Data Processing in Databricks

Leveraging Pandas, PySpark, and SQL

Marcelino Mayorga Quesada

databricks

Marcelino Mayorga Quesada

- 14 Years of Experience in Software in finance, marketing and video games sectors and 3 Years focused in AI
- Experienced in technical and management roles for delivery
- Technical Instructor on GCP and .NET courses
- Fun Fact: Training RL agents in video games

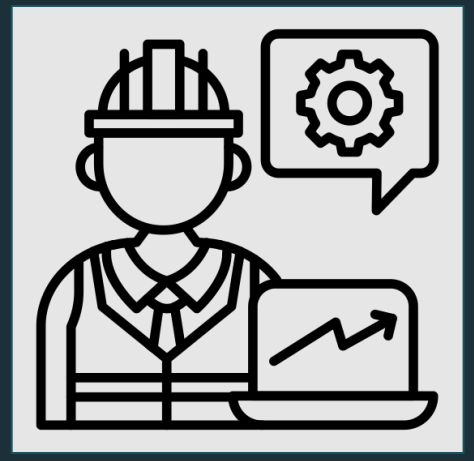
Agenda

1. Introductions
   - Instructor, Topic, Audience
2. Data Processing
   - Concept, Operations
3. Databricks
   - Solution, Pyspark, Pandas, SQL
4. Demo
   - Community Version
   - Use case on NLP
5. Key Takeaways
6. Q & A

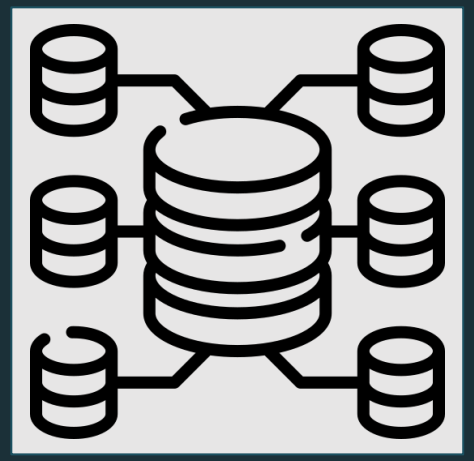
Data Processing

Concept

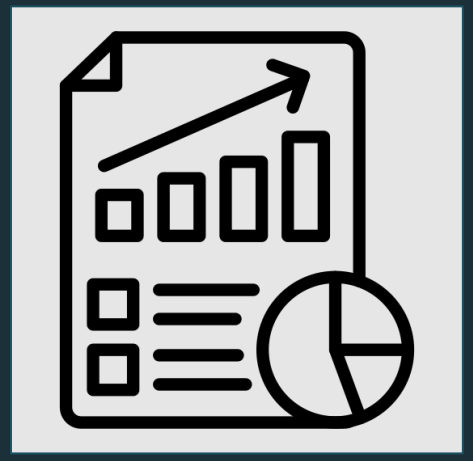
A series of operations to convert raw data into **meaningful information**



