Project Description

Overview

In this project you will implement, evaluate, operate, monitor, and optimize an end-to-end ML pipeline. The details of your assigned pipeline: datasets, model choices, metrics are provided to you in an accompanying README.

The focus of the project is to implement the pipeline choices, evaluate which configuration of the pipeline is best, under what data distributions, operationalize the pipelines on the cloud, monitor their real-time behavior, and optimize the choice of a pipeline configuration.

The project entails many concerns, including deployment, scaling, reliability, drift and feedback loops. It also has three milestones including a final project presentation and report.

Overall mechanics and infrastructure

Teamwork: You will work on this project in your assigned teams. As a team, you will use a shared GitHub repository and a virtual machine to coordinate your work. Please establish this repository, and a way of communication and collaboration that works for your team – for example, a Teams channel. Please agree on tasks and responsibilities. We expect that team members will have different backgrounds and different amounts of experience with machine learning and engineering – this is intentional. Use the first milestone to identify strength and weaknesses, fill missing gaps in background knowledge, and teach each other about tools and practices that are effective for this task. Finally, be a good team citizen.

Milestones: For each milestone and the final presentation, there are separate deliverables, described below. The milestones are checkpoints to ensure that certain functionality has been delivered.

Milestones are graded on a pass/fail scheme for the criteria listed with each milestone. A fail turns into a pass if it is achieved by the end of the next milestone. All milestones must be delivered by the final due date. Note, Milestones will build on each other.

Pipelines: Each pipeline has two stages except ICP which has three. Each pipeline has input datasets and variety of models for each stage. Your assigned pipeline has been communicated you over email. The pipleline repository is here.

Infrastructure: For each pipeline, you will find its Dockerfile in the specific repository. The Dockerfile specifies datasets and models to download. You should build the Dockerfile into a container and peruse the data and model files.

> You may request more computing resources from the course staff if the virtual machine's resources are insufficient — we may or may not be able to accommodate such requests, it may take a few days, and will require a system reboot.

Data For Milestone 1 you use all provided data. This section will be revised for subsequent milestones.

Languages, tools, and frameworks: Your team is free to chose any technology stack for any part of this project. You have root access to your virtual machine and are free to install any software you deem suitable. You also may use external data and services (e.g. cloud services) as long as you can make them also accessible to the course staff. For example, you can use the free cloud credits that companies like Microsoft, Google, and AWS provide to students or reserve Chameleon credits for this project. Whenever you set up tools or services, pay some attention to configuring them with reasonable security measures; past student solutions have been actively exploited in past projects and this can lead to data loss or loss of internet access for your virtual machine.

Documentation For all milestones, we ask you to discuss some aspects of your design decision and implementation. It may be a good idea to write general documentation that is useful for the team in a place that is shared and accessible to the team (e.g., README.md or wiki pages on GitHub). Conversely it may be a good idea to include text or figures you write for reports as part of the project documentation. Feel free to link to more detailed documentation from your report or simply copy material from existing documentation into the report.

In general, we do not much care about the format or location of where we can find parts of your answers, such as code and screenshots, as long as we can find it. Please make an effort to be clear where to find content if it is not copied directly into the report, preferably with direct links to individual files or even lines

Milestone 1

Learning Goals:

The primary task is to set up multiple pipeline configurations, where in each pipeline configuration provides some result to compare each configuration. Note we do not care about the accuracy of fidelity of the result at this point. At this point we care whether a pipe is functional or not. For example, in ID card processing, one pipeline configuration is [orientation, red channel, Tesseract]. Define the accuracy of this pipeline and compute it. Obtain the end-to-end latency of your pipeline configurations.

Tasks:

T1. Meet with your team members. For this milestone it may be helpful to work on task individually for a while, and then get together to create one working notebook that is documented. Alternatively, you may decide to code up each configuration individually and then merge your configurations into a single pipeline. In either case, decide how you divide the work (minimum: who is going to do what and by when). Share team skills and responsibilities.

T2. Setup pipeline configurations in a Jupyter notebook. This is a complex task, which will require you to connect the input datasets to the first step of the pipeline, the first step to the second step, and so on, and finally output a basic accuracy number for the pipeline. We have provided one accuracy metric, which can be used as an assertion. It is upto you to think of other improved accuracy metrics and provide justification for their use. Once you have one single pipeline working, experiment by changing models at each step and set up as many pipeline configurations as possible.

Run each pipeline configuration and obtain pipeline latency and accuracy. Report as a table for each configuration.

T3. Write a report. In this report describe:

1. Write and submit a short report that describes the pipeline, its objective and architecture. Architecture includes description of datasets, models, and assertions used, and the resulting pipeline configurations.
2. Mention how many choices of datasets and models were used and the number of resulting pipeline configurations that are possible in your pipeline.
3. For each pipeline configuration, report the accuracy (in terms of number of assertions), latency, and whether SLA was met, as a table. (We will release the SLAs for each pipeline soon.)
4. Ensure that your notebook is well-documented starting with the name of the pipeline used, and report is detailed and understandable.
5. Briefly describe in your report how your team organizes itself. What communication channels do you use? How have you divides the work? Did you encountered any teamwork problems and what steps are you planning to take in future milestones to avoid them?

Deliverables.

Submit your Notebook which consists of code for pipeline configurations to GitHub. Provide a Dockerfile of your container including the notebook on your Team GitHub and share with us for grading. Include your report also on Github.

Grading.

This milestone is worth 10 points. 3 points for producing a notebook, 3 points for sharing a Dockerfile that is downloadable, runnable into a container, and regenerating the results reported in the document. 4 points for the report that is clear, complete, and precise.