Day 1: Pathology Identification with off-the-shelf LLMs

And running it locally

Dr. Marie Hoffmann, 02. December 2025

Learning Objectives

- Use the Python OpenAI API module to interact with generative LLMs

- Use a locally hosted LLM via LM Studio

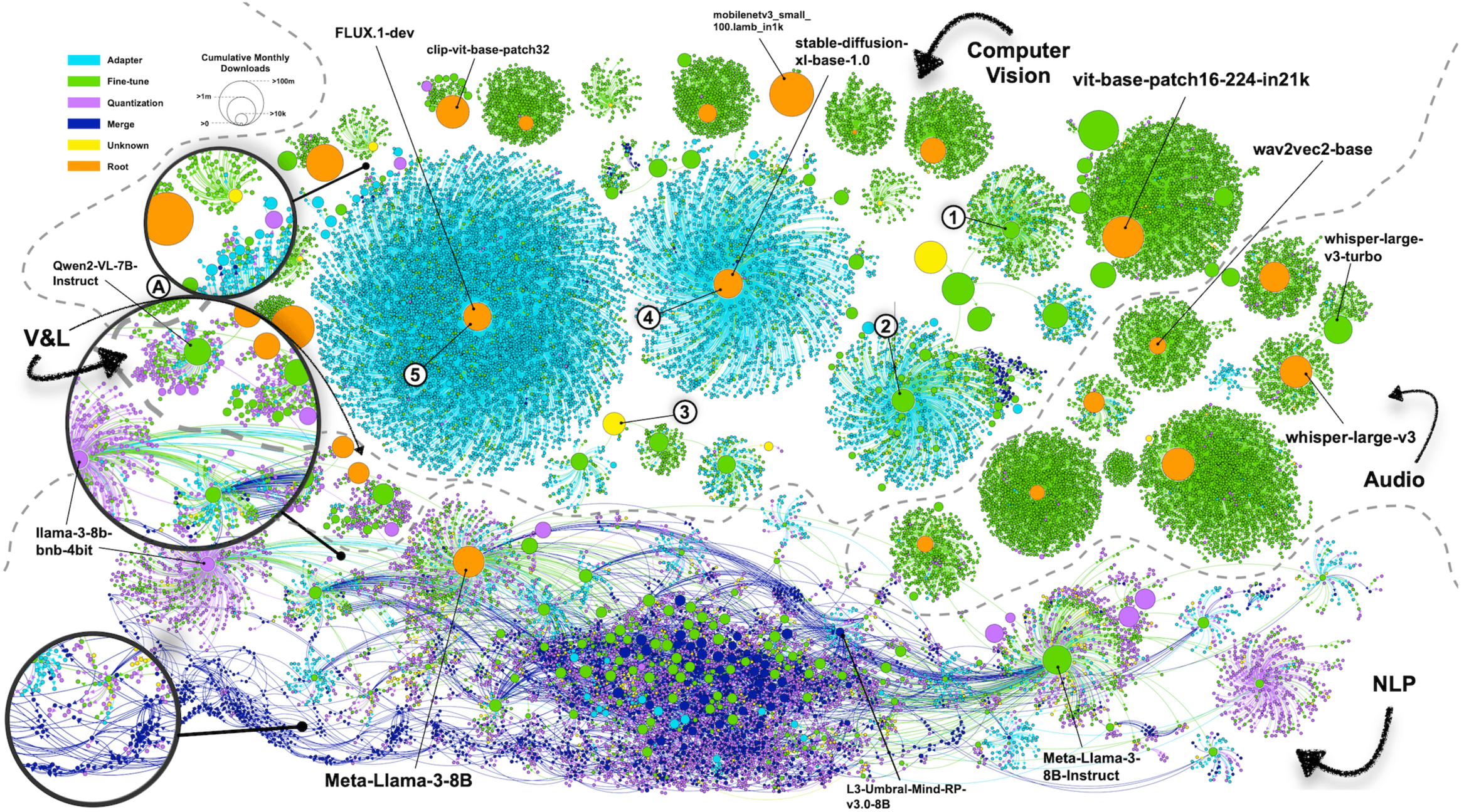
- Learn a typical workflow of

    - prompt templating

    - structured LLM output generation

    - label extraction

- Basic logging, checkpointing, metric tracking



State 10th Dec 2025: 2,270,081 models

Source: https://horwitz.ai/model-atlas

Prerequisites

What you need (all free)

