Day 2: Pathology Identification with RAG-augmented off-the-shelf LLMs

run locally

CHARITÉ
UNIVERSITÄTSMEDIZIN BERLIN

Dr. Marie Hoffmann, 12. December 2025

Learning Objectives

- Use the Python OpenAI API module to interact with generative LLMs
- Use a locally hosted LLM via LM Studio
- **Build and cache vector store**
- Learn a typical workflow of
    - prompt templating **with augmentation**
    - structured LLM output generation
    - label extraction
- Basic logging, checkpointing, metric tracking

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

Database vs Vectorstore

What do you need?

Database vs Vectorstore

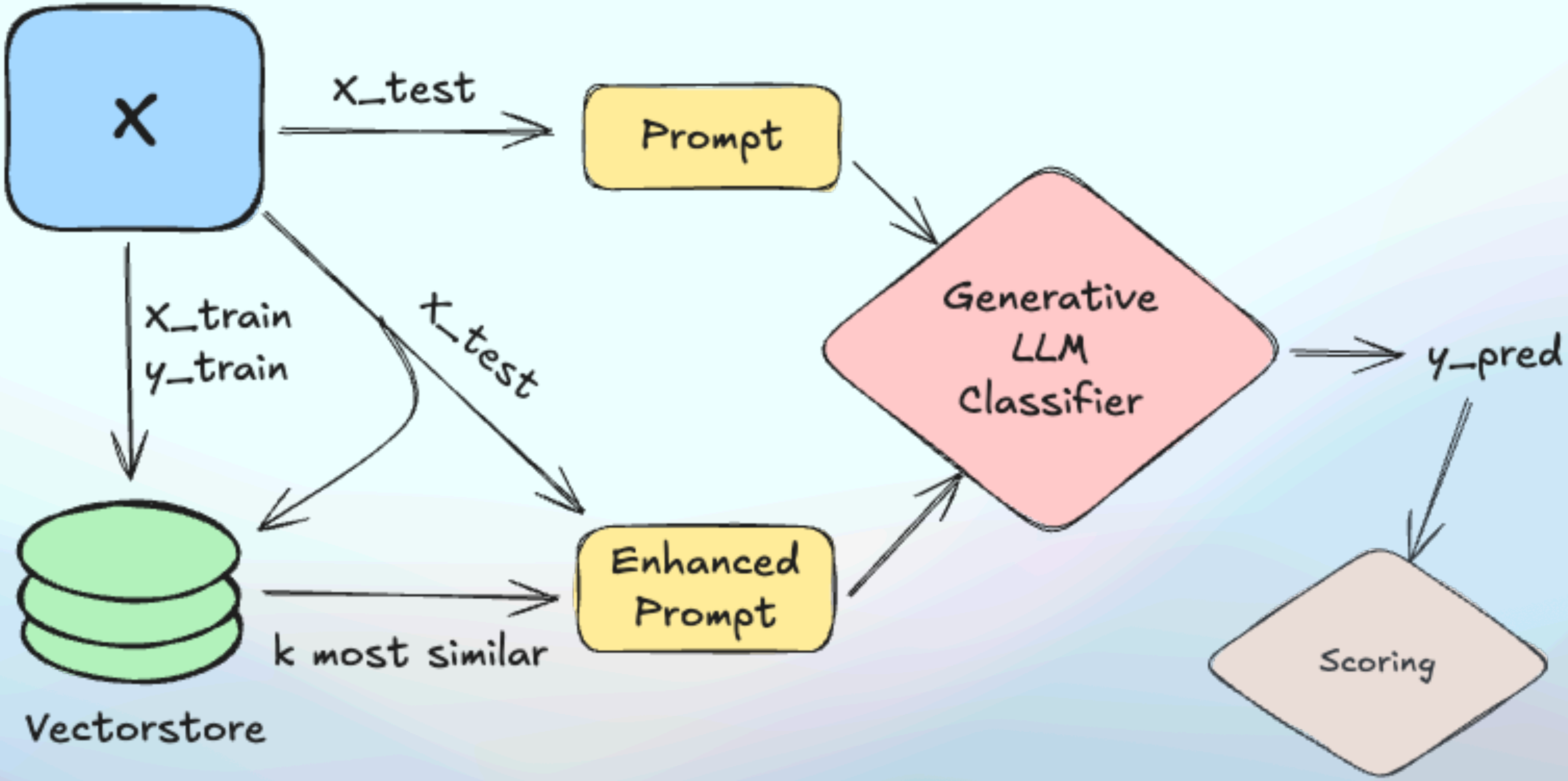
What do you need?

- Primary Data: Structured rows, JSON docs, numeric/categorical cols
- Semantics: Explicit schema & constraints (types, keys)
- Queries like "all patients with EF < 35% and NYHA ≥ II, ordered by admission date".
- ACID Properties (Atomicity, Consistency, Isolation, Durability)
- Examples: Postgresql, MySQL

- Primary Data: Dense vectors (embeddings), often plus small metadata
- Semantics: Implicit semantics encoded in embeddings
- Application: similarity search, semantic retrieval, retrieval augmentation
- Examples: FAISS, Qdrant, Milvus, Weaviate, Chroma

The Task: Pathology Classification with an LLM

Experimental Design



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 using Python >= 3.10

```
$ python -m venv .env
$ source .env/bin/activate  (Windows: $ .\.venv\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 llm_env --display-name "Python (llm_env)"
```

5. Open notebook and solve the task

```
$ jupyter notebook
```

Hands-on: Get Precomputed Vectorstore

Or partially compute locally in Notebook Day 2

1. Get via Dropbox link

   https://www.dropbox.com/scl/fi/9x6wy4s7ug3l0d70rncxl/vectorstore.zip?rlkey=1mb8hpmekxw8uufmz5og5iw0f&st=eu0t47t4&dl=0

2. Place unzipped version in your workshop_llm_classifier repository

```
/workshop_llm_classifier
            |— .env
            |— Notebook Day 1.ipynb
            |— Notebook Day 2.ipynb
            |— slides
            |— vectorstore
```