Dataset & Task Instructions

4.1 Dataset Overview

You are given a sample of real-world **project records** in CSV format.
Each row is **one record** that may or may not be a true infrastructure project.

Key columns (simplified):

- **Basic description & source**
  - `description` – free text describing the project or program
  - `source` – where the record comes from (DODGE, TxDOT, Infrastructure Canada, EU portal, US Grants, etc.)
  - `title` – short project/program/title name
  - `url` – link to the original source page
- **Location & geography**
  - `country_name`, `state_name`, `city_name`, `county_name` – location fields
  - `location` – sometimes free-text place description
- **Budget & currency**
  - `currency` – currency code (USD, CAD, PLN, EUR, …)
  - `budget` – numeric value from source (may be raw / noisy)
  - `ml.budget_ccy_bot.results.budget_usd` – model-derived budget in USD (when available)
  - `budget_label` – text label (e.g., “Estimated Construction Cost”, “Total Project Cost”, “overall_budget”, “Net revenues increase”)
- **Project status**

    - `ml.status_simplified.results.identified_status` – simplified status (Closed, Under Construction, Failure, Unknown, etc.)
    - `status` – human-readable status (bid results, completed, ongoing, abandoned, etc.)

- **Timestamps**
    - `timestamp_label` – describes which timestamp is used (e.g., `publish_date`, `estimated_start_date`, `project_start_date`, `info_updated_date`)
    - `timestamps` – JSON-like field with multiple date keys:
        - e.g. `{'estimated_start_date': '2016-09-09', 'estimated_end_date': '2017-06-02', ...}`
    - `timestamp` – a single extracted date (string like `2023-04-10`)

- **Sector & tags**
    - `sector` / `subsector` – sector labels if available (e.g., “Drinking Water and Wastewater”, “Manufacturing”, “Public Transit”)
    - `ml.sector_tags_method.results.level1_sector` – higher-level sector tags (e.g., Road, Water Supply and Storage, Commercial, etc.)

- **Procurement & entities**
    - `ml.procurement_method_redo2.results.method` – delivery / procurement method (Design-Bid-Build, Design & Build, PPP, etc.)
    - `ml.entity_enrich_redo1.results.*` – multiple columns marking presence of different entity types:
        - `ai_entities__government`, `ai_entities__epc`, `ai_entities__banks`, `ai_entities__infrastructure_investors`, `ai_entities__architecture_design`, etc.

Some records are clearly **infrastructure assets** (roads, bridges, water plants, hospitals, etc.). Others are **not** (scholarship programs, corporate revenue, regional education projects, research grants).

Your job is to **clean**, **filter**, and **enrich** this dataset into a high-quality "infrastructure projects" dataset and then analyze it.

4.2 Objective of the Task

You will:

1. **Clean the dataset**
2. **Filter to true infrastructure projects only**
3. **Create a binary variable for "megaproject"**
4. **Use web browsing + (optional) genAI tools to fill missing attributes**
5. **Produce a final, enriched dataset** (CSV)
6. **Write a short PDF report** with:
    - Failure rates
    - Analysis by delivery/procurement method
    - Key uncertainties and data gaps

You will submit:

- A **folder** inside the provided Google Drive
- Your **code** (Colab notebook or script)
- Your **final cleaned dataset** (CSV)
- Your **PDF report**

4.3 Step-by-Step Instructions

Step 1 – Create Your Working Folder

1. Open the Google Drive link:

   `https://drive.google.com/drive/folders/1iFHfid3SYIq7JzvcPi3TSLGUQdlGBY70?usp=drive_link`

2. Create a new folder named:
   `YourName_ResearchAnalyst_Task`

3. All your outputs will go into this folder.

Step 2 – Load & Inspect the Dataset

1. Load the CSV into Python (pandas) in a Colab notebook or your environment.
2. Inspect:
   - Column names
   - Sample rows
   - Data types
   - Missing values
   - Examples of obvious non-infrastructure records

You may work in **Colab** or locally, but the final code must be shareable.

Step 3 – Clean the Dataset

Create a clean working table with at least the following fields:

- `project_id` (you can create this: e.g. hash or row ID)
- `project_name` (from `title`, cleaned)
- `description`

- `country_name`, `state_name`, `city_name`
- `currency`
- `budget_raw` (copy of `budget`)
- `budget_usd_clean` (your cleaned value)
- `sector_main` (your consolidated sector)
- `timestamp` (primary date used)
- `timestamp_label`
- `source`
- `url`

Budget normalization:

- If `ml.budget_ccy_bot.results.budget_usd` is non-null → use it as `budget_usd_clean`.
- Else if `currency == 'USD'` and `budget` looks numeric → use `budget`.
- Else → set `budget_usd_clean` to `NaN` (missing).

Keep notes / comments on any additional rules you apply.

Timestamps:

- Parse the `timestamps` JSON-like field to usable Python dates (where possible).
- Decide a simple rule for “primary date”:
    - e.g. if `project_start_date` exists, use that; else `estimated_start_date`; else `publish_date`; etc.
- Store that in `timestamp` and keep the type in `timestamp_label`.

Duplicates & obvious noise:

- Remove exact duplicates and obvious “duplicate reports” (e.g. `DUPLICATE REPORT` in title).
- Keep a simple `is_duplicate` or `drop_reason` if you want to track it.

