General guidance for every project group

1. **Choose one clear task + dataset first**
    - Decide **exactly what you will predict** (labels, regression target, reconstruction, ...) **and from which dataset**.
    - Check that the dataset truly contains the labels you need (several proposals paired the wrong task with the dataset).
    - Write down: "Input → Target → Metric".
2. **Build a *minimal, end-to-end baseline* within the first week**
    - A small, well-known model (EfficientNet-B0, Xception, 3-layer GCN, U-Net-small, YOLOv8-nano, etc.).
    - One-pass data loader, single training script, single evaluation script.
    - Run on a subset of data so the whole loop finishes in **< 1 h**.
    - Save the baseline metrics and a few qualitative outputs (images, audio, Grad-CAM, etc.)—they will be your reference.
3. **Keep compute realistic**
    - Free Colab / Kaggle GPU ⇒ T4 16 GB; design your model to stay under 8 GB VRAM and ≤ 4 h GPU total per experiment.
    - Prefer light variants ("-tiny", "-nano", "-small") or frozen backbones + small heads; use AMP (`torch.cuda.amp`) and small batch sizes.
4. **Automate pre-processing once**
    - Write a `prepare_data.py` (or notebook) that downloads, cleans, splits and caches the dataset into a format your model can load quickly.
    - Version the resulting CSV / JSON split files so results are reproducible.
5. **Fight class imbalance up front**
    - Inspect the label histogram; if the largest class is > 5× the smallest, use class weights**,** focal loss, or oversampling.
    - Report macro metrics (macro-F1, macro-AUPRC) or balanced accuracy, not only overall accuracy.
6. **Use subject-/patient-/speaker-wise splits where leakage is possible**
    - Never mix frames or slices from the same individual across train, val, test.
    - For time-series, split by time blocks (past weeks for train, future weeks for test).
7. **Track the right metrics. Examples:**

Task	**Primary metric**
Binary medical diagnosis	AUROC ± 95 % CI
Multi-class imbalance	Macro-F1 + confusion matrix
Object detection	mAP@[.5:.95] + per-class P/R
Reconstruction / SR	SSIM + PSNR
Regression (affinity, traffic)	RMSE + MAE

    - **Plot learning curves**; early-stop on the primary metric, not on loss alone.
8. **Document experiments succinctly**
    - Each run: dataset + commit hash + config (lr, batch, epochs, augmentation, seed) + metrics.
    - Keep a `results.csv` or a simple README log; it saves hours when comparing variants.

9. **Add one interpretability element**
   - Grad-CAM, SHAP, attention heat-map, reconstruction error map—depending on your modality.
   - Show at least one correct and one failure example; these often expose data or label issues.
10. **Iterate surgically**
    - Change one thing at a time (learning-rate sweep, backbone swap, new loss).
    - If a new idea adds < 2 pp to your main metric, keep it; otherwise revert quickly.
11. **Plan your GPU budget**
    - Baseline (week 1) → 1 h
    - Two–three controlled improvements (weeks 2–3) → 5 h
    - Final model + ablations + visuals (week 4) → 3 h
    - Leave slack for bugs and paper/poster figures.

Remember that you can use PCs from 1.3.30 for training. Not all, but some of them have GPU Nvidia RTX 3060.

Following these steps ensures every group delivers a working, reproducible pipeline early, keeps within compute limits, and leaves time for meaningful experimentation instead of last-minute debugging.