Data Engineering and MLOps in Business

Feature Selection, Batch Inference Pipelines, Model Registry

Primoz Konda

AAUBS

March 25, 2024

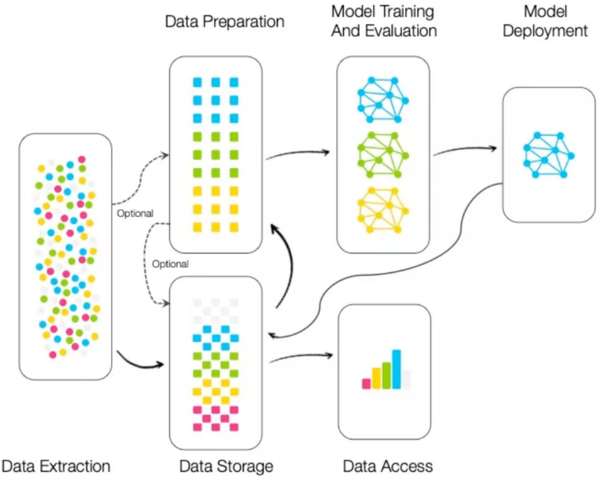
pk@business.aau.dk

Plan for today

Where are we now in MLOps journey?

- Technical issues?
- A bit different approach
- Simultaneously learning
- Parallel exercises

ML & Data Pipeline



From Feature Group to specific Feature View

Feature Group

- All data we can get for our project
- "The more the merrier"
- We are always open to get more data

Feature View

- Specific set of features for a specific model
- We can have multiple Feature Views for one project
- We exclude Redundant, Irrelevant, and Prohibited features

Influences on Feature Selection by Project Type (Part 1)

Real-time Predictions:

- Requires features that can be computed quickly.
- Emphasis on streaming data compatibility.

Static Models:

- Can utilize more complex features that are compute-intensive.
- Less concern for data freshness.

Highly Regulated Industries (e.g., Healthcare, Finance):

- Feature selection must consider compliance and ethical considerations.
- Transparency and interpretability become more critical.

Influences on Feature Selection by Project Type (Part 2)

Consumer Applications:

- User experience drives the need for quick, relevant feature computation.
- Privacy considerations may limit available features.

Research and Development:

- Feasibility of broader feature experimentation.
- Tolerance for longer model training and refinement cycles.

Feature Evaluation for MLOps (Part 1)

Aspect	**Details**
Feature Name	Name of the feature
Description	Brief description
Source	Data source
Availability	Frequency of updates
Accessibility Rating	1-10
Cost	Estimated cost
Required Permissions	Permissions needed

Feature Evaluation for MLOps (Part 2)

Aspect	Details
Accuracy	Data accuracy
Relevance	Relevance to the model
Historical Stability	Stability over time
Resource Intensity	Computational resources needed
Expected Model Impact	Potential impact
Previous Use Cases	Examples of past impact
Include in Model?	Yes/No
Rationale	Reason for decision

Model-Specific Transformations: Introduction

- **Definition:** Tailored preprocessing steps designed to optimize data for specific model requirements.
- **Purpose:** Enhance model performance, manage diverse data types, and improve learning efficiency.
- Examples include scaling for distance-based models, embedding for categorical data in neural networks, and time series decomposition for forecasting models.

Model-Specific Transformations: Examples

- **Scaling/Normalization:** Essential for models like SVMs and k-NN, where distance metrics are used. Prevents features with larger scales from dominating the learning process.
- **Text Embeddings:** Converts text into numerical vectors for NLP tasks, crucial for models such as RNNs and Transformers.
- **Time Series Decomposition:** Separates trends and seasonality in data for time series forecasting models, enhancing predictive accuracy.
- **Feature Encoding:** Different models require different encoding techniques. Decision trees handle categorical data naturally, while logistic regression may benefit from one-hot encoding.

Choosing the Right Transformations (Tips)

- Consider the **model type** and its mathematical underpinnings. Does it rely on distances or probabilities?
- Understand the **data structure and type**. Are you working with text, numerical, categorical, or time series data?
- Evaluate the **model's sensitivity** to feature scales, outliers, and missing values.
- Aim for transformations that **preserve important relationships** in the data while making it more digestible for the model.

Batch Inference Pipelines: Overview

- **Purpose:** Efficiently process large volumes of data to make predictions or inferences, typically in **a non-real-time** environment.
- **Use Cases:** Financial transaction processing, end-of-day stock analysis, large-scale image or document processing.
- **Advantages:** Can leverage economies of scale, optimize resource utilization, and process data during off-peak hours to reduce operational costs.

Key Components of Batch Inference Pipelines

- **Data Storage:** Repositories for storing raw and processed data (e.g., databases, data lakes).
- **Batch Processing System:** The engine that processes data in large batches.
- **Model Serving:** Mechanism to load and serve the machine learning model for inference.
- **Orchestration and Scheduling:** Tools to manage job sequences, dependencies, and timing.
- **Monitoring and Logging:** Systems to track pipeline performance, errors, and resource usage.

Challenges in Batch Inference Pipelines

- **Data Volume and Velocity:** Managing and processing large datasets within acceptable time frames.
- **Integration Complexity:** Ensuring compatibility between different components of the pipeline.
- **Model Versioning and Management:** Keeping track of model versions and updates in a scalable manner.
- **Cost Optimization:** Balancing computational resources with operational costs.

Model Registry: Overview

- **Definition:** A centralized hub for managing the lifecycle of machine learning models, including versioning, storing, and accessing models.
- **Purpose:** Facilitates collaboration among teams, ensures model traceability, and streamlines model deployment and monitoring.
- **Functions:**
  - Version Control: Keeps track of different versions of models.
  - Model Staging: Manages model stages (development, staging, production).
  - Metadata Storage: Stores model metadata, including training data, parameters, and evaluation metrics.

Benefits and Key Features of a Model Registry

- **Benefits:**
  - Streamlined model deployment and rollback processes.
  - Enhanced collaboration and governance through access control.
  - Improved model performance tracking over time.
- **Key Features:**
  - Integration with ML pipelines for automatic versioning and tracking.
  - APIs for model deployment, retrieval, and monitoring.
  - Support for annotations and comments to enhance collaboration.
  - Compatibility with various machine learning frameworks and environments.

Lets start coding...

First, we will do Module 3 LAB.

Second, we will start working on the two-day project!