MLOps Framework Overview

- Product Operations
  - Product Concept

    What is Product, its parts and borders
  - Products Landscape

    How DS activities are classified
  - Product Management

    Backlog, priorities, development
  - Product Pipeline

    Embraces the whole pipeline

    "*Data ingress*" → "*Features Eng*" → "*Model or Algorithm creation*" →

    "*Model evaluation*" → "*Monitoring*" → "*Apply Model*" → "*Results Delivery*" → "*API*"
  - Assets and Delivery

    Overview of all assets along the DS Product Pipeline path and how they get delivered
- MLOps Capabilities

  Overview of tech capabilities to address DS SDLC specifics
  - Deployment Scenarios

    Deploying code vs Deploying models
  - Typical Developer's Scenarios

    What do we do with a Product pipeline
  - Product AB Testing

    Approach to organize hypotethes testing
  - Product Monitoring

    Keep track of resources utilisation for each Product|
  - Execution environments

    First of it is a logical separation.

    Basically that is a workspace with data, storage and

    compute resources governed accordingly
    - Development
    - Staging/Preproduction
    - Production
    - Connection between environments
- Development Operations

  What do we have to enable development? What templates and tools?
  - Fast Experimentations
  - Project code management
    - Workspace folders
    - Repositories

      Repos types, structures, principles
    - CICD Principles and Automation
    - ML Workflows CICD
    - Python libs CICD
    - Typical Scenarios - real examples

      Feature Implementation or Bug Fix in DEV and STAGING

      Workflow Change in DEV and STAGING

      Promotion of a Code Change to PROD
  - Execution machines
    - Runtimes
    - Clusters Policies
    - Clusters
  - Infrastructure Management

    Just a few words about what is included and that it is under a DevOps control
- Model Operations
  - Experiments tracking
    - MLFlow experiments
  - Model Registry

        - MLFlow Registry and Model stages
    - Model Deployment and Serving
        - Deploying code vs Deploying models
        - Real-time and Batch Serving
        - Model HTTP API Endpoints
- Data Operations
    - Data Architecture, Groups, Permissions
    - Unity Catalog
    - Raw, Bronze, Silver, Gold data layers
    - AP Data and External world

      Integration with external systems, what we call data import/export
        - SharePoint
        - Teams
        - Product specific destinations
    - Feature Store

      Whatever it is, aims to solve duplication and real-time serving issues.
    - Vector Store
    - Semantic Layer

      Move from fragmented data tables to connected, coherent Domain Knowledge layer
- Best practices

  Considering MLOps Capabilities above

  one's best use of them in the following principles
    - Documentation
    - Process data incrementally
    - Pipeline is in code
    - Specify configuration details once
    - Partition data
    - Conditional execution
    - Repeatability, Reproducibility, Reliability, Determinism, Idempotency

      Transfer some of a scientific

      relative principles onto computer systems to reduce error
    - Data Quality Monitoring

      DQ issue - incomplete/incorrect/full of duplicates data.

      Done for base tables, bronze layer. DQ team does:

      - DQ check the data meets specification.

      - The data was entered incorrectly.

      - Quality control failed to root out data quality problems.

      - Duplicate records were created.

      - The data isn’t used or interpreted correctly.

      - All known data about an object wasn’t integrated.

      - The data is too old to be useful anymore.
    - Data Drift Monitoring

      Data Drift issue - distribution changes, changes in covariances,

      changes in category ranges, changes in relations between features and target.

      Can be checked by statistical tests and using integration tests.
    - Rest data and models between tasks
    - Use job clusters and Service Principals
    - Unit and Integration Tests
    - EngX practices in action: real use-cases (formatting, understanding, repeatability, modularization, correctness)
    - Key Principles for Managing Data within Databricks
    - Data Architecture principles
    - Deduplicate efforts using "Feature Store"

      Have one place that described key business entities and ML tasks objects as much as possible
    - Experiments or model development tracking: real use-cases
    - Model deployment
- RACI Matrix/WoW

  How do we work together? Analytics Platform vs Analytics Products co-operation