Projects
**Machine Learning Course**
Fall 2023

**EPFL**
School of Computer and Communication Sciences
**Nicolas Flammarion & Martin Jaggi**
www.epfl.ch/labs/mlo/machine-learning-cs-433

# Class Project 2
# Submission Deadline: Dec 21st, 2023

## Introduction

In this project, you will learn to use the concepts we have seen in the lectures and practiced in the labs on a real-world dataset, start to finish. You will be doing exploratory data analysis to understand your dataset and your features, do feature processing and engineering to clean your dataset and extract more meaningful information, implement and use machine learning methods on real data, analyze your model and generate predictions using those methods and report your findings.

**Grading.** Project 2 counts 30% to your final grade in the course. We offer you three interesting directions of doing this main project.

## Logistics

**Group formation.** For Project 2, you again form a team of 3 members on your own. It can be different from your group in Project 1. If you are still searching for teammates, please use the discussion forum and discord. In exceptional cases, if you didn't manage to form a team of 3 by Dec 1st, contact us.

**Submission deadline.** Dec 21st, 2023 (at 16:00 afternoon, sharp)

## Option A - Machine Learning for Science

Pick a real-world challenge offered by any research group of the (extended) EPFL campus, subject to availability. You learn about an interesting application domain, and collaborate with the lab to apply machine learning methods to their specific research question, on real data provided by the lab - an exciting option to follow an interdisciplinary approach to find new insights with your team.

`https://www.epfl.ch/labs/mlo/ml4science/`

**Deliverables at a glance.** (More details and grading criteria further down)

- **Written Report.** You will write a maximum 4 page PDF report on your findings, using LaTeX.
- **Code.** In Python. External libraries are allowed, if properly cited.
  The guidelines for the projects are the same as in the standard tasks explained below.

**Participation of the Hosting Lab.** The only major difference to the other options is that here, you do this project in collaboration with another lab on EPFL campus. The lab hosting you will help us grading the domain-specific merit of your contribution.

We encourage you to reach out to any research group of the EPFL campus, or any academic institution in Switzerland (such as EPFL, UniL, CERN, CHUV, Idiap, and others). Labs who are interest to host a student group can contact us here to offer their project idea, dataset or task.

Here is a list of labs which already have proposed several exciting project ideas. The list will be updated once project 2 start comes closer.

`https://docs.google.com/spreadsheets/d/1Mav7vND2dYghHQxLEXqnvLZ1z9GKRcNynOAjwcQpMPo/`

Other labs are possible as well and you're encouraged to reach out to any lab of your choice (even outside EPFL), as mentioned above.

Important Logistics: You can only sign up for this option if the *professor* of that lab agrees to host your group of 3 students, and will confirm this by filling the mandatory registration form for labs, by November 12 (ideally):

`https://goo.gl/forms/0ElZtbKGz4U0lCag1`

Submission URL for the final project: `http://mlcourse.epfl.ch` (same as for option B)

Your final submission to the online system (a standard system as used for scientific conferences) must consist of the following:

- **Report:** Your 4 page report as `.pdf`. References are allowed to be put on an extra page. An appendix is possible but might not be graded.
- **Code:** The complete executable and documented Python code, as a github repository link (here is the invite link: https://classroom.github.com/a/fEFF99tU). Rules for the code part (see also the guidelines further below)
  - *Reproducibility:* Please provide well-documented and reproducible code. This can be in form of Python notebooks or other form. Include a clear `ReadMe` file explaining your code organization, and how to reproduce your setup. Include pre-trained models when possible, as well as additional code to produce them, including all data preparation, feature generation as well as cross-validation steps that you have used. Include all needed data whenever possible (the CS433 github repositories remain private). We nevertheless encourage you to open-source your code publicly as well.
  - *External libraries, models and datasets* are allowed, as long as accurately cited and documented.

