EXPERIMENT MANAGEMENT AND TRACKING

[1]Muhammad Kabir Hamzah, [1]Tuhinangshu Gangopadhyay, [1]Sarita Mourya,
[2]Muhammad Qasim & [2]Muhammad Momin Naveed

[1]Medical Imaging and Applications (MaIA)
[2]Medical Image Computing (MIC)

E-Health

Universitat de Girona

VICOROB

Universitat de Girona

1. Introduction

The usage of structured, scalable, and maintainable codebases in data science, particularly when shifting from an exploratory Jupyter notebook-based workflow to a more robust, production-ready system, has become an essential component of research and machine learning initiatives. This is especially important for graduate students working on thesis, as coordinating several experiments, maintaining parameters, and guaranteeing reproducibility are critical. In order to construct a production-ready system, it is critical to structure the codebase by isolating files based on the unique functions they perform. This modular approach not only keeps the code clean and understandable, but it also contributes significantly towards maintenance. When working with vast and complicated datasets, organizing the system makes it much easier to track issues, debug difficulties, and make necessary modifications. Furthermore, this structure assures that the code may be easily improved, extended, or troubleshooted in the future, adding long-term value to both current and future development efforts. Proper organization enables easy access to essential files, resulting in smoother updates and collaboration.

In our student lab, we integrated MLflow for tracking experiments and managing information related to experiment results, alongside implementing a structured codebase to enhance organization and maintainability.

2. Problem Statement

When working with huge datasets, various models, each with multiple combinations of parameters and hyperparameters, it can be difficult to determine which combination of datasets, parameter values, and model configurations produces the best results. The complexity of monitoring and analyzing several experimental setups frequently results in confusion or loss of vital information, especially when dealing with a large number of parameters and combinations. Identifying the ideal collection of parameter values and model architecture that produces superior outcomes can be difficult without a methodical approach.

To solve this issue, solutions such as MLflow [1] have been created to improve experiment tracking and management. MLflow, introduced in 2018 by Zaharia et al., is an open-source platform designed to manage the entire machine learning lifecycle. It offers a comprehensive platform for recording and organizing information about each experiment, including the datasets, parameters, and model versions utilized. It also provides tools for experiment tracking, project packaging, model versioning, and deployment, ensuring reproducibility and scalability in data science workflows. By centralizing records, MLflow enables users to rapidly compare and assess various experiments' performance indicators. This functionality not only minimizes data loss, but also allows for a clear knowledge of which parameter values produce the greatest outcomes. Furthermore, MLflow's experiment tracking function improves reproducibility by allowing researchers to revisit previous experiments and accurately duplicate their results. With its ability to save a well-structured history of trials, MLflow is invaluable for data scientists and researchers looking to optimize models and ensure consistency throughout their workflows.

Over the years, multiple research studies have been conducted to provide detailed analysis on the usefulness of MLflow in tracking experiments and its capabilities in streamlining machine learning workflows, managing model lifecycle, and ensuring reproducibility in machine learning projects [2, 3, 4, 5]. For instance, Idowu et al. provided a detailed review of experiment tracking tools, including MLflow, Weights & Biases, and Comet, emphasizing their role in enhancing reproducibility, scalability, and collaboration in data science workflows. The authors discuss how these tools effectively manage parameters, metrics, and model artifacts, addressing key challenges in organizing experiments, debugging issues, and tracking provenance. By streamlining the research-to-production pipeline, these tools not only improve workflow efficiency but also enhance transparency and traceability, which are critical for collaborative projects and regulatory compliance.

3. Brainstorming

A significant aspect of the lab was the brainstorming session, which served as a platform for generating and exploring a wide range of ideas for potential activities. We discussed different ideas until we got enough topics. Once we had compiled a substantial number of ideas, we carefully evaluated each by analyzing their potential advantages and disadvantages. This thorough evaluation allowed us to identify the most feasible and impactful activities for implementation. Below is an overview of the ideas we considered and their objectives:

- Experiment Management and Tracking: To prepare students for conducting experiments in structured, GPU-enabled environments, commonly used in professional research.
- Diffusion Models for Medical Image Generation: Introduce Students to Diffusion Models, which is a state-of-the-art methodology For Medical Image Generation And Reconstruction Tasks In Medical Imaging
- An interactive platform for medical image segmentation of different body parts: To provide an interactive session where students can experiment with segmenting different body parts in medical images, focusing on quick training, testing, and preprocessing techniques.

