Project Guidelines – COMP 432

The "big picture" for project grading is this: at the end of the term, the instructor will look over the projects and, for each student, consider the question "*Does this student seem competent at machine learning and at communicating about machine learning?*" The more you can convince the instructor that this is the case, the more you merit a high grade. The instructor and TAs will look at your submission carefully.

Students have two options for their projects:

1. Instructor-Proposed Project: The instructor will provide a project within the first few weeks of the course. If you opt for this choice, the section on ”project proposal” below is not applicable to you

2. Student-Proposed Project: Students have the opportunity to suggest a project that they personally find engaging and educational. In that case, you need to write a project proposal and get it approved by the instructor.

Project Proposal

Please, fill out the project proposal form that you find here. The form is designed to encourage you to think carefully about your project before starting. As you can see, the proposal has to consider many aspects of the project, including goals, data, challenges, computational resources, literature review, and originality. Key aspects to consider for the project proposal are the following:

- **Clarity.** Make sure the proposal is well-written, fluent, and logically consistent.
- **Problem statement**. Clearly describe the problem that you are going to solve. Mention why it is important.
- **Solution**. Describe as precisely as possible the solution that you would like to explore. Explain why you think the solution can solve the problem. Mention explicitly the machine learning technique that you want to explore.
- **Originality**. How does the proposed solution differ from what has already been done by others? Where is the novelty of your work? Note that working on a project based on a Kaggle competition is discouraged due to a lack of originality. If you opt for that, you have to explain clearly what is the difference between your project and those already available online.
- **Data**. Clearly mention the dataset that you would like to use. Explain why it is relevant to your problem. The use of open data is encouraged. In any case, you have to make the data available to the instructors to allow replicability.
- **Measurable Performance**. Make sure the outcome of the project is measurable. Clearly state which performance metrics you would like to consider and why they are meant for your problem.
- **Computational Resources**. Describe the computational resources that you plan to use for the project. Make sure the project is feasible with your current computational resources.
- **Feasibility**. Make sure the project is feasible by the end of the course.
- **Challenges**: Describe the main expected challenges. Consider reporting a risk-management plan that explains how to address challenges and risks.

The deadline for the project proposal is **February 16**. However, we encourage students to start thinking about their projects at the beginning of the course. The project proposal will be reviewed by your project advisor. The instructor will review the final project proposal as well. The outcome of the final review could be: “*accepted*”, “*revision required*”, or “*rejected*”. If the project is accepted you can move on with the project. If the project needs revision, you have to fine-tune the proposal based on the comments of the project advisor and instructor. If the project proposal is rejected, you need to

submit a new project proposal. The project proposal is not graded. However, having a project proposal accepted is mandatory to get the project scores.

Project Submission

At the end of the course, each student must submit on Moodle a Colab notebook similar to those used in the lab sessions. The notebook should contain a mix between text, code, plots, and tables. Sometimes, it is not convenient to put all the code in a single notebook (e.g., because the code is too long, has too many functions, etc). In this case, you can put the code in an external repository (e.g, Github or Gitlab) and import it into Colab. This way, the notebook can be more compact. However, it should be runnable, well-commented, and allow result replication anyway.

You can find a template for the colab here: https://colab.research.google.com/drive/1MtSbf1n67Sb-I3wq_ATi2QmPiGLfe8nR?usp=sharing

You have to make sure the results that you report are reproducible. We encourage using open data or making the datasets available to the project advisor and the main instructor.

As for the text part of the report, the following aspects will be considered in the final evaluation:

Clarity. The most common failure mode of a project report is a lack of clarity. Even if you are good at machine learning, if you cannot describe the "what" and the "why" of what you did at the appropriate level of detail, then this will make it hard for you to work in the industry or succeed in research.

Logical consistency. The second-most common failure mode of a project is a disconnect between the stated goal(s) and the actual system that was built, or the experiments that were performed. For example, if your goal is classification, then saying "we used K-means" does not make logical sense on its own, and would require further explanation.

