[Edit: 07/04/2024]

INFO9023 - Project description

Spring 2024

Introduction

During the class Machine Learning Systems Design you will get to implement one large **group project**.

The goal of this course is to give you the skills required to build real world ML applications. The group project's main goal is to teach you new tools and best practices of MLOps.

You are in control of your project and will be able to make choices in terms of design and tooling.

Most importantly, find a project that excites you, maybe that even serves a real world use case. Pick a cool name and work in teams.

Practicals

- Form teams of **3 - 5 students**
- There will be **2 milestone meetings** where you can present your results
- The topics covered in lectures and in labs are split in 5 sprints. The components to be implemented during the group project **follow the same 5 sprints.** We recommend following the pace of the sprints (so implement the components in the same sprint), but as long as you have them implemented for the **milestone meetings** it's fine.
- Each sprint has several work packages.
- Note that all work packages are not mandatory. Some components are just optional. You can achieve a perfect score without them if you show great quality in the required work packages. But the optional ones can help you increase your score if not.
- There will be some time after each class to sit down with your team and make progress on your projects. The teaching staff will be in the same room and can provide support.

Handovers

You will get to show your work in two ways:

1. Milestone presentations

There will be 2 milestone presentations where you will get to present your work.

- The practicals for the project are **updated in the Github repo**. Please use that as the central place for organization information concerning the project.

MS	Topics	Sprints
1	Present your general *use case*, the *data preparation* and the result from your model *experimentation*.	Sprint 1 & 2
2	Present your overall project work. You can present a demo of your model/use case and any other topic you think is relevant. **This MS presentation will be in person and all students are welcomed to join and watch other project results.**	All sprints

2. Code submissions

The way to share your implementation will be through **code reviews**.

- Make sure to share your github repo with the teaching staffs
- Once done with a building block create a **pull request** from the develop branch to the master branch and include the teaching staff as reviewers
- Your team can arrange going from feature to develop branch as you prefer
- The **README** and other markdown files will be used as the main documentation points. All important aspects of your project should be explained there.
    - We recommend using the main README to explain the project structure and have different markdown files per building blocks (e.g. EXPERIMENTATION.md, DEPLOYMENT.md, …).
- The teaching staff won't run the codes, so make sure to heavily document all important points in markdown files.
- Once ready, send a link to the pull request to the teaching staff **by email**. Tagging the teaching staff in Github is not enough.

Building blocks

This section covers the components you should implement for your project. The structure follows the overall course sprints.

Sprint 1: Project organization

This sprint will focus on overall project setup and organization. It mostly covers functional aspects of a project as well as collaboration and communication tools.
Note that for each sprint some concepts are explained in the 2nd week. The W*X* next to a task indicates in which week we'll cover that topic in lecture/lab.

#	Week	Work package	Requirement
1.1	W01	Pick a **team** ● Try to mix skills and experience ● If you didn't find one let one of the teachers know and we'll allocate you to one	Required
1.2	W01	Setup **communication channel** ● Discord ● Trello board (optional) ● …	Required
1.3	W01	**Select** a use case Source options ● Previous course ● https://www.kaggle.com/datasets ● … Make sure to pick a use case where **data is available**. Note that the ML modeling itself won't be a big part of the course. If you can pick data from one of your previous course projects that's perfect. Ideally pick something with nice data and a useful/interesting usage.	Required
1.4	W01	**Define** your use case ● Fill in a ML Canvas template page ● See https://madewithml.com/courses/mlops/product-design ● **Attention**: You can skip the *Inference* part as we will tackle that in a later sprint.	Required
1.5	W02	Setup a **code** versioning **repository** ● We recommend Github as we will cover Github Actions during sprint 5	Required
1.6	W02	Find a cool name for your project ✨	Required

Sprint 2: Model development

The goal of this sprint is to prepare your data and train and optimize your ML model.