3.1 Chosen Idea

After discussing various ideas for the lab sessions, we collectively decided to focus on **Experiment Management and Tracking**. This choice stood out for its practical relevance and the significant learning opportunities it offers. Our goal is to provide hands-on experience in structuring a codebase effectively and to demonstrate how MLflow facilitates streamlined experiment tracking, ensuring easier debugging and future modifications.

This idea was chosen for its ability to address critical challenges in iterative development and research. A systematic approach to managing experiments minimizes confusion, reduces redundancy, and fosters collaboration among team members. Additionally, it introduces participants to best practices in software development, promoting organized workflows and effective use of industry-standard tools.

While other ideas had their strengths, they lacked the practicality and immediate applicability. Experiment management and tracking emerged as the ideal choice, combining technical learning with real-world utility, making it a valuable focus for our lab sessions.

4. First Feedback: Five minutes presentation

After the first presentation, we received very positive and suggestive feedbacks, which is important for modifying, adjusting and refining the idea. We read through all feedbacks and accordingly adjusted the ideas for improving the lab session in order to progress in the development of the lab. Some of the important feedbacks suggested:

- Emphasizing the importance of using Linux and how is it better than windows.
- Adding an extra layer of engagement by integrating a gamified component to make it more interactive and enjoyable.
- Adding some visual examples or simple diagrams to clarify how the setup and tracking works.
- Adding small collaborative challenges or peer reviews to simulate team-based research settings, and maybe adding an element where students present and analyze each other's results for an added layer of learning.
- Making it smoother, prepare a detailed notebook and share the tools/libraries participants need to download beforehand.

5. Lab Planning

A significant aspect of the lab was the planning session, which served as a platform for generating and exploring a wide range of ideas for potential activities to incorporate in order to make it interactive as well as intuitive. This creative process enabled us to come up with diverse and innovative concepts to enhance the learning experience in our lab. Some of the activities we have added in while planning are:

- Storyline development for the presentation: We designed the presentation to clearly convey the importance of managing experiments in machine learning. The slides were structured to explain why it is crucial to keep track of experiments, the challenges involved, and the solutions available for effective experiment tracking. The presentation also highlighted the role of tools that log experiments automatically, emphasizing their impact on improving the efficiency and reproducibility of machine learning workflows.
- Interactive Trivia Challenges: One of the ideas was to introduce trivia challenges directly related to the codebase. This activity would not only be interactive and engaging but also help participants better understand how the code is organized and structured. By incorporating questions about the logic, structure, and functionality of the codebase, participants could enhance their comprehension and reinforce their coding skills in a fun and collaborative manner.
- Skin Lesion Classification Dataset for Experiments: Another idea involved leveraging the available dataset for skin lesion classification. This dataset could be utilized for conducting experiments that focus on applying machine learning techniques to medical image analysis.

By working with real-world datasets, participants could gain valuable hands-on experience in solving practical problems, further enhancing their understanding of data preprocessing, model building, and evaluation.

- Local Server Setup for MLflow: To streamline experiment tracking, we proposed setting up a local server for MLflow. This approach would ensure accessibility for everyone using their personal laptops, making it easy to track experiments and manage results efficiently. A locally hosted MLflow server offers simplicity, flexibility, and cost-effectiveness while maintaining all essential functionalities for logging parameters, metrics, and models.

Through this structured brainstorming process, we aimed to identify activities that were not only feasible but also impactful, providing participants with practical insights into machine learning workflows, experiment tracking, and codebase management. These ideas align with the goals of fostering interactive learning and equipping participants with the tools and skills needed for real-world applications. Our planning phase is summarized with the Gantt Chart in Fig 1.

