Data Processing in Databricks

Leveraging Pandas, PySpark, and SQL

Marcelino Mayorga Quesada

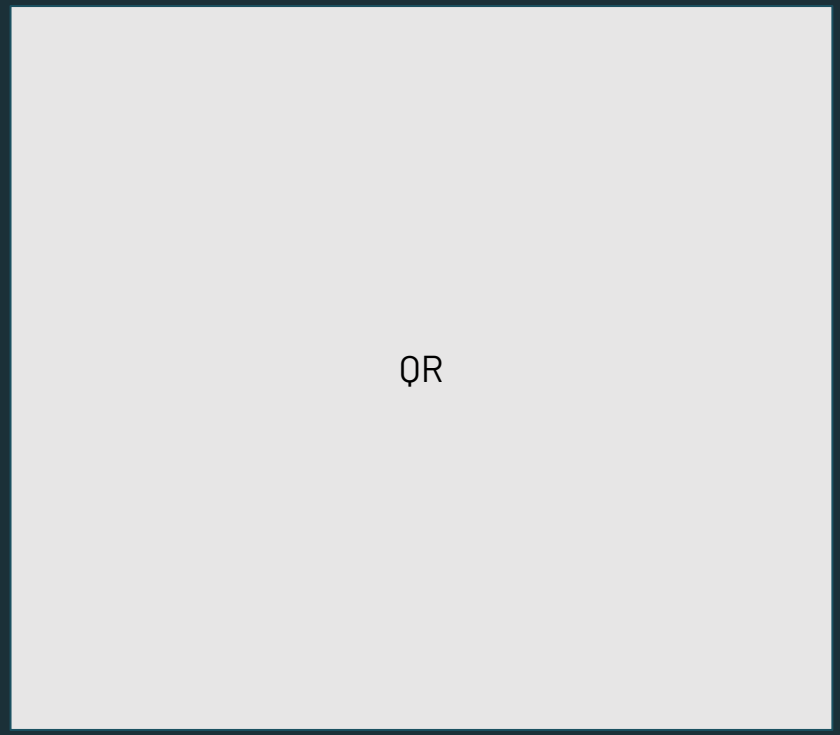
databricks

Marcelino Mayorga Quesada

- 14 Years of Experience in Software in finance, marketing and video games sectors, and over the last 3 years in AI.
- Experienced in technical and delivery management roles.
- Clients: Blackstone, Cambridge Associates, EA Sports, Citibank and BAC Credomatic.
- Technical Instructor on GCP and .NET courses.
- Passionate about Artificial Intelligence🤖, music 🎹, and video games 🎮.

Agenda

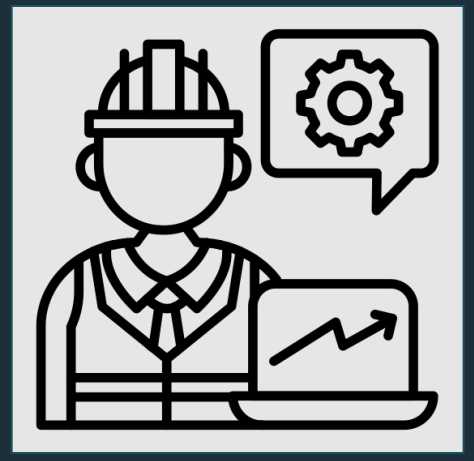
1. Introductions
   - Instructor, Topic, Audience
2. Data Processing
   - Concept, Operations
3. Databricks
   - Solution, Pyspark, Pandas, SQL
4. Demo
   - Community Version
   - Use case
5. Key Takeaways
6. Q & A



