Lab 7b: Serving & Deploying an ML Model

Overview

In this lab you will:

- **Train a Simple Model:** Train a logistic regression model on the Iris dataset and save it.
- **Build a FastAPI Backend:** Create an API that accepts Iris measurements and returns a species prediction.
- **Develop a Streamlit Frontend:** Build an interactive interface that collects input, calls the FastAPI endpoint, and displays the result.
- **Containerize the Application with Docker:** Package both the backend and frontend into one Docker container, deployable on Hugging Face.

Note: The frontend will launch the FastAPI backend as a background process so that both run in a single container.

2. Prerequisites

1. **Python 3** environment and relevant packages. We'd recommend creating a new virtual environment using:
    - Navigate to the cloned folder lab7: `cd <your-path>/lab7b`
    - Create environment: `python -m venv env`
    - Activate environment: `source env/bin/activate`
2. Download the lab7 zipped folder, containing following files, from LMS.
    - requirements.txt (for installing packages)
    - train.py (for model development)

    - backend.py (for backend server and model)
    - frontend.py (for frontend server)

3. Model Training

The training script is already given as train.py. Run it to train and save the model.

- python train_model.py
- You should see iris_model.pkl file in your directory

4. Create the FastAPI backend

The server script is also given as backend.py.

- Start the streamlit server by running streamlit run app.py in your directory
- You should see your application deployed at http://localhost:8501/

5. Build an Interactive Interface using Streamlit

The frontend script is also given as frontend.py.

- Start the streamlit server by running streamlit run app.py in your directory
- You should see your application deployed at http://localhost:8501/

The frontend launches the FastAPI backend as a background process, then makes HTTP requests to it when the user clicks "Predict."

6. Containerize the Application

Dockerfile is given with appropriate steps. Make sure requirements.txt file is placed in the same directory.

- **Docker image**: Open a terminal and execute docker build -t iris-app (make sure docker engine is running)
- **Run the image:** Run the application with docker run -p 8501:8501 iris-app
- You should see your application deployed at http://localhost:8501/

7. Submission

Submissions are in-class.