Title

Delivery: (initial or final)

Max 15 pages for initial report, 30 for the final report

▪Name of the team/project

▪GitHub repository

▪Others resources we need to access (e.g., Taiga, Trello)

▪Contact mail (or whatever means) for the whole team

|  |  |  |  |
| --- | --- | --- | --- |
| Surname | Name | UPC email | GitHub account |
|  |  |  |  |

<Index>

# Introduction

## Goal of the project

Brief description of what the project is about and what the problem is to be solved. It must include the success criteria of the project.

## Teammates’ evaluation

Option 1: A complete team consensus on the following statement:

**“All team members agree that they had an equal contribution to this delivery and project: completing a fair share of the team's work with acceptable/high quality, keeping commitments, and completing assignments on time, helping teammates who are having difficulty when it is easy or important”.**

Option 2: There is no consensus on the above statement. Each team member evaluates (see scale 1 to 5 below) his/her peers’ contributions in each delivery according to their contribution, based on CATME ratings.

Figure 1. CATME ratings (from <https://info.catme.org/features/catme-five-dimensions/>)

|  |  |
| --- | --- |
| Rating | Description of Rating |
| 5 | Does more or higher-quality work than expected.  Makes important contributions that improve the team's work.  Helps teammates who are having difficulty completing their work. |
| 4 | Demonstrates behaviors described immediately above and below. |
| 3 | Completes a fair share of the team's work with acceptable quality.  Keeps commitments and completes assignments on time.  Helps teammates who are having difficulty when it is easy or important. |
| 2 | Demonstrates behaviors described immediately above and below. |
| 1 | Does not do a fair share of the team's work. Delivers sloppy or incomplete work. |

**Fill in the following table (received evaluations in rows):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Name 1 | Name 2 | Name 3 | Name 4 | Name 5 | Average |
| Name 1 | - |  |  |  |  |  |
| Name 2 |  | - |  |  |  |  |
| Name 3 |  |  | - |  |  |  |
| Name 4 |  |  |  | - |  |  |
| Name 5 |  |  |  |  | - |  |

# Methodology

## Milestone 1: Inception

### Selection of problem and requirements engineering for ML

How, and why has the problem been selected, and what are the system requirements?

### Dataset card

Brief description of the main points of the dataset card, and link to the file in the repository.

### Model card

Brief description of the main points of the model card, and link to the file in the repository.

### Project coordination and communication

What tools have been selected, why, and how are they used?

### Selection of the cloud provider

Justification of the selection (to be used in Milestone 4)

## Milestone 2: Model Building – Reproducibility

Describe how the different software engineering practices and tools have been applied.

### Project structure

(e.g., Cookiecutter)

### Code versioning

(e.g., GitHub Flow)

### Data versioning

(e.g., DVC)

### Experiment tracking

(e.g., MLflow)

## Milestone 3: Model Building – Quality Assurance

Describe how the different software engineering practices and tools have been applied.

### Energy efficiency awareness

(e.g., Codecarbon) Reason about the energy efficiency of your ML component.

### Static code analysis

(e.g., pylint)

### Model testing

(e.g., pytest)

### Data testing

(e.g., Great Expectation, Deepchecks)

## Milestone 4: Model deployment – API

Describe how you have deployed your model and the technologies used.

### ML-based component/system architecture

Description of the physical and logical architectures of the system, including diagrams. Design patterns used. Selection and justification of design decisions (e.g., cloud provider)

### API design

(e.g., FastAPI)

### API testing

(e.g., FastAPI, pytest)

## Milestone 5: Model deployment – Model packaging

Describe how the different software engineering practices and tools have been applied.

### Container and Orchestration

(e.g., Docker, Docker compose)

### CI/CD

(e.g., GitHub Actions)

## Milestone 5: Monitoring

Describe how the different software engineering practices and tools have been applied.

### Resource monitoring

(e.g., Grafana, Prometheus)

### Model performance

(e.g., Alibi detect)

### Cycle: Feedback loops and retraining

Description of all the stages of the system pipeline, and how they are connected. Explain how the pipeline can be used to automate the retraining process.

# Self-evaluation of the project (retrospective)

To do in the final deliverable

Explain the main challenges, barriers, and opportunities you encountered during the project. Describe what you have learned, what you would do differently, etc.

Explain what concepts you have used from other courses.