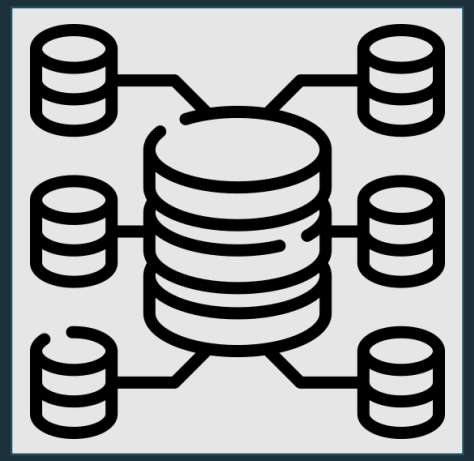
Data Processing

Concept

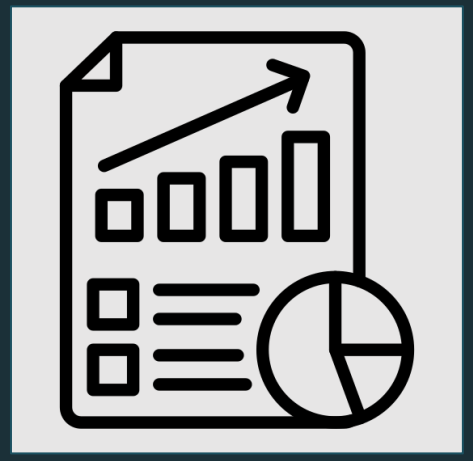
A series of operations to convert raw data into **meaningful information**.