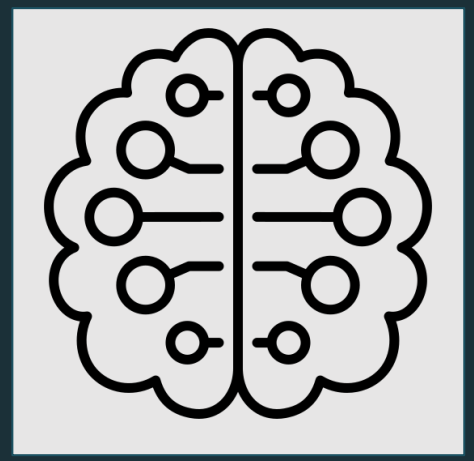
Data Engineers



Storage



Analytics

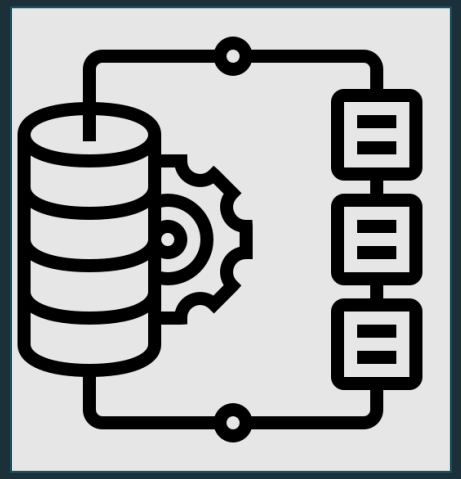


DL & ML Models

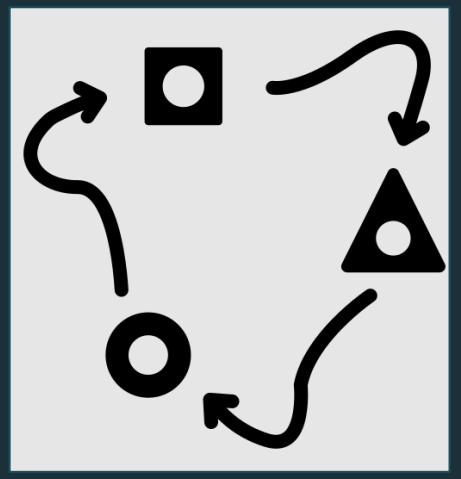
Data Processing

Operations

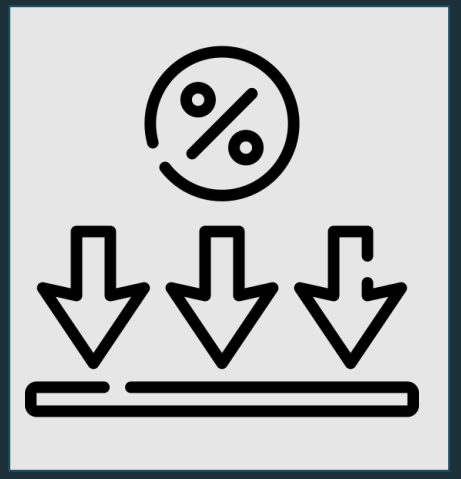
Cleaning



Integration



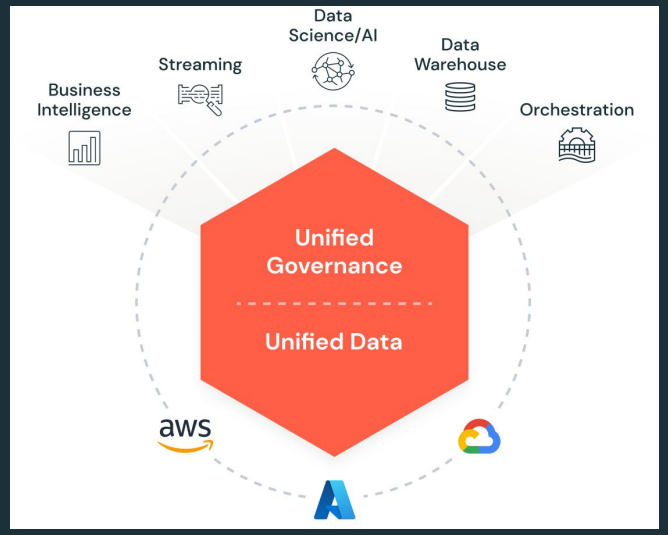
Transformation



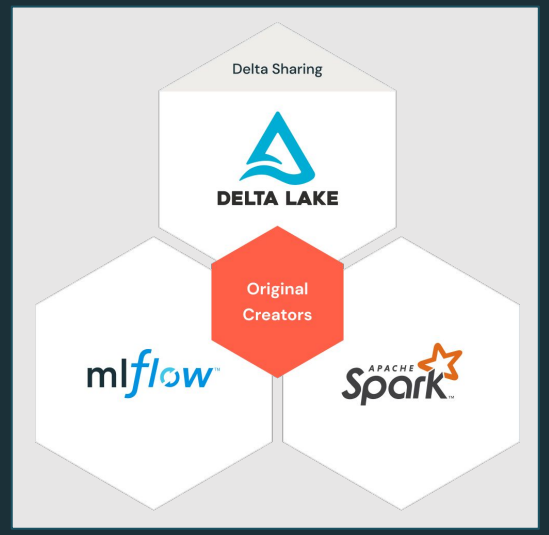
Reduction

Databricks

Data Lakehouse Architecture



Unified



Open

Scalable

Pandas

Data Analysis and Manipulation Library

- Data structure, cleaning, transformation and analysis
  - Pandas Dataframes
- Prototyping on low-volume data for single-node computing
- Available on:
  - Stand alone library
  - Pandas API (Pandas-on-Spark)

PySpark

Python API for Apache Spark

- Unified analytics engine for large-scale data processing
  - High volume data (TB, PB)
- Distributed computing (Clusters)
  - Parallel processing
  - Lazy Evaluation
  - Fault Tolerance
- Used via Spark Session & Context and Spark Dataframes

