

DATA**LAB**
COURSE INTRODUCTION — MASTER IASD

Alexandre Vérine

Université Paris Dauphine – PSL / ENS – PSL

1 October 2026



WHAT IS DATA LAB?

Three projects, one research workflow

Choose an approach, implement it, compare it with a baseline, and explain what the experiments teach you.

1. **Collaborative filtering:** predict missing ratings and recommend items.
2. **Generative models:** study the trade-off between quality and diversity through precision and recall.
3. **Adversarial robustness:** understand attacks and build more robust classifiers.

WHAT YOU SHOULD LEARN

- ▶ Turn concepts from your other courses into working experiments.
- ▶ Read a scientific paper and identify what is needed to reproduce its ideas.
- ▶ Design fair comparisons: data splits, baselines, metrics and computational budget.
- ▶ Explain a result, including limitations and unsuccessful experiments.
- ▶ Collaborate through Git and communicate clearly in a report and a short talk.

WHERE TO FIND THE COURSE

One route to remember

www.alexverine.com → Teaching → DataLab

▶ Open the DataLab website

- ▶ The course website is the entry point for course materials and project links.
- ▶ Resources will be released progressively throughout the semester.
- ▶ While the website is being updated, use the 2026 dates in these slides.

Research practice

AI IS ALLOWED. RIGOUR IS ESSENTIAL.

You remain responsible for everything you submit

AI tools are welcome. They make rigorous verification of your code, report and oral presentation even more important.

- ▶ Understand and be able to explain every method, equation and result you use.
- ▶ Check generated code, calculations, claims and references against evidence.
- ▶ Be ready to defend your choices and answer questions without an AI assistant.
- ▶ Use AI to support your thinking; generic prose is not scientific analysis.

WRITE CODE THAT MAKES YOUR RESEARCH REPRODUCIBLE

- ▶ Write readable code with clear names and a simple structure.
- ▶ Make experiments almost autonomous: one documented command, minimal manual steps.
- ▶ Document installation, data, dependencies, configurations and how to reproduce results.
- ▶ Use useful comments and docstrings to explain assumptions and non-obvious choices.
- ▶ Record seeds, parameters and the code version associated with each result.

For this course, code is not graded

Reproducibility supports your experiments and conclusions. The required interface for automated tests must still work.

DESIGN EXPERIMENTS THAT ANSWER A QUESTION

- ▶ Start with a question, a baseline and a fixed evaluation protocol.
- ▶ Vary one parameter or component at a time to isolate its effect.
- ▶ Run ablations: remove components to understand what each one contributes.
- ▶ Study scaling: how do results and computational cost change with data or model size?
- ▶ Check variability across runs when relevant; distinguish a trend from noise.

Avoid unsupported causal conclusions

If you change several things at once, you cannot attribute the outcome to one of them. Study interactions with controlled comparisons.

WRITE A REPORT THAT INTERPRETS THE RESULTS

- ▶ You have five pages and a reader who knows the topic: go straight to your work.
- ▶ Avoid generic introductions and trivial filler generated by ChatGPT or Claude.
- ▶ Describe the experimental protocol precisely enough to assess the comparisons.
- ▶ Include useful evidence and details you could not cover during the talk.
- ▶ **Interpret, do not just describe:** why might this result occur? What does it imply?
- ▶ Offer your expert judgement: separate observations, hypotheses and limitations.

GIVE A CLEAR, ADAPTABLE PRESENTATION

Respect your allocated time

Rehearse, prioritise your message and finish on time. This is a core requirement.

- ▶ Choose visuals that support your message: plots for trends, tables for exact comparisons.
- ▶ Use readable figures and typeset equations; avoid pixelated screenshots.
- ▶ Proofread slides: no typos, tiny labels or unexplained axes.
- ▶ Adapt to the room: shorten background already covered by previous groups.
- ▶ Spend your time on your experiments, interpretation and take-home message.

EXPECTATIONS INCREASE THROUGHOUT THE SEMESTER

	Research workflow	Presentation	Report
A1	A simple notebook is acceptable.	Clean and readable; no typos or pixelated screenshots.	A conventional, mostly descriptive report is acceptable to start.
A2	A cleaner Python repository with some documentation.	Better visuals and an adaptable delivery.	More technical detail and stronger interpretation.
A3	Reproducible, carefully controlled experiments.	Precise, polished and within time.	Rigorous evidence, critical analysis and clear limitations.

A3 will be assessed very strictly. Code itself remains ungraded across all three projects.

Course organisation

HOW A PROJECT WORKS

1. **Start:** introduction to the problem, baseline and expected outputs.
2. **Explore:** read, implement, experiment and ask focused questions.
3. **Present:** one presentation per group and per project.
4. **Submit:** code, a report and presentation slides in your GitHub repository.

