The background image shows a high-tech laboratory or office environment. In the foreground, a man in a white lab coat is seated at a desk, interacting with a tablet that displays a bar chart. On the desk, there are various items including a stethoscope, a notebook, and several small containers. A large, transparent, blue-tinted holographic display in the center of the desk shows a complex network graph with a central node and many branching connections. To the right, a robotic arm with two grippers is positioned over the desk. In the background, there are several computer monitors displaying code or data, a server rack with glowing blue lights, and a large whiteboard covered in a detailed flowchart or organizational chart. Potted plants are scattered throughout the space, adding a touch of greenery to the tech-heavy environment.

# Data Science & Machine Learning Lab

---

Exam rules  
(A.Y. 2025/26)

Flavio Giobergia

# Overview

- The exam consists of two independent parts:
  - **Project** (10 points)
  - **Written Test** (22 points)
- You may complete the two parts in **any order** throughout the 3 exam session:
  - **Winter session:** 1 project assignment, 2 written exam dates
  - **Summer session:** 1 project assignment, 1 written exam date
  - **Fall session:** 1 project assignment, 1 written exam date
- Scores from both parts remain valid until the **end of the Fall 2026 session**

# Partial and final grades

- The partial grades obtained will **not** be rounded
- The final score = written score + project score will be rounded to the closest integer
  - 17.49 → 17, 19.5 → 20
- The exam is considered pass if:
  - Written score  $\geq 12/22$
  - Project score  $\geq 6/10$
  - $\text{round}(\text{Final score}) \geq 18/32$
- The grade 30 with honor is assigned if the final score, **before rounding**, is  $\geq 31$



# Project

# Info

- Each project requires designing and implement a complete data science pipeline for a *classification* or *regression* task
- The pipeline will be used to predict data points for which a ground truth is not known ("evaluation set")
- A public leaderboard will show your performance and that of your colleagues
- A timespan of **at least 14 days** will be available to solve the problem
  - Typically, the written exam date will be in the middle of this timespan, if possible

# Groups

- The project can be delivered:
  - **individually**, or
  - as a group of up to **2 students**
- Team composition may **change across exam sessions**
- Each project session requires groups to register their team composition before the project begins
  - No need to register the group if you will work on the project individually
  - A link to an online form will be provided in the days prior to the exam
  - Teams must be registered **before** the project begins
- Only **one report** is submitted per group

# Deliverables

- A valid project submission includes **all** the following:
  1. **Prediction output** submitted to the online leaderboard platform
  2. **Project report** following the required structure
  3. **Software/code**, provided as Python notebooks and/or scripts in a single ZIP file
  4. **LLM usage declaration form**

# Project evaluation

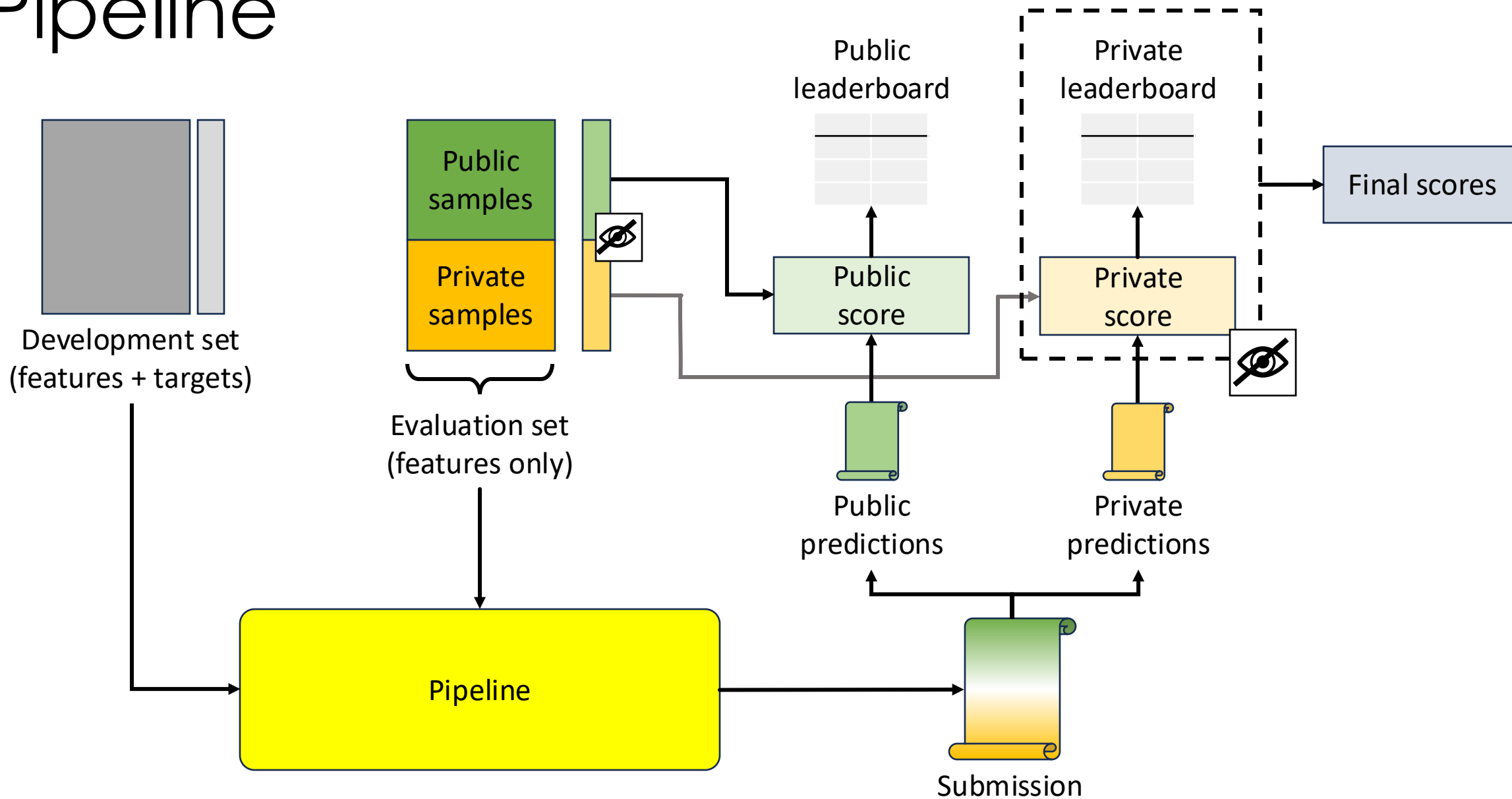
- The project score consists of:
  - **Report grade:** up to **7 points**
  - **Performance grade:** up to **3 points**
- A project is **sufficient** if the *project score*  $\geq 6$ .



