Laboratory. Third semester

Machine Learning Systems in Production (MLOps)

Silverio Martínez-Fernández, Santiago del Rey, Matias Martinez

@silveriomf

Some parts of this lecture are adopted with permission from lectures given by Prof. Filippo Lanubile at University of Bari (Italy)

Outline

- A short introduction to MLOps
- Motivation and Purpose
- Organization
- Schedule
  - Detailed weekly schedule
- Evaluation

Short introduction – Learning Objectives

1. Interpret the basic concepts of Software Engineering for ML systems, especially in relation to the use and exploitation of MLOps practices.
2. Apply and analyze MLOps practices to build ML models, fostering reproducibility and quality assurance.
3. Apply and analyze MLOps practices to deploy ML models, fostering API development and component delivery.
4. Describe concepts and methods related to monitoring data obtained during the use of ML systems, in order to enable feedback loops in response to changes.

Software Engineering for ML (1/3)

Goal: to teach MLOps practices and provide hands-on experience with MLOps tools

Milestone 1

Project inception

- **ML problem spec**
  - Model cards
  - Dataset cards
- **Project coordination and communication**

Milestone 2

Model building: *Reproducibility*

- **Project structure**
  - Cookiecutter DS
- **Versioning**
- **Experiment Tracking**

Filippo Lanubile, Silverio Martínez-Fernández, Luigi Quaranta: Teaching MLOps in Higher Education through Project-Based Learning. SEET@ICSE (2023)

Filippo Lanubile, Silverio Martínez-Fernández, Luigi Quaranta: Training future ML engineers: a project-based course on MLOps. IEEE Software, in press

Software Engineering for ML (2/3)

Goal: to teach MLOps practices and provide hands-on experience with MLOps tools

Milestone 3

Model building:
Quality Assurance

- **Energy efficiency**
- **Static analysis**
- **Testing**

Milestone 4

Model Deployment:
API Development

- **Physical Architecture**
- **API Development**

Filippo Lanubile, Silverio Martínez-Fernández, Luigi Quaranta: Teaching MLOps in Higher Education through Project-Based Learning. SEET@ICSE (2023)

Filippo Lanubile, Silverio Martínez-Fernández, Luigi Quaranta: Training future ML engineers: a project-based course on MLOps. IEEE Software, in press

Software Engineering for ML (3/3)

Goal: to teach MLOps practices and provide hands-on experience with MLOps tools

Milestone 5

Model Deployment

Component Delivery

- **Containerization**

- **CI/CD for ML**

Milestone 6

Monitoring

- **Resource monitoring**

- **Model performance monitoring**
  - Alibi Detect

Filippo Lanubile, Silverio Martínez-Fernández, Luigi Quaranta: Teaching MLOps in Higher Education through Project-Based Learning. SEET@ICSE (2023)

Filippo Lanubile, Silverio Martínez-Fernández, Luigi Quaranta: Training future ML engineers: a project-based course on MLOps. IEEE Software, in press

Purpose

- Discussing and complementing the content of the lecture
- Gathering experience in software engineering and software quality for data science and ML projects
- Learning to:
  - build ML models components following software engineering practices
  - deploy ML models components following software engineering practices

Organization – Overview

- Students will build **teams of approximately 5 members**
- During the semester, each team will be responsible for taking part in a **data science project**, discussing the data analysis, documenting and presenting their results:
  - The team will present a report, written in English, summarizing the main aspects of the practice, for example, the process of building an ML component of an ML-based system, and an evaluation of the accuracy of the models and algorithms used.
  - The resulting software, duly documented, will be uploaded to a repository.

Organization – Weekly milestones and feedback

- Teams are expected to:
  - Solve the assigned tasks planned for each week.
  - Document their results according to a predefined template.
  - Present their solutions during the exercise class.
- Students and lecturer will discuss the proposed solutions together in the exercise class.
- Teams are expected to enhance their solutions based on the previous discussions.

Organization – Project goal

- Each team has to
  build and deploy
  an ML component
  following software engineering best practices.

- Each team is free to choose their own ML component goal (either eliciting their own ML component or defining one from the initial ideas):
  - Tip: think in the ML component deployed and used in operation
  - In the same group, teams cannot repeat the very same goal (recommended to use a unique dataset)
  - First-in, first-out!