During support sessions

Bring an idea, an experiment, a result or a specific technical problem. Be ready to explain what you have already tried.

CALENDAR — OCTOBER

Date	Session
Thursday 1 October	DataLab introduction + project A1
Thursday 8 October	Slurm introduction + technical support
Thursday 15 October	A1 preliminary presentations
Thursday 22 October	A1 final presentations + introduction to A2 and precision/recall
Thursday 29 October	No class

CALENDAR — NOVEMBER AND DECEMBER

Date	Session
Thursday 5 November	A2 preliminary presentations
Thursday 12 November	A2 final presentations + introduction to A3
Friday 13 November	Guest lecture: Lucas Gnecco Heredia
Thursday 19 November	No class
Wednesday 25 November	No class
Thursday 3 December	A3 final presentations: all groups

PRELIMINARY OR FINAL: WHO PRESENTS WHEN?

A1 and A2

Each group presents **once**: either in the preliminary session or in the final session.

Provisional allocation: 19 groups in preliminary; 18 in final. **5 minutes of presentation per group, with an alarm at 5 minutes.** Allow a further 2 minutes for questions and transition.

A3

Every group gives a **final presentation** on 3 December.

The detailed allocation and speaking time will be announced once the number of groups is known.

Presenting early does not replace the final project deliverables.

A FINAL SESSION ALSO LAUNCHES THE NEXT PROJECT

Time budget	Activity on 22 October and 12 November
-------------	--

2 h 15 min	Student presentations, questions and transitions
------------	--

5 min	Transition margin
-------	-------------------

Last 40 min	Introduction to the next project
-------------	----------------------------------

Three hours in total

Respect the announced speaking time. The next project starts during the same class, so all groups should stay until the end.

GITHUB REGISTRATION: COMPLETE EVERY STEP IN ORDER

1. Fill in your GitHub ID in the registration sheet.

2. Accept the invitation to join Master-IASD.

If you have already added your ID, you should have received an invitation email. Check your inbox.

3. Register for the DataLab classroom.

4. Register for assignment A1.

Every student must complete all four steps. Each link above is clickable.

FORM YOUR GROUP BEFORE MONDAY 5 OCTOBER

Exactly three students per group

Agree on your partners and group name before registering your group.

1. **One student creates the group** when accepting A1.
2. **The other two students join that same group.**
3. Check that all three members can access the same GitHub repository.

Do not create three separate groups for the same team.

A1 AND A2: FIVE-MINUTE PRESENTATIONS

Your presentation lasts exactly 5 minutes

An alarm rings at 5 minutes. Finish your talk when it rings.

Current A1 count: 37 groups (7 October).

Session	Groups	Talk per group	Total with questions
Preliminary	19	5 minutes	133 minutes
Final	18	5 minutes	126 minutes

- ▶ Allow a further 2 minutes per group for questions and transition.
- ▶ Within 2 h 15: 2 minutes of margin in preliminary; 9 minutes in final.
- ▶ Provisional allocation: it may change after incomplete groups are reorganised.

A3: with 37 groups, these slots require 4 h 19. Its presentation format will be announced separately.

Presentations and submissions

WHAT MAKES A USEFUL PRESENTATION?

Preliminary presentation

- ▶ State the problem and your research question.
- ▶ Compare the approaches you are considering.
- ▶ Explain what you have implemented so far.
- ▶ Describe the evaluation protocol and next experiments.
- ▶ Show initial results, if available.

Final presentation

- ▶ State the problem briefly.
- ▶ Explain your contribution and baselines.
- ▶ Describe the experimental protocol.
- ▶ Present readable tables or plots.
- ▶ Interpret the results and discuss limitations.

Focus on your work: do not repeat the lecture. Use typeset equations and tables, not screenshots of code or equations.

REPORT FORMAT AND DELIVERABLES

Format

One group cover page (group name and student names) + **at most five pages of content**. Optional appendices will not be read.

- ▶ Explain what you implemented and point to the relevant source files.
- ▶ Identify important external code or libraries you used.
- ▶ Describe the experiments, comparisons and conclusions.
- ▶ Discuss what worked, what failed and what you learned.

Avoid long recaps of the lecture and extensive code listings. Keep essential results in the main content.

SUBMIT THE RIGHT FILES, IN THE RIGHT PLACE

At the repository root

```
your-group-repository/  
  slides.pdf  
  report.pdf  
  requirements.txt  
  ... project code ...
```

- ▶ Use these exact filenames: `slides.pdf` and `report.pdf`.
- ▶ Commit **and push** the files to GitHub.
- ▶ Open the repository on GitHub and verify both PDFs.
- ▶ Submit through the repository, not by email.

Submission requirements

Deadlines and filenames are mandatory. Late submissions will not be accepted; missing or incorrectly named files will be penalized. A local file that has not been pushed is not a submission.