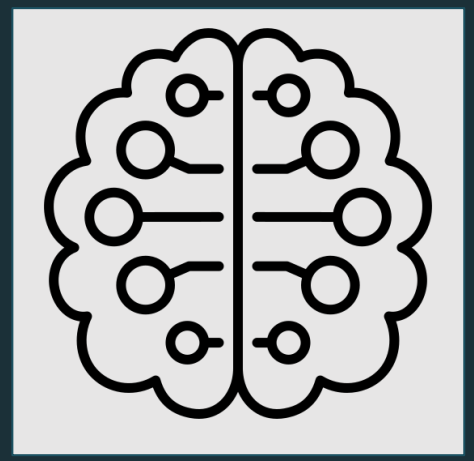
Data Engineers



Storage



Analytics



DL & ML Models

Data Processing

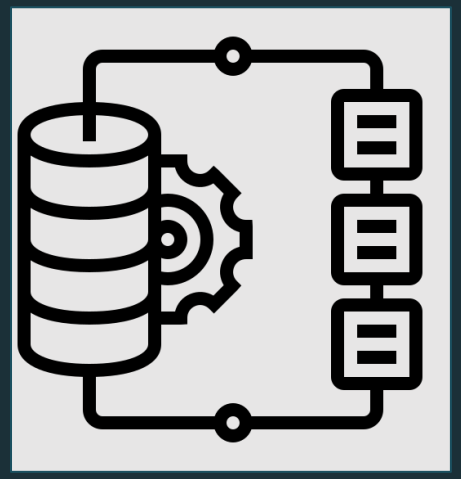
Operations

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

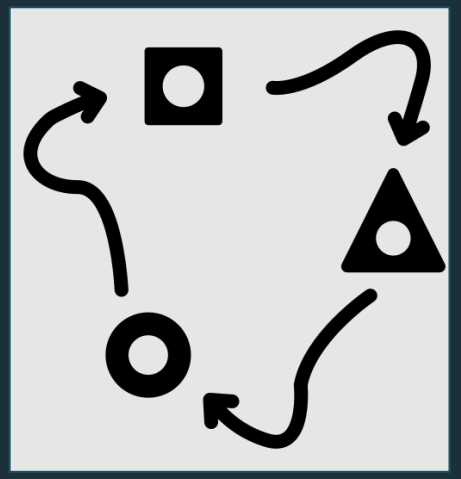
Data Processing

Operations

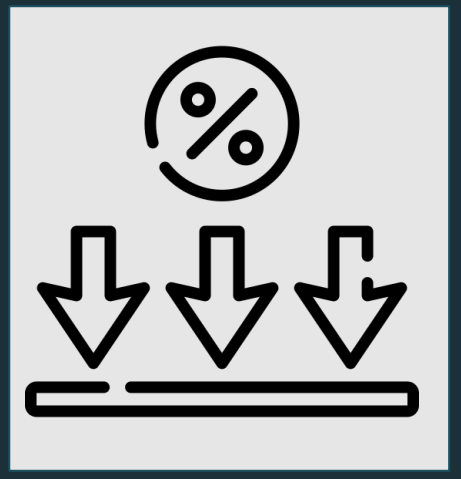
Cleaning



Integration



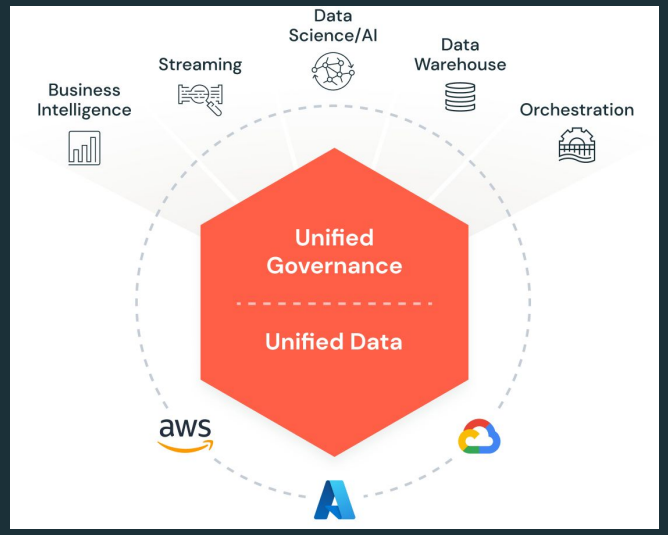
Transformation



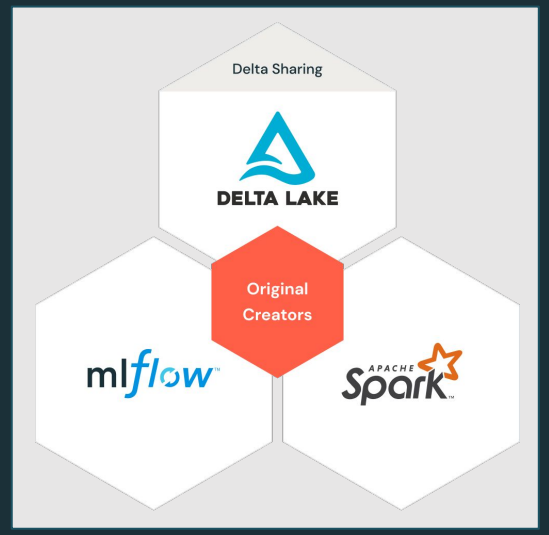
Reduction

Databricks

Data Lakehouse Architecture



Unified



Open

Scalable

Pandas

Data Analysis and Manipulation Library

- Data cleaning, transformation and visualization
- Low volume data for single computing
- Prototyping
- Available:
  - Stand alone library
  - Pandas API in Spark
  - Pandas Dataframes

PySpark

Big Data Processing Framework

- Python API for Apache Spark
- High volume data (TB, PB)
- Distributed computing (Clusters)
  - Parallel processing
  - Lazy Evaluation
  - Fault Tolerance
- Complex data transformations
- Used via Spark Session & Context and Spark Dataframes

SQL

Structure Query Language

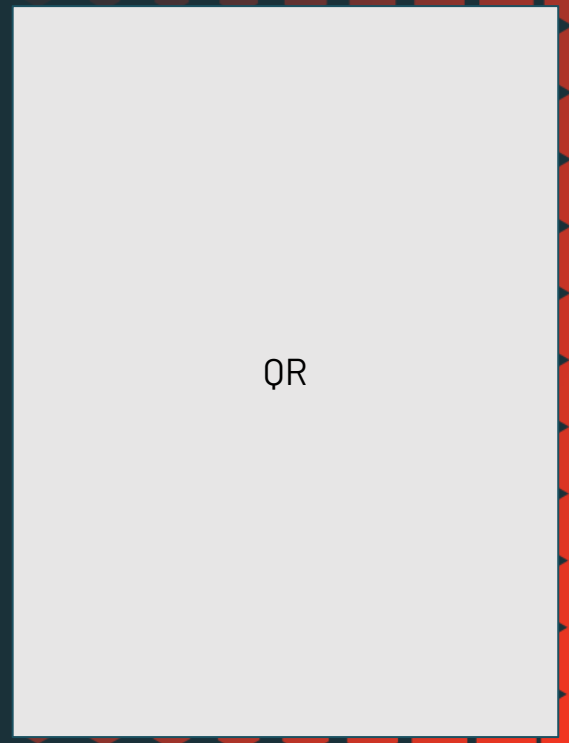
- Managing and manipulating relational databases
- Queries and Transactions over multiple tables.
- Available:
  - Stand alone DBMS (mysql)
  - SQL in Spark
  - SQLContext

