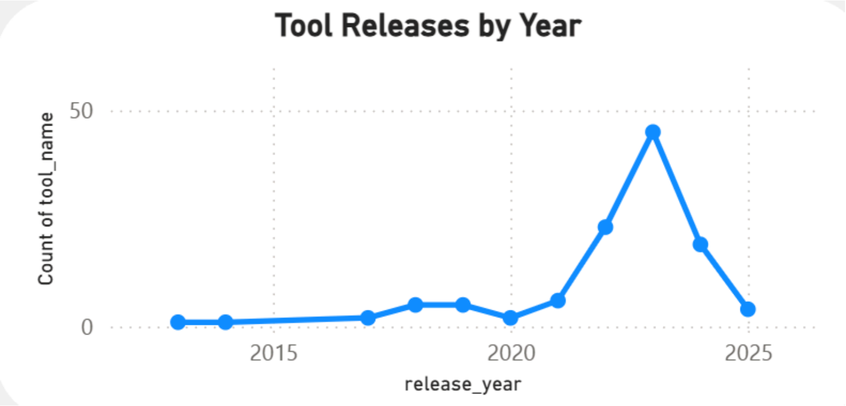
This slicer filters tools by their functional category.This allows users to study which categories dominate in certain years or companies.

4.2.4 Openness

The Openness slicer categorizes tools by licensing type. Users may select Open Source, Closed Source, or both.This helps analyze how licensing choices vary by company, year, and category.

4.3 Visualizations Included in the Dashboard

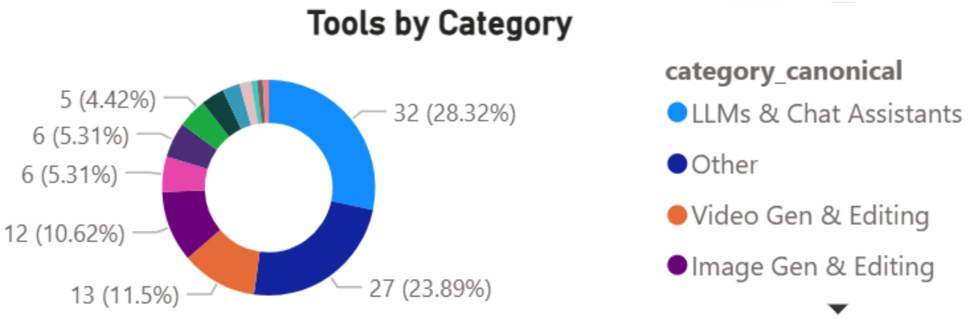
4.3.1 Visualization 1: Tool Releases by Year



This line chart illustrates the number of AI tools released each year within the selected time range.It helps identify trends such as growth periods, peak innovation years, or slowdowns in tool development. When combined with slicers, users can analyze release patterns for specific companies, open-source vs closed-source release trends and category-specific growth patterns.

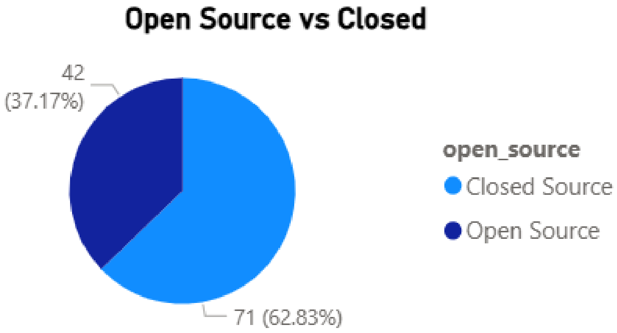
The number of AI tools released increased steadily from 2020 to 2022, followed by a major surge in 2023, which appears to be the peak of AI tool development in the dataset. After 2023, there is a noticeable decline, with releases dropping to 19 in 2024 and further to 4 in 2025. This may indicate market saturation, stricter regulations, strategic consolidation by major companies, or incomplete data for 2025.

4.3.2 Visualization 2: Tools by Category



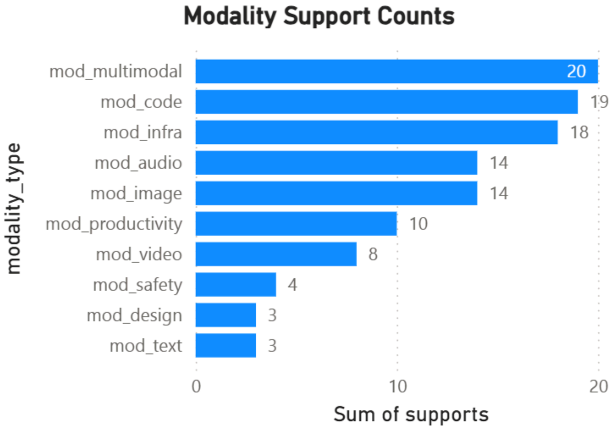
This donut chart shows the distribution of tools across different categories.It provides a quick overview of which categories dominate the ecosystem.It shows that LLMs & Chat Assistants dominate the dataset (28.32%), highlighting the strong industry focus on conversational AI. This is followed by Other AI categories (23.89%), reflecting broad innovation across diverse tool types. Video Generation & Editing tools (11.5%) also make up a large share, showing rapid growth in AI-powered content creation, while Image Generation & Editing (10.62%) remains important but smaller in comparison. Overall, the distribution suggests that text-based and multimedia generative tools are currently leading the AI landscape.