Realistic conclusions. The third most common failure mode of a project is when there is a disconnect between what the conclusions claim and what the experiments actually show. It is much better to depict an honest assessment of what you could / couldn't conclude, or what you did/didn't succeed at than to try to impress the instructor with broad sweeping claims that are not justified by your analysis. When your conclusions overreach or misrepresent, that indicates to the instructor (or, someday, to your supervisor) that your conclusions are not to be trusted. Remember that TAs will run your code and will try to replicate your results.

Best practices. Finally, if you demonstrate that at least some ML "best practices" were applied during your project, then this can strengthen your grade. For instance, make sure the comparison between different machine learning techniques is fair. Explain clearly why you explored some machine learning techniques and not others. Make sure your dataset is properly split into training, validation, and test. Make sure there is no data leakage between the sets. If you can perform hyperparameter tuning, make sure it is done based on the performance of the validation set (not on the training set or test set).

Originality. Make sure the report clearly explains what are the original aspects of your project and in which way your effort differentiates from related attempts. The project must be something novel for you. Reusing projects from previous courses is not acceptable and will be reported. We discourage you from proposing projects already addressed in a Kaggle Competition or already over-explored by other people online.

As for the code part of the notebook, we will consider the following aspects:

Reproducibility. You have to make sure your results are reproducible by the project advisor and the main instructor. At a minimum, the teaching team should be able to achieve results similar to those reported in your official tables.

Acknowledge sources. Using code from the internet as part of your project is not a major problem. Using code from the internet without acknowledging the source is a major problem. The absolute worst thing you can do for yourself is to paste chunks of code from the internet into your project and then pretend that you wrote the code. That is a clear violation of academic integrity. It is thus important to clearly state which parts of the code are original and which parts are taken from existing repositories. We suggest writing a docstring for each function/class available in the code (see documentation) where you claim it. In any case, we expect you to write most of the core code yourself. We cannot accept projects where all the code is fully imported from other projects (even though you acknowledge them properly).

Documentation. Properly documenting your code is very important. For each function/class, we ask students to write short docstrings reporting:

1- Description of the functionalities implemented.
2- Description of the input arguments (with their type).
3- Description of the output arguments (with their type).
4- A working example that shows how to use the function or class.

This is the type of docstrings used in popular projects such as PyTorch (e.g. ,https://pytorch.org/docs/stable/_modules/torch/nn/modules/batchnorm.html#BatchNorm2d) or SpeechBrain (e.g., https://github.com/speechbrain/speechbrain/blob/develop/speechbrain/nnet/CNN.py#L21). Feel free to use the same template for your docstrings. Beyond that, make sure to use proper in-line comments every time you think it is needed.

Code organization. Take time to think about a meaningful, effective, and intuitive code organization. This aspect will be considered for grading.

Modularity. Split your code into meaningful functions or classes that can be combined and reused. Avoid writing too long pieces of code (that are often hard to follow). Write short functions with clear and intuitive meanings.

Code Style. We ask students to make the code PEP8 and flake8 compliant. To make it PEP8 compliant you have to use the following python packages: autopep8 or black. These tools will automatically convert your code into a PEP8-compliant one. Flake8 is another python tool that also checks for programming errors that the students are supported to address.

Computing:

If you work on a project involving deep neural networks trained on a large dataset, you might need to use a GPU-based machine. Google Colab gives you access for free to GPU machines as well. You just have to go to "runtime=>change runtime type=> GPU". You will have access to an NVIDIA K80 with 12GB of memory. This can work for many projects. However, the free version of Google Colab has some limitations. For instance, Colab is not providing guaranteed or unlimited resources. This means that overall usage limits as well as idle timeout periods, maximum VM lifetime, GPU types available, and other factors vary over time. More info here: https://research.google.com/colaboratory/faq.html
An alternative is using Google Colab Pro (https://colab.research.google.com/signup) which gives you access to faster GPUs with more memory (13.99$/a month). There is now also Google Colab Pro + ($67.20/month) that allows using even more resources (and running your jobs in the background). A

valid alternative to Google Colab is Gradient (https://www.paperspace.com/gradient/free-gpu). You can have access to good GPUs for just $8/month (https://www.paperspace.com/pricing) or better GPUs for $39 per month. Many of my students and collaborators found gradient very useful (and better than Colab).
An alternative is to use Google Cloud and take advantage of its 500$ initial bonus.
In any case, we suggest using non-free computational resources only if the free version of Colab turns out to be not enough for your project. If you have a GPU available on your computer, you can use it. Note that you are also free to use the GPU machine that is available in our labs. The main limitation is that you have to stop using them during the lectures and you cannot run jobs in the background.

