Semester Project 2022

The semester project is an integral part of the CIL course. Participation is mandatory. Failing the project results in a failing grade for the overall CIL course.

You work in groups of three to four students (no more, no less) to develop novel solutions to one of three topics. You may use Moodle to find team members or join an existing team.

Building on the implementations you develop during the semester, you and your teammates create a novel solution by combining and extending previous work. You compare your solution to at least two baselines and submit it to the online ranking system for evaluation. Finally, you write up your methodology and present your experimental results in the form of a short scientific paper.

Project Reports are due on 31. July 23.59. Competitive submission deadlines are given on Kaggle.

Students repeating the course can either carry over their grade from the previous year's project as-is, in which case they have to inform us in advance. Alternatively, they can also resubmit a new project in a (new) regular group.

As part of the semester project, you and your teammates are expected to

- Develop a novel solution, e.g. by combining and extending methods from the programming exercises.
- Compare your novel solution to at least two baseline algorithms.
- Submit your novel solution for evaluation to the Kaggle online ranking system.
- Write up your findings in a short scientific paper.

As a rough guide, you may approach the problem as follows: (i) study the project description sheet, (ii) download the training data and implement the baselines, (iii) develop, debug and optimize your novel solution on the training data, (iv) submit your solution for online evaluation on test data, (v) see where you stand in a ranking of all submissions.

Developing a Novel Solution

You are free to exploit any idea you have, provided it is not identical to any other group submission or existing implementation of an algorithm on the internet.

Comparison to Baselines

You must compare your solution to at least two baseline algorithms. For the baselines, you can use the implementations you developed as part of the programming exercises or come up with your own relevant baselines.

Ranking of Novel Solution

You must submit your novel algorithm to the Kaggle online ranking system. See project descriptions.

Scientific Report

For instructions on how to write a scientific paper, see the following PDF, source. The write-up must be a maximum of 4 pages long (excluding references).

Project Submission

To submit your report, please go to {TBD}, register, and follow the instructions given there. You can resubmit any number of times until the deadline passes.

When finally uploading your report, you are also required to upload the Python code that you have used for your final Kaggle submission as supplementary material. The code should be well-documented and generate the predictions in the required format as uploaded to Kaggle. For reproducibility, you should also include additional code which you have used to produce plots and additional experiments, etc. The size limit for the supplementary material is 100MB. If there is anything that exceeds this limit (e.g., additional data used), please leave clear instructions in the readme of the code on how to download the remaining code/data.
Include the name of your group in the header of the submitted PDF file, e.g: author{Author1, Author2 & Author3, group: cil_nerds, Department of Computer Science, ETH Zurich, Switzerland}
Attach the signed plagiarism form at the end of your paper (scan).

Project Grading
The project grade is composed of a competitive (30%) and a non-competitive (70%) part.

Competitive grade (30%): The competitive grade uses the following formula: Grade = 4 + 2 * (x - baseline_score) / (max_score - baseline_score), where the baseline score is provided by the TA team and the max_score is the score of the winning team of the competition.

Non-competitive grade: The following criteria are graded based on an evaluation by the teaching assistants: quality of paper (30%), the creativity of solution (20%), quality of implementation (20%). Each project is graded by two independent reviewers. The grades of each reviewer are de-biased such that the average grade across all projects that the reviewer graded is comparable for each reviewer.

Computational infrastructure

Use ETH's new Leonhard cluster. You will be given access to Leonhard during the course of the semester.

Report Grading Guidelines

Your paper will be graded by two independent reviewers according to the following three criteria:

1. Quality of paper (30%)

6.0	Comparable to a (workshop) submission to an international conference (features discussed in the last lecture).
5.5	Background, method, and experiment are clear. May have minor issues in one or two sections. Language is good. Scores and baselines are well documented.
5.0	Explanation of work is clear, and the reader is able to identify the novelty of the work. Minor issues in one or two sections. Minor problems with language. Has all the recommended sections in the howto-paper.
4.5	Able to identify contributions. Major problems in the presentation of results and or ideas and or reproducibility/baselines.
4.0	Hard to identify contribution, but still there. One or two good sections should get students a pass.
3.5	Unable to see novelty. No comparison with any baselines.

2. Creativity of solution (20%)

6.0	Elegant proposal, either making a useful assumption, studying a particular class of data, or using a novel mathematical fact.

5.5	A non-obvious combination of ideas presented in the course or published in a paper (Depending on the difficulty of that idea).
5.0	A novel idea or combination not explicitly presented in the course.
4.5	An idea mentioned in a published paper with small extensions/changes, but not so trivial to implement.
4.0	A trivial idea taken from a published paper.

3. Quality of implementation (20%)

6.0	Idea is executed well. The experiments make sense in order to answer the proposed research questions. There are no obvious experiments not done that could greatly increase clarity. The submitted code and other supplementary material are understandable, commented, complete, clean and there is a README file that explains it and describes how to reproduce your results.
Subtractions from this grade will be made if	● the submitted code is unclear, does not run or experiments cannot be reproduced or there is no description of it ● experiments done are useless to gain understanding or of unclear nature or obviously useful experiments have been left undone ● comparison to baselines are not done