SQL

Structure Query Language

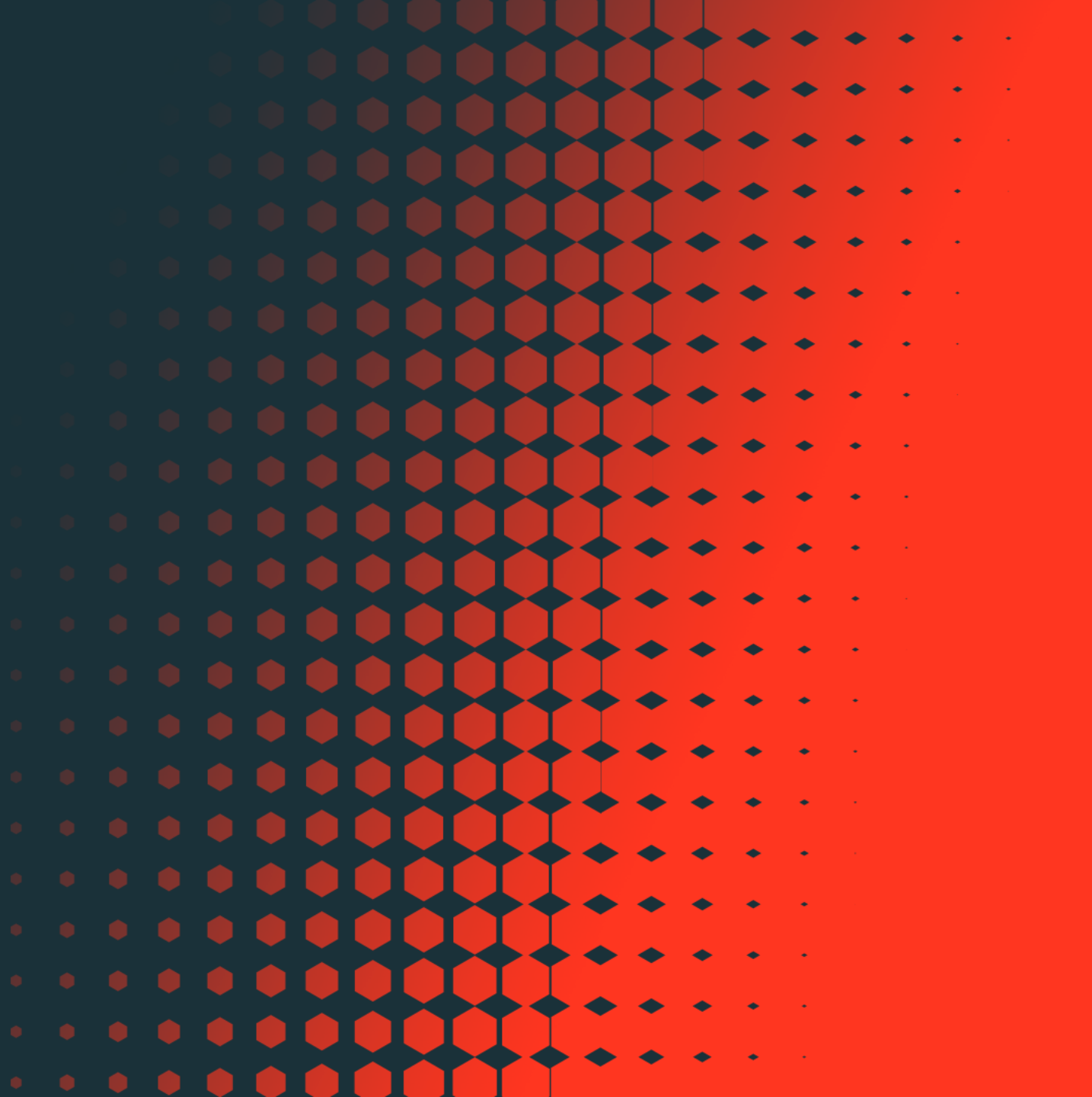
- Managing and manipulating relational databases
- Queries and Transactions over multiple tables
- Available on:
  - Stand alone (mysql, MS SQL)
  - SQL in Spark via SQLContext

Databricks

Summary

Name	Type	Purpose	Usage via	Ideal
Pandas	Data Analysis and Manipulation Library	• Data Handling and Transformation for Single-Node computing • Eager Execution	• Standalone Library • Pandas API on Spark • Dataframes	• Low Volume Data • Prototyping
PySpark	Python API for a Unified analytics Engine for large-scale data processing	• Distributed Computing • Parallel Processing • Lazy Evaluation • Fault Tolerance	• Pyspark's Context & Session • Dataframes	• High Volume Data • Scalability and Performance • Integration with multiple sources
SQL	Structured Query Language	• Querying • Transformations • Transactions • Storage	• SQLContext • Datasets	• Querying and Analysis • Managing relational databases

POP QUIZ!



