

Data Processing in Databricks

Leveraging Pandas, PySpark, and SQL

Marcelino Mayorga Quesada



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 Agents in Video games.

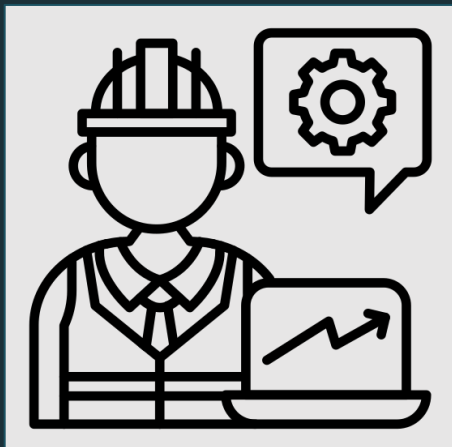
Agenda

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

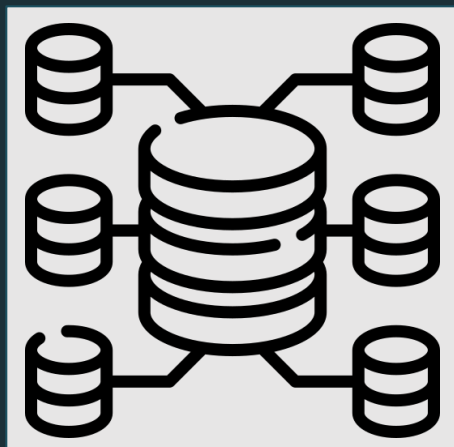
Data Processing

Concept

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



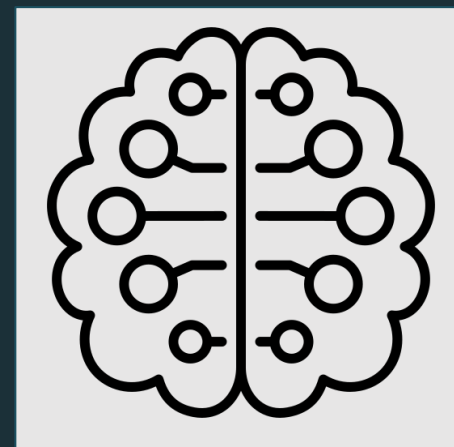
Data Engineers



Storage



Analytics



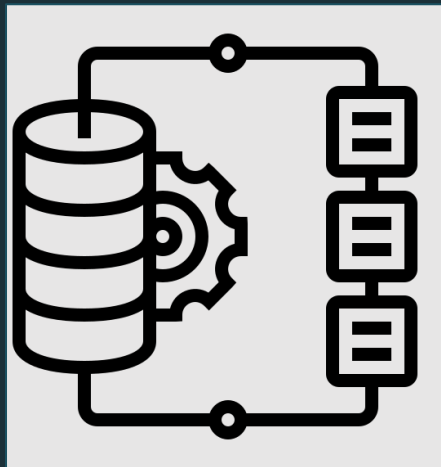
DL & ML Models

Data Processing

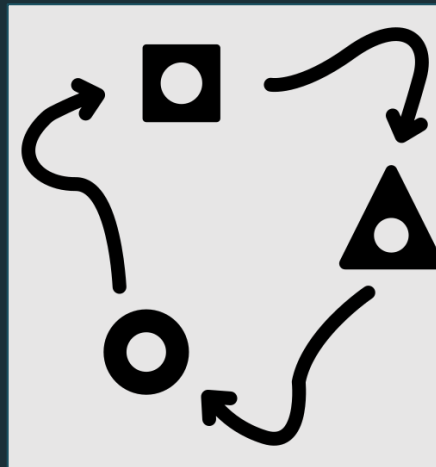
Operations



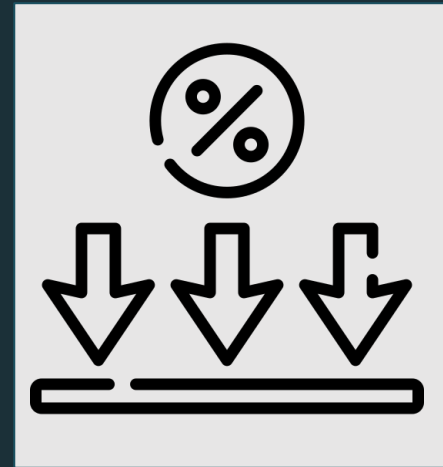
Cleaning



Integration



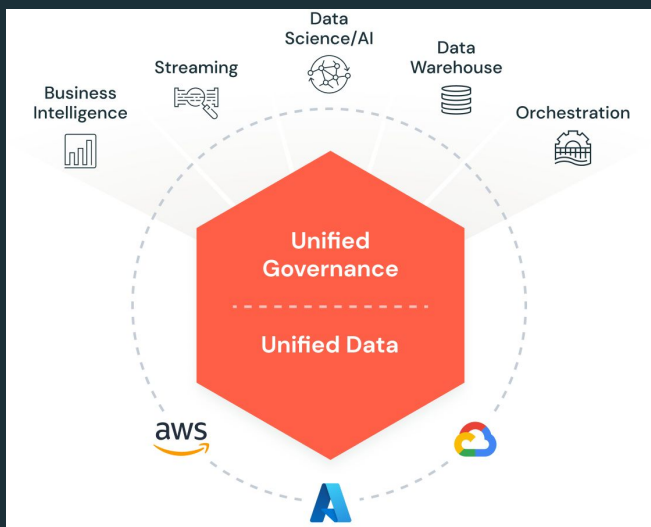
Transformation



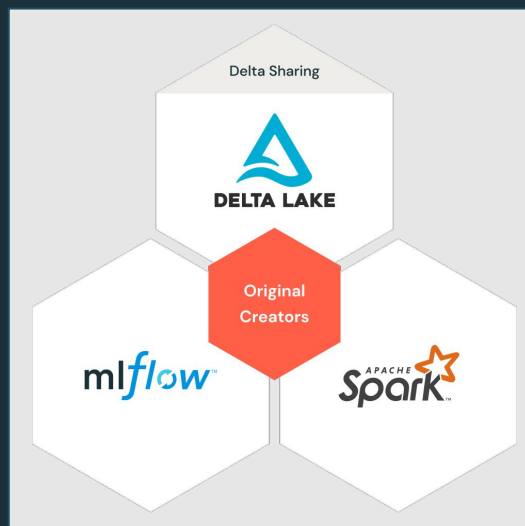
Reduction

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:
 - 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:
 - Stand alone RDBMS(mysql)
 - SQL in Spark via **SQLContext**



Databricks

Summary

Name	Type	Purpose	Usage via	Ideal
Pandas	Data Analysis and Manipulation Library	<ul style="list-style-type: none">• Data Handling and Transformation for Single-Node computing• Eager Execution	<ul style="list-style-type: none">• Standalone Library• Pandas API on Spark• Dataframes	<ul style="list-style-type: none">• Low Volume Data• Prototyping
PySpark	Python API for a Unified analytics Engine for large-scale data processing	<ul style="list-style-type: none">• Distributed Computing• Parallel Processing• Lazy Evaluation• Fault Tolerance	<ul style="list-style-type: none">• Pyspark• Context & Session• Dataframes	<ul style="list-style-type: none">• Big Data• Scalability• Performance• Integration with multiple sources
SQL	Structured Query Language	<ul style="list-style-type: none">• Querying• Transformations• Transactions• Storage	<ul style="list-style-type: none">• SQLContext• Datasets	<ul style="list-style-type: none">• Querying• Managing relational databases

POP QUIZ!