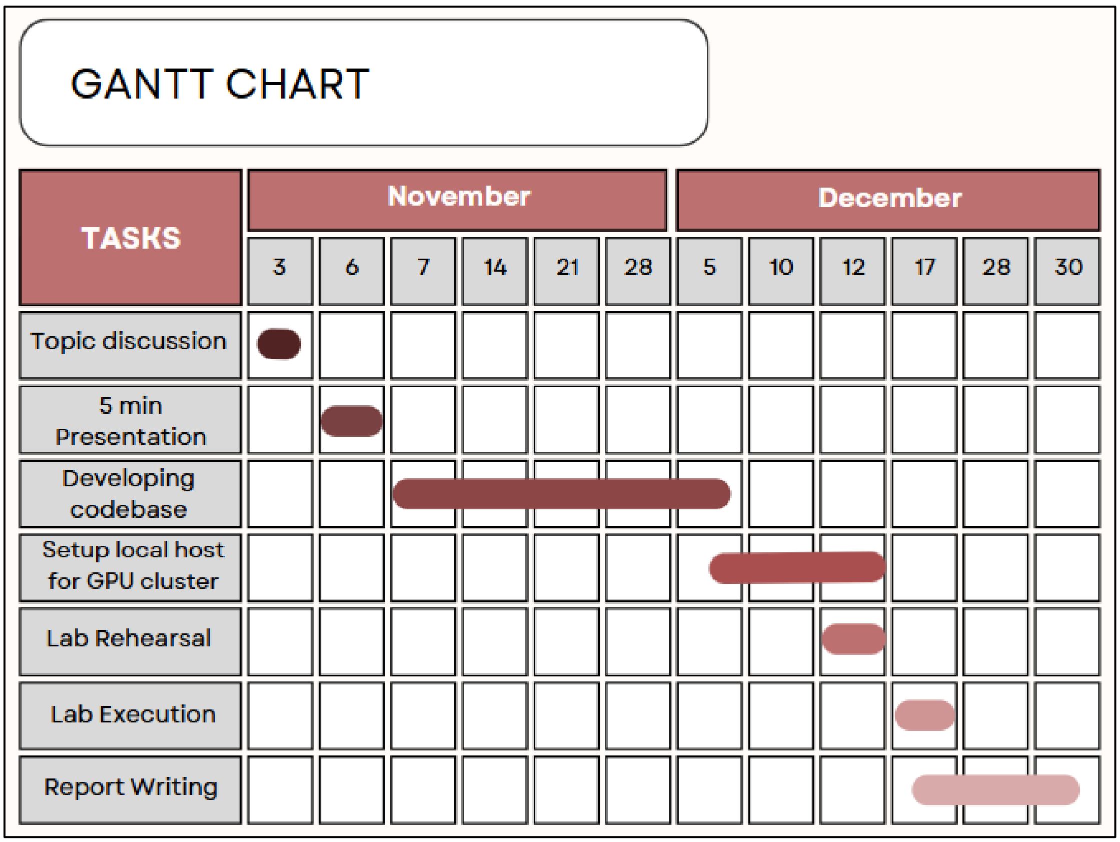

Fig. 1: Gantt Chart presenting our lab progress

6. Lab Implementation

6.1 Preparation

This section provides a clear summary of the lab preparation, outlining the tools we utilized and the main components of each part of the activity.

6.1.1 Tools

To develop and design the experiments and lab sessions, we utilized a range of tools that played a crucial role in creating an engaging and interactive learning experience. Our primary goal was to make the lab sessions enjoyable and educational, allowing participants to learn in a game-like manner. Below is a comprehensive overview of the tools we used:

