Tên: Đặng Trần Nam

Lớp: MSE20HCM

MSSV: 24MSE23149

Môn: DDM501c

# CÁC BƯỚC THỰC HIỆN DỰ ÁN MLOPS

## 1. Thiết lập cấu trúc dự án

- Tạo cấu trúc thư mục chính: src/, data/, app/, memory-bank/

- Tạo file requirements.txt với các thư viện cần thiết

- Thiết lập môi trường MLflow để theo dõi thí nghiệm

## 2. Xây dựng module tạo dữ liệu (src/data\_utils.py)

- Sử dụng scikit-learn make\_classification để tạo dữ liệu tổng hợp

- Chia dữ liệu thành tập huấn luyện và kiểm thử

- Lưu dữ liệu vào thư mục data/ để sử dụng cho các bước tiếp theo

## 3. Xây dựng mô hình (src/model.py)

- Thiết kế mô hình MLP sử dụng PyTorch

- Làm cho kiến trúc mô hình có thể cấu hình được (số lớp ẩn, số đơn vị, v.v.)

- Đảm bảo mô hình hoạt động với cả CPU và GPU

## 4. Xây dựng quy trình huấn luyện (src/train.py)

- Xây dựng hàm tải dữ liệu và khởi tạo mô hình

- Thiết lập hàm huấn luyện với bộ tối ưu, hàm mất mát và các hyperparameter

- Sử dụng MLflow để ghi lại các thông số, chỉ số và artifact

## 5. Tối ưu hóa hyperparameter (src/tune.py)

- Thiết lập quy trình tìm kiếm hyperparameter hiệu quả nhất

- Thử nghiệm với các cấu hình khác nhau (số lớp, tốc độ học, v.v.)

- Đánh giá mô hình bằng F1-score và các chỉ số khác

- Đăng ký mô hình tốt nhất vào MLflow Model Registry

## 6. Xây dựng ứng dụng web (app/)

- Tạo ứng dụng Flask để phục vụ dự đoán từ mô hình

- Thiết kế trang chính (index.html) để nhập đặc trưng

- Thiết kế trang kết quả (result.html) để hiển thị dự đoán

- Thêm nút "Ngẫu nhiên" để tạo đặc trưng ngẫu nhiên cho việc kiểm thử

## 7. Giải quyết vấn đề tương thích thiết bị

- Đảm bảo mô hình hoạt động trên cả CPU và GPU

- Xác định thiết bị hợp lý (CUDA nếu có, nếu không thì CPU)

