Machine Learning in Production Final Project

The purpose of this project is to combine all the knowledge acquired in the course, and build a web app that serves a machine learning model, with all the associated CI/CD processes, versioning, and testing. Finally, the web app should be deployed to a cloud platform.

You are to work in groups of 4

Project Topic

The project can be about any topic of interest for you. It could be about a hobby, a problem you face, or something that you are passionate about. Think about building your startup, or something interesting that you believe is worthy of an app.

You will be judged based on your execution, not the idea itself.

Technical Overview

The project is based on a web app that serves a machine learning model. The technology stack is up to you, but it's recommended to use Python for the backend that runs the ML model, and a NodeJS framework for the frontend, like ReactJS or NextJS. You can have more than one backend if you see the need to, and include a database if it is required by your app.

A good online service for a DBMS is Supabase or NeonDB.

Code architecture

Project

- Code (managed by Git, saved on Github)
  - app code (frontend, backend, ...)
  - tests (unit, integration, ...)
  - Dockerfiles (and docker-compose file)
  - Github workflow files (CI/CD)
- ML (managed by MLFlow, saved on DagsHub)
  - Notebooks
  - MLFlow registry
- Data (managed by DVC, saved on S3 or any data store you choose)
  - Raw data

Github workflow

On pull request to dev branch

- Build the app (if applicable)
- Run integration tests
- If all good, manually merge into dev branch

On push to dev branch

- push to staging branch

On push to staging branch

- Build the app (if applicable)
- Run all tests
- push to production branch (main)

On push to production branch (main)

- build Docker images
- deploy Docker images to DockerHub
- Deploy to production

How deployment should work

After the tests are ran, the docker image is to be built with the latest model version included in the build. The latest model should be fetched from the MLFlow instance running on your DagsHub repo.

After the Docker image is pushed to DockerHub, it should be deployed to the cloud platform of your choice.

Some platforms that can be used for deployment:

- koyeb.com
- railway.app
- aws.amazon.com
- gcp.google.com
- azure.microsoft.com

At the end, there needs to be an accessible url publicly available, with the app running and serving the ML model.

Evaluation

You will be evaluated on the following criteria:

- proper branching of your github repo
- the inclusion of the MLFlow registry and DVC for data management
- the inclusion of 3 unit tests, 3 integration tests, and 3 end-to-end tests
- a fully running CI/CD platform, that handles the previously mentioned steps in Github workflow
- Docker images are built and pushed to DockerHub
- the app is deployed to a cloud platform, with a publicly accessible url
- the app is running and serving the ML model