MLOps Features: Model Comparison and Auto-Documentation

Victor Absil, Armand Chambaud, Ivan Leboucher, Faustin Mounier

11/04/2025

We implemented two simple but very useful tools on the DSBA MLOps platform: a model comparison function and an automatic documentation generator.

1. Model Comparison Function The function `compare_models_simple()` allows users to load and evaluate several models stored in the registry. Given a list of model IDs and a test dataset, it computes selected metrics (currently accuracy and F1-score) using scikit-learn and returns a comparison table.

This function provides:

- A quick and standardized way to evaluate the relative performance of models.
- The ability to test naive baselines (random, always-positive, always-negative) alongside more developped classifiers (Random Forest, XGBoost).

2. Documentation Generator We developed a Python script that read all source files under `src/` and extracts the names of all defined classes, methods and functions using Python's `inspect` module. It then creates a text summary listing the architecture of the system.

This tool serves several purposes:

- It helps to quickly understand where are each definitions located.
- It reduces the effort of onboarding or debugging in larger teams.
- It supports maintainability and makes the platform more explainable.

Instead of relying on human-written documentation this script provides a snapshot of the current implementation and help IT teams to understand quicker the code.

Why It Matters in MLOps In a production MLOps environment model comparison is essential for tracking progress and preventing performance regressions. Plus the auto-documentation supports auditability and reduces adaptation time for new contributors or reviewers. These two tools though minimal in code enable scalable practices aligned with MLOps principles:

- **Comparison**: Brings reproducibility and visibility.
- **Documentation**: Encourages code clarity and supports rapid team transitions.