- **Python:** Used as the core programming language for developing a structured and maintainable codebase for the experiments.
- **VS Code:** Employed with a local Python virtual environment to install and manage all required libraries, extensions, and dependencies necessary for running the experiments. VS Code also facilitated easy configuration and debugging.
- **Git and GitHub:** Used for collaborative development and version control. GitHub allowed team members to share progress, review changes, and work together seamlessly on designing and developing the codebase.
- **Screen Recording Tools:** To manage time effectively during the final lab sessions, we created explanatory short videos using screen recording tools. These videos provided clear demonstrations of the key concepts and steps, ensuring a smoother lab experience.
- **Experiment Tracking and Logging:** MLflow: This tool was chosen for tracking and logging experiments, providing a streamlined way to monitor parameters, metrics, and results.
- **Dependency Management with pipreqs:** Used to handle dependencies and generate a `requirements.txt` file, ensuring that all necessary libraries and packages were accounted for and could be easily installed.
- **Code Refactoring with ruff:** A tool utilized for code refactoring, improving the readability and maintainability of the codebase.
- **Configuration Management with Omegaconf and Hydra:** These tools were employed to manage configuration files efficiently. They provided flexibility in handling different experimental setups and parameter tuning.
- **Model Development and Evaluation with PyTorch, TensorFlow, and scikit-learn:** Used for training, fine-tuning, and evaluating machine learning models. These frameworks enabled the implementation of robust and scalable models for our experiments.
- **Interactive Quizzes with Kahoot:** Used to conduct quizzes during the lab sessions. This tool added an interactive and fun element, encouraging active participation and reinforcing key concepts.

By leveraging these tools, we were able to create a well-structured and efficient workflow for our lab sessions, ensuring an engaging and educational experience for all participants.

6.1.2 Setting up the Questionaries for the Quiz

One of the most challenging tasks was designing the right questions for the quiz to ensure that the participants go through the code files and become familiar with how the code is organized. The goal was to help everyone understand how a well-structured codebase improves code-readability and thus, simplify the debugging and modifications.

To create interactive and meaningful questions, we thoroughly reviewed all the configuration files, data loader files, and training and trainer scripts that we had developed for the experiments. We crafted the questions to focus on key elements like specific keywords or hyperparameter values, and then encouraged the participants to explore the code and learn how to change any parameter value that could help them run a totally different experiment and save these changes, which can later be helpful in retrieving the values that returned the best performance for the machine learning experiments.

To make the quiz even more engaging, we locked access to certain files, which could only be unlocked using indices derived from the quiz answers. This approach added an exciting and interactive element while reinforcing the importance of understanding and navigating the codebase effectively.

6.2 Plan Execution

To execute the lab session successfully, we collected ideas from all the group members, keeping in mind the time constraints and available resources. We accordingly planned everything to ensure the lab could be completed appropriately. Table 1 shows the time constraints we put for each component of our student lab.

Table 1: Time distribution for the student lab

Activity	**Time (min)**
Introduction	5
Solving question and running the experiment	30
Video on using Linux GPU server	5
Quiz on Kahoot	5
Conclusion	5
Total	50

On the day of the lab session, we divided the class into six groups, providing each group with a laptop preloaded with the codebase, required libraries, dependencies, and an instruction sheet to guide them through the code and experiments (Fig 2).

We began with a brief 5-minute introduction using a storyline presented through slides. This presentation highlighted the importance of structured codebases and experiment tracking in machine learning. After the introduction, the team members spread around the class in order to supervise the progress of the groups and also to assist them in case they faced any difficulty.

This file contains an information guide about the tasks that will be carried out in this lab. The image below contains the structure of the code/repository. Take a moment to understand what each file or folder does.

Repository Structure

```
project/
├── data/              # Directory for storing datasets.
├── notebooks/         # Jupyter notebooks for experimentation and exploration.
├── src/               # Source folder for all Python scripts.
│   ├── data.py        # Script for data loading and processing.
│   ├── train.py       # Script for training models (could be called main)
│   ├── trainer.py     # Script containing the custom training functions.
│   ├── models.py      # Script containing the model architectures.
│   └── utils.py       # Utility script and helper functions.
├── requirements.txt   # Project dependencies.
├── configs/           # Configurations for model parameters, data paths, and experiments.
└── README.md          # Project documentation (this file).
```

Explore the cloned repository; you will find the configuration folder (configs) that contains the files responsible for model and training parameters. In this exercise, we will investigate what each file or parameter does by answering some questions.

