Week11 - Deploy Model

Ngày 26 tháng 11 năm 2025

1 Tổng quan

1.1 Flask

Flask (2010): triết lý của Flask là đơn giản và linh hoạt. Nó cung cấp những thế cơ bản nhất để xây dựng một web server, còn lại (database, validation, authentication) sẽ tùy chọn thư viện để cài thêm.

Flask dựa trên kiến trúc WSGI (Synchronous - Đồng bộ). Xử lý từng request một cách tuần tự.

1.2 FastAPI

FastAPI (2018): Triết lý của FastAPI là tốc độ (hiệu năng tốc độ code) và chuẩn hóa. Nó tích hợp sẵn những công cụ mạnh mẽ nhất để làm API (như Pydantic, Swagger).

2 So sánh

Tiêu chí	Flask	FastAPI
Kiến trúc cốt lõi	WSGI (Synchronous - Đồng bộ). Xử lý từng request một cách tuần tự (mặc định).	ASGI (Asynchronous - Bất đồng bộ). Xử lý song song nhiều request nhờ tận dụng thời gian chờ I/O.
Tốc độ xử lý	Khá. Phù hợp với đa số web app thông thường.	Rất nhanh. Ngang ngửa NodeJS và Go nhờ xây dựng trên Starlette.
Cú pháp (Syntax)	Python truyền thống. Không bắt buộc khai báo kiểu dữ liệu.	Python hiện đại (Type Hints). Bắt buộc khai báo kiểu input/output.
Kiểm tra dữ liệu (Validation)	Không có sẵn. Phải code tay hoặc cài Flask-Marshmallow.	Tích hợp sẵn Pydantic. Tự động validate dữ liệu cực mạnh.
Tài liệu API (Docs)	Không có sẵn. Phải cài Flasgger hoặc viết tay.	Tự động 100%. Sinh ra Swagger UI / ReDoc ngay khi chạy code.

Khi nào nên dùng Flask:

- Ứng dụng đồng bộ (synchronous): không cần xử lý nhiều request đồng thời.
- Web application truyền thống, template HTML
- Ứng dụng cần nhiều plugin/extensive ecosystem: Flask có cộng đồng lớn, nhiều extension

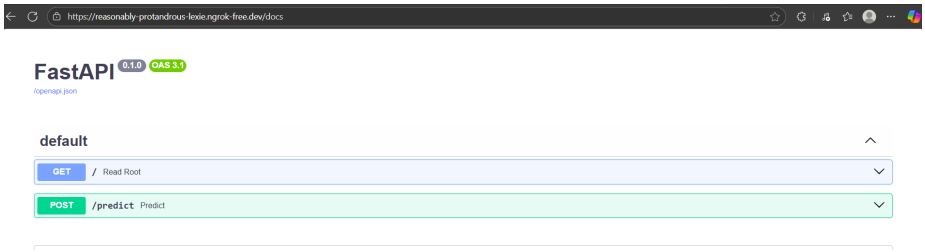
Khi nào nên dùng FastAPI:

- API / microservices hiệu năng cao: xử lý nhiều request đồng thời nhờ async/await.
- API cần validation và docs tự động: tích hợp Pydantic, Swagger UI tự động
- Code type-safe, maintainable: hỗ trợ type hints, autocomplete IDE, static checking

3 Thực hành

Trong bài tập thực hành này, mục tiêu là triển khai (deploy) một mô hình học máy dưới dạng dịch vụ web API bằng cách sử dụng FastAPI và Ngrok được sử dụng để tạo đường hầm (tunnel) từ máy cục bộ ra Internet, đóng vai trò như một API Gateway tạm thời, giúp các ứng dụng bên ngoài có thể truy cập API mà không cần cấu hình server hoặc triển khai lên cloud.

Sau khi triển khai ứng dụng FastAPI trên máy cục bộ và sử dụng Ngrok để tạo đường hầm ra Internet, người dùng có thể truy cập tài liệu API tự động qua địa chỉ công khai mà Ngrok cung cấp. Trong hình là giao diện Swagger UI tại URL:



Hình 1: Swagger UI hiển thị các endpoint của dịch vụ FastAPI

Đây là đường dẫn mặc định FastAPI tạo ra nhằm hỗ trợ kiểm thử và tương tác trực tiếp với API.

Trong giao diện tài liệu API (Swagger UI), hệ thống cung cấp hai endpoint chính như sau:

(1) `GET /` – Read Root Đây là endpoint gốc (root) của ứng dụng FastAPI. Mục đích của endpoint này là kiểm tra nhanh trạng thái hoạt động của server. Khi gửi yêu cầu `GET` tới đường dẫn "/", hệ thống sẽ phản hồi một thông điệp JSON đơn giản, ví dụ:

```
{
    "message": "Hello World"
}
```

Sự tồn tại và phản hồi thành công của endpoint này cho phép xác nhận rằng dịch vụ FastAPI đã khởi động đúng cách và quá trình chuyển tiếp yêu cầu thông qua Ngrok hoạt động ổn định.

(2) `POST /predict` – Endpoint dự đoán Đây là endpoint quan trọng nhất của ứng dụng, được sử dụng để gửi dữ liệu đầu vào tới mô hình học máy và nhận lại kết quả dự đoán. Người dùng hoặc các hệ thống bên ngoài có thể gửi yêu cầu `POST` kèm theo payload chứa các đặc trưng đầu vào.