1. Slides available on GitHub, get them:

```
git clone https://github.com/aieoa/workshop_llm_classifier
```

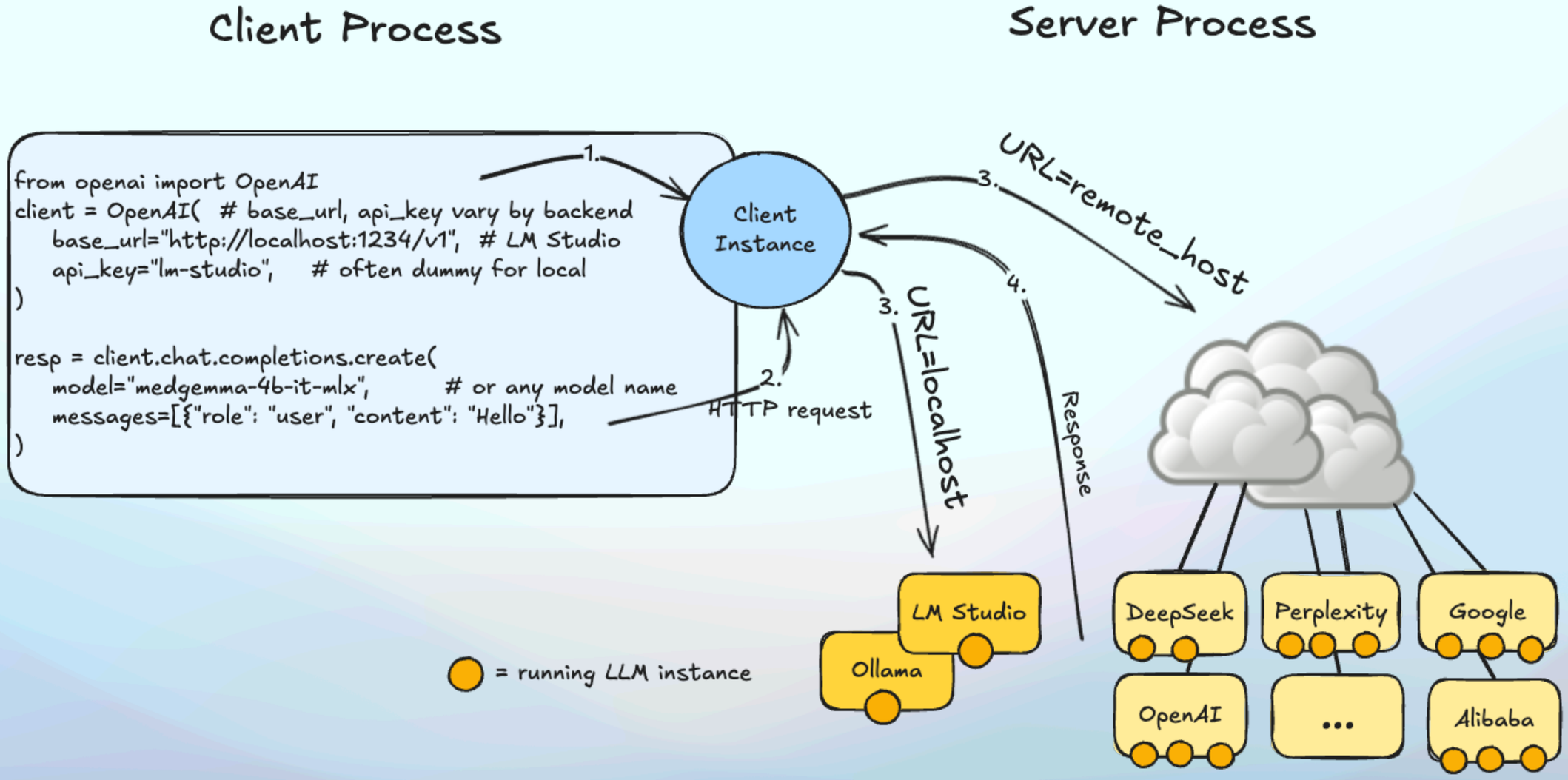
2. Install LM Studio (MacOS, Windows, Linux)