DEADLINES: PDFs AND CODE HAVE DIFFERENT CUTOFFS

Deliverable	Deadline (Paris time)
<code>slides.pdf</code>	23:59 the day before your presentation
<code>report.pdf</code>	23:59 the day before the final project session
Project code	Changes accepted until 05:00 on the day of the final session

Tests start around midnight. Code updated during the night may need to be tested again.

The 05:00 code cutoff does not extend the PDF deadlines.

YOUR REPOSITORY IS TESTED EVERY MORNING

Daily run: around 05:00, Paris time

The platform pulls your group's GitHub repository and runs the project tests on the code you have pushed.

1. **Push your changes:** local commits alone are not available to the platform.
2. **Automatic tests:** the platform runs your code and measures its results.
3. **Check the results:** use the feedback to identify problems and improve your experiments.

► Example: last year's platform for another course

This illustrates the principle. The DataLab URL and visual design will change. Tests start around 05:00; results become available after the runs finish.

SUBMISSION DATES AT A GLANCE

	A1	A2	A3
Preliminary slides	14 Oct, 23:59	4 Nov, 23:59	—
Final slides and report	21 Oct, 23:59	11 Nov, 23:59	2 Dec, 23:59
Final code	22 Oct, 05:00	12 Nov, 05:00	3 Dec, 05:00

Each group submits slides for its assigned presentation session. The final report is required from every group.

THIS YEAR, SUBMISSION DEADLINES ARE STRICT

Experience from previous cohorts

Around **60% of students submitted their repositories late**. With a larger cohort, I can no longer accommodate late submissions.

Example	Repository status	Deadline status
Group A	PDF uploaded	On time
Group B	PDF uploaded	Late: uploaded the next morning
Group C	File saved locally only	Missing from GitHub

Check before the deadline

Correct filename, correct repository, readable PDF, successful push. Do not wait until the start of the class.

The approximate 60% figure is an instructor observation; the table uses illustrative groups.

Compute resources

COMPUTE RESOURCES FOR YOUR PROJECTS

MesoNET and the Juliet GPU cluster

We will use shared compute resources for the projects. Slurm schedules jobs and allocates the requested CPU, GPU, memory and running time.

- ▶ **This week:** request your account and complete your profile.
- ▶ **Before 8 October:** request access to the IASD project and prepare your SSH key.
- ▶ **8 October:** practical introduction to Slurm and technical support.

Start now: account approval and project access are separate steps.

CREATE YOUR MESONET ACCOUNT NOW

1. Open **iam.mesonet.fr** and sign in with **eduGAIN**, using your institutional account.
2. Check your name and email; mention that you are a student in the **IASD DataLab course** in the registration note.
3. Validate your email and wait for account approval. The confirmation includes your MesoNET username.
4. Sign in to **your MesoNET profile** and complete it **before emailing me**.

► [Official account creation guide](#)

REQUEST ACCESS TO THE IASD COMPUTE PROJECT

Once your account is approved and your portal profile is complete, email:

alexandre.verine@dauphine.psl.eu

- ▶ Include your name, MesoNET username and the email used for registration.
- ▶ Mention **IASD / DataLab** so I can add you to the course project.
- ▶ Follow any invitation instructions and check your project access in the portal.

Be ready for the tutorial on 8 October

Account approved, profile complete, project access requested, SSH public key prepared. The next slides provide the SSH setup to try before class.

BEFORE YOU LEAVE TODAY

1. Add your GitHub ID to the sheet, accept the Master-IASD invitation, then register for the classroom.
2. Accept A1: one student creates the group, the other two join. Finish before Monday 5 October.
3. Read the starter repository and identify the required input and output.
4. Request your MesoNET account now; complete your profile and email me for IASD project access.
5. Plan a baseline experiment and prepare questions for 8 October.

SSH setup: reference

SETUP REFERENCE: GENERATE AN SSH KEY

Run these commands on **your own computer**. If you already have an SSH key, you can use it; do not overwrite it.

1. Generate a key if needed

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Accept the default path only if no key exists there. A passphrase is recommended.

2. Display the public key

```
cat ~/.ssh/id_ed25519.pub
```

3. Register it in the MesoNET portal's SSH keys section.

Upload only the `.pub` key. Keep the private key on your computer.

SETUP REFERENCE: CONNECT TO JULIET

Prerequisites: an approved account, access to the course project, and your public SSH key registered on MesoNET.

```
ssh username@juliet.mesonet.fr
```

Replace username with your MesoNET login. Optionally add this to ~/.ssh/config:

```
Host juliet
  Hostname juliet.mesonet.fr
  User username
  IdentityFile ~/.ssh/id_ed25519
  IdentitiesOnly yes
```

Then connect with `ssh juliet`. Training jobs will be submitted with Slurm during the tutorial.

► [Official Juliet connection guide](#)