Databricks

Summary

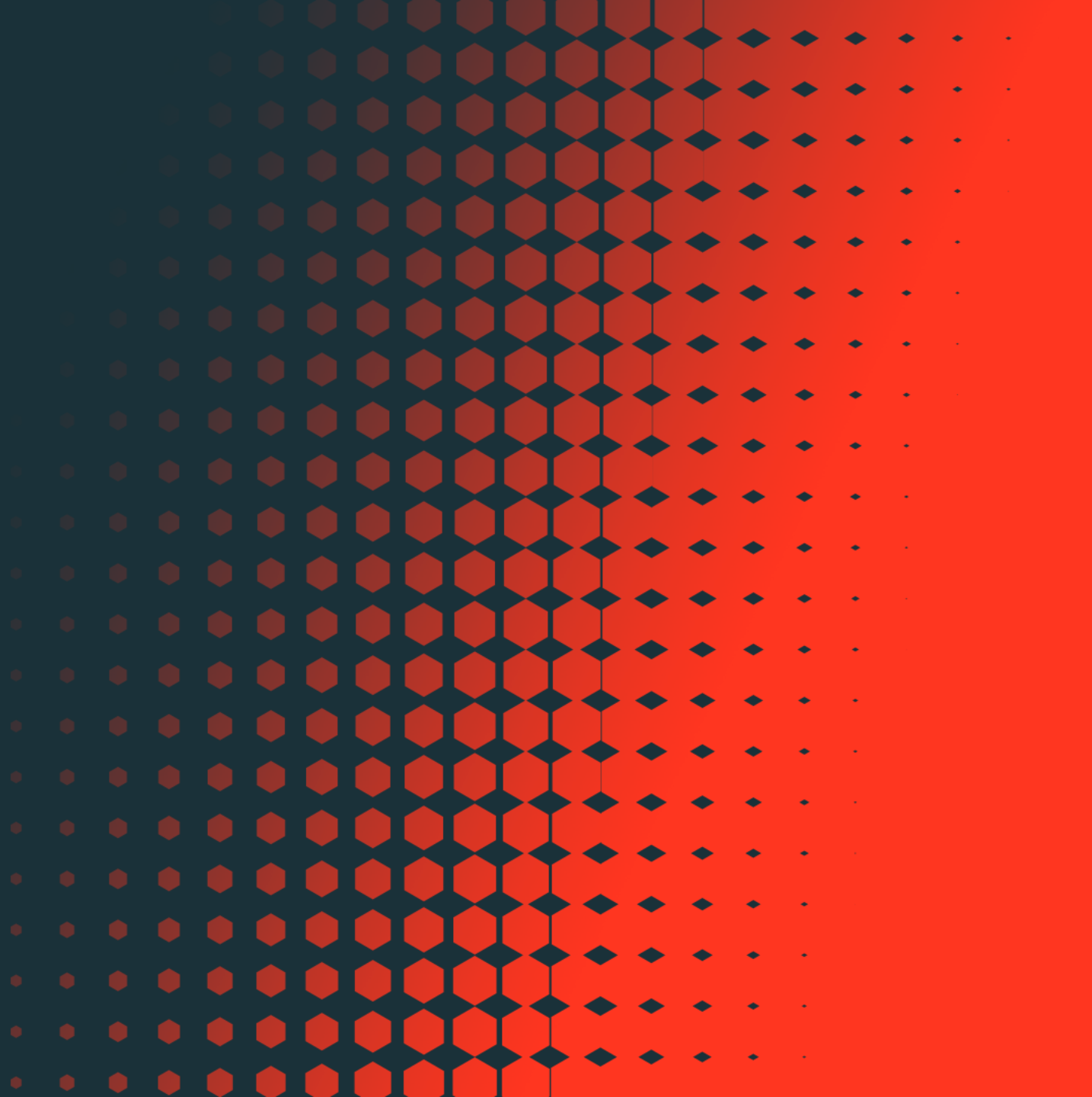
Name	Type	Purpose	Usage via	Ideal
Pandas	Data Analysis and Manipulation Library	• Data Handling • Transformation • Eager Execution	• Standalone Library • Pandas API on Spark • Dataframes	• Low Volume Data • Prototyping
PySpark	Python Interface for Big Data Processing **Framework**	• Distributed Computing • Parallel Processing • Lazy Evaluation • Fault Tolerance	• Pyspark • Context & Session • Dataframes	• Big Data • Scalability • Performance • Integration with multiple sources
SQL	Structured Query Language	• Querying • Transformations • Transactions • Storage	• SQLContext • Datasets	• Querying • Managing relational databases

POP QUIZ!