Key things to note:

Running an experiment is as easy as running the command ``` python train.py```. This simply means that we want to run the experiment with all the default configurations. It is the same as writing ``` python train.py model=default train=default mlflow=default```.

Part 1: Models and Data
Remember to mark your answers, you will need the indices to unlock the zipped files.

A. Assume that we want to run an experiment using the default model and mlflow parameters but with some modifications to training parameters e.g. number of epochs. Which of these would be appropriate?
   1. python train.py model=default train.epochs=50
   2. python train.py train=default model.epochs=50
   3. python train.py model=default config.epoch=50
   4. python train.py train.epoch=50

Fig 2: The instruction sheet with the quiz

The main activity was allocated 30 minutes, during which participants worked on answering the quiz, unlocking specific files, and conducting two experiments. The session was designed to be engaging and interactive, encouraging participants to explore the codebase and learn through hands-on experience. Following this, we conducted the quiz session using Kahoot, where we recorded the answers and determined the winning group. To add an element of excitement, we announced a prize for the winners, which motivated everyone to participate enthusiastically and give their best effort.

The session turned out to be both productive and enjoyable, with everyone actively engaged and immersed in solving the quiz and conducting the experiments. We successfully completed the session within the allocated time, achieving positive outcomes and leaving participants with a meaningful and fun learning experience. In Fig 3, we have included some pictures of students working in groups to solve the quiz, unlock all the files and run the experiments. Fig 4 shows the results from the Kahoot challenge, ending the lab session.

Fig 3: Some pictures from our student lab

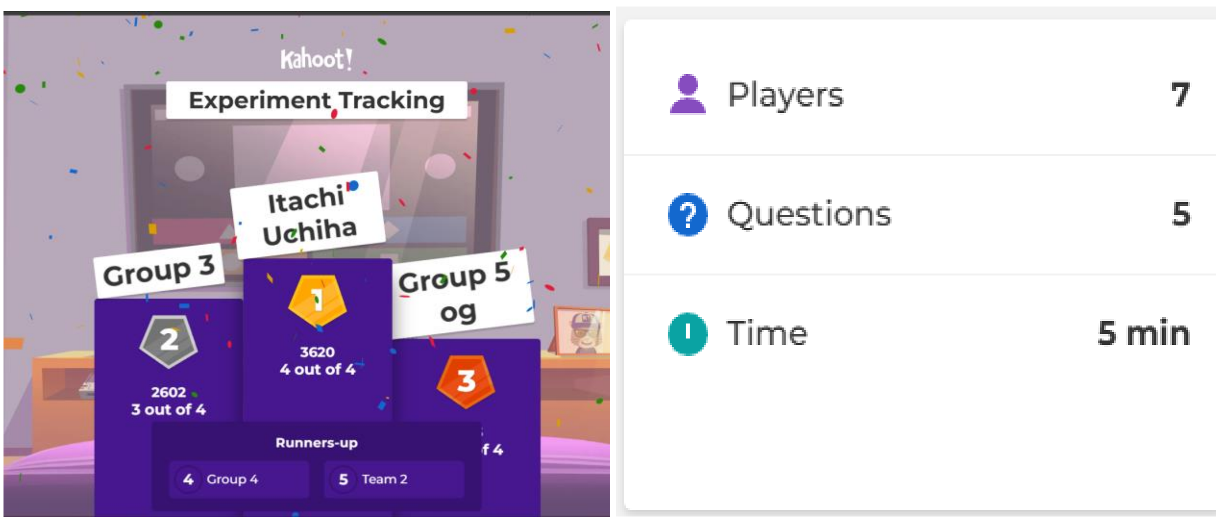

Fig 4: Results from the Kahoot Challenge

7. Second Feedback

Feedback is an essential component of the learning process, especially when it comes to evaluating the impact of lab sessions. It provides valuable insights into how effective the session was and helps us understand the participants' experiences while working through the activities.

