# Project Guidance – COMP 432

The instructor will ask himself the following question when grading: "Is this group competent at machine learning *and* at communicating about machine learning?" The more you can convince the instructor that this is the case, the more your group merits an A / A+ grade.

The instructor will look at your report carefully, but will only glance at your code. The TA will look at your code carefully, and only glance at your report.

## Report guidance

Here are some considerations:

- **Clarity.** The most common failure mode of a project report is lack of clarity. Even if you are good at machine learning, if you cannot describe the "what" and the "why" of what you did, and at the appropriate level of detail, then this will make it hard for you to work in industry or to succeed in research.
- **Logical consistency.** The second-most common failure mode of a project is a disconnect between 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.
- **Best practices.** Finally, if you demonstrate that at least some ML "best practices" were applied during your project, then this can strengthen your grade. Were application-driven and data-driven asymmetries considered and accounted for, if applicable? Was feature preprocessing considered and, if so, was its impact assessed? Was a simple baseline, such as a dummy classifier or a linear model, included? What ML algorithms were evaluated and why? How was hyperparameter search conducted, if applicable? Was the cross-validation scheme(s) a good representation of how the model would be used at test-time?
- **Length.** Aim for 4 pages at a density that approximately matches the report template (font size, margins). However, it is OK to write a report that is under 4 pages if you think that it is clear and complete.
- **Supplementary pages.** You will not be penalized for including extra pages in your report, but your report should still 'conclude' within the page limit, and the instructor is not obliged to look at extra before grading your work. If you do include supplementary pages, best to fill them with results that are easy to understand at a glance, like extra examples of data or predictions, or raw tables that were already summarized in the main text of the report.

## Presentation guidance

The purpose of the presentation is to give the instructor, and everyone in the class, a high-level preview of

1. *What* goal you were trying to achieve.
2. The kind of *data* you were working with.
3. *How* you tried to achieve your goal.
4. To what degree you *did* or *did not* achieve your goal.

For fun you can finish your presentation a sentence or two on something like "If we were to do this project again, the biggest thing we would do differently is ..."

Considerations:

- **Length.** 2 minutes max. Shorter is OK, but only if the above points are still made clear.
- **Use slides.** You probably only need 4-6 slides total. Consider using Google Slides so that team members can collaborate remotely. Avoid recording a video of you scrolling through Jupyter notebook code. If you really think it's important to show code during your presentation, you can put a screenshot in your slides.
- **Use Zoom.** You can record the presentation however you want, but the easiest way to record, especially if you record as a group, is to have someone share the slides and begin Zoom local recording. You can start/stop for each attempt, and when the session is over Zoom will create a separate file for each one.
- **Faces are optional.** Audio is obviously required, but whether you show your faces is up to you. Video with faces results in slightly larger video files.
- **Multiple speakers is optional.** It would be nice if each group member would speak during the presentation, but it is not required. For example, a poor microphone may impede presentation quality, or it may be too difficult for the entire group to coordinate recording together.
- **Show the fun stuff.** You don't have to show all your results. Instead, just focus on the main points, and on getting the viewer excited to read your report.
- **Relax.** You will not be penalized for "ums" and "uuuuhhs" or for correcting yourself, so do not fall into the trap of "restart recording on first mistake"; just make sure the material is clear, and be done with it.

# Code guidance

Some criteria / suggestions:

- **Acknowledge sources.** For the purposes of a project, using code from the internet as part of your project is not a problem. Using code from the internet *without acknowledging the source* is a major problem. The absolute worst thing you can do for you and your team is to paste chunks code from the internet into your project and then pretend that you wrote the code — that is a clear violation of academic integrity.
- **Avoid copy/paste.** If the TAs see that long blocks of code were copied and pasted over and over, across scripts or notebooks, without any attempt at modularizing that code for re-use, then you may lose marks.
- **Avoid hard-coded paths.** If your code has things like `open("/home/me/ml_project/data/foo.txt")` then there's no way it's going to run on another computer. Try to organize your data and configuration files so that your scripts can refer to the files using relative paths, rather than absolute paths.
- **Code organization.** The TA will assess whether the code was reasonably organized. This assessment is somewhat subjective, but you just have to live with that.
- **Code comments.** Try to comment important functions or important computations in the code.
- **Reproducibility is good.** Reproducible code is more likely to get a full grade. This includes simple universal best practices like setting the relevant random seeds (Numpy, PyTorch, TensorFlow) or, if code is intended

to run on a local machine, exporting the conda/pip environment needed to run the code. However, if it is too late for your team do think about reproducibility, you can still get a good code grade.
- **Easy to run.** Even if your code is not fully scripted/automated, the fewer steps a TA would have to understand to run your code, and the clearer it is how to do those steps, the better.
- **Submit notebooks with results included.** In case a TA cannot run your code, it is better to submit your notebooks with the output cells (plots, images, output) included, so that the TA can at least see what the output was when *you* ran them.
- **Package small datasets with the project code.** Moodle accepts projects up to 250 MB in total size. If your compressed data set fits well within that limit, then it should be directly included in the submission.
- **Link to large data sets.** If your data is too large to include, you must provide a link (Dropbox, Google Drive, etc.) where the file's modification date is clearly visible. See "Submitting files."