- Cập nhật mã để xử lý đúng việc tải mô hình từ registry

## 8. Cải thiện trải nghiệm người dùng

- Thêm JavaScript để tạo đặc trưng ngẫu nhiên với một click

- Cải thiện xử lý lỗi để hướng dẫn người dùng nhập đúng định dạng

- Làm cho giao diện trực quan và dễ sử dụng

## 9. Tạo script chạy toàn bộ quy trình (run.py)

- Xây dựng script để chạy từng bước của quy trình

- Cho phép chạy các bước riêng lẻ: generate, train, tune, app

- Đảm bảo đường dẫn Python đúng để tải mô-đun

## 10. Kiểm thử

- Tạo predict\_client.py để kiểm thử API dự đoán

- Xây dựng các script kiểm thử để xác minh tải mô hình và dự đoán

- Đảm bảo ứng dụng hoạt động đồng bộ từ đầu đến cuối

## 11. Tài liệu

- memory-bank với mô tả dự án

- Tạo hướng dẫn chạy từng bước của quy trình

- Ghi lại các quyết định kỹ thuật và mô hình tư duy

## Quy trình chạy hoàn chỉnh

1. Tạo dữ liệu: `python run.py --step generate`

2. Huấn luyện mô hình: `python run.py --step train`

3. Tối ưu hyperparameter: `python run.py --step tune`

4. Khởi động ứng dụng web: `python run.py --step app`

5. Truy cập giao diện web tại http://localhost:5000

6. Kiểm thử API: `python predict\_client.py`

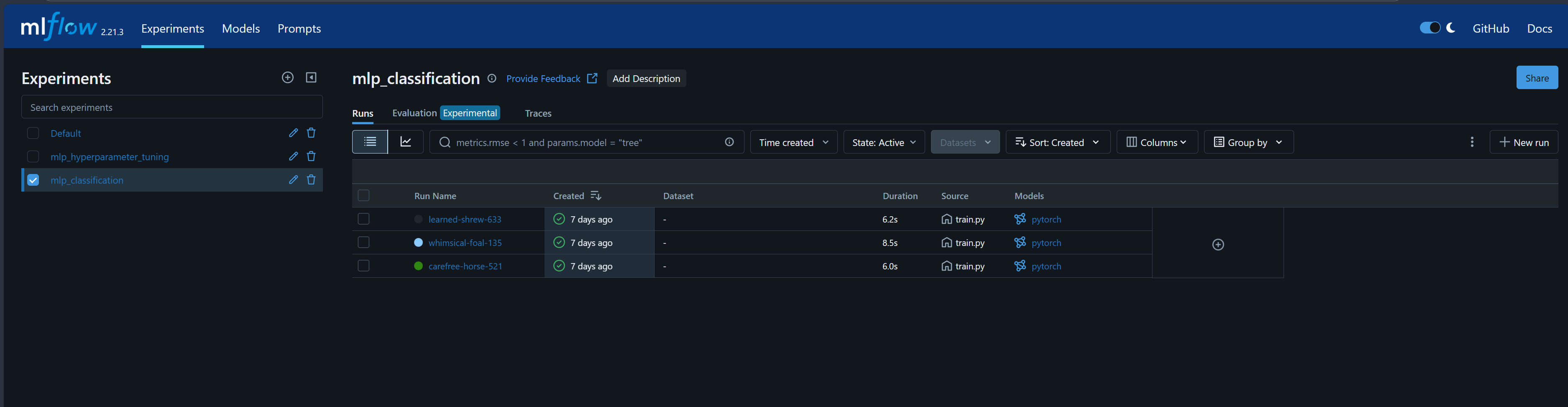


