

Advanced Machine Learning (DA633E) Labs resolution

You report should have following instructions: (you can upload a pdf file (contains explanations) + your code in a zip file or a colab notebook with > explanation including the following topics)

- How to run your code (especially, if not using notebooks and using other libraries)
- Describe your dataset and explain whether it's a classification or regression problem.
- Any library you are using that we did not cover, so I can download the right one (with the right version!). Appropriate pip commands in your notebooks > would be the best.
- Custom folder structures. If I need to download some specific folder from drive or something. Please make sure that how to connect it in the code is clear > on how I can change it to assess.
- Anything you think that must be written to complement the code and the report

The labs are available at <https://github.com/notPlancha/aml-homework>.

Setup

For these labs, [pixi](#) was used to manage dependencies and their versions, leveraging the conda ecosystem. To install:

```
# Windows
powershell -ExecutionPolicy ByPass -c "irm -useb https://pixi.sh/install.ps1 | iex"
# Linux & macos
curl -fsSL https://pixi.sh/install.sh | sh
# You might need to restart your terminal or source your shell for the changes to take effect.
```

To run the code:

```
pixi install # installs the environment with the dependencies in pixi.toml
pixi run jupyter lab # vscode detects the environment automatically
```

Alternatively, you can load it conda:

```
conda env create --name envname --file=environment.yml
```

Finally, you can just use pip. The versions are specified in `pixi.toml`.

The labs are inside `src/`. The repository (which includes everything besides the data) is available at <https://github.com/notPlancha/AML-homework>

Datasets used

Lab 1, 2 and parts of lab 4

The dataset used is [Sign Language MNIST](#). It is composed of grayscale pictures of handwritten letters (in csv), with the objective of classifying the letters.

Complementary task: segmentation

The dataset used is Ultralytics's COCO8, "a small, but versatile object detection dataset composed of the first 8 images of the COCO train 2017 set" [Ultralytics/COCO8](#). The COCO Object Detection Task is "designed to push the state of the art in object detection forward, [...] containing more than 200,000 images and 80 object categories, [...] annotated with a detailed segmentation mask" [COCO 2017 Object Detection Task](#).

Complementary task: finetuning any of the huggingface models

The dataset used is RecipeNLG, a novel dataset of cooking recipes, introduced by Bien et. al. in (RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation)[<https://aclanthology.org/2020.inlg-1.4/>], available in <https://recipenlg.cs.put.poznan.pl>.