Project : Collaborative Pipeline for Sentiment Analysis Partie 2

Aperçu

Following on from your work on the sentiment detection project, you will now complete your MLOps pipeline by integrating containerization with Docker and automation with GitHub Actions.
The goal of this second part is:

- Ship your app inside one or multiple containers.
- Implement a CI/CD Github Actions pipeline.
- Automate tests, evaluation and deployment of the model.

Part 1: Containerization with Docker

1. Creating the Dockerfile
   - Create a `Dockerfile` at the root of your project
   - Configure it to install all necessary dependencies
   - Expose an entry point for predicting sentiments (API or CLI)

2. Volumes for data and models
   - Identify data/models that need to persist
   - Set up a Docker volume to store them

3. Docker Compose
   - Create a `docker-compose.yml` file
   - Define necessary services (API, DB for logs, etc…)
   - Configure network between services

Part 2: Automation with GitHub Actions

1. Test workflow
   - Create a single workflow `.github/workflows/test.yml` that:

     - Runs on every push and pull request
     - Runs all your unit and integration tests
     - Checks code quality (linting)

2. Model evaluation workflow
   - Create a workflow `.github/workflows/evaluate.yml` that:
     - Runs after successful testing
     - Evaluates model performance on a test set
     - Stores performance metrics as artifacts
     - Fails if performance is below a threshold

3. Docker build workflow
   - Create a workflow `.github/workflows/build.yml` that:
     - Builds the Docker image
     - Publishes it to the Dockerhub Registry

Deliverables

Documentation – Update the README to include a brief description of each component

Project Report – Document the complete MLOps architecture, explaining your technical choices (Docker, Github Actions) and describe the complete workflow (you can use diagrams). URL to your github repository, screenshots of github actions runs. Do not forget to add your names.