4.3.3 Visualization 3: Open Source vs Closed Source



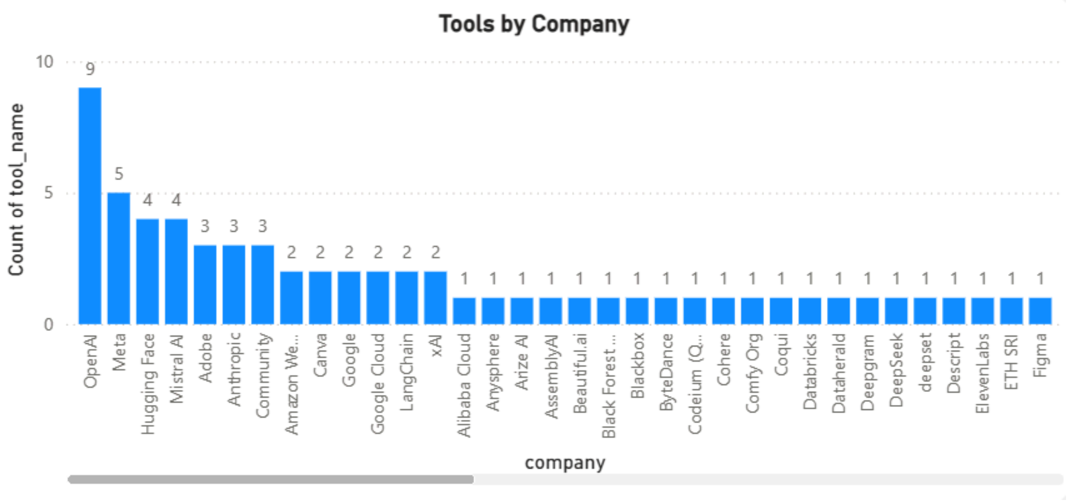
This visualization compares the number of open-source tools against closed-source tools. It helps identify whether companies or categories tend to prefer open-source or closed-source models. The pie chart shows that the majority of AI tools in the dataset are closed source, making up 62.83%, while only 37.17% are open source. This indicates that most companies still prefer to release AI tools under proprietary models rather than openly sharing their code. With the Openness slicer, users can filter and explore how openness varies across different companies, categories, and release years.

4.3.4 Visualization 4: Modality Support Counts



This horizontal bar chart shows how many tools support each modality based on the dataset's modality columns, where each column contains a value of 1 if a tool supports that modality and 0 if it does not. By summing these values across all tools, Power BI calculates the total number of tools that support each modality. The chart reveals that multimodal support is the most common (20 tools), followed closely by code (19 tools) and infrastructure-related features (18 tools). Meanwhile, more specialized modalities such as text-only capabilities (3 tools) appear far less frequently. This pattern suggests that modern AI tools increasingly focus on broad, multi-capability functionality rather than single-modality designs.

4.3.5 Visualization 5: Tools by Company



This bar chart ranks companies by the number of AI tools they have developed. According to the dataset, OpenAI leads with 9 tools, followed by Meta (5 tools), HuggingFace (4 tools), and Mistral AI (4 tools). The chart highlights the major contributors to the AI ecosystem and allows users, through the slicers, to explore category-specific contributions, open-source vs closed-source preferences, and yearly trends in company output. It provides a quick view of which companies dominate tool development and innovation within the AI industry.