# Option B - One of our Pre-defined Challenges

**Deliverables at a glance.** (More details and grading criteria further down)

- **Written Report.** You will write a maximum 4 page PDF report on your findings, using LaTeX.
- **Code.** In Python. External libraries are allowed, if properly cited.
- **Competitive Part.** To give you immediate feedback and a fair ranking, we use the competition platform AIcrowd.com to score your predictions. You can submit whenever and almost as many times as you like, up until the final submission deadline.

**Pick Your Favorite Among Two Challenges.** Pick your favorite competition among the following two online competitions. Don't be influenced by their seemingly different difficulty level, since your contribution as compared to standard approaches will be taken into account in the grading.

## Step 1 - Getting started

In order to be able to access the challenges, please first create an account at AIcrowd.com using your @epfl.ch email address. Pick your favorite competition among the following two. To read the description and download the dataset, please follow the corresponding links:

`https://www.aicrowd.com/challenges/epfl-ml-text-classification`

`https://www.aicrowd.com/challenges/epfl-ml-road-segmentation`

For the two possible tasks, we provide some additional description and sample code on the course github:

`https://github.com/epfml/ML_course/tree/master/projects/project2`

### Step 2 - Implement ML Methods

You are allowed to use any external library and ML techniques, as long as you properly cite any external code used.

### Step 3 - Submitting your Predictions

Once you have a working model (using the above methods or a modified one), you can send your predictions to the AIcrowd competition to see how your model is doing against the other teams. You can submit whenever and as many times as you like, until the deadline.

Your predictions must be in `.csv` format, see `sample-submission.csv`. You must use the same datapoint ids as in the test set `test.csv`. To generate `.csv` output from Python, use our provided helper functions in `helpers.py` (see Project 1 folder on github).

After a submission, AIcrowd will compute your score on the test set, and will show you your score and ranking in the leaderboard.

Do not use the AIcrowd score as the only estimate of your error. Instead, always estimate your test error by using a local validation set, or local cross-validation! This is important to avoid overfitting the test set online. Also, it allows you to make experiments faster, and save uploading bandwidth :). You are only allowed a maximum of 5 submissions per person to AIcrowd per day.

### Step 4 - Final Submission of Your Project

Your final submission to the online system (a standard system as used for scientific conferences) must consist of the following:

- **Report:** Your 4 page report as `.pdf`
- **Code:** The complete executable and documented Python code, as a github repository link with your group of students (here is the github classroom invite link).
  Rules for the code part:
  - *Reproducibility:* In your submission, you must provide a script `run.py` which produces *exactly* the same `.csv` predictions which you used in your best submission to the competition on AIcrowd. This includes a clear `ReadMe` file explaining how to reproduce your setup, including precise training, prediction and installation instructions if additional libraries are used - the same way as if you would ideally instruct a new teammate to start working with your code.
  - *Documentation:* Your ML system must be clearly described in your PDF report and also well-documented in the code itself. A clear `ReadMe` file must be provided. The documentation must also include all data preparation, feature generation as well as cross-validation steps that you have used.
  - *External ML libraries* are allowed, as long as accurately cited and documented.
  - *External datasets* are allowed, as long as accurately cited and documented.

Submission URL: `http://mlcourse.epfl.ch`

Don't forget to provide all author names with their EPFL email, as well as the (correct one and only one!) submission ID appearing in the AIcrowd ranking, and select the right one of the A,B,C tasks. You can update all parts of your submission anytime until the deadline.

## Option C - ML Reproducibility Challenge

Your team participates in the **Reproducibility Challenge**. Here the goal is to select a recently submitted paper, and try to reproduce (parts of) its experiments:

`https://reproml.org/`
(the papers can be from any conference including NeurIPS, ICML, ICLR, ACL-IJCNLP, EMNLP, CVPR, ICCV, AAAI and IJCAI)

# Ethical Risk Assessment in the Written Report

You should evaluate the ethical risks of your project at least once, preferably in the early phases. To assist in this process, we suggest that you use the Digital Ethics Canvas, a risk assessment grid with a series of questions that guide you in the analysis of software-specific ethical risks. A procedure for using the canvas is available online. You can use the provided examples as well as the references and information from the slides on the workshops page.

