

Project 2: Containerization

This is a single-person project. It is fine to discuss your process with other students but make your own decisions and write your own code and report.

You are still an ML engineer at Newcrosoft. Your company has decided that everyone should use Docker from now on, in order to homogenize the local development environment with the production environment in the cloud. You have been tasked with converting a project to a containerized app.

Deliverables

Submit a report documenting how you solved the tasks below, especially tasks 2 and 3. The report should be at most 2 A4 pages, in PDF format. No appendix. Choose a reasonable font and size. At the end, include a reflection of what went well and what you would improve next time. In the report, include a link to your public GitHub repo for the last task.

Tasks

Task 1: Adapt the training notebook from the last project (use your own code, or the [original](#)) into one or more Python scripts. It should be possible to start a training run with a single call, passing the hyperparameters and save path as arguments (e.g. `python main.py --checkpoint_dir models --lr 1e-3`). Log your training run with your favorite experiment tracking tool.

Task 2: On your local machine, create a Docker image that runs a single training run with the best hyperparameters you found in Project 1. You typically won't have GPU support on your machine, so this will take longer than on Colab.

Task 3: Run your Docker image on GitHub Codespaces or the Docker playground. Did it all run without changes, or did you have to make adaptations? Verify the training in your favorite experiment tracking tool. Is the performance exactly the same as on your local machine?

Task 4: Create a public GitHub repository with all your project files. Include the Dockerfile. Make it as easy as possible for someone to clone your repository, build your Docker image and run it. Describe all necessary steps in the project's README file, using proper Markdown.

Grading

The following criteria will determine the grade:

- Structured and organized approach
- Clear documentation of how tasks were solved
- Reflection on what went well and what you would improve next time
- Clean GitHub repository with useful README
- Report focuses on tasks 2 and 3
- Report adheres to specifications

Hints

- Maybe someone will download and try to run your project in the near future...
- Docker Playground has very little memory. Think about reducing your image size (e.g. just installing PyTorch for CPU) if you want to use it.