# Project delivery

- You may deliver any of the projects by the provided deadlines.
  - (Deadlines are strict, no exceptions will be made)
- Delivered projects are kept “on hold”, and will only be evaluated when one of the team members achieves a sufficient (and non-rejected) grade in the written exam.
- You may deliver at most one report per project
- You may submit reports for multiple projects (one in Winter, one in Summer, one in Fall)
  - Only the latest delivered project is ever considered

# Pipeline



# Leaderboard

- Each project uses an online submission platform
  - Similar to the one you will use for Labs 9 and 10
- Each student will receive a personal key (“password”) to be used for submissions
  - The key assigned before Lab 9 will be the one you will also use for the exams – do not lose it!
- For each project, you may submit up to **200** predictions
- There is a minimum **1-minute delay** between submissions
- You may select up to **2 submissions** for the final evaluation
  - If none are specified, the solutions with the two highest public scores are used
- Only the **private leaderboard score** contributes to the performance grade

# Report – requirements

- Must use the [IEEE conference LaTeX template](#) provided
  - (You can use LaTeX either locally, or on Overleaf)
- Maximum **4 pages** (excluding references)
- Must include the required sections:
  1. Problem Overview
  2. Proposed Approach
    1. Preprocessing
    2. Model Selection
    3. Hyperparameter Tuning
  3. Results
  4. Discussion
  5. References

# Report – grading criteria

- The grading is based on:
  1. Relevance of the contents
  2. Quality of the presentation
  3. Significance and relevance of images and tables
  4. Presence of key steps for the successful completion of the task
  5. Additional steps that helped achieve improved performance
- Non-compliance with the template results in **0 points** for the report
  - This includes changing aspects of the template such as font sizes, margins, or using different templates

# Software – requirements

- Python only
- Notebooks and/or scripts packaged in a **single ZIP file**
  - Other formats will not be accepted and will be assigned 0 points
- Execution must be clear and reproducible
- A clear entry point must be provided (e.g., main.py or instructions in README)
- All experimental results must appear in the report, not generated at grading time
- Students must deliver original content produced by themselves. Plagiarism is not allowed in any form
  - (including copying from websites)

# Other details

- **Out-of-syllabus methodologies** are allowed.
  - The proposed approach must be motivated in the report
  - A further oral examination may be requested
- **Additional (public) data sources**, unless otherwise stated in the project description, can be used in the proposed pipeline
  - A clear description and motivation must be provided in the report
- **Further assessments** (oral or written) may be requested by the teachers, regarding the delivered report and/or software, in specific scenarios
- LLM usage:
  - LLMs may be used **only for writing the report**
  - LLMs may **not** be used for coding or experiments
  - All participants must submit the LLM declaration form



# Written test



# Info (1)

- The written exam lasts **2.5 hours** and is divided into two parts:
  - “Quiz” section (1 hour – 14 points)
  - “Coding” section (1.5 hours – 8 points)
- The written exam will take place *in presence* in the premises of Politecnico di Torino
- You will use your own laptop computer (Moodle + Lockdown Browser)
- You must be regularly booked through the “Portale della Didattica” to be allowed to take the written test
  - You will be asked to present your Student Card + an ID document
- Seating arrangements will be communicated prior to the exam