By collecting feedback, we can gauge the strengths of the session—what worked well and resonated with the participants—as well as identify areas that might need improvement. This not only enhances the quality of future sessions but also ensures that everyone's perspectives are taken into account, fostering an inclusive and collaborative environment.

Additionally, feedback allows us to understand how participants felt during the execution of the session. Did they find it engaging and informative? Were there any challenges or obstacles they faced? By addressing these questions, we can make necessary adjustments to better cater to the group's needs and enhance the overall learning experience.

Ultimately, feedback serves as a bridge between planning and execution. It helps refine our approach and ensures that each lab session becomes more effective, enjoyable, and beneficial for everyone involved. Some of the feedback were:

"Well prepared and informative. The activities were a little bit challenging but was fun at the same time."

"The activity was fun and interactive, and I actually learned something, if I had to give a negative feedback it would be the lack of entertainment, but who cares about that if you are presenting an actual lab. Good Job!"

"Congratulations on executing the lab systematically. After the lab, I realise that MLflow is actually something that we all will use a lot for our thesis. Thanks for this informative lab"

"The approach towards the student lab was methodical, and the presentation and codebase your team implemented was both efficient and informative. The notebook was also clear and easy to follow."

8. Group Members Participation

In this section, we have described how the tasks were distributed among the group members and accordingly, it is important to note that every member made valuable contributions at every step of the lab but we present the most significant contributions made by each member; as shown in Table 2.

Table 2: Contribution of each group member

Group Member	**Task**
Muhammad Kabir Hamzah	Focused on developing the code base, gpu server setup, and creating the instructions manual.
Tuhinangshu Gangopadhyay	Developed the starter notebook, set up the experiments, organized feedback and writing the report.
Sarita Mourya	Responsible for designing quizzes and refining the instructions, ensuring the comprehensiveness and ease of use and contribution in writing the report.
Muhammad Qasim	Worked on proof testing the experiments, ensuring the compatibility on different devices.
Muhammad Momin Naveed	Worked on the ideation and creation of the introductory slides—developing the overarching story.

9. Conclusion

The lab session on experiment tracking and management successfully demonstrated the importance of structured and maintainable codebases in machine learning workflows. By integrating MLflow and a well-organized codebase, participants gained hands-on experience in managing complex experiments, tracking parameters, and ensuring reproducibility—essential skills for both research and production environments.

The session provided a balance of theoretical knowledge and practical application, enabling participants to understand the significance of tools like MLflow and Python-based configuration management in streamlining machine learning experiments. Activities such as interactive quizzes, trivia challenges, and collaborative problem-solving reinforced learning in an engaging and intuitive manner.

The feedback received highlighted the session's strengths, including its informative and interactive nature, while also providing constructive insights for improvement. Participants appreciated the well-prepared content, hands-on exercises, and the structured approach to teaching experiment management and code organization.

Overall, the lab session achieved its objectives of equipping participants with essential tools and techniques for machine learning workflows, fostering collaborative learning, and creating a productive and enjoyable environment. These efforts have laid a strong foundation for future enhancements, ensuring that subsequent sessions will continue to offer value and relevance in the field of data science and machine learning.

References

1. Zaharia, M., et al. (2018). MLflow: A Tool for Managing the ML Lifecycle. Proceedings of the 2018 ACM/IEEE International Conference on Big Data Computing, Applications and Technologies.

2. Idowu, A., Kuhn, T., Reimer, D., & Wilke, C. (2022). Machine Learning Experiment Tracking and Management Tools: A Comparative Review. IEEE Transactions on Machine Learning in Practice.

3. Erden, Caner. "Machine Learning Experiment Management with MLFlow." Encyclopedia of Data Science and Machine Learning. IGI Global, 2023. 1215-1234.

4. Spadari, Vincenzo, et al. "An MLOps Framework for Explainable Network Intrusion Detection with MLflow." 2024 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2024.

5. Grossi, Maria Aparecida de Faria, et al. "The influence of ML Flow test in leprosy classification." Revista da Sociedade Brasileira de Medicina Tropical 41 (2008): 34-38.