Project handout — Data Acquisition, Processing and Mining for AI (M1 AI, S7)

The current handout and data-card template are published on the course's **IM2AG Moodle** page: https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=721.

Read it before Milestone M0, due before the Week 3 lab.

---

1. What you are building

In a fixed pair, you will build one pipeline that acquires data from an API, cleans and explores it, and produces an interpretable mining result. Most labs end with a short task that applies the week's method to your project. Pairs are fixed once you register a scope at M0.

2. Milestones

The project has four Moodle submissions. M0–M2 are marked complete/incomplete; M3 is graded out of 20.

Milestone	Due	You submit on Moodle	Grading
M0	before the W3 lab	Registration form	pass/fail
M1	before the W4 lab	Acquisition log + script + Parquet shapes	pass/fail
M2	before the W6 lab	Data card + EDA notebook	pass/fail
M3	defense in the W8 lab; files W8 + 1 week	Pipeline notebook + report + data card	graded /20

Submission contents:

- **M0** — both names, source, scope in one sentence, and confirmation of the required accounts: GitHub for everyone; a TMDB key or Twitch app with 2FA only if you chose that source. You create the GitHub token in Week 3.
- **M1** — `acquisition_log.json`, the acquisition script or notebook, and the shape of each Parquet file. Do not submit the data itself. A revised scope goes in the log.
- **M2** — your one-page data card (`data_card_template.md`) and your EDA notebook. The cleaned dataset stays on your machine.
- **M3** — final pipeline notebook, report of about 3 to 5 pages, and updated data card. The 5-minute defense is in the Week 8 lab; files are due one week later.

How a milestone is checked

- Submit by **20:00 on the day before the relevant lab**.
- A missing M0, M1 or M2 submission costs **1 point out of 20** on the final project mark.
- If an external problem blocks you, submit a sentence explaining the problem and what you tried.
- The M3 defense presents an almost finished pipeline. You may change it with the final submission files one week later.

3. Choosing a source and scope

At M0, choose one source from the list below and define the exact slice you will collect: entities, time window and filters. Scopes must be distinct even when two pairs use the same source.

Look at the source before you register. Open one endpoint in a browser and read what comes back: which fields are actually populated, and how much data one request returns. Every source below except TMDB and Twitch answers a browser request without credentials (GitHub: 60 per hour, enough to look).

Example registered scopes:

- "GitHub: issues of CNCF orgs, 2024–2026",
- "Hacker News: front-page stories and their top-level comments, January–June 2026".

Record the scope precisely.

The M0 scope is a first estimate. You may revise it once, at M1: put the new scope in `acquisition_log.json` with one sentence on what the first real pull showed. After M1 the scope is fixed.

4. Curated project sources

Every source supports an acquire → clean → mine pipeline. Use the suggested mining steps to judge its fit. Run bulk acquisition **outside** class hours.

Crossref REST API — https://www.crossref.org/documentation/retrieve-metadata/rest-api/

- *Auth:* none. Include your address in `mailto=` and `User-Agent` to use the polite pool: 3 requests/s, 3 concurrent.
- *Data:* scholarly metadata: titles, journals, authors, funders and citation counts. It uses cursor pagination (`cursor=*`).
- *Mining step:* clustering works by journal / year / citation-count features; **co-authorship** co-occurrence → itemsets.

GitHub REST API — https://docs.github.com/en/rest

- *Auth:* **personal access token (PAT) required** — unauthenticated requests are limited to 60/hour per IP, unusable for real work in class.
- *Data:* repositories, issues, users and events.
- *Mining step:* clustering repos by language/activity features; label co-occurrence on issues → itemsets.

Hacker News API — https://github.com/HackerNews/API

- *Auth:* none, no rate limit.
- *Data:* hierarchical stories, comments and users, served via Firebase. Endpoints end in `.json`, e.g. https://hacker-news.firebaseio.com/v0/topstories.json or https://hacker-news.firebaseio.com/v0/item/8863.json.
- *Mining step:* clustering stories (TF-IDF on titles, score/time features).

Open-Meteo — https://open-meteo.com/en/docs

- *Auth:* none for non-commercial use.
- *Data:* forecast and historical weather at hourly or daily resolution.

- *Mining step:* clustering locations/days by weather profile.

TMDB API — https://developer.themoviedb.org/docs/getting-started

- *Auth:* free API key — register with a TMDB account.
- *Data:* movies, television programmes and people.
- *Mining step:* clustering films by genre/metadata; genre co-occurrence → itemsets.

MediaWiki Action API (Wikipedia) — https://www.mediawiki.org/wiki/API:Main_page

- *Auth:* none.
- *Data:* page content, search, geosearch and revisions. `api.php` needs a query string: the bare https://en.wikipedia.org/w/api.php shows the API help page, not data. Try https://en.wikipedia.org/w/api.php?action=query&list=search&srsearch=grenoble&format=json (without `&format=json` you get the HTML result browser).
- *Mining step:* clustering articles (TF-IDF); category co-occurrence → itemsets.

data.gouv.fr API — https://guides.data.gouv.fr/api-de-data.gouv.fr/reference

- *Auth:* none for read access. A browsable endpoint: https://www.data.gouv.fr/api/1/datasets/?page_size=1.
- *Data:* the French open-data catalog. Choose either (a) catalog metadata—datasets, organizations and tags—or (b) a dataset discovered through the catalog.
- *Mining step:* (a) tag co-occurrence → itemsets, organization clustering;
  (b) depends on the dataset you choose.

Twitch API (Helix) — https://dev.twitch.tv/docs/authentication/getting-tokens-oauth/

- *Auth:* OAuth2 client-credentials flow (token endpoint https://id.twitch.tv/oauth2/token, which is **POST-only** — opening it in a browser returns 404); registering an app requires a Twitch account with **two-factor authentication enabled**.
- *Data:* games, streams and clips retrieved with a server-to-server token.
- *Mining step:* clustering games/streams by viewer counts, tags, schedule features; tag co-occurrence → itemsets.

5. Rules for every source

- **Use the API, not scraping.** Scraping is only taught for illustration purposes.
- **Run bulk acquisition outside class.** Test the script in the lab, then run the full pull at home.
- **Never submit credentials.** Read tokens and keys from environment variables or another unsubmitted local file; do not place them in notebook cells or outputs.
- **Run from a fresh kernel.** The final notebook must run top-to-bottom with no manual steps or hidden state. It may load the raw Parquet produced by your acquisition code.
- **Minimize personal data.** The report and data card must state what personal data you collected, what you excluded, and why the remainder is justified.

6. M3 checklist

Submit one week after the Week 8 lab:

1. **Pipeline notebook** that runs from cached raw Parquet to final results on a fresh kernel.

2. **Report** of about 3 to 5 pages: acquisition, cleaning decisions, EDA findings, mining method, interpretation and the GDPR/data-minimization statement.
3. **Updated one-page data card.**

The **5-minute defense** is held in the Week 8 lab. Both partners must be able to explain any part of the pipeline.