Organization – Goal and datasets suggestions

- Computer vision / image classification. Datasets:
  - German Traffic Sign Recognition Benchmark (GTSRB)
  - MNIST
  - Cifar-10
- Natural language processing / text classification: Datasets:
  - Sentiment140
  - Amazon Reviews
  - IMDB Movie Reviews
- Code generation, summarization, etc. Datasets:
  - HumanEval
  - GitHub Code
  - rStar-Coder
- Or suggest to the lecturer:
  - Another ML goal for your chosen ML application (examples of ML applications)
  - Another dataset, for instance from Kaggle or Hugging Face
  - Another dataset or ML model in which you have previously worked

Analyzing the Evolution and Maintenance of ML Models on Hugging Face

Joel Castaño
Universitat Politècnica de Catalunya
joel.castano@upc.edu

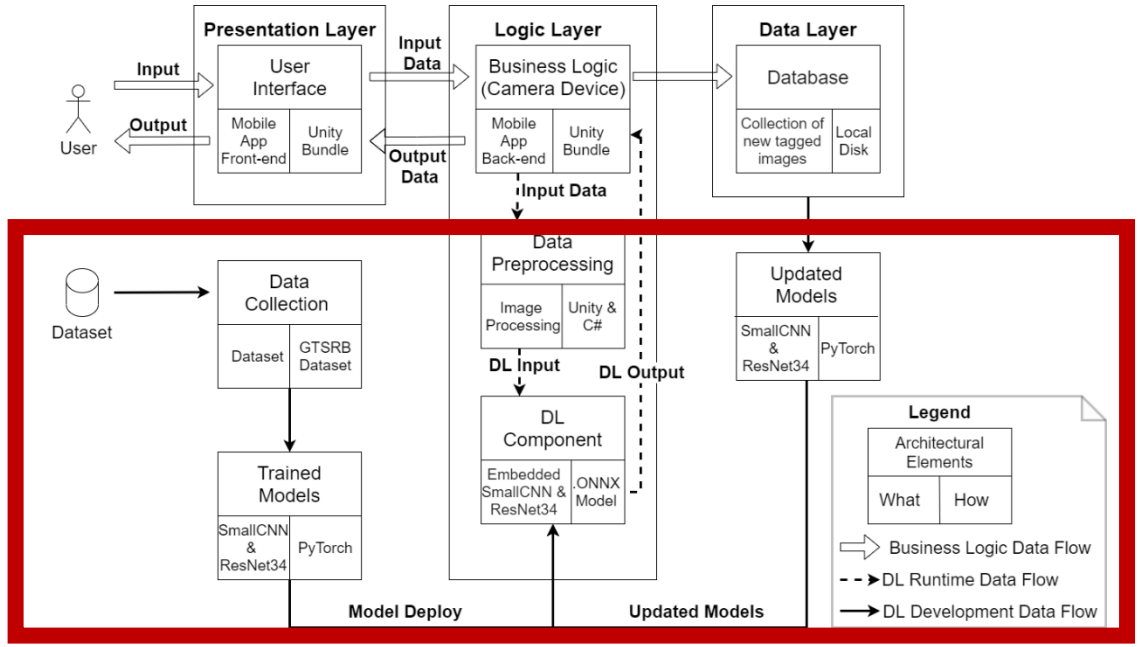
Silverio Martínez-Fernández
Universitat Politècnica de Catalunya
silverio.martinez@upc.edu

Xavier Franch
Universitat Politècnica de Catalunya
xavier.franch@upc.edu

Justus Bogner
Vrije Universiteit Amsterdam
j.bogner@vu.nl

Organization – Scope

- We focus on quality aspects of building the ML model (e.g., reproducibility, quality assurance) and deploying it as an ML component (e.g., API), rather than a whole ML-based system
  - Example:



Roger Creus Castanyer, Silverio Martínez-Fernández, Xavier Franch: Integration of Convolutional Neural Networks in Mobile Applications. WAIN@ICSE 2021: 27-34

Organization – Documentation (1/3)

- Main report: a single pdf file with all the information
  - To be uploaded in Atenea
  - Template in Atenea
- A cover page clearly stating:
  - Name of the team/project
  - Which deliverable (initial or final)
  - Links to resources:
    - GitHub repository
    - Others resources we need to access (e.g., Taiga, Trello)
  - Contact mail (or whatever means) for the whole team
- Team members' description:

Surname	Name	UPC e-mail	GitHub account
Fuertes	Dolores	…	…
Lotas	Felipe	…	…
…			

Organization – Documentation (2/3)

1. Introduction
   - Goal of the project (with success criteria)
   - Teammates' evaluation (by default, all happy)
2. Methodology
   - A subsection with an individual description for each software engineering practice

Organization – Documentation (3/3)

3. Self-evaluation of the project (retrospective)
   - main challenges, barriers, opportunities, …
   - what have you learned, what would you do differently, …
   - what concepts have you used from which courses

Very important:

- Maximum of 15 pages for the initial report, and 30 pages for the final report.
- Update previous sections if necessary (e.g., model card).
- Check the slides 21-32 for SE practices for section 2.