Step 4 – Filter to True Infrastructure Projects

Create a new column:

- `is_infrastructure_project` – boolean (True/False)

Mark as TRUE when the project is clearly about a physical asset such as:

- Transport: roads, rail, metro, ports, airports, parking, bridges
- Water/environment: water supply, wastewater, drainage, flood, desalination, dams, pipelines
- Energy: power plants, transmission lines, renewable projects
- Social infrastructure: hospitals, schools, universities, prisons, government buildings, community centers
- Data/industrial: data centers, industrial plants, logistics hubs (if clearly a facility)

Mark as FALSE when it is **not** a specific infrastructure asset, e.g.:

- Scholarship & education support programs
- Training programs, employment activation schemes
- General grants or baskets (FASE, education reforms, etc.)
- Corporate revenue or financial results
- Purely research or academic projects without a build/asset

You may use **keywords (description, sector, subsector, level1_sector, source)** plus simple rules.

Output:
Create a filtered dataset **`infra_projects`** = all rows with `is_infrastructure_project == True`.

Step 5 – Create a "Megaproject" Flag

Define a simple rule for **megaprojects**, for example:

- `is_megaproject = True` if `budget_usd_clean >= 500,000,000`
- Otherwise `False`

You may adjust the threshold if you explain it in the report.

Add this as a new column: `is_megaproject`.

Step 6 – Enrich the Dataset (Web + Optional GenAI)

For a **subset of projects** (e.g. all megaprojects + a sample of others), use **online research** to fill missing fields.

Allowed tools:

- Web browsing (official project pages, government portals, PDF reports)
- Optional: GenAI **only** for:
    - translating text
    - summarizing long project descriptions
    - inferring high-level sector labels or lifecycle phase

Not allowed: using AI to auto-write your entire report. That is easy to detect and will disqualify you.

Try to fill:

- `project_type` (transport, energy, water, social, industrial, etc.)
- `sector_main` (if missing or messy)
- `estimated_construction_start_date`
- `estimated_completion_date`
- `actual_construction_start_date` / `actual_completion_date` (if available)
- `contract_award_date` (if available)
- `delivery_method` / `procurement_method` (use `ml.procurement_method…` + web info)
- `final_actual_cost` (if different from early estimates)
- Any relevant risk/failure notes (e.g. cancellation, delay, litigation)

Optional extra fields (only if you can find them quickly):

- `identified_risks` (short list)
- `failure_event_date` (if the project fails / is cancelled)
- `data_completeness_score` — e.g. High / Medium / Low

Step 7 – Failure & Delivery Method Analysis (for the PDF Report)

Using your **infra-only** dataset:

1. Define a simple binary `failed_or_problematic` flag, e.g.
    - True if status is Failure / Abandoned / Canceled / Delayed
    - False otherwise

2. Use `ml.procurement_method_redo2.results.method` (and/or your enriched `delivery_method`) to group projects by delivery method, for example:
    - Design-Bid-Build
    - Design & Build / Design-Build
    - PPP / Concession
    - Other / Unknown
3. Compute for each delivery method:
    - Number of projects
    - Number and percentage that are `failed_or_problematic`
    - Any visible patterns by sector or region
4. Create at least **one visualization**, for example:
    - Bar chart: failure rate (%) by delivery method
    - Table: delivery method × sector × failure count
5. In your PDF report, **explain what you see**:
    - Any methods that appear riskier in this sample?
    - Any caveats (small sample size, biased sources, missing data)?

Include a final section on **uncertainties**:

- What you could not know from the data
- How sample and source bias might affect conclusions
- Where you would improve the dataset (more fields, more sources, better labels)

4.4 Deliverables & How to Submit

Inside your Drive folder `YourName_ResearchAnalyst_Task`, include:

1. **Code**
    - File: `notebook.ipynb` (Colab) or `analysis.py`
    - Include all data processing steps: loading, cleaning, filtering, enrichment, analysis
2. **Cleaned Infrastructure Dataset (CSV)**
    - File: `infra_projects_clean.csv`
    - Must include at least:
        - `project_id`, `project_name`, `description`
        - `country_name`, `state_name`, `city_name`
        - `budget_raw`, `budget_usd_clean`
        - `sector_main`, `project_type`
        - `is_infrastructure_project`, `is_megaproject`
        - `delivery_method`
        - `timestamp`, `timestamp_label`
        - `source`, `url`
        - Any enriched fields you computed
3. **(Optional but strong plus) – Future Major / Mega Projects Table**
    - File: `future_major_projects.csv`
    - Projects with future start dates or major planned investments, with inferred budgets where possible
4. **PDF Report (2–4 pages)**

- File: `report.pdf`
- Must contain:
  - Brief description of your approach
  - Key cleaning + filtering steps
  - Summary stats: e.g. number of total records, infra records, megaprojects
  - Failure vs delivery method analysis + at least one chart/table
  - Limitations & uncertainties
  - Short note on how you used web / GenAI (if you did)

4.5 Making Evaluation Easy (What We'll Look At First)

This is how your work will be evaluated quickly:

1. **Does `infra_projects_clean.csv` exist and open without errors?**
2. **Do we see meaningful values in `is_infrastructure_project` and `is_megaproject`?**
3. **Does the PDF report clearly explain:**
   - cleaning logic,
   - infra filtering logic,
   - megaproject definition,
   - failure vs delivery method results,
   - uncertainties?
4. **Is the code readable and reproducible?**
5. **Did you use research/AI in a thoughtful way (and not to auto-write the report)?**

If you can do this clearly and coherently, that’s a very strong signal for this role.