databricks

Demo



databricks

Databricks

Community Edition

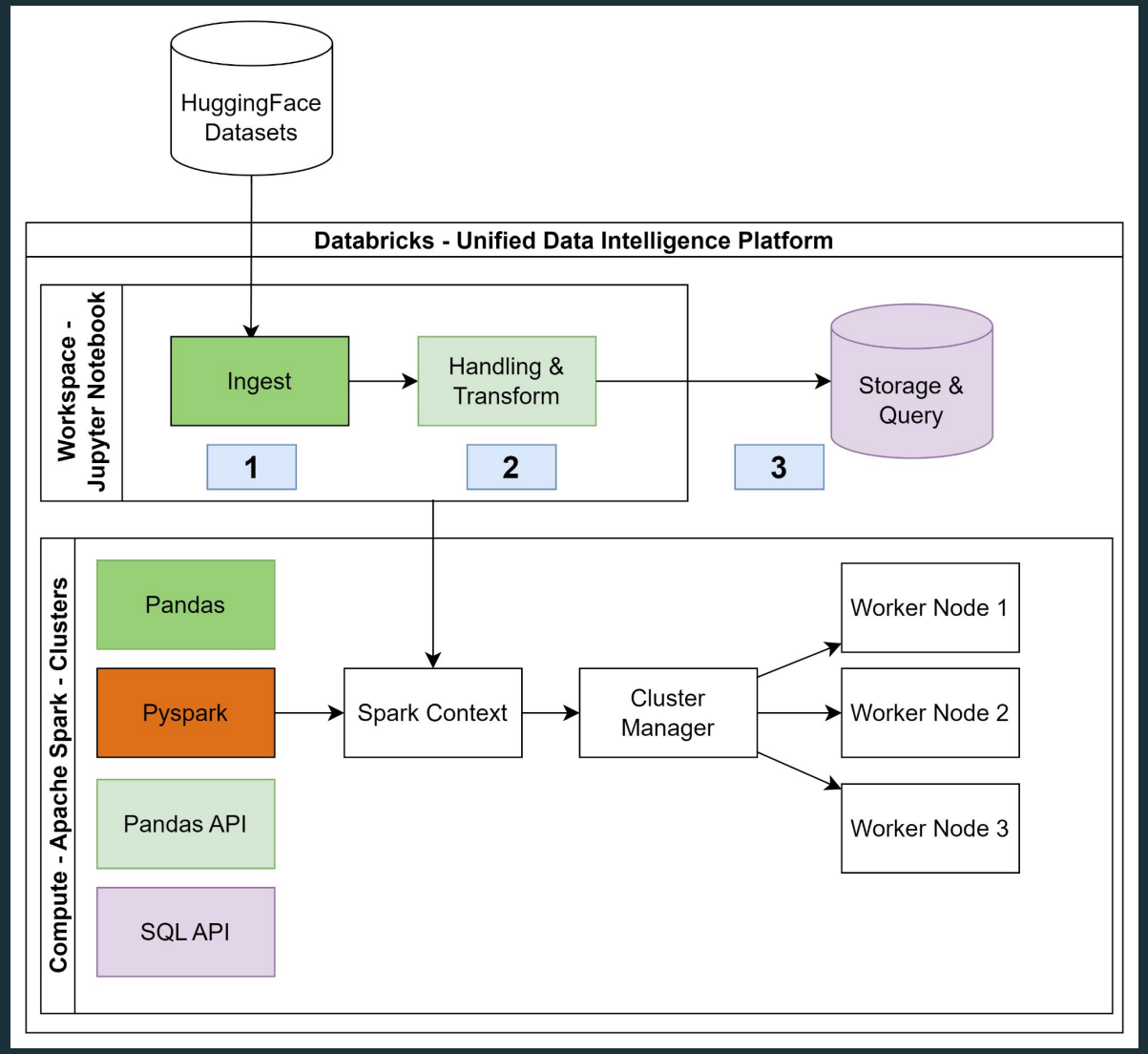
- Free access: https://community.cloud.databricks.com/
- Collaborative Notebook Environment
- Cluster Management
  - 1 Driver
  - 15.3 GB Memory, 2 Cores, 1 DBU
- Experiment Tracking

Data Workflow

Stages

1	**Ingestion**	• **Batch** Load from various sources • **Stream** in real-time
2	**Transformation**	• **Clean, transform, balance** techniques over raw data
3	**Storage**	• **Data Lake Integration** with AWS, Azure, GCP
4	**Analysis**	• **Statistics** • **Visualization:** Dashboards
5	**Machine Learning**	• **MLOPS**