3. Account on HuggingFace for model download

4. Python >3.10

OpenAI API: One Interface

For local and remote serving



The Task: Pathology Classification with an LLM

Dataset

- Chest X-rays paired with **diagnostic text sections**

- Each data point labeled either “no findings” or at least one out of 13 possible pathology classes ("Lung Lesion", "Pneumonia", etc.)

- Derived from CheXpert fully available here

  - Original: 224,316 chest radiographs of 65,240 patients

The Task: Pathology Classification with an LLM

Example: first patient

Column section_findings
X_test_0

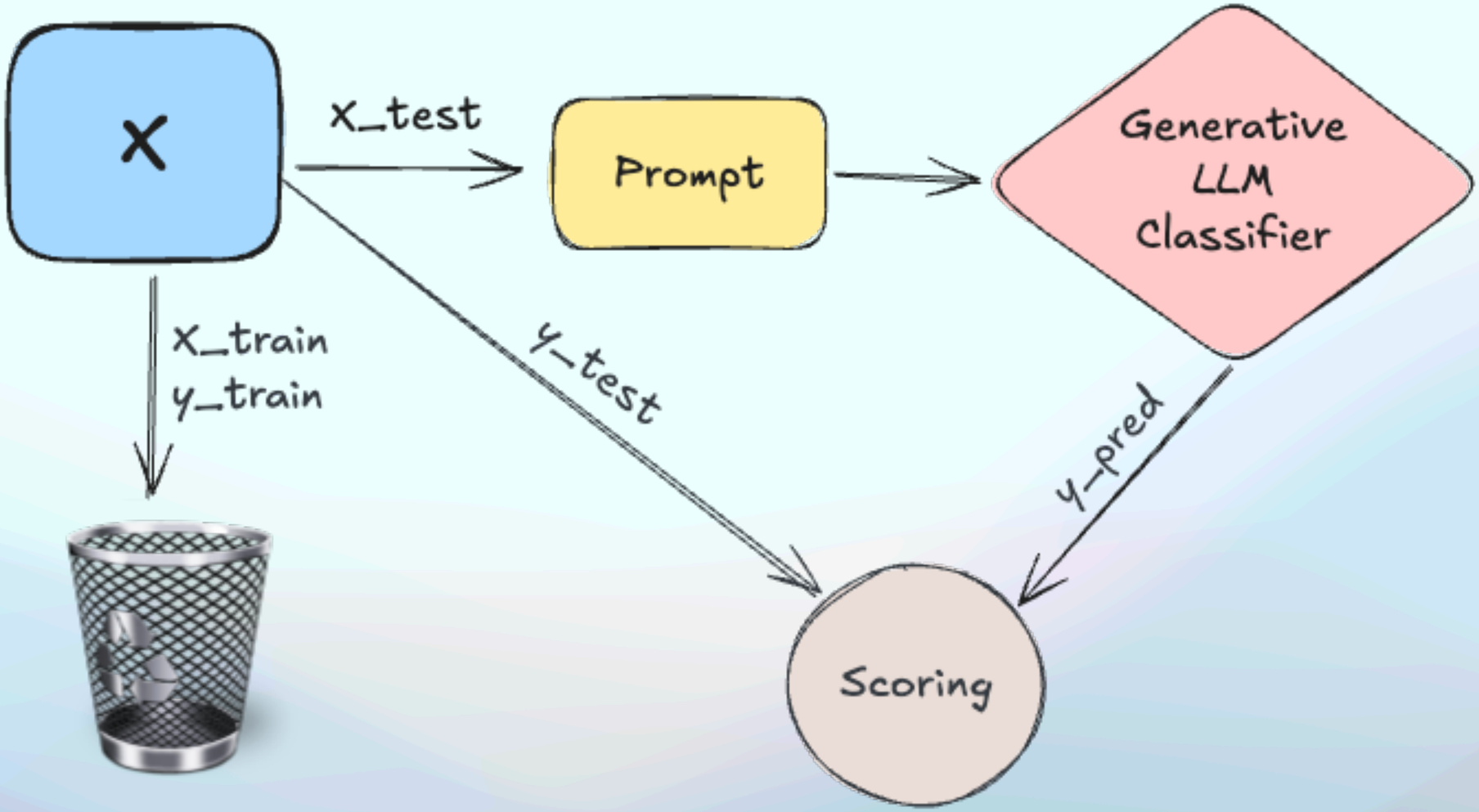
'Unchanged position of the left upper extremity PICC line. Again seen are surgical clips projecting over the right hemithorax. The cardiomediastinal silhouette is stable in appearance. Increased stranding opacities are noted in the left retrocardiac region. Subtle stranding opacities in the right upper lung zone are unchanged.. There are no pleural or significant bony abnormalities. Absence of the right breast shadow compatible with prior mastectomy.'

