ETH zürich

ETH AI CENTER

Data Science Lab
Spring 2025 Course

DSL Best Practices

Giovanni Acampa & Arnout Devos

Ensuring success in your project

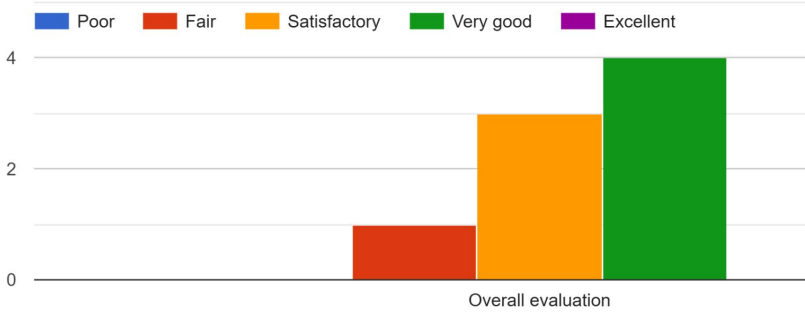
What this session covers:

- **Tips and tricks** for a successful Data Science Lab project.
- How to **meet expectations and deadlines** effectively.
- **Technical suggestions** to help you with your coding setup.
- Help you with **experience from previous Data Science Labs**.

But first: how was the feedback for DSL from last semester's students?

Last semester feedback

How would you evaluate the course overall?



What aspects of this course were most useful or valuable?

- hands on experience training machine learning models on the cluster
- diverse range of industry partners is very exciting
- Insights into real world problems, new experiences with different ML fields
- opportunity to put theory into practice
- Opportunity to apply knowledge in a new setting

This is what last semester's students liked!

Last semester feedback

How would you improve this course?

clearly define passing criterion, provide paper/poster templates at the start of the course, start course earlier in the first week to have more time for the project, provide technical advisors that can actually help, we got an advisor from a pure mathematics background with no practical ML experience

Respond faster to questions on slack. Share templates for final presentation and report earlier (we also have other things to do at the end of the semester and thus might want to start earlier).

Better communication! The course overall was really cool but especially at the beginning nobody really knew what to do and what are the goals.

And this is what **they didn't like**. We are trying to improve it this semester: *this presentation already contains some of the solutions!*

Tips & Tricks

Tips & Tricks - Define your goals

For how naive it may sound, remember to define your goals clearly.

- **Why?** Clear goals provide direction and help prioritize tasks.
- **Start small**. Maybe you will run out of time or find difficulties along the way: you should still have some results, such as a small model trained or your experiment's results on a subset of the data.
- **Have a long term vision.** Divide your work in subtasks, but always keep in mind why you are accomplishing the tasks. Keeping an agenda with weekly updates on the project can be useful.

Tips & Tricks - Literature review

The literature review is a step that is too often overlooked or conducted too late in the project.

- **Why?** You want to understand the state of the art research and build on that, rather than duplicating existing papers. You also want to understand which approaches are more likely to fail.
- **How?** Read recent and highly cited papers, for example from Arxiv or Google Scholar. Save the references to these papers and show that you are building your project keeping the SOTA research into account.

Tips & Tricks - Establishing baselines

Remember to always compare your results with some baselines. Two suggestions are:

- **Compare with results from other papers**, that worked with similar problems, datasets or methods.
- **Build your own baselines**: before running experiments with a complex technique, try to solve the problem with a linear regression, decision tree or a simple perceptron.

Show how your final results are an improvement over the baseline. Take any difference between experiments into account, like dataset, metric or hyperparameters.

Tips & Tricks - Consult with the coaches

You can benefit from an academic coach and a challenge giver.

- **Academic coach.** Have a group meeting with the academic coach every 1 or 2 weeks. They can help you to understand if the technical approach is correct and if there are some important steps you are skipping.
- **Challenge giver.** Have a group meeting with the challenge giver every 3 or 4 weeks. You should update them on your findings and understand together what are the next steps.
- **Come prepared to the meetings.** Come with specific questions and with some slides on your current results and future expectations.

Tips & Tricks - Don’t be scared of negative results

A negative result can be a great result!

- **Justify the results.** If the experiments are not validating the hypothesis, try to show that maybe it’s impossible to predict the variable Y from X. Show that you followed every step correctly and your experiments are reliable, or double check your pipeline to find mistakes.
- **Find an alternative way.** From your wrong results you may get a new idea on how to solve the problem. Maybe you can predict the variable Y from Z instead of X.