Note that you will not be graded on the performance of your model, only on your development methodology.
So do **not spend much time optimizing your data or model**.

#	Week	Work package	Requirement
2.1	W03	Prepare your data and run an Exploratory Data Analysis.	Required
2.2	W04	Train your ML model	Required
2.3	W04	Evaluate your ML model	Required
2.4	W04	Use Weights & Biases for step 2.2 - 2.3 (see Lab)	Optional

Sprint 3: Model deployment

This sprint focuses on building a **model serving API**, a **Docker container** hosting the model serving and **deploying** your model serving API.

On top of that, *if you are interested*, you can experiment with managed services such as Sagemaker Predict or Vertex Predictions.

#	Week	Work package	Requirement
3.1	W05	Build an **API** to **serve your model** and any extra logic that is needed to serve it (e.g. using Flask). You should be able to run the API locally.	Required
3.2	W05	Package your model serving API in a **Docker container**. This too should be run locally.	Required
3.3	W06	**Deploy** your model serving API in the Cloud. You should be able to call your model to generate new predictions from another machine. **Attention**: This can incur Cloud **costs**. Make sure to use a platform where you have credits and not burn through them. You can ask for support from the teaching staff in that regard.	Required

Sprint 4: Model pipeline

Here you have the opportunity to finish the work of the previous sprint. You can also *optionally* implement two other things (not required).
I*f you are interested*, you can run your model training in the Cloud. This allows you to **automate** your model training and to use **stronger compute**.
If you decide to do so, you have multiple options:

- Containerise your training script and run it on a VM in the Cloud
- Use a managed service such as Vertex Training or Sagemaker Training

On top of that, you also have the option to run the training and potentially the deployment as an **orchestrated pipeline**. Many tools exist to package more complex training as a multi-step pipeline (e.g. Kubeflow Pipelines, Docker Compose, AWS Sagemaker, GCP Vertex.).

#	Week	Work package	Requirement
4.1	W07	Package your **model training script** in a **Docker container**. You should be able to run it locally.	Optional
4.2	W07	Run your model training as a job in the Cloud. You can implement this in different ways: ● Containerise your training script and run it on a VM in the Cloud (e.g. on EC2 or on Cloud Run, example,) ● Use a managed service such as Vertex Training or Sagemaker Training **Attention**: This can incur Cloud **costs**. Make sure to use a platform where you have credits and not burn through them. You can ask for support from the teaching staff in that regard. **Note:** This is really **optional** and only do it for your self learning if you are interested in this tech. The teaching team is there to support you if you get started with this and have any questions.	Optional
4.3	W08	Build a **pipeline** to automatically run different sequential components such as training your model and deploying your model. For it you can use orchestrated pipeline tools such as Kubeflow Pipelines, Docker Compose, AWS Sagemaker, GCP Vertex. **Attention**: This can incur Cloud **costs**. Make sure to use a platform where you have credits and not burn through them. You can ask for support from the teaching staff in that regard. **Note:** This is really **optional** and only do it for your self learning if you are interested in this tech. The teaching team is there to support you if you get started with this and have any questions.	Optional

Sprint 5: Monitoring & CICD

This sprint focuses on two things: CICD and monitoring. Note that as part of monitoring we will talk about dashboarding as a whole.
Note that only the CICD part is mandatory. Even then, you can decide which components are relevant.

#	Week	Work package	Requirement
5.1	W09	Build a dashboard that runs either locally on in the Cloud to show your results	Optional

5.2	W10	Build a CICD pipeline using Github Actions (or other tool) to automatically run some of the following steps. Include at least one step. The rest is optional. Up to you to decide what is relevant.	Required
5.3	W10	Include step in CICD: Automatically launch model training pipeline	Optional
5.4	W10	Include step in CICD: Automatically launch model deployment	Optional
5.5	W10	Include step in CICD: Pylint	Optional
5.6	W10	Include step in CICD: Pytest for any unit test you think is relevant	Optional