y_test_0

Enlarged Cardiomediastinum	Cardiomegaly	Lung Opacity	Lung Lesion	Edema	Consolidation	Pneumonia	Atelectasis	Pneumothorax	Pleural Effusion	Pleural Other	Fracture	Support Devices	No Finding
0	0	1	0	0	0	0	0	0	0	0	0	1	0

The Task: Pathology Classification with an LLM

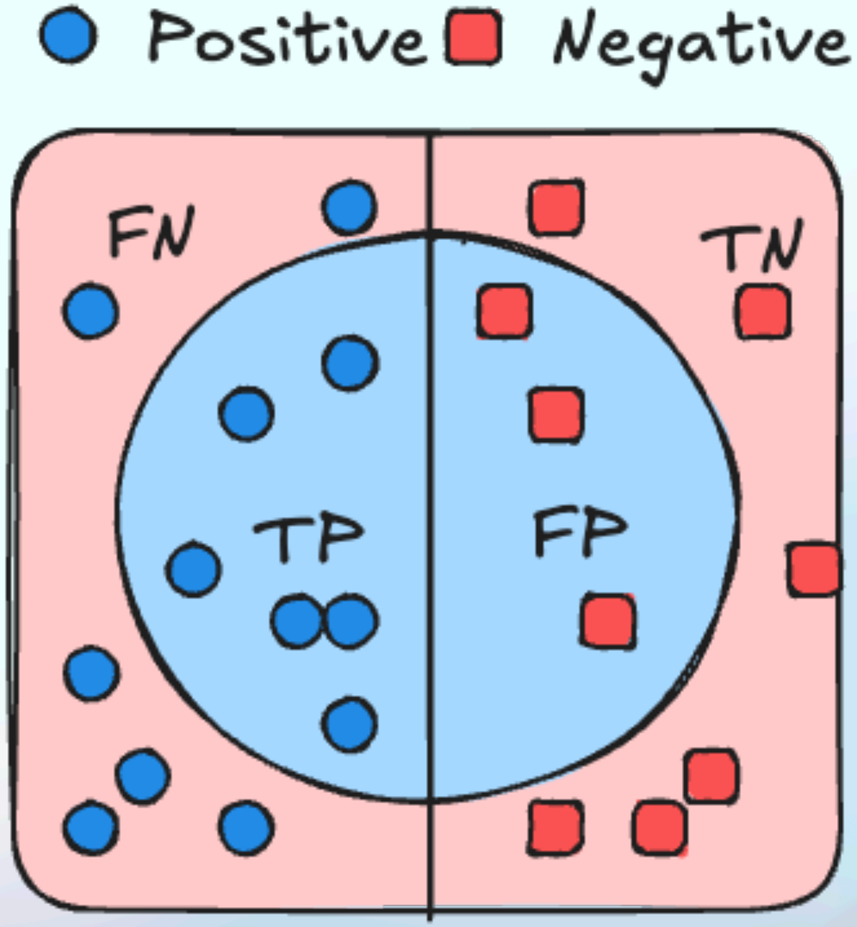
Experimental Design



Scoring Metrics

Multilabel Classification

- TP: true positives
- FP: false positives
- FN: false negatives
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 = 2 * Recall * Precision / (Recall + Precision)