Organization – Software and Replication package

- The resulting **software**, duly structured, will be **uploaded to a repository (preferably a link inside to a public repository inside our GitHub organization)**
  - https://github.com/mlops-2526q1-mds-upc
  - Contact: santiago.del.rey@upc.edu and matias.martinez@upc.edu
  - Repo. name: MLOps_<name of the team>
- To foster **correctness** and **reproducibility**, it has to follow this project structure (justified deviations allowed!): https://drivendata.github.io/cookiecutter-data-science
  - Collaborate more easily with you on this analysis
  - Learn from your analysis about the process and the domain
  - Feel confident in the conclusions at which the analysis arrives

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── setup.py           <- Make this project pip installable with `pip install -e`
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.testrun.org
```

Organization – Communication

- General questions about the documentation or technical issues (git, dvc errors, configuration of the VM, etc.)
  - **Use the forum "FAQ"** in Atenea.
  - Check if your question has already been discussed. If not, create a new discussion for your question.
- Team-specific doubts or any other business
  - Send an email to the professor.
  - Every email sent about MLOps, should have as prefix: "[MLOps] *<team name>*" (square brackets included)
  - Kindly provide some mechanism to your teacher so that s/he can send an email and all the members of the team receive this email
- Every document or repository should have as the prefix MLOps_<team name> and then a self-explanatory name (e.g., Initial report, final report, etc.)
- Inform your teachers about any team problem you may experience as soon as possible

Schedule – Laboratory

Session	Assignments / Outputs for each laboratory session	Tools	Date	Delivery (one day before the session)
1	Milestone 1 – Project kick-off and inception.	GitHub repository creation (for model and dataset cards)	Sep. 9	
2	Milestone 2a – Model building: reproducibility	Git with GitHub Flow, DVC	Sep. 16	
3	Milestone 2b – Model building: reproducibility	MLFlow	Sep. 23	
4	Milestone 3a – Model building: quality assurance	Pylint, Pytest, Great Expectations	Sep. 30	
5	Milestone 3b – Model building: quality assurance		Oct. 7	
6	Presentation Milestones 1–3		Oct. 14	**1st (Milestones 1–3)**

Schedule – Laboratory

Session	Assignments / Outputs for each laboratory session	Tools	Date	Delivery (one day before the session)
7	Milestone 4a – Model deployment: API	ML serving (Virtech, AWS,…)	Oct. 21	
8	Milestone 4b – Model deployment: API	Fast API	Oct. 28	
-	There is no class		Nov. 4	
9	Milestone 5a – Model deployment: Model packaging	Docker and Docker Compose	Nov. 11	
10	Milestone 5b – Model deployment: Model packaging	GitHub Actions	Nov. 18	
11	Milestone 6a – Monitoring	Better Uptime, Prometheus, and Grafana, Alibi detect	Nov. 25	
12	Milestone 6b – Monitoring		Dec. 2	
13	Presentation Milestones 4–6		Dec. 9	**2nd (Milestones 4–6)**
14	Review Milestones 4–6		Dec. 16	

Schedule – Team

- The team:
  - Identified by a name chosen by students
  - Ideally 5 members
    - One member of the team sends an e-mail to the lecturers, with cc the whole team
      - Subject: [MLOps] <name of the team>
      - Content:
        - Name, surname, **GitHub username**, and **UPC e-mail** of each member
        - Proposal of ML component to develop and dataset
          (it can be refined until the 2nd week)
      - **Deadline: Team created in today's laboratory session!!**

Schedule –Milestone 1

- Milestone 1: Inception
  - Each team chooses and downloads a dataset for the ML component
  - Each team defines a dataset card and model card about their ML component
  - Each team creates its collaborative working space (e.g., GitHub repository, installing Jupyter Notebook, a discord server…)

Practice	Tool(s)
MILESTONE 1: INCEPTION	
Selection of problem and requirements engineering for ML	Model and dataset cards (by Hugging Face)
Project coordination and communication	Taiga, Trello, slack

Milestone 1 – evaluation criteria

- Goal Definition and Requirements Engineering
  - How clearly does your project definition and requirements show what makes your system unique and complete?
  - How well does your dataset and model documentation communicate the choices you made and their implications for others?
- Project Coordination and Communication
  - How effectively do your coordination and communication practices support collaboration and leave a useful trace of your progress?

Schedule –Milestone 2

- Milestone 2 - Model building: reproducibility
  - Each team applies practices for reproducibility of the project:
    - Project structure
