Introduction to Machine Learning (67577)
Hackathon 2023

Teachers: Dr. Gabriel Stanovsky, Gilad Green

TAs: Eitan Wagner, Gili Lior, Noa Moriel, Reshed Mintz

Tzars: Itai Alon, Nitay Alon

Second Semester, 2023

1 General Instructions

Welcome to the IML Hackathon 2023!
Please read the entire document carefully. Failing to comply with the following instructions might result in rejecting your submission

- Hackathon kickoff Wednesday 07/06/2023 19:00
- Hackathon times: Wednesday 07/06/2023 19:20 until Friday 09/06/2023 13:00
- Hackathom team size: **exactly** of 4 students
- The instructions and data for both tasks will be available on the course Moodle.
- After reading through the entire document, as well as the tasks' explanation documents, select **one** of the tasks and solve it as best as you can.
- Please use the proper moodle forum for clarification questions.

Both tasks are equally hard. Choose the one that *you* find more interesting and fun to work on. You are far more likely to have good results on a task you find fun and interesting; Also, the performance you can hope to achieve is different between the two tasks.

- Be sure to write good, clean, and documented code. Our ability to understand what you are performing and why you did it is important for receiving a high grade.
- Work iteratively - we advise starting of with a basic solution that works. Then try to improve it as much as possible in the given time.

Please read the following guidelines in detail before engaging in the task solving.

2 Submission Instructions

Be sure to follow the described structure precisely. We will have a **zero tolerance** policy and there are no presubmission tests - submissions that fail to comply with the requested format will receive zero points.

Submit a single `.tar` file containing the following files and structure:

- `README.txt` — contains a file list and a brief description of each file.
- `USERS.txt` — contains the team members' logins and IDs. Use one line per student, in the format: `login, ID`.
- `project.pdf` - submit **a description of your project**, up to 2 pages long, as a PDF file. Explain your solution, describe your work process, and do your best to justify any design decisions you have made. Feel free to include supporting figures if you want.
- `predictions` – place all your predictions (.csv files) in this directory. Name each file to match the question number. Make sure to name the file's columns as instructed
- `answers` – if the question asks for descriptive answer instead of prediction, place the corresponding .pdf files in this directory. Make sure to name the files as instructed.
- Submit **your code** in the following manner. To ensure that we can run your code on the CS computers without errors, please submit your environment's `requirements.txt` file. Instructions on the freeze command are available on Moodle. This documents all the packages used by you and guarantees reproducibility of your results. If we cannot replicate your submission, you'll get 0 grading - so be sure to follow this step. Your code should have a `main.py` file, with proper executable call (`if __name__ == "__main__"`...). All calls to methods, classes and whatnot should be included in this file. Place the rest of your code in a directory titled `code`. That is, your code directory has contains a single file `main.py` and a single code folder `code`. Make sure to use relative paths and not absolute paths when importing modules or files. Solutions must be written in Python3. To use Python 3 on CS computers, use python3 instead of Python.

File or directory	**Description**
`<student-id>.tar`	.tar file **only**
└ `USERS.txt`	
└ `README.txt`	
└ `project.pdf`	
└ `task_number`	***Task related files***
│ └ `predictions`	
│ │ └ *`<csv files>`*	
│ └ `answers`	
│ │ └ *`<pdf files>`*	
│ └ `code`	
│ │ └ `requirements.txt`	
│ │ └ `main.py`	
│ │ └ `hackathon_code`	
│ │ │ └ *`<all python files>`*	

3 Evaluation

Your grade will be determined by two factors (equally important):

1. The quality of your prediction on the test set. Try to provide the best prediction possible.

2. The quality and depth of your written work description in the PDF. We will grade your written description based on the following:
   - Did you describe the dataset and any challenging characteristics it has?
   - Did you describe (briefly) the data cleaning and preprocessing?
   - Did you describe the considerations that guided your design of learning systems?
   - Did you describe (briefly) various methods you tried and the results you obtained?
   - Did you describe the learning system you ended up using?
   - Did you provide a prediction (and explanation) of the generalization error you expect your system to have?

3.1 Verify Before Submission

Before you submit your results, please verify the following check list to make sure your submission is correct:

- □ Your files are organized in the defined file folder structure described in 2
- □ Your output data matches the format defined in the tasks
- □ Your code only calls files present in the `code` folder. When doing so, use a relative path. For example : `"weights.txt"` is good while `"C:\Users\name\IML\weights.txt"` is bad.
- □ You have a `requirements.txt` file
- □ You included all the outputs in your directory - we will not accept corrections or errata past submission
- □ Your code doesn't throw any exception or error when running - if it does you'll get no grade
- □ Your code has to finish prediction (including any necessary loading or initialization) in **5 minutes** when running on the CS computers
- □ After zipping your project, it should have a **maximum size of 20 MB**
- □ Submit only **one** of the challenges (see directory structure)
- □ Submit the zip (or tar) file through the Moodle of only **one** of the team members
- □ **Your work is original** - any suspicious of copying will result in severe disciplinary action

4 Hackathon FAQ

Please review these questions carefully as we will not address them separately.

Q: We've started working on task A and after a day we decided to work on task B, can we do that?
A: Yes, you will be graded for one task only (either Task A or Task B) but you may switch between them. If you submit 2 tasks (or parts of them) you'll get no grade.

Q: Can we use external python libraries?
A: Yes, as long as you follow the instructions above and make sure you provide a working `requirments.txt` and verify that these libraries do not require additional sources (`c++ libraries`, drivers, etc.).

Q: Can we use external data to solve the tasks?
A: NO. You are allowed to use the web for inspiration and code ideas but you are not allowed to augment the data provided with external data. Groups that will do so will get no grade.

Q: If we're able to run the code but it failed to run on evaluation time, do we get full grading?
A: NO. One of the demands of this Hackathon is to be replicable. If we can't run your code, we can't verify your predictions and your submission is disqualified.

Q: We need a place to sit quietly, where can we find it?
A: There's a list of available offices during the Hackathon - please use them considerably - others would like to use them too.

Q: Can we use LLM (GPTChat, AutoPilot) to generate code?
A: Yes you may, but make sure to indicate that you've used it in the main .pdf file (it has no affect on the grading). If you decide to use it, we'll appreciate if you can describe the experience and add some print-screens of the prompts and output.

Q: We want to consult the course staff, how can we reach them?
A: The course staff will be on campus after Hackathon kickoff (until 21:00), on Thursday between 9:00-18:00 and Friday from 9:00 until submission time. You may find them at C112-C113 and they will walk around the CS building. There will be **no** remote support.

Q: Something went wrong and we can't submit on time - is there an extension
A: No. The Hackathon is a bonus and there will be no extension to deadline - plan you time wisely.

Good Luck!