# Info (2)

- Books, notes, electronic devices of any type **are not allowed**
  - This includes smartphones, smart watches, programmable calculators)
- A single scientific calculator, non-programmable, is allowed
  - A calculator will also be made available in-browser

# Quiz section

- Will include up to **10 open- or closed-ended questions**
- All topics covered during the course are included:
  - Data preparation and preprocessing
  - Association rules (extraction algorithms, itemset types, quality indices)
  - Classification: algorithms, quality indices, validation strategies
  - Regression: algorithms, quality indices, validation strategies
  - Clustering: algorithms, quality indices
  - Anomaly Detection: algorithms, quality indices
  - Python notions and operations (+ libraries)
- Closed-ended questions will include penalties for incorrect answers
- If needed, relevant formulas will be provided as a part of the exercises
  - You will still be expected to know *basic* formulas

# Coding section

- You will solve a small machine learning problem using a provided dataset. Tasks will include:
  - Classification, Regression, Anomaly Detection
- You will receive:
  - A **development set**, used to build your model
  - An **evaluation set**, containing new samples to be predicted with your model
    - (**Note:** for Anomaly Detection tasks, development and evaluation sets will coincide)
- You must design a preprocessing + modeling pipeline and produce predictions in a CSV file ("**submission.csv**")

# Coding environment

- Conducted in **CrownLab + Lockdown Browser**
- No Internet access
- The environment will be based on Visual Studio Code
  - With relevant Python libraries (NumPy, pandas, scikit-learn, torch, SciPy)
  - With access to Jupyter Notebook inside of Vscode
  - With relevant offline documentation for the available libraries

# Required files

- When the exam ends, the contents of the working directory will be stored in a zip file. It must include:
  - **main.py**: loads data, preprocesses, trains a model from scratch, predicts, saves to submission.csv
  - **submission.csv**: the predictions for all evaluation rows
- Note: main.py must run in its entirety in at most 150 seconds.
- Hyperparameter tuning operations must be done separately, and only the “best” solution must be executed in main.py

# Evaluation process

- The evaluator will:
  1. Re-run main.py in a clean environment
  2. Verify it completes within **150 seconds**
  3. Validate submission.csv (headers, row count, IDs)
  4. Compare the results with the ones reported in submission.csv
  5. Compute the final performance on the evaluation set

# Invalid or penalized submissions

- Penalties will be introduced if any of the following occurs:
  1. `main.py` crashes or fails to run
  2. execution exceeds time limit
  3. `submission.csv` missing/malformed
  4. wrong filenames or wrong directory location
  5. non-reproducible results (mismatch between uploaded csv and generated csv files)



# Restrictions

- No pre-trained models or pre-computed artifacts can be used
- All artifacts must be created at runtime
- Randomness should be controlled via seeds

# Resubmission policy

- The results of the evaluation process (final score + penalties) will be published after the exam ends
- You will be given access to your solution (zip containing main.py and submission.csv)
- You may change your code (e.g., to improve performance, or to address the problems that introduced penalties) and re-submit it through the “Elaborati” (Homeworks) section
- Your new submission will be re-evaluated, with a penalty term proportional to the significance of the introduced changes

# Scoring criteria

- The final score will be assigned according to the following criteria:
  1. Performance achieved on the evaluation set (w.r.t. a naïve and a “full” pipeline)
  2. Presence of penalties in the final submission
  3. Extent of the changes applied during the resubmission
  4. Relevance of the solution to the specific task

# Timeline

- **Day 0** (exam day, at the end of the exam): your solution is uploaded as a ZIP file at the end of the exam (available to you in the Elaborati/Homeworks section)
- **Day 1** (by end of day): publication of the initial results (scores + penalties)
- **Day 4** (by end of day): deadline for the re-submission of the solution
- **Day 6** (by end of day): publication of the final evaluation
- (Note: Day 1 and Day 6 deadlines may need to be postponed according to the number of participants – in that case, the other deadlines will be updated accordingly)



# Exam “pipeline”

# Exam (1)

- Student presence
  - If a student is absent on the exam day, they are not considered in any of the next steps (as if they never enrolled – and will be recorded as absent)
  - If a student is present on the day of the exam, their exam is evaluated
    - Go to “exam outcome”
    - There is no option to withdraw from the written exam
- Exam outcome
  - If the exam is insufficient, the student must retake the written test in a future date (even if they had passed a previous written exam)
  - If the exam is sufficient, students may decide to reject their written score
    - If rejected, the student will need to retake the written test at a future date)

# Exam (2)

- Project evaluation (applicable if a student passes the written test, and does not reject the score)
  - If they did not submit any project, their written score is kept “on hold”, until they deliver a project
  - If they submitted multiple projects, only the latest one is considered
  - If they submitted a single project, that project is considered
- Project grading
  - If the report had already been graded (e.g., because their teammate passed the exam in a previous session), the same grade is assigned
  - Otherwise, the report is graded
- Final score
  - If the report grade is sufficient, the final score = written + project is recorded
- Rejecting the final score
  - If the final score is not satisfactory, students may reject the final score from the Portale della Didattica, as per the Politecnico rules