Use Case

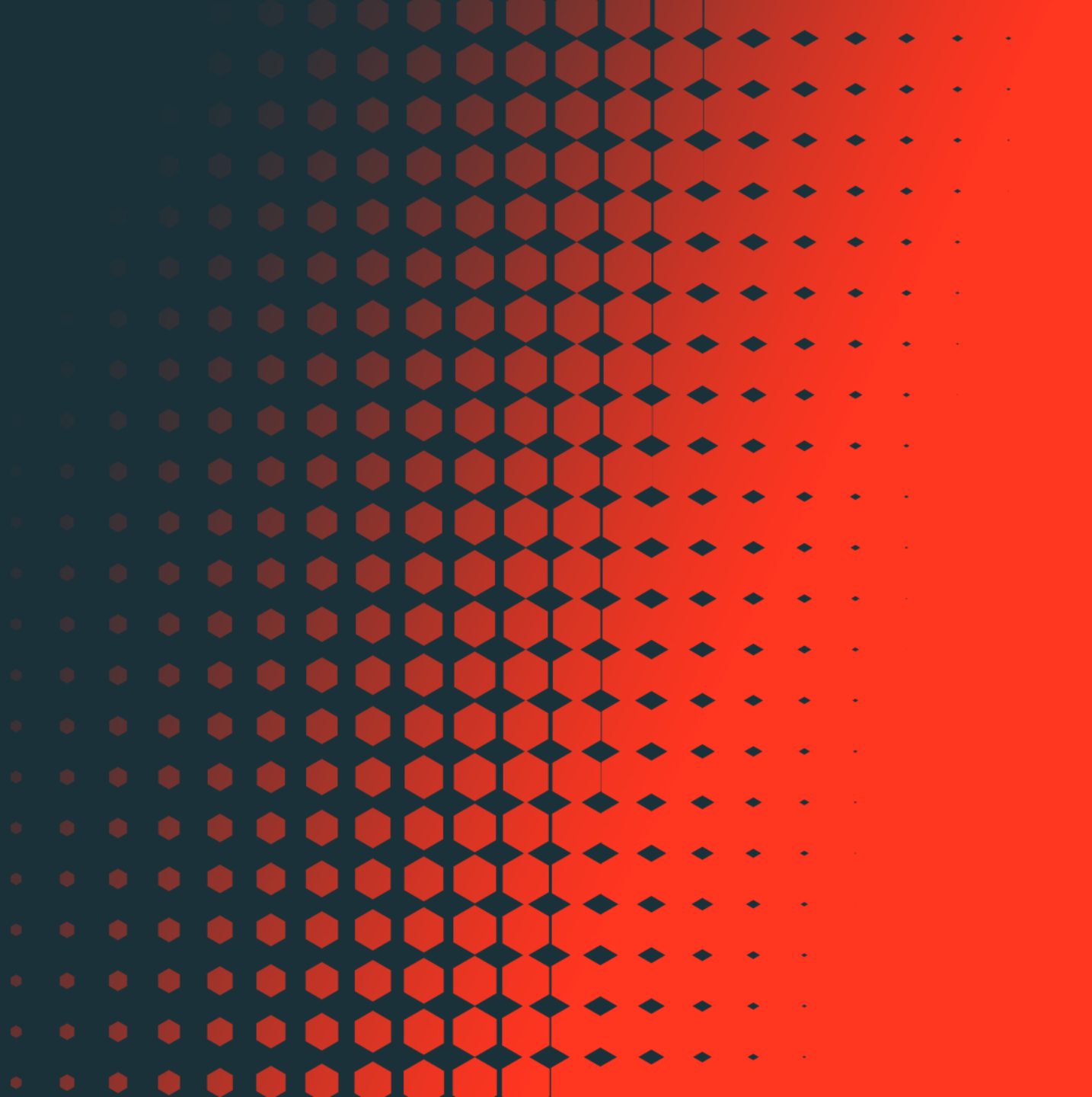


databricks

Key Takeaways

- Data Processing crucial step in Data Engineering
- Databricks Data Intelligence Platform
- Pandas is a powerful for data handling library aimed for single computing.
- Pyspark supports and enhances Pandas and SQL

Q&A



databricks

databricks