In your report, you **must include an "Ethical risks" section** (200 to 400 words max., does not count in the 4-page limit), which counts for **part of the grade of the project report** and should consist of one or two short paragraphs. You may choose one of the two options based on your unique project topic:

## Situation A: You have identified at least one ethical risk

Focus on *one* of the ethical risks you identified and answer the following questions:

- Describe the risk
  - Who is impacted (which type of stakeholder)?
  - What is the negative impact?
  - How significant is the risk, both in terms of severity and likelihood of occurrence?
- How did you evaluate this risk (research you did, metrics you measured)?
- How have you taken this risk into account in your project?
  - If you were able to take it into account, what did you change?
  - If you were not able to take it into account, what were the barriers?

## Situation B: You have not identified any ethical risk

Please describe the process you followed to rule out the presence of ethical risks:

- Which types of stakeholders did you consider when evaluating the ethical risks of your project? You should mention at least 2 categories, including some indirect stakeholders and/or the environment, and provide a description of their roles.
- How did you rule out the different ethical risks for these stakeholders (research you made, metrics you measured)?

You are welcome to attach your Digital Ethics Canvas as an Appendix to the report to help illustrate your risk analysis process, but it is not required.

# Appendix

## Grading Criteria and Some Advice

We are aware that not all tasks have the same difficulty (especially there will be larger variance in options A) and C). We will take the difficulty of the question into account when grading, so none of the choices will have any disadvantage for you.

- **Code.** In Python. You are allowed to use external libraries, as long as properly cited, documented and referenced. Similarly as in Project 1, your project submission will be graded by two assistants independently. As some additional advice for the code part,
  - Double-check that the archive you submit actually contains everything needed to run your solution. If your project is for option B), check that your code indeed exactly reproduces your best AIcrowd submission (and select the submission on AIcrowd accordingly).
  - Make sure any additional installation needed to run your solution is fully described in the accompanying `ReadMe` file, including version numbers if necessary.
  - Try to follow your own instructions to get your code running at least once.
  - Try to make your code as readable as possible. Reduce copy-pasted code to a minimum, use helper functions and clearly name your files, functions and variables.
- **Written Report.** You will write a maximum 4 page PDF report on your findings, using LaTeX. We will grade you on the scientific contribution you made, that is on the improvement achieved over the standard baseline methods, as well as the rigorous and fair measuring of the claimed improvements. The criteria are

  - **Solid comparison baselines supporting your claims**
    Quantify the benefits of your method by providing clear quality measurements of the most important aspects and additions you chose for your model. Start with a very basic baseline, and demonstrate what improvements your contributions yield.
  - **Reproducibility**
    Your classmates should be able to reproduce your results based on your report only. Describe what preprocessing is required, what hyper-parameter values you selected and why, and clearly describe the overall pipeline you used.
  - **Scientific novelty and creativity**
    You will likely be using more than the standard methods we saw in the first half of the course. To communicate that your methods work and that you understand them, you should make sure that your report makes clear the following points.
    - What *specific* problem your method is intended to solve.
      By specific, we do not mean "image classification" but what specific issue with your current model are you trying to improve with this method.
    - Why is this an important problem? Why are you solving this one instead of something else?
    - How is your method helping?
    - What are the results of your method? Compare the error before and after.
  - **Ethics component**
    Situation A:
    - The risk should be described specifically in the context of the project and should include elements on the affected stakeholder(s), the type and importance of the potential impact(s).
    - The description of the risk evaluation process should include references to existing sources documenting the risk or refer to metrics used to evaluate the risk in the context of the project.
    - The text should describe how the results of the risk analysis have been considered in the project, either by presenting specific changes made to the project or by detailing specific barriers to mitigation.

    Situation B:
    - At least 2 different categories of project stakeholders should be presented with a description of their role, and should include some indirect stakeholder(s) and/or the environment.

  - The description of the risk evaluation process should include references to existing sources ruling out the risks or refer to metrics used to rule out the risks in the context of the project.