databricks

Demo - IMDB's Movie Reviews

Databricks

Community Edition

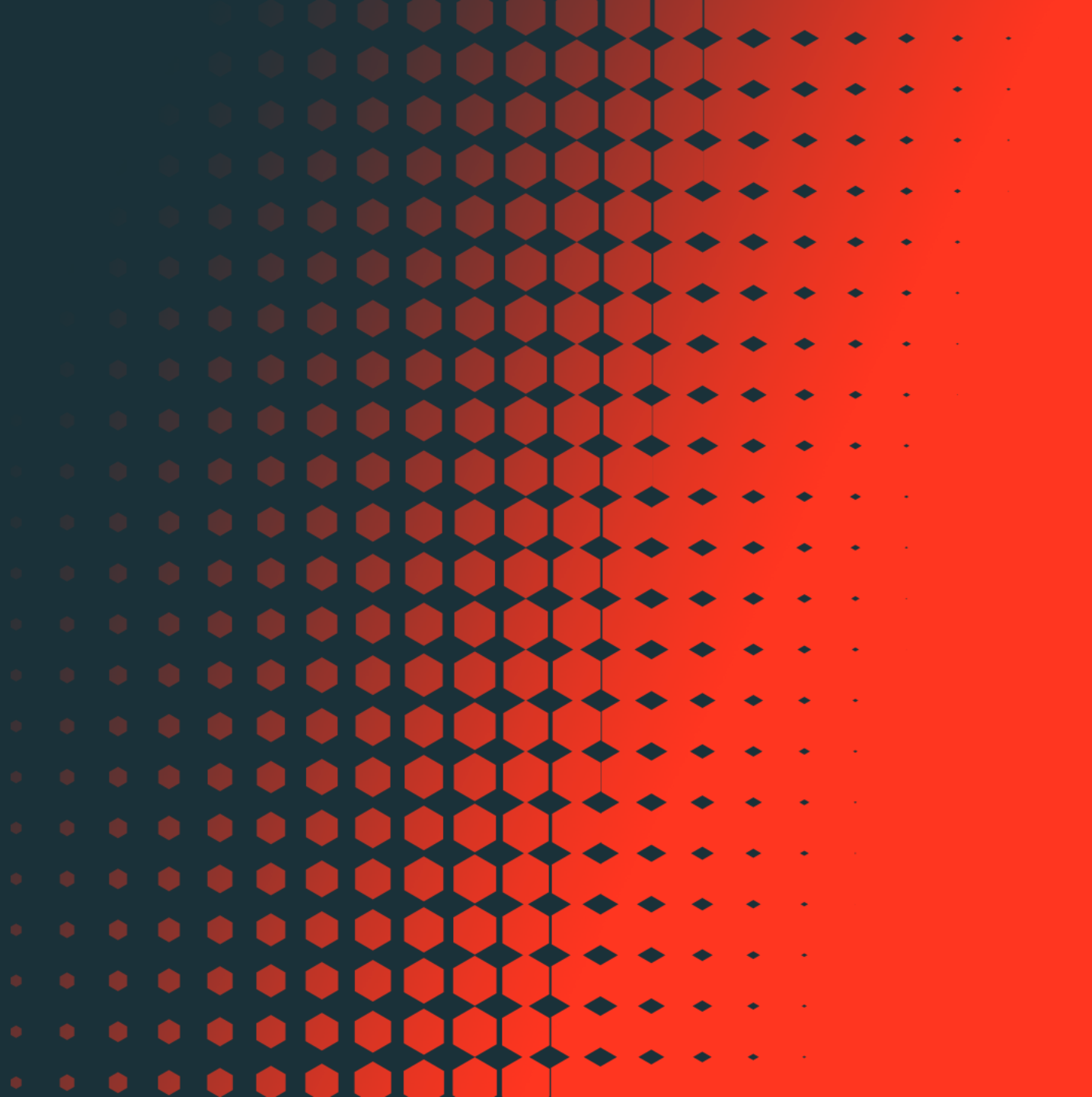
- Free access: https://community.cloud.databricks.com/
- Workspace - Notebooks
- Experiment Tracking
- Cluster Management
  - 1 Driver
  - 15.3 GB Memory, 2 Cores, 1 DBU
- Be aware of Idling resources

Demo - Data processing in Databricks

Key Takeaways

- Data Processing is a **crucial step** in Data Engineering
- Databricks Data Intelligence Platform is **powerful at scale**
- Pandas is an **easy to use data handling library** aimed for single-node computing
- Pyspark **enhances** Pandas and SQL to distributed-computing
- All of these tools have a **specific purpose and are flexible** hence the confusion when to use them

Q&A



databricks

databricks