Expectations & Deadlines

Expectations & Deadlines - Presentations

There will be 3 important deadlines (more info on the exact dates will be given later):

- **Mid-term check-in**: mid-semester. Presence is mandatory.
- **Final presentation**: end-of-semester. Presence is mandatory.
- **Report submission**: end-of-semester. Submission is mandatory.

Expectations & Deadlines - Presentations

How does it work?

- **Mid-term check-in**: you need to prepare a few slides (usually 3) focusing on:
  - Introduction to the project
  - Initial approach to the project and what you have done so far
  - Being quantitative (numbers, graphs, ...) / scientific (related work, baselines, comparison, ...) is recommended
  - What are your planned next steps

Expectations & Deadlines - Presentations

How does it work?

- **Final presentation**: it will be a poster session. Each group will prepare a poster and will explain it to academic coaches, challenge givers and other students attending the event
- **Report submission**: you have to prepare a report written in the form of a scientific paper, that your supervisors will grade with a pass/fail grade

Expectations & Deadlines - Templates

For the poster and the report, we provide two useful templates at the following links:

- **Poster**: https://www.overleaf.com/read/xzrpcdjscrkm#4a8d04 (template.tex)
- **Report**: https://www.overleaf.com/read/hknhbrbwcvcb#55ba68 (template.tex; don't touch acmart.cls)

Expectations & Deadlines - Templates

Notes about the templates:

- You must write a **2-columns report of up to 6-pages**, excluding references and appendices. Make sure to be scientific
- The report includes a **necessary contributions footnote**: update and check it with everyone involved
- The report includes a **necessary preprint footnote**: please link your public repository (e.g., GitHub) including an MIT license there.
- The poster has to be **printed in A1 format**. On the poster, only include the ~3 most important references

Expectations & Deadlines - Reproducible code

The code that you develop should be easily reproducible. We don't want to force you to use a specific repo structure, but please follow the following advice:

- **Create a readme.md** file.
  - You can quickly explain how the code is organized what it is supposed to do and what are the most important .py files.
- Include a few **running examples** in the readme to test your code.
  - Always make sure that your examples run in a short time. You can make examples of **training with a dummy dataset** or **evaluation**.
- **Write clean code**. Remove unused variables, write functions to make it more readable and format it appropriately.

Expectations & Deadlines - Reproducible code

The code that you develop should be easily reproducible. We don't want to force you to use a specific repo structure, but please follow the following advice:

- Run your experiments in a **conda/venv environment**, to keep track of the packages you are using. In the readme, specify the python version you used.
- Create a **requirements.txt** file or similar (e.g. see poetry) with all the packages needed to run your code.
- At the end of the project **test the code yourself**! Download your own repo on your computer, follow your instructions from the readme and see if the running examples work.

Technical suggestions

Technical suggestions - Understanding the Data

In most of the projects the challenge giver will provide you with a dataset:

- **Understanding the data**: what are the data representing? Are they already clean? Do some variable appear useless or extremely useful? You can formulate hypothesis before running the experiments.
- **Start with a subset**: especially if the dataset is large, perform your experiments on a subset first. This will save you time and give you good insights on what works and what doesn't.
- **What can I do with these data?** Do I have enough data to train a deep learning model? Should I use something simpler?

Technical suggestions - Data processing

- **Data cleaning:** almost every dataset needs to be cleaned, using steps such as data normalization, data imputation and outliers removal.
- **Data splitting:** remember to split your data into training/test/validation set (usually 80/10/10) early in your project, to avoid any data contamination.
- **Preliminary analysis:** Observe your data. Building visualizations and basic statistics can help you to discover flaws or oddities in the data in an early stage of the project, saving you time.

Technical suggestions - Use of existing infrastructure

- There are several **software tools and packages** that can help you in your project, such as PyTorch, TensorFlow and scikit-learn.
- Depending on the topic of your project, e.g. NLP, Computer Vision…, there are several **specific libraries** that you can find online.
- These resources can help you a lot in your project. Whatever you decide to use, make sure to **mention it** in your presentations and report.

Technical suggestions - Computing resources

- ETH provides a student cluster with GPUs for INFK courses. You can find here the **full guide** to the student cluster: https://www.isg.inf.ethz.ch/Main/HelpClusterComputingStudentCluster

Q&A

Any questions?