

Final Project Submission Guideline

Release Date: November 13, 2025

1 Submission Overview

This semester, we will evaluate your code on the Hugging Face platform. The link for two leaderboards:

1. Project A: IMG2GPS
2. Project B: News Headline Classifier

For both project tracks, submit the following files:

- `model.py`: your model implementation (must be importable and instantiable).
- `preprocess.py`: your preprocessing function(s).
- `model.pt` (optional): your trained weights; required if your model needs them to evaluate.

When you submit, include your Group ID and an Alias string. By **December 15, 2025**, each team must submit their code, collected data and a 5-page project report.

For your collected data, create a Hugging Face Dataset and include its link in your report. You can find the tutorial here. [Huggingface Tutorial](#)

We provide submission templates in python and local evaluation script similar to our backend. We also provide a small reference data for Project A. You can download the resources here: [New Resources](#)

2 Environment

Our backend environment includes the following:

- `numpy`, `pandas`, `torch==2.9.1`, `torchvision`, `scikit-learn`, `opencv-python`

While we strongly encourage implementations with Pytorch, people who want to use other frameworks are welcome to leave an Ed post to let us know. If your code requires a library not in this list and no simple workaround exists, leave an Ed post as well; we will decide whether to add it to the environment.

3 Data Streams and Contracts

The backend runs your submitted `preprocess.py` and `model.py` with a strict I/O contract. Do not attempt to cheat the leaderboard. TAs will have access to your source code.

3.1 Project A: Img2GPS

Data. A CSV with images under our backend. The form is similar to our provided `reference/` folder (e.g., `reference/metadata.csv`) containing:

- Image path column (one of: `image_path`, `filepath`, `image`, `path`, `file_name`)
- Latitude column (one of: `Latitude`, `latitude`, `lat`)
- Longitude column (one of: `Longitude`, `longitude`, `lon`)

Preprocess. Your `preprocess.py` must expose:

```
def prepare_data(csv_path: str) -> (X, y)
```

- `X`: sequence/array/tensor of inputs suitable for `model.predict(batch)` or `model(batch)`.
- `y`: sequence/array/tensor of target pairs [lat, lon] in *degrees* (raw, not normalized).

If you use normalization, **hard code the stats** in your `model.py`.

Model. Your `model.py` must provide either:

- `get_model()` -> `model_instance`, or
- a class named `Model` or `IMG2GPS` that can be instantiated without arguments.

At inference, the backend calls `model.predict(batch)` if available; otherwise it calls `model(batch)`. Outputs must be [lat, lon] in degrees (raw, not normalized).

Weights. If you submit `model.pt`, we will load it into your model via `torch.load(..., map_location='cpu')` and a robust `load_state_dict` routine. Ensure your checkpoint keys match your model's parameter names.

3.2 Project B: News Headline Classifier

Data. A CSV similar to `url_data_only.csv` under our backend.

Preprocess. Your `preprocess.py` must expose:

```
def prepare_data(csv_path: str) -> (X, y)
```

- `X`: sequence/array/tensor of inputs suitable for your model.
- `y`: sequence of labels (strings or integer class ids).

Model. Your `model.py` must provide either:

- `get_model()` -> `model_instance`, or
- a class named `Model` or `NewsClassifier` that can be instantiated without arguments.

At inference, the backend calls `model.predict(batch)` if present; otherwise it calls `model(batch)` and falls back to `argmax` over the final dimension if a tensor of logits is returned.

Weights. If you submit `model.pt`, the backend will load it as a state dict into your model before evaluation.

4 Evaluation Metrics

4.1 Project A: Img2GPS

We evaluate predictions strictly against the *raw latitude/longitude* from the CSV (degrees). Let predicted pairs be $\hat{y}_i = (\hat{\phi}_i, \hat{\lambda}_i)$ and ground-truth be $y_i = (\phi_i, \lambda_i)$.

Leaderboard metric: *Average Haversine distance (meters)*. Lower is better.

$$d(a, b) = 2R \arcsin \left(\sqrt{\sin^2 \frac{\Delta\phi}{2} + \cos \phi_1 \cos \phi_2 \sin^2 \frac{\Delta\lambda}{2}} \right), \quad (1)$$

where $R = 6,371,000$ m, $\Delta\phi = (\phi_2 - \phi_1)$ in radians, and $\Delta\lambda = (\lambda_2 - \lambda_1)$ in radians. We report $\frac{1}{N} \sum_i d(y_i, \hat{y}_i)$.

4.2 Project B: News Headline Classifier

Let predictions be \hat{y}_i and ground-truth labels be y_i . We compute *accuracy*. If your model outputs integer ids while labels are strings (or vice versa), we apply a robust 2-class mapping when possible; otherwise we compare as strings.

5 End-to-End Flow

Project A (Img2GPS)

1. Read CSV (e.g., `test/metadata.csv`).
2. `preprocess.prepare_data(csv)` returns (X, y) .
3. Instantiate `model` from `model.py`.
4. Load `model.pt` (if provided).
5. Run batched inference via `model.predict(batch)` or `model(batch)`.
6. Compare predictions to *raw lat/lon* from the CSV; compute metrics.

7. Write JSON results and update leaderboard (backend).

Project B (News Headline Classifier)

1. Read CSV (e.g., `url_test.csv`).
2. `preprocess.prepare_data(csv)` returns (X, y) .
3. Instantiate `model` from `model.py`.
4. Load `model.pt` (if provided).
5. Run batched inference via `model.predict(batch)` or `model(batch)`.
6. Compare predictions to labels from `y`; compute accuracy.
7. Write JSON results and update leaderboard (backend).

6 Local Sanity Checks (Strongly Recommended)

We provide lightweight local evaluators to mimic backend behavior:

- Project A: `eval_project_a.py`
- Project B: `eval_project_b.py`

Run them with your paths to verify `preprocess.py`, `model.py`, and optionally `model.pt` before submission.

7 Packaging and Submission Tips

- Keep imports standard; avoid non-listed dependencies unless approved.
- Ensure `preprocess.prepare_data` and your model entry points exist and match the contracts above.
- If providing `model.pt`, ensure its keys match your model parameters.
- **Project A:** outputs must be in degrees and will be compared against raw labels.
- Include your Hugging Face Dataset link in the report by the deadline.