# Submitting the presentation

The presentation deadline was extended to Dec 6 @ 11:59pm EST. Find the "project presentation" item in the course Moodle page. Your group should submit a single file `GXX_presentation.mp4`. The instructor would prefer to receive `.mp4` files, like Zoom creates, but other formats (`.mov`, `.wmv`) may work as well. Do not submit a `.pptx` file.

# Submitting the project files

The "project submission" link in the course Moodle page accepts one file per group.

Submit a single file `GXX_project.zip` (<250 MB), where GXX is your group ID. Unzipping should give:

```
$ unzip GXX_project.zip
GXX_project/GXX_report.pdf           <-- your report
GXX_project/GXX_code_and_data.zip    <-- code and data that
GXX_project/GXX_readme.txt           <-- overview of code and data
```

You do *not* need to submit the slides from your presentation video.
You do *not* need to submit a copy of your original proposal.

**GXX_report.pdf** This file is your report, exported as PDF.

**GXX_code_and_data.zip** This file should contain all your configuration files, scripts, modules, and notebooks needed to run your code. Projects with small datasets should include the data files as well. For example, Excel-style tabular data is often small enough, whereas computer vision or music projects may require too much data to fit into Moodle. Projects that only run in Google Colab or similar should include downloaded copies of the notebooks, but not of the data (instead, a link to data should be given).

**GXX_readme.txt** This file is important for the TA and instructor. It is the first thing they read before looking at your code. It should contain a brief overview of the key files in your project code, and explain to the instructor/TA how the code and data is organized, and what the project dependencies are.

For example, something like this is all you need:

```
README for GXX
--------------

Our project relies mainly on sklearn, PyTorch, and JAX.
Our code submission contains the following files:

  ./code/environment.yaml    <-- conda environment for our project
  ./code/util/*.py           <-- some utility functions we wrote
  ./code/preprocess.py       <-- script to preprocess our data
  ./code/train_svm.py        <-- script to train our SVM
  ./code/train_rf.py         <-- script to train our random forest
  ./code/train_nnet.py       <-- script to train our neural net
  ./code/report.ipynb        <-- notebook to generate figures and tables
  ./data/housing_prices.csv  <-- raw data set

* The files should be run in the order:
   preprocess.py
   train_*.py
   report.ipynb
* GPU is not required.
* Training takes ~2 hours.
* The report notebook saves files to an "out" directory.

The housing_prices data was downloaded from https://kaggle.com/blahblah

For preprocessing and training the SVM we relied heavily on the tutorial
found at https://towardsdatascience.com/blahblah
```

If you used Google Colab, still explain what the included notebook files do, and provide a shared link to the live notebooks which have access to the data. Do not alter the live notebooks after the deadline.

# Using LaTeX (Optional)

The vast majority of research papers in the engineering and mathematical sciences are written in LaTeX. LaTeX is a "compiled" document format, unlike for example Micorosft Word.

The example LaTeX file `GXX_report.tex` is based on the submission template for *CVPR* (the IEEE International Conference on Computer Vision and Pattern Recognition). CVPR is the top conference in the field of computer vision and machine learning. It uses a compact two-column format, but you are free to use a single-column format with similar density if you wish (e.g., NeurIPS style, or a non-LaTeX Microsoft Word template).

In terms of software tools, you have multiple options:

- **Overleaf.** This is an online collaborative LaTeX document authoring service. The free version only allows for one official `collaborator' per document, but the creator of the document can still share it by link with others. That way, several people can still edit and preview the document online. Nothing to install, but you need to upload your figures to their system. See overleaf.com.
- **Visual Studio Code.** This is an integrated development environment (IDE) that runs locally on your computer and supports many types of files (including `.py` and `.tex`) through extensions that are very

easy to add. The LaTeX extension supports PDF preview. If you go this route, you'll need to find a way for multiple people in your group to collaborate on `GXX_report.tex`, such as Dropbox or Github. See code.visualstudio.com and add the LaTeX Workshop extension.

- **MikTeX.** This provides a set of local command-line tools for compiling `.tex` files to other formats like `.pdf`. It comes with a pre-packaged LaTeX text editor with PDF preview. If you go this route, you'll need to find a way for multiple people in your group to collaborate on `GXX_report.tex`, such as Dropbox or Github. See miktex.org