    - Code versioning
    - Data versioning
    - Experiment tracking

Practice	Tool(s)
MILESTONE 2 – MODEL BUILDING: reproducibility (2 weeks)	
Project structure	Cookiecutter data science template
Code and data versioning	Git with GitHub Flow, DVC
Experiment tracking	MLflow

Milestone 2 – evaluation criteria

- Project Structure
  - How well does your project structure support clarity, maintainability, and adaptation to your project's needs?
- Code and Data Versioning
  - How do your version control practices ensure collaboration is smooth and mistakes are minimized?
  - How effectively does your data versioning strategy guarantee reproducibility and scalability for your project?
- Experiment Tracking
  - How does your experiment tracking help you and others compare, understand, and extend your results?

Schedule – Milestone 3

- Milestone 3 – Model building: quality assurance
  - Each team applies practices for quality assurance of the project:
    - Reporting CO_2 emissions
    - Static analysis (e.g., Pynblint linter)
    - Testing model
    - Testing data

Practice	Tool(s)
MILESTONE 3 – MODEL BUILDING: QA (2 weeks)	
Energy efficiency awareness	Code Carbon
Quality assurance for ML (static analysis + testing data and model)	Pynblint (notebook + repository QA), Pylint or flake8, Pytest and Great Expectations

Milestone 3 – evaluation criteria

- Energy Efficiency Awareness
  - How do your energy tracking practices help you reflect on the efficiency and sustainability of your choices?
- Quality Assurance for ML
  - To what degree do your use of linting contribute to the overall quality and consistency of your codebase?
  - To what extent do your tests give confidence that your code and data behave as expected under different conditions?
  - How effectively does your data validation strategy anticipate and prevent potential issues in your pipeline?

Schedule – Milestone 4

- Milestone 4 - Model deployment: API
  - Select the cloud provider where you will deploy your ML component
    - A virtual machine from FIB is available for each team if necessary
  - Each team designs the deployment of the model as an ML component
  - The ML component is available via an API
  - The ML component is tested

Practice	Tool(s)
MILESTONE 4 – MODEL DEPLOYMENT: API (2 weeks)	
ML system design	Cloud platform selected by the students (VMs, Heroku, DigitalOcean, etc.)
APIs for ML	FastAPI Pytest (to test the API endpoints)

Milestone 4 – evaluation criteria

- ML System Design
  - In what ways does your system design balance clarity and justified complexity while applying suitable architectural patterns and principles?
- APIs for ML
  - To what extent does your API design, together with its documentation, make the system robust, intuitive, and easy to adopt by others?

Schedule – Milestone 5

- Milestone 5 - Model deployment (component delivery):
  - Each team creates a component ready to be delivered and deployed
  - Each team defines a process for automatizing the ML pipeline

Practice	Tool(s)
MILESTONE 5 – MODEL DEPLOYMENT: components (2 weeks)	
Container and Orchestration	Docker
CI/CD for ML	Github Actions

Milestone 5 – evaluation criteria

- Container and Orchestration
  - In what ways does your containerization approach guarantee portability, modularity, consistency, and efficiency across environments?
- CI/CD for ML
  - What aspects of your CI/CD pipeline demonstrate that automation and collaboration are fully integrated into your process?

Schedule – Milestone 6

- Milestone 6: Monitoring of deployed ML components:
  - Each time has to design and implement solutions for monitoring:
    - Resources
    - Model performance

Practice	Tool(s)
MILESTONE 6 – Monitoring	
Resource Monitoring	Grafana, Prometheus
Model Performance	Alibi detect

Milestone 6 – evaluation criteria

- Resource Monitoring
  - To what extent does your monitoring anticipate real-world challenges and provide actionable insights into system performance?
- Model Performance
  - To what extent does your monitoring provide timely, actionable signals about data changes, model performance degradation, and user impact, enabling confident investigation and response?

Schedule – Feedback and Presentations

- Presentations after every three milestones.
- Two presentations:
  - M1-M3 presentation (15 mins). 6th week – Oct. 14
    - Focus on inception (ML component definition and success criteria), and model building: reproducibility & quality assurance
  - M5-M6 presentation (15 mins). 13th week – Dec. 9
    - Focus on model deployment: API & CI/CD and monitoring.

Schedule – Deliveries

- Two deliveries
  - When?
    - 1st report (Milestone 1-3): Oct. 13, 23:55.
    - 2nd report (Milestone 4-6): Dec. 8, 23:55.
