Enhanced MLOps Workflow: Experimentation and Benchmarking

Sofia de Tremiolles

April 11, 2025

Overview

This project builds upon the starter code from the course repository to implement some useful MLOps functions. I prioritized a simple project scope to really master the basics while focusing on realistic and useful functionalities. The goal is to enable Data Scientists to experiment with different input datasets while maintaining a fixed model architecture, thus providing insight into how data variations can affect performance. The added functionalities include running experiments, benchmarking model performance, and producing CSV reports. The API and Docker integration ensure that these operations are reproducible and production-ready.

Key Enhancements

- **Experiment Functionality:**
  - Added a Python function to run experiments using different datasets.
  - The experiments use a pre-defined model architecture and focus on assessing performance differences based on dataset granularity (fine-grained vs. large datasets) and other data variations.
- **Benchmark Reporting:** Generates a CSV report summarizing the performance of all models for a specified metric, making it easy to share results with management.
- **API Integration:** Built FastAPI endpoints to run experiments and generate benchmark reports.
- **Dockerization:** Containerized the full application to support seamless deployment and reproducibility.

Practical Considerations

- The core purpose is to provide a reproducible process where model architecture is fixed and only the input data is varied.
- The approach demonstrates how even a simple, focused toolset can offer valuable insights into model performance when different datasets are used.
- While the functionalities implemented are modest, they mirror real-world operations by streamlining experiment management and reporting.

Conclusion

The enhancements made to the codebase introduce a practical MLOps workflow. By integrating experiment execution and benchmark reporting with API endpoints and Docker, the project

reinforces the concept that effective MLOps is about building a robust infrastructure for managing models in production. This toolset allows a Data Science team to quickly iterate on input data configurations, obtain clear performance benchmarks, and communicate results in a straightforward, reproducible manner.

Discussion

Two main improvements for this project:

- Add unit and integration tests to ensure everything runs smoothly, since this is mostly an internal MLOps tool (not production ready) I did not consider this as a priority so far.

- Deploy on the cloud to enable users to connect to the Docker container without cloning the repository.