- **Writeup quality**
  Some advice:
  - Try to convey a clear story giving the most relevant aspects of your approach, in a reproducible way. Learning what has not worked can additionally help the reader (and help them better understand *why* you have made the many choices you did), but focus on what is most relevant for your final solution.
  - Before the submission, have an external person proofread your report. It is easy to write a sentence that makes perfect sense to you since you wrote it but is actually hard to parse. Use a spell-checker.
  - Plots are great way to share information that might be hard to convey by writing. Make sure that your plots are understandable, have labels for axes, a title, correct axes limits, add a description of what your plot is about and what can be learned from it.

- **Competitive Part (only for option B)).** The final rank of your team in the (private) leaderboard will be translated linearly to a scale from 4 to 6.

As usual, your code and report will be automatically checked for plagiarism.

## Guidelines for Machine Learning Projects

Now that you have implemented few basic methods, you should use this toolbox on the dataset. Here are a few things that you might want to try.

**Exploratory data analysis** You should learn about your dataset - figure out which features are continuous, which ones are categorical, check if there are obvious relationships between the features, take a look at the distribution of each feature, and so on. Check `https://en.wikipedia.org/wiki/Exploratory_data_analysis`.

**Feature processing** Cleaning your dataset by removing useless features and values, combining others, finding better representations of the features to feed your model, scaling the features, and so on. Check this article on feature engineering: `http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/`.

**Determining whether a method overfits or underfits** You should be able to diagnose the whether your model is over- or underfitting the data and take actions to fix the problems with your model. Recommended reading: *Advice on applying machine learning methods* by Andrew Ng: `http://cs229.stanford.edu/materials/ML-advice.pdf`.

**Applying methods and visualizing** Beyond simply applying the models we have seen, it helps to try to understand what the ML model is doing. Try to find out which datapoints are wrongly classified and, if possible, why this is the case. Then use this information to improve your model. Check Peter Domingo's *Useful things to know about machine learning*: `http://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf`

**Accurately estimate how well your method is doing** By applying cross-validation and estimating the test error.

## Report Guidelines

In addition to finding a good model for the data, you will need to explain your methodology in a report.

Clearly describe your used methods, state your conclusions and argue that the results you obtained make (or do not make) sense, and the reasons behind it. Keep the report short and to the point, with a strict limit of 4 pages. References are allowed to be put on an extra page.

To get started more easily with writing the report, we provide you a LaTeX template here

github.com/epfml/ML_course/tree/master/projects/project1/latex-example-paper

The file also contains some more helpful information on how to write a scientific report or paper. We will also help you learn it during the exercise session and office hours if you ask us.

For more guidelines on what makes a good report, see the grading criteria above. In particular, don't forget to take care about

- *Reproducibility:* Not only in the code, but also in the report, do include complete details about each algorithm you tried, e.g. what lambda values you used for ridge regression? How exactly did you do that feature transformation? how many folds did you use for cross-validation? etc...
- *Baselines:* Give clear experimental evidence: When you added this new combined feature, or changed the regularization, by how much did that increase or decrease the test error? It is crucial to always report such obtained differences in the evaluation metrics, and to include several properly implemented baseline algorithms as a comparison to your approach.

Some additional resources on LaTeX:

- `https://github.com/VoLuong/Begin-Latex-in-minutes` - getting started with LaTeX
- `http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/` - tutorial on LaTeX
- `https://wch.github.io/latexsheet/` - cheat sheet collecting most of all useful commands in LaTeX
- `http://en.wikibooks.org/wiki/LaTeX` - detailed tutorial on LaTeX

## Producing figures for LaTeX in Python

There are some good visualization tools in Python. "matplotlib" is probably the single most used Python package for 2D-graphics. The relevant tutorials are as follow:

- Matplotlib tutorial: `https://github.com/rougier/matplotlib-tutorial`
- Matplotlib tutorial other useful Python data visualization libraries: `http://jakevdp.github.io/mpl_tutorial/`