QR

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

Demo - Data processing in Databricks

Key Takeaways

- Data Processing is a crucial step in Data Engineering
- Databricks Data Intelligence Platform is powerful
- Pandas is easy to use data handling library aimed for single-node computing.
- Pyspark enhances Pandas and SQL to distributed-computing
- Understanding and leveraging these tools for data processing

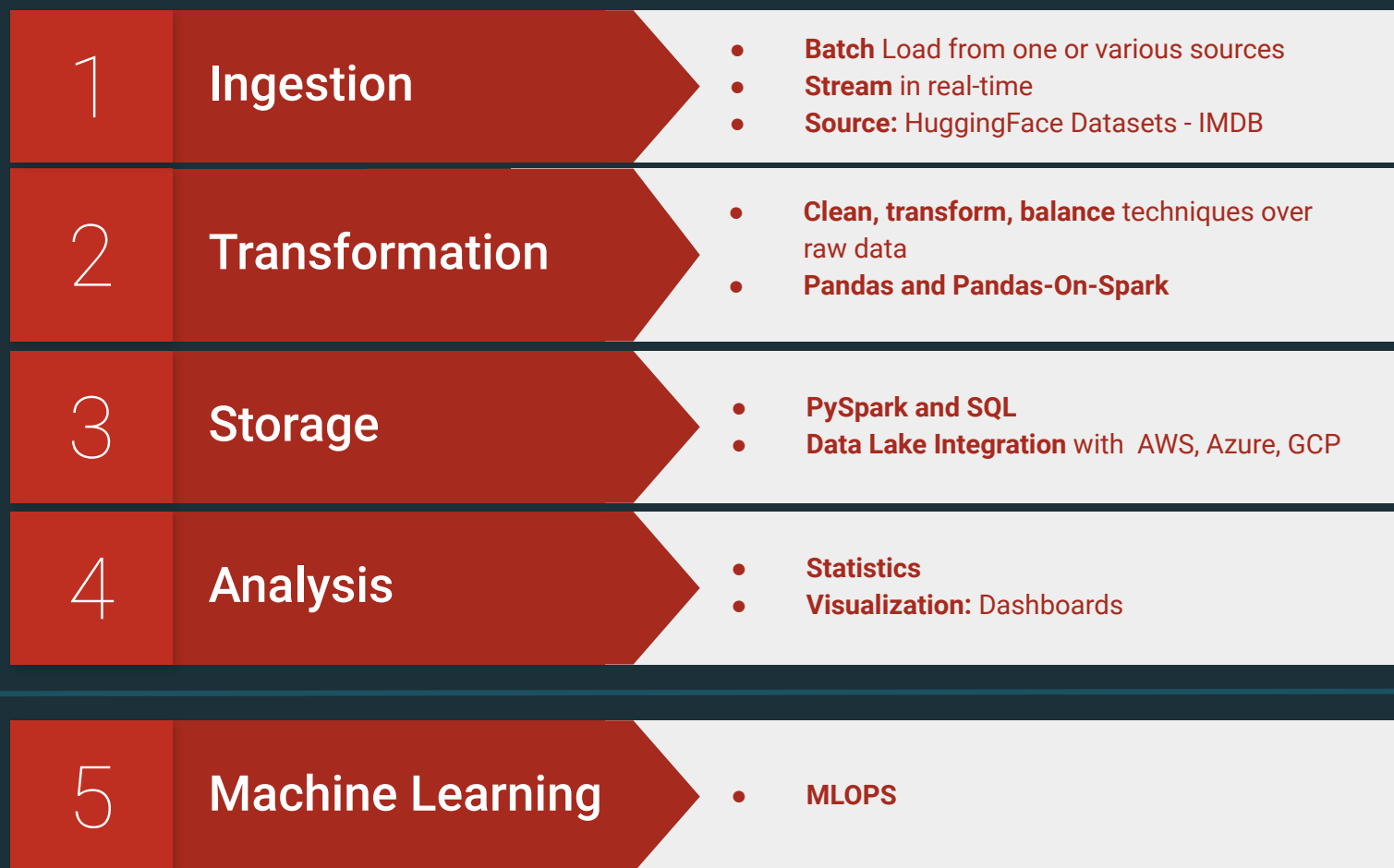
Q&A



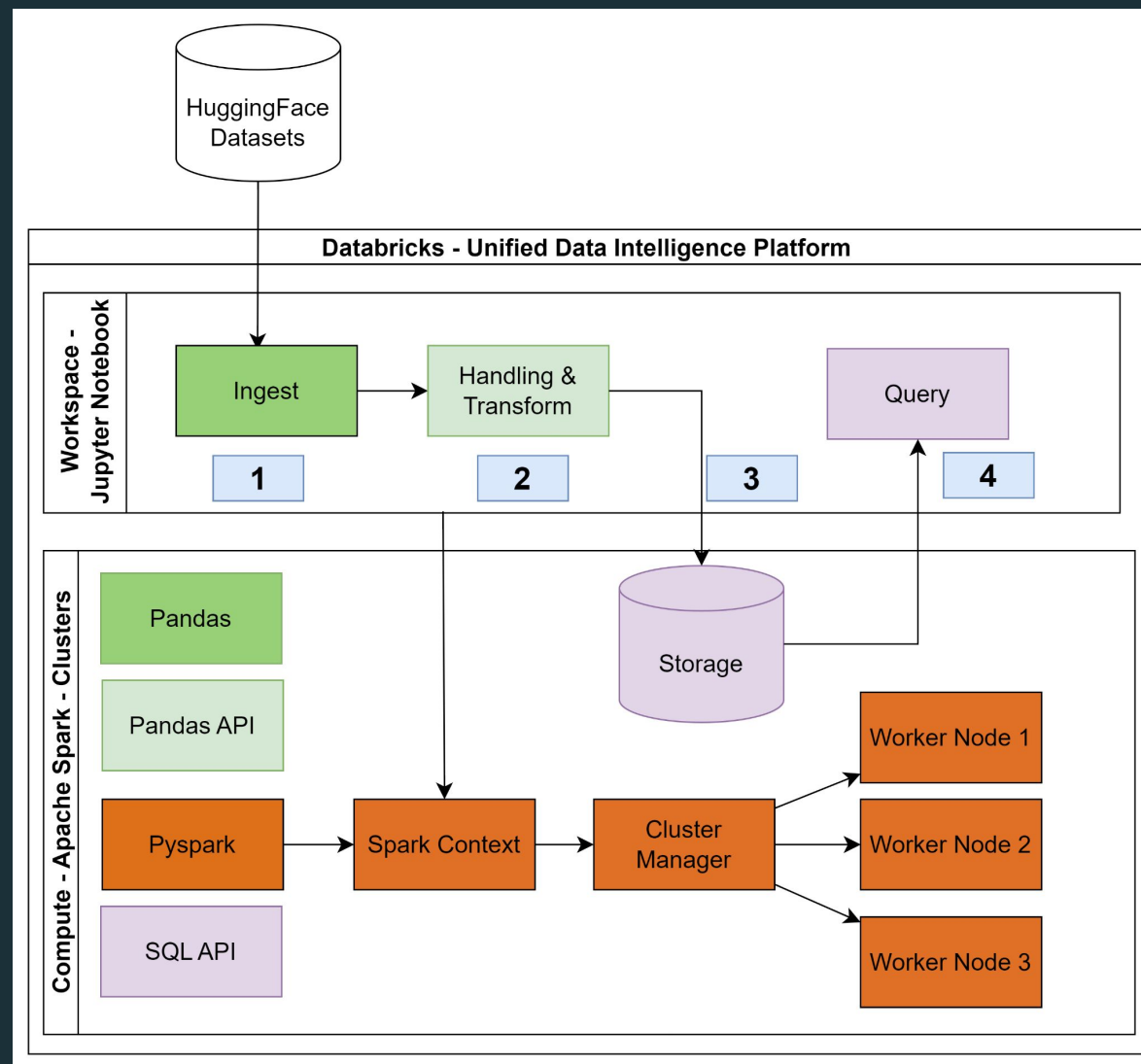
databricks

Data Workflow

Stages



Data Workflow



Data Processing

Operations

1. Cleaning

- Removing duplicates
- Impute or delete missing values
- Correct errors and inconsistencies

2. Integration

- ETL (Extract Transform Load)
- Merge and Join data warehousing
- Augmentation

3. Transformation

- Normalization and Standardization
- Aggregation (Summing, Averaging)
- Pivoting tables
- Encoding categorical values

4. Reduction

- Dimensionality Reduction: PCA, t-SNE
- Feature Selection & Extraction
- Sampling
- Compression