4.4 Conclusion

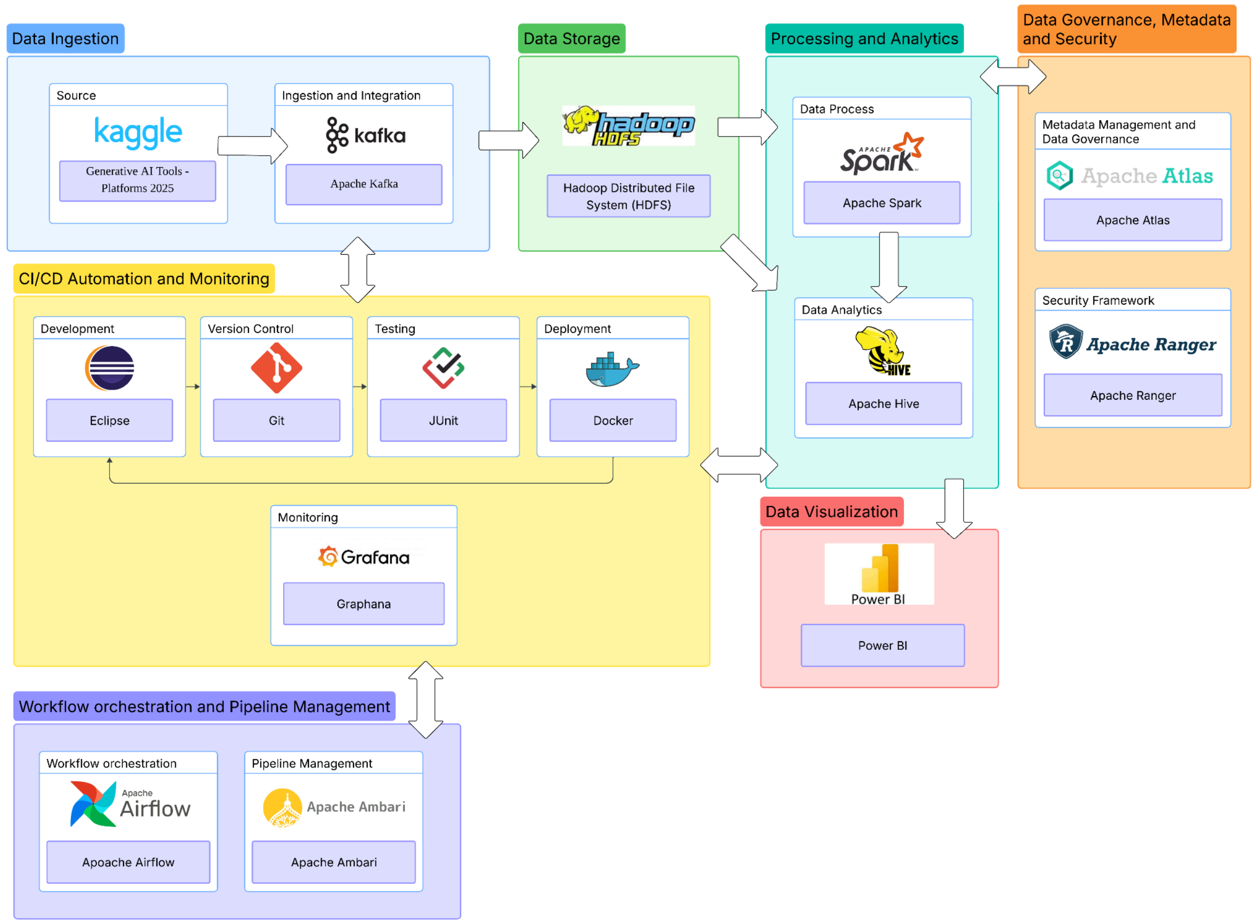
In a nutshell, the dashboard shows that the AI landscape is mainly driven by a few major companies such as OpenAI, Meta, HuggingFace, and Mistral AI, which contribute many of the tools in the dataset. There was a clear rise in new AI tool releases from 2020 to 2023, showing rapid growth in the industry, followed by a drop in 2024 and 2025 that may be due to market slowdown, changing priorities, or limited data. The category breakdown highlights that LLMs and chat assistants are the most popular, while the modality results show that many tools now support multiple capabilities, especially multimodal and code-related functions. The comparison between open-source and closed-source tools shows that most companies still prefer closed-source models. Overall, the dashboard helps us understand how different companies, categories, and design choices shape current AI development, giving a clear picture of the trends and direction of the industry.

5.0 DataOps Architecture

DataOps is a modern approach to managing data and analytics that emphasizes close collaboration between people, processes, and technology to deliver reliable data products efficiently. It supports the rapid creation and deployment of data-driven applications, enabling organizations to extract greater business value from their data. The DataOps framework is built upon three principles: Agile practices, DevOps methodologies, and lean process thinking. Agile practices enable analytics teams to work iteratively, delivering insights in short cycles while quickly responding to evolving business needs. DevOps introduces automation and continuous integration and deployment techniques to streamline the development, testing, and release of data pipelines and analytics solutions. Lean process thinking focuses on minimizing waste, improving efficiency, and maintaining consistent quality across the data lifecycle through continuous monitoring and measurement. By combining these principles, DataOps helps organizations improve operational efficiency, strengthen data quality and reliability, enhance governance and security, and significantly shorten the time required to deliver data science and analytics solutions.

5.1 Architecture Diagram

The diagram below shows a complete view of our DataOps Architecture.



5.2 Layers and Components Explanation

1. **Data Ingestion**

   This layer is responsible for acquiring and importing data into the analytics platform. It ensures that raw datasets are consistently ingested and traceable for downstream processing.