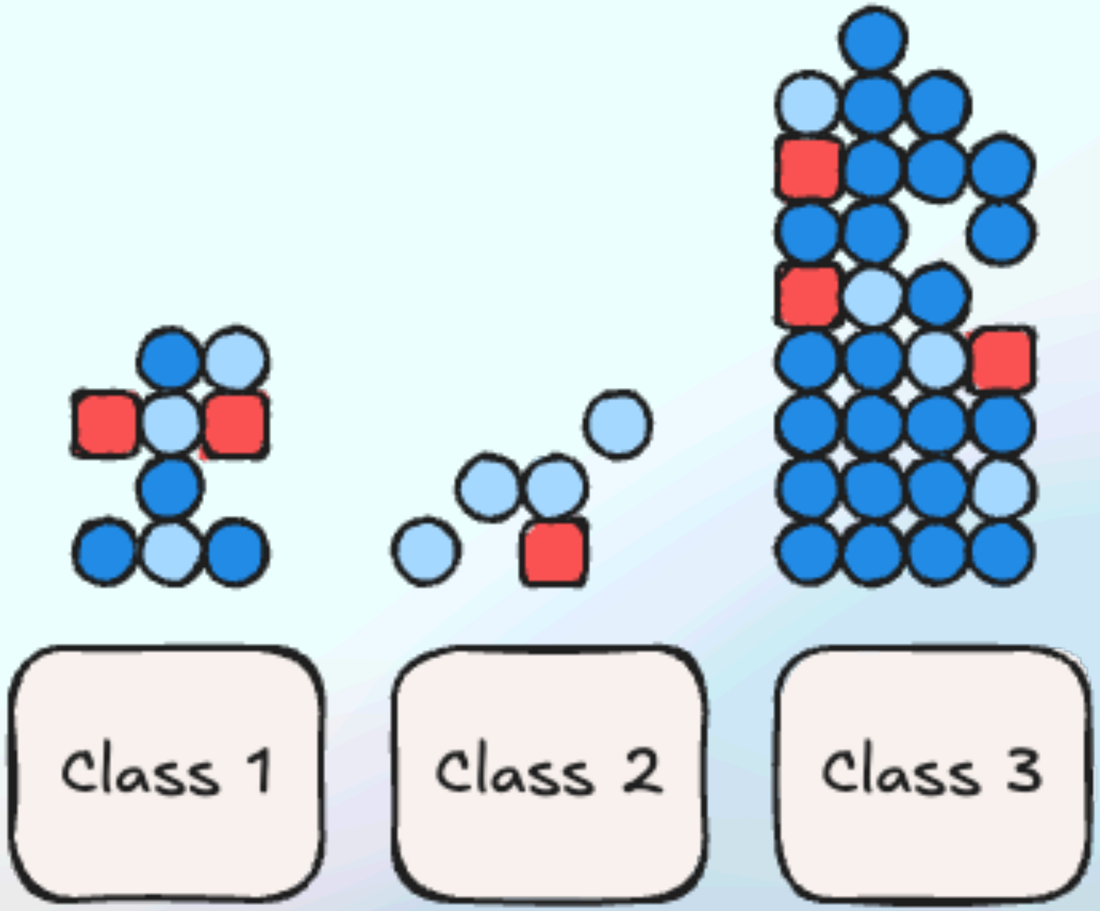


Scoring Metrics

Multilabel Classification

Average Methods:

1. Micro - global pooling of TP, FP, FN
    - Precision = TP / (TP + FP)
    - Recall = TP / (TP + FN)
    - F1 = 2 * Precision * Recall / (Precision + Recall)