Figure 1: training experiments

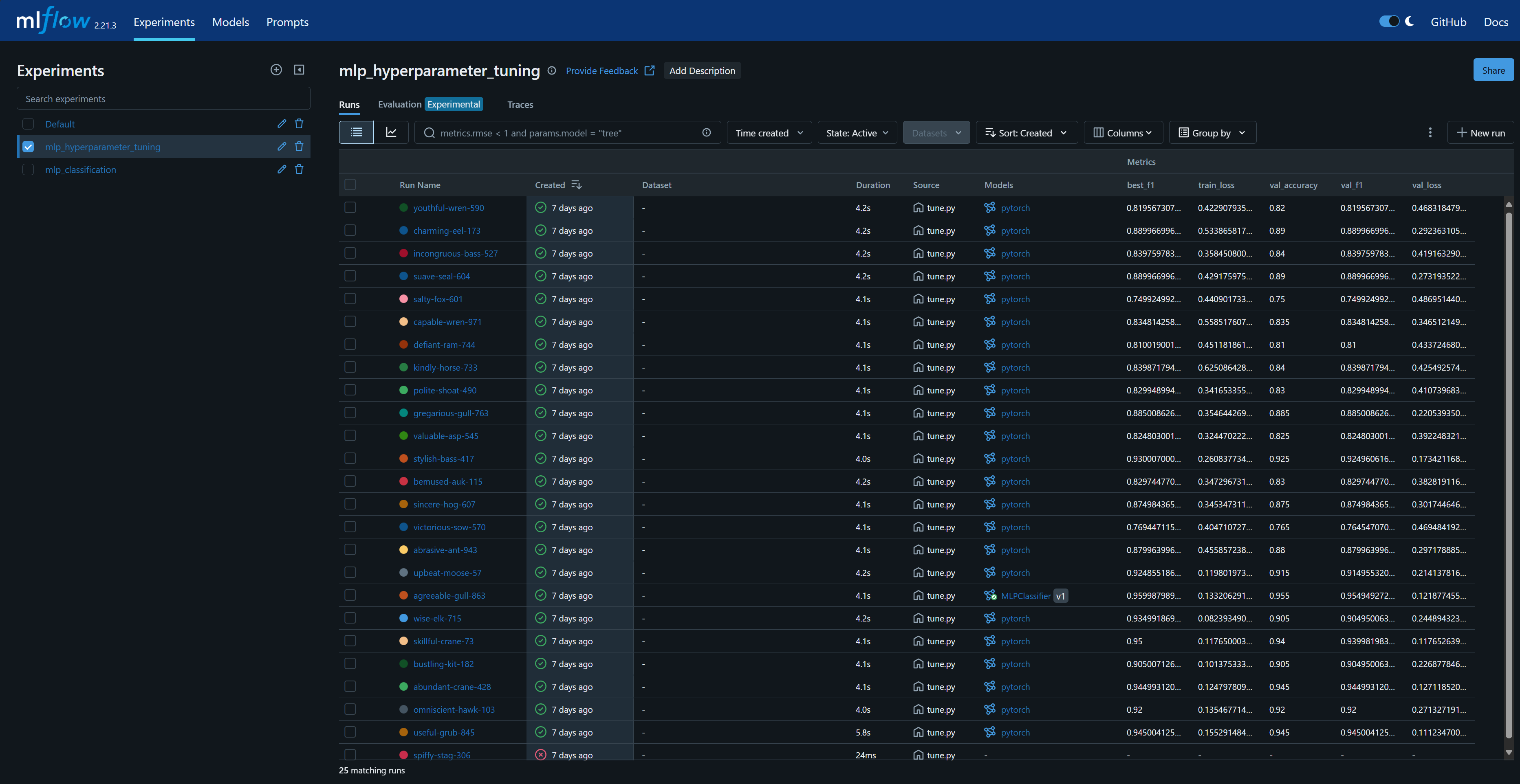


Figure 2: Fine-tunning

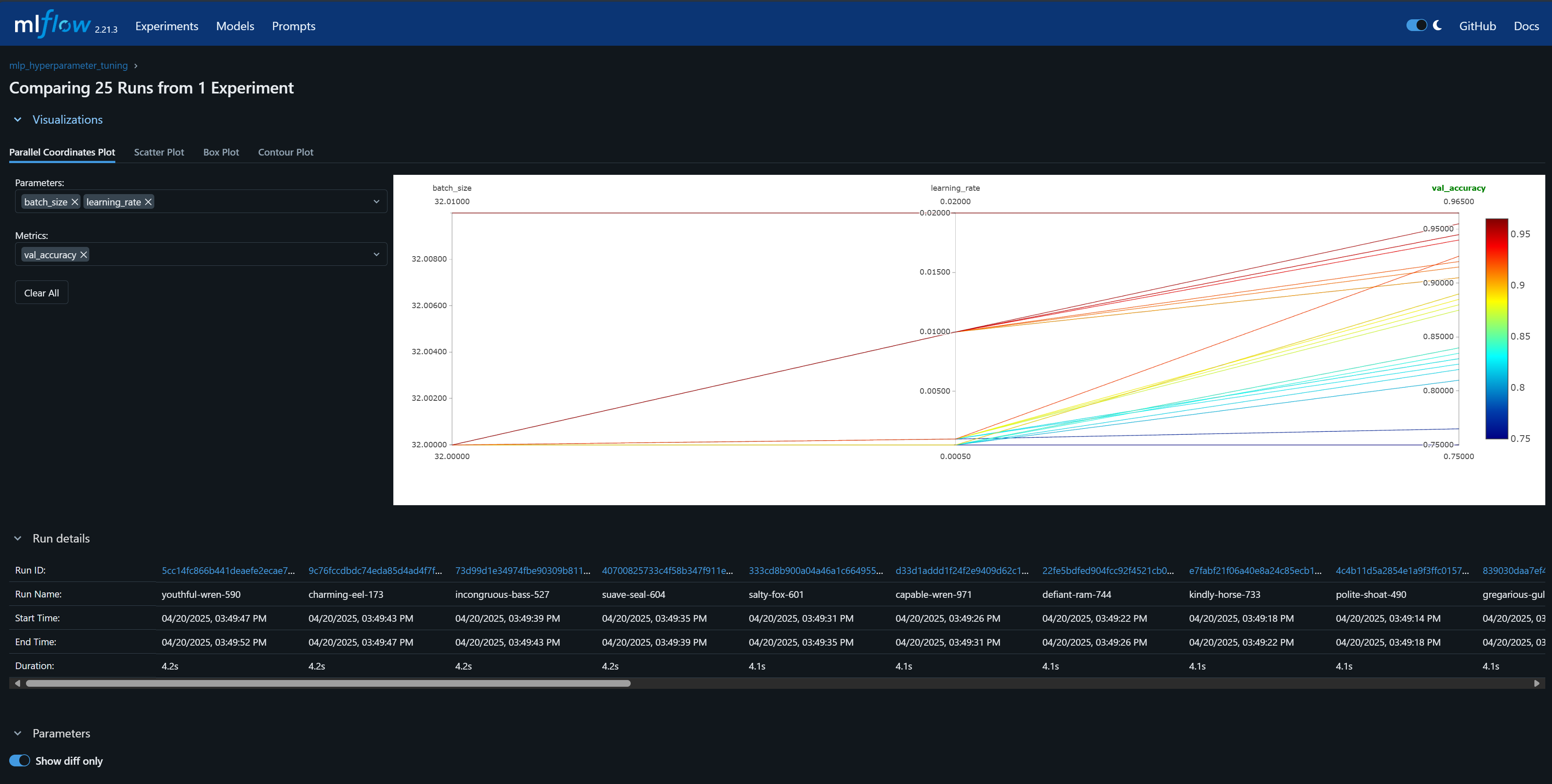


Figure 3: Fine-tunning comparison

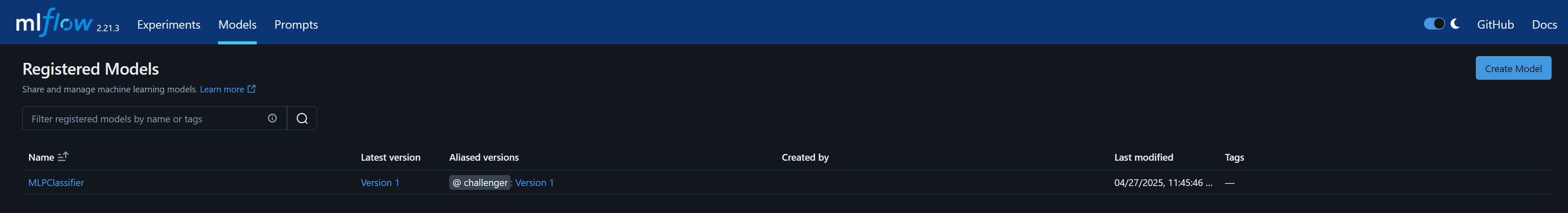


Figure 4: Saved models

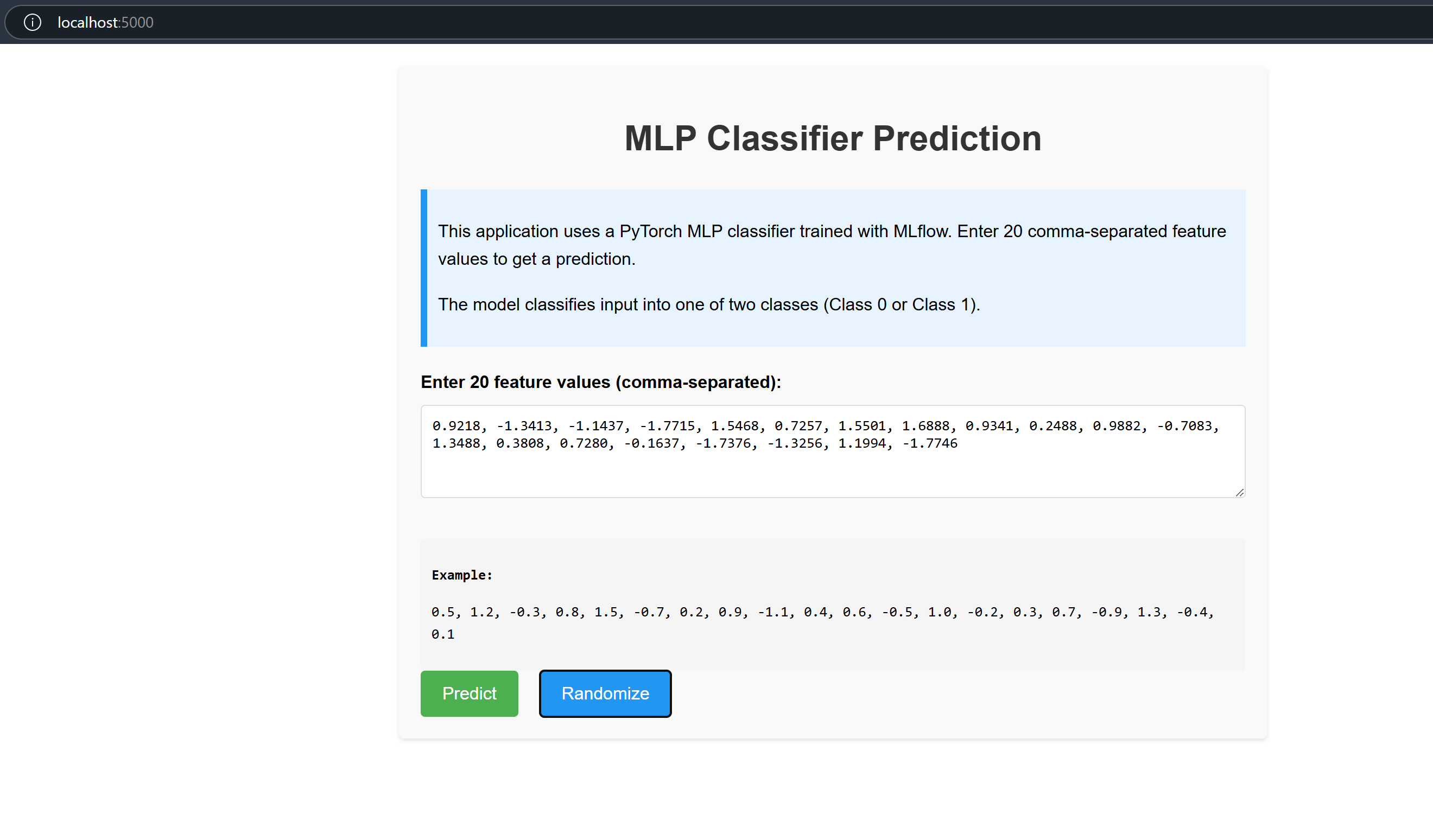


Figure 5: UI prediction

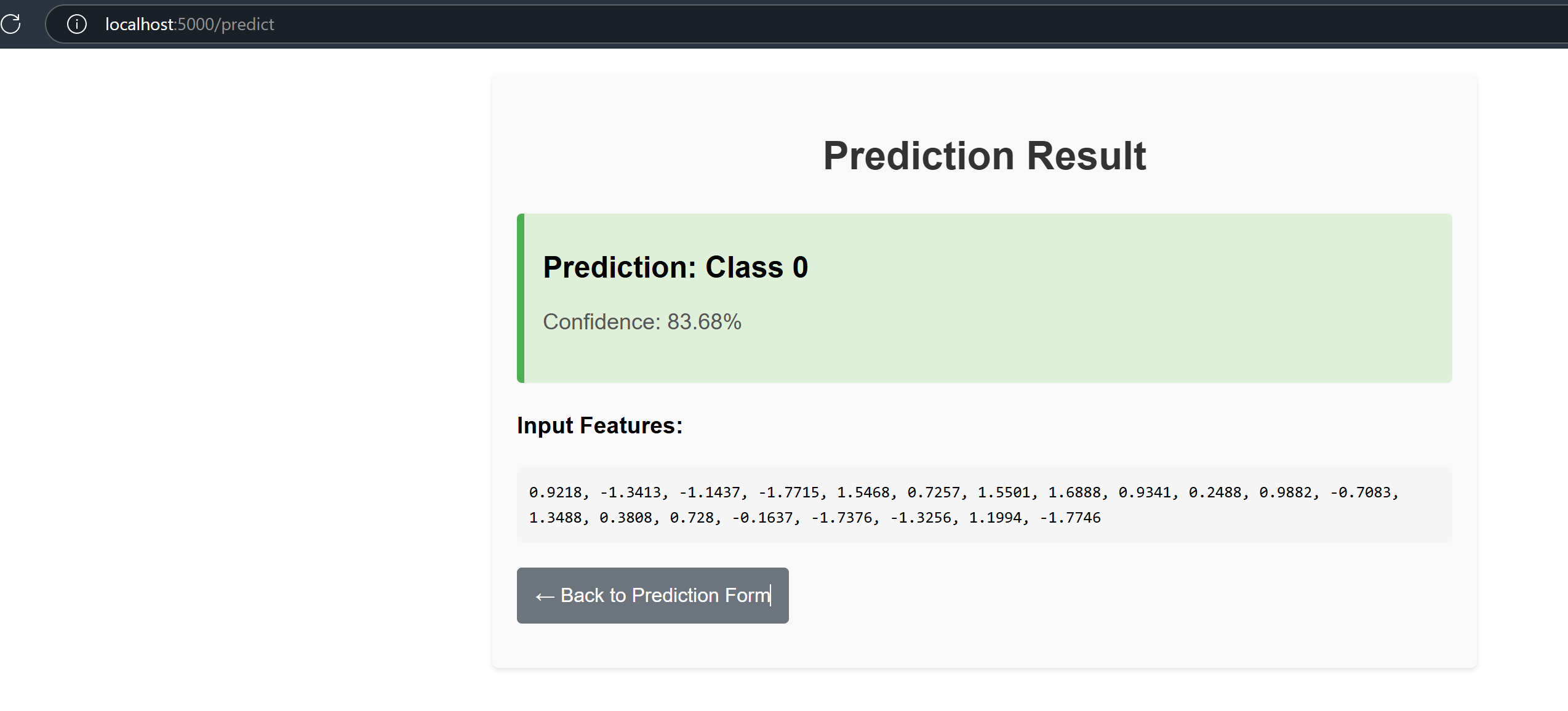


Figure 6: UI Prediction Result