Evaluation

The project grade counts for 30% of the total grade. The grade will consider the aforementioned aspects.

We will use the following evaluation grid:

- **Clarity** (5 points): The teachers will assess the clarity of the notebook that the student has submitted. Is the problem explained clearly? Is the solution implemented explained? Are there tables and plots that help understand the solution and its performance?

- **Originality** (5 points): If the student has utilized a Kaggle dataset or any other publicly available dataset, it is essential to ensure that the code used has not been directly copied from an existing code repository. However, it is acceptable if the student has reused code from an existing repository, provided that they acknowledge the source and specify any modifications that have been made. Therefore, we will verify whether the student has properly reported the source and documented any changes made when reusing code.

- **Code Quality** (5 points): We here assess the quality of the code. Is the code well-commented? Is there a meaningful split in terms of functions/classes? Is the class or function implemented compactly? Does the student seem fluent in Python?
  Is the code running? Did you notice some bugs? If you find major bugs that prevent the model from learning, you should assign a low score in the Machine Learning Best Practices voice as well.

- **Machine Learning Best Practices** (15 points): We here assess how the project is implemented and if the best machine learning practices have been put in place. Your evaluation should consider the following aspects:

Machine learning models:
- Are the machine learning models adopted meaningful for the proposed tasks?
- Is the student comparing many different machine learning models?

Results and Performance:
- Are the results (and comparison with other models) critically analyzed?
- Are the major bugs that prevent the model to learn or the results to be meaningful?
- Does the student provide a convincing justification If the performance is not good?
- Is the performance metric relevant to the task of interest?

Data:
- Are the training, validation, and test datasets composed of different samples (data leakage)?
- Is the data processed correctly before feeding them into the machine learning model?

- If a hyperparameter tuning is done, is it conducted on the validation set or test (the latter is a bad practice)?
- Are the features selected relevant to the task of interest?

Other:
- If the model overfits, does the student consider regularization techniques?
- If the problem involves highly unbalanced classes, is that considered in the performance metrics or during model training?

Submissions:

Each student needs to submit on Moodle the following documents:

- **Project Proposal (Deadline February, 17)**. You have to submit a single PDF file (Proposal_title_of_the_project.pdf). This only applies if you did not choose the instructor-proposed project.

- **Project Notebook (Deadline April, 21)**. You should submit on Moodle a single notebook file (*.ipynb). This applies to all the students.

Example of Past Projects:

- Predicting hourly solar power with machine learning.
- Improving Carrier Recommendation with machine learning.
- Garbage classification with deep learning.
- Traffic Sign Classification.
- Lunar Lander: Can we land on the moon with reinforcement learning?
- Online Anomaly Detection in a cloud system using log analysis

Important Dates

February, 16 (11.59)	Deadline for Project Proposals
April, 21 (Tentative)	Deadline for Project Submissions

Frequently Asked Questions (FAQ)

- *Can we do a group project?*

No. Group projects can be a good learning opportunity. However, it is very hard to assess the real contribution of each team member. In this course, we want to evaluate your machine learning skills in a way that is as fair as possible.

- *Can students work independently on the same project?*

This is discouraged as well. It would force students to exactly explain the differences between what he/she has done and what done by other students.

- *What is the role of the project advisor?*

Each student will be assigned to a project advisor. The advisor will monitor the evolution of the project and can provide feedback along the way. You can contact your project advisor if you need support.

However, the advisor will not solve the problem for you and will not solve your bugs. He/She will only provide suggestions and hints.

- *Why should I propose a project?*

In this course, there won't be a single project imposed by the instructor. Instead, we ask the students to think about a machine learning project. This step forces students to play an active role and be creative. Carefully thinking about what can be done with machine learning, which data are needed, how to measure the performance, which kind of machine learning model can work, etc is very formative. It also gives you the possibility to work on a topic that is within your interests.