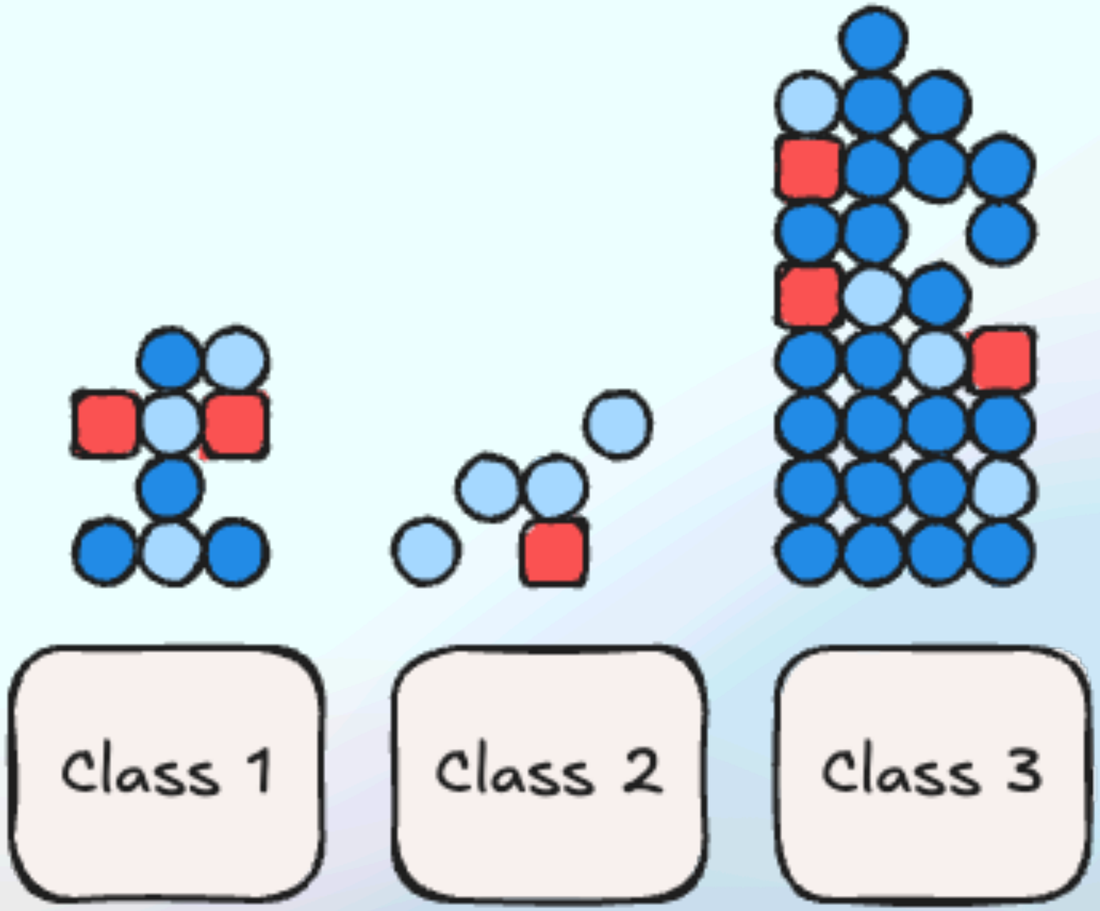


Scoring Metrics

Multilabel Classification

Average Methods:

1. Micro - global pooling of TP, FP, FN
2. Macro - per class scores are averaged
   - mean(metric(c_i)) for i in [1:3]

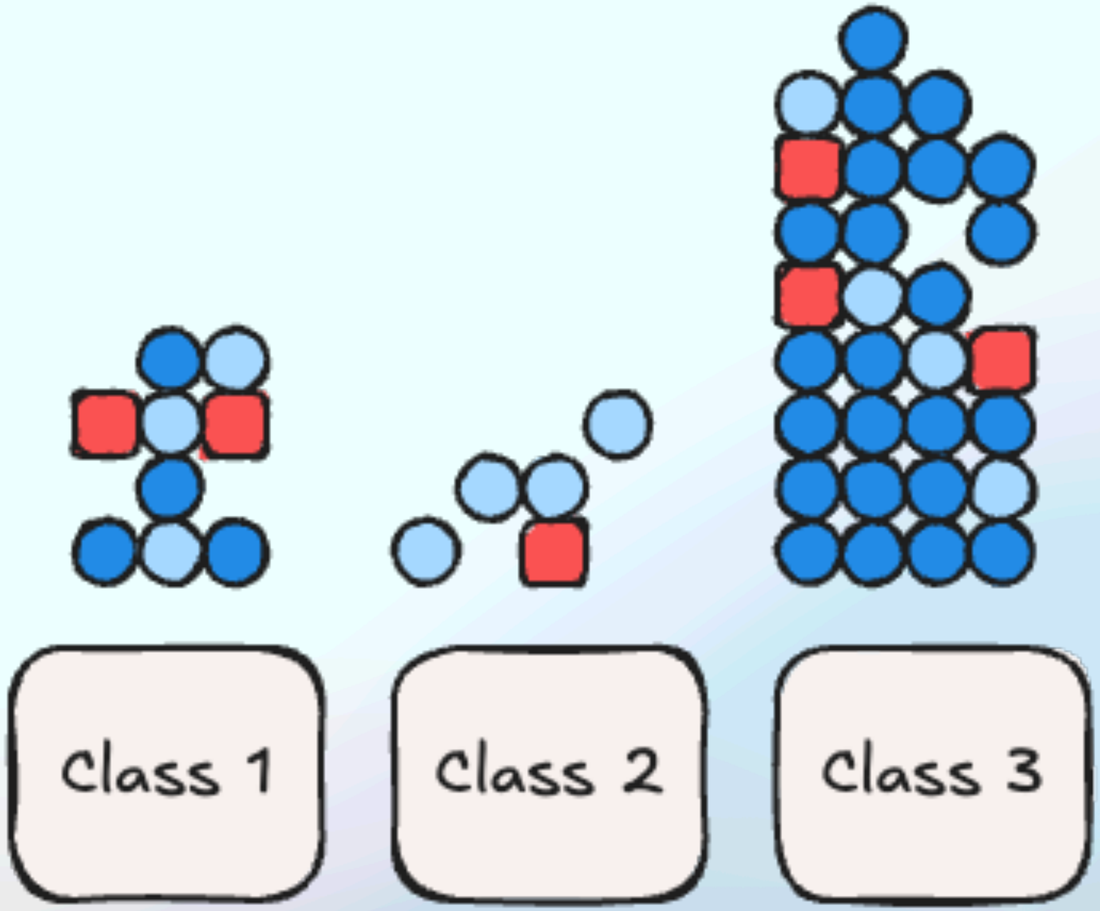


Scoring Metrics

Multilabel Classification

Average Methods:

1. Micro - global pooling of TP, FP, FN
2. Macro - per class scores are averaged
3. Weighted - per class scores are weighted by class support and averaged
    - mean(|c_i| / n * metric(c_i)) for i in [1:3] and n number of samples



Hands-on: Get the Classifier Code

Pull from repo, create environment in Terminal (Windows: Powershell)

1. Clone repository with notebook

   $ git clone https://github.com/aieoa/workshop_llm_classifier

2. Navigate to cloned repository

   $ cd ~/git/workshop_llm_classifier (Windows: $ cd C:\Users\<you>\git\workshop_llm_classifier)

2. Create & activate virtual environment

   $ python -m venv .env

   $ source .env/bin/activate (Windows: $.\.env\Scripts\Activate.ps1 or .\.env\Scripts\Activate.ps1)

3. Install (or update) requirements

   $ pip install -r requirements.txt

4. Register environment as kernel

   $ python -m ipykernel install --user --name .env --display-name "Python (llm_env)"

5. Open notebook and solve the task

   $ jupyter notebook

Now Hands on

Pull a Model & Run Server

Till 04:45 pm, then wrap up

1. Open LM Studio
2. Go to Discover (magnifying glass) tab
3. Choose a smallish model (4-8 GB) and click Download
4. After download, verify that it appears under “My Models” / “Local Models”
5. Go to Developer tab, select model, start serving
6. Check if serving URL identical to the one in the notebook

Outlook

Usage of remote frontier models

1. Data must be anonymized
2. Re-use core code, but change **URL**, possibly provide **additional keys**
3. Some provide **tooling**, eg, for structured JSON output: `response_format`