   - **Source**: Kaggle datasets or other sources related to Artificial Intelligence tools are used as the primary data source.
   - **Ingestion and integration**: Apache Kafka can be used to automate data flow, handle data routing, and ensure reliable data pipelines from source into the storage system.

2. **Data Storage**

   This layer provides scalable and fault-tolerant storage for raw and processed data. HDFS enables distributed storage and supports large-scale data processing experiments.

   - **Hadoop Distributed File System (HDFS)** is used to store raw Kaggle datasets in their original format.

3. **Processing and Analytics**

   This layer focuses on transforming raw data and processing performance to derive insights and support decision-making. Both engines operate on the same dataset, but for different purposes.

   - **Data Process:** Apache Spark is used in processing big data because it utilizes in-memory processing that speeds up overall processes.
   - **Data Analytics:** Apache Hive is used to define schemas on top of processed datasets stored in HDFS. And also uses SQL-based queries to execute aggregate, filter and prepare analytical datasets.

4. **Data Visualization**

   This layer focuses on presenting insights to end users.

   - **Power BI** is used to create dashboards and reports.
   - Interactive dashboards enable stakeholders to explore results dynamically.

5. **Data Governance, Metadata, and Security**

   This layer ensures data quality, managed metadata, and controlled access. It improves transparency, trust, and compliance.

   - **Metadata Management and Data Governance**: Apache Atlas facilitates metadata management and data governance, allowing easy cataloging of assets and trace data lineage effectively.

- **Security Framework**: Apache Ranger offers a centralized management of data security, and auditing across many different platforms.

6. **CI/CD, Automation, and Monitoring**

   This layer enforces DataOps best practices. Automation reduces errors and accelerates development cycles.

   - **Development**: IDEs like Eclipse allow application development in java and other languages.
   - **Version Control**: Git is used to manage processing scripts, configuration files, and workflow definitions.
   - **Testing**: JUnit tests small pieces of code, verifying individual components before execution.
   - **Deployment**: Docker containers ensure consistent execution environments.
   - **Monitoring**: Grafana monitors system performance, job execution time, and resource usage.

7. **Workflow Orchestration and Pipeline Management**

   This layer automates and manages data workflows. Task dependencies and scheduling improve reliability and repeatability.

   - **Apache Airflow** is used to orchestrate: Data ingestion from Kaggle, data preparation for visualization.
   - **Apache Ambari** is used to manage Hadoop clusters and pipelines.

6.0 References

Databricks. (2021, December 8). What is MapReduce? https://www.databricks.com/glossary/mapreduce#:~:text=MapReduce%20is%20a%20Ja va%2Dbased

DataKitchen. (2020, October 20). What is DataOps? https://datakitchen.io/what-is-dataops/

Gaur, C. (2022, November 2). RDD in Apache Spark Advantages and its Features. Xenonstack. https://www.xenonstack.com/blog/rdd-in-spark/#:~:text=Resilient%20Distributed%20Data set%20(RDD)%20is

Gaur, C. (2023, November 24). DataOps- Principles, Tools and Best Practices. Xenonstack. https://www.xenonstack.com/insights/data-operations

Guller, M. (2015). Big data analytics with spark. ISBN-13 (pbk), 978-1.

IBM. (2023). What is Apache MapReduce? https://www.ibm.com/topics/mapreduce

Nicolas. (2015, October 31). Mini-Cluster Part IV : Word Count Benchmark. Nico's Blog. https://blog.ditullio.fr/2015/10/31/mini-cluster-part-iv-word-count-benchmark/

Nicolas. (2015, December 24). Hadoop Basics- Filter, Aggregate and Sort with MapReduce. Nico's Blog. https://blog.ditullio.fr/2015/12/24/hadoop-basics-filter-aggregate-sort-mapreduce/

O'Reilly. (2023). 4. Working with Key/Value Pairs- Learning Spark [Book]. https://www.oreilly.com/library/view/learning-spark/9781449359034/ch04.html#example4-1

Tutorialspoint. (2023). Apache Spark- RDD. https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm

Tutorialspoint. (2023). Hadoop- MapReduce. https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm

Verma, A., Mansuri, A. H., & Jain, N. (2016, March). Big data management processing with Hadoop MapReduce and spark technology: A comparison. In 2016 symposium on colossal data analysis and networking (CDAN) (pp. 1-4). IEEE.

X-TechStacks. (2023, March 8). DataOps & MLOps Tech Stack. Linkedin. https://www.linkedin.com/pulse/dataops-mlops-tech-stack-g-labs-innovation/

Zhasa, M. (2023, January 30). What is Hadoop? Components of Hadoop and Its Uses. Simplilearn. https://www.simplilearn.com/tutorials/hadoop-tutorial/what-is-hadoop