  - Where?
    - Atenea
  - What?
    - Current/Final version of your team **report (keeping the maximum number of 15 pages for the first report and 30 pages for the second report)**, written in English, summarizing the main aspects of the practice.
      - See slide: Organization – Documentation
    - The resulting **software**, duly documented, is included in the cover page as a link to the code repository.
      - See slide: Organization – Software and Replication package
    - Slides of the presentations.

Lab Evaluation (1/6)

- Laboratory grade: ∑ SE practices * Individual factor
  - The sum of the software engineering practices of the next slide will be assessed based on:
    - the report
    - the replication package (the software repository)
    - the discussions in the laboratory, including presentations

Lab Evaluation: ∑ SE practices (2/6)

Practice	Tool(s)	Points (Out of 100)
MILESTONE 1: INCEPTION (2 weeks)		
Selection of problem and requirements engineering for ML	Model and dataset cards (by Hugging Face)	6
Project coordination and communication	Taiga, Trello, slack	4
MILESTONE 2 – MODEL BUILDING: reproducibility (2 weeks)		
Project structure	Cookiecutter data science template	5
Code and data versioning	Git with GitHub Flow, DVC	15
Experiment tracking	MLflow	5
MILESTONE 3 – MODEL BUILDING: QA (2 weeks)		
Energy efficiency awareness	Code Carbon	5
Quality assurance for ML (static analysis + testing data and model)	Pynblint (notebook + repository QA), Pylint or flake8, Pytest and Great Expectations	10

Lab Evaluation: ∑ SE practices (3/6)

Practice	Tool(s)	Points (Out of 100)
MILESTONE 4 – MODEL DEPLOYMENT: API (2 weeks)		
ML system design	Software architecture, cloud platform selected by the students (VMs, Heroku, DigitalOcean, etc.)	15
APIs for ML	FastAPI Pytest (to test the API endpoints)	10
MILESTONE 5 – MODEL DEPLOYMENT (2 weeks)		
Container and Orchestration	Docker	5
CI/CD for ML	GitHub Actions	10
MILESTONE 6 – MONITORING (2 weeks)		
Resource Monitoring	Grafana, Prometheus	5
Model Performance	Alibi detect	5

Lab Evaluation: ∑ SE practices (4/6)

- Each practice is evaluated according to these criteria:
  - **Poor:** the students were not able to apply the recommended MLOps practice;
  - **Fair:** the students implemented the MLOps practice by replicating the provided examples with minor changes;
  - **Good:** the students implemented the MLOps practice by replicating the provided examples with major changes;
  - **Excellent:** the students implemented the MLOps practice in an extended or innovative way

Lab Evaluation: Individual factor (5/6)

- Individual factor between 0 and 1.2 (exceptional cases)
  - Normally 1.0 for all team members. Indeed, this is a team goal!
    - The whole team has the same grade. Teachers encourage students to identify any team problems that may arise and to solve them in time!
  - The individual factor depends on your contribution to your team. Quantified by the teacher based on:
    - Peer review by the team members following the CATME rubric (see next slide)
    - Individual continuous work and contributions at the lab are seen by the professor based on:
      - the report
      - replication package (number of commits in the main branch and total lines of code changed in the software repository)
      - the discussions in the laboratory, including presentations
  - Inform both your team members and your teachers about any team problems you may experience as soon as possible and report them in the deliveries!
  - This communication shall not be seen as a punishment, but as a way to motivate team communication and enhance the ability to resolve conflicts.
  - A lower individual factor for a team member does not *automatically* imply an increase for another member, nor the other way around.
  - It is ok for a team member to work/contribute less (for personal, professional, or motivational issues) on the project, but this can affect the individual factor.

Lab Evaluation: Individual factor (6/6)

- Peer review by the team members following the CATME rubric
  - Peer review by the team members following the CATME rubric **with two deadlines (within the two deliveries)**. Two options:
    - 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 (from https://info.catme.org/features/catme-five-dimensions/). This assessment is public within the team to enhance team communication, as the final goal is to have an equal contribution to the project

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 the following table (received evaluations in rows):

	Name 1	Name 2	Name 3	Name 4	Average
Name 1	-				
Name 2		-			
Name 3			-		
Name 4				-	

Questions

- By e-mail
  - santiago.del.rey@upc.edu
  - matias.martinez@upc.edu