

## Workshop project: Reproduce published results

### Where to start?

1. Select a paper: Done ✓
2. Make sure that you have all relevant data: Done ✓
3. Make sure the methodology and experiments are clearly described, and that you have sufficient details for your implementation: Done ✓
4. Make sure that you have the needed computational resources.
5. In many cases you can find the source code on-line, don't cheat 😊

### What to submit?

**Project proposal:** describe the selected paper, available data, available resources, suggested timeline and so on. **Quick presentation in class:** Jan 04, 2026 (a brief description of the paper, your task, the data, what blockers can hold you back, see **Guidelines for Paper Presentation**)

**Project document:** a “paper-like” document describing your results in comparison to the original paper results, explain differences if exist. You dont need to cover all analyses in the paper, and you can add analyses if you think they are appropriate and helpful.

Dates:

**Working code and result:** Jan 28, 2026

**Code review:** Jan 18-25, 2026

**Final Submission:** Feb 5, 2026

### Grading for the Analyst side (75%)

Criterion	Weight	Breakdown
I. Analytical Correctness	40%	Correct understanding of the project goals and implementation, as described in the paper.
II. Technical Proficiency	30%	As described in the <b>Project Submission Guidelines</b>
III. Complexity level	15%	The amount of required “Thinking Power”, data, architecture, code, and so on.
IV. Final Presentation	10%	Clarity, professionalism, and summary of findings.
V. Structure and Readability	5%	Your final grade may be adjusted based on the quality of the "Client" review you receive.

## Project Submission Guidelines

Your project submission should include the following:

- **Package Submission:**
  - A toml file containing the project name.
  - Required dependencies.
  - Development dependencies (e.g., pytest).
- **README File:**
  - **Project Documentation:**
    - Project description: main objectives, assumptions, hypothesis, etc.
    - Folder/module structure, including sub-modules.
    - Key stages and what was done in each (e.g., data import → data processing → modeling → analysis → visualization/graphs).
    - Important definitions and explanations of key parameters/configurations.
  - **Data Description and Link to the Dataset.**
  - **References to Papers or Articles Used.**
  - **Instructions for Running the Project (Commands).**
- **Tests Directory:**
  - Include tests covering each major stage of the project. Successful execution of these tests should confirm that each stage meets its intended assumptions. Please carefully design the tests to reflect expected outcomes.
- **Main Script (main file):**
  - Should contain only a minimal amount of code.
  - Calls to key functions, each representing a major stage in the process. (Think of it as a sequence of large building blocks, each leading to the next.)
- **Logging Best Practices:**
  - Use a logger instead of print statements.
- **Naming Conventions:**
  - The project name (on GitHub) should reflect its purpose. A generic name like *FinalProject* does not provide meaningful information.
  - Folder, file, and function names should be **consistent** (e.g., camelCase vs. snake\_case) and **clear/readable**.
  - Avoid passing raw numbers or strings directly into functions. Instead, use meaningful variable names, e.g.:

- `function_a(binary_num, number_to_print, number_dic)` instead of `function_a(1, "2", {1, 2})`
- **Keep Functions Short:**
  - If you find yourself writing a comment above a block of code, consider extracting it into a separate function.
  - Functions should not exceed **50 lines of code**.

#### **Guidelines for Paper Presentation:**

1. Introduce Your Paper: Provide a clear and concise summary of the paper.
2. Anticipated Challenges: Identify the key challenges you expect to encounter during the implementation.
3. Potential Unexpected Challenges: Discuss any unforeseen difficulties that might arise and how you plan to address them.
4. Work Plan: Outline your strategy for tackling the project. If working in pairs, specify each team member's roles and responsibilities.
5. Effort Estimation: Provide an estimate of the time and coding effort required to complete the implementation.
6. Performance Evaluation: Explain how you intend to evaluate your results and compare them with the outcomes presented in the paper.