**Overview**

In this challenge, you’ll demonstrate your ability to build an end-to-end data analytics solution using Python. You will: scrape real-world movie data from IMDb.com; ingest and clean that data in Pandas DataFrames; and then generate, report, and visualize key performance indicators (KPIs) that a streaming service or studio might use to guide content strategy and marketing. ​

❗ **Important Note:**  
Complete this task independently. You may consult official documentation or tutorials on Python, Pandas, and BeautifulSoup, but all scraping logic, analysis, and reporting must be your own work.

**Business Scenario**

FilmFlix is a growing streaming platform seeking to understand broader audience preferences and market trends. Rather than relying on proprietary data alone, the analytics team supplements their insights with publicly available information from IMDb. Leadership has tasked you to deliver automated, actionable reports on viewer-relevant metrics—such as the most popular genres, director and actor performance, and box office success—so they can make data-driven decisions about licensing and original content production.

**Provided Data**

Students will scrape the IMDb “Top 250 Movies” page (or a comparable public listing) to collect, at minimum, the following fields for each title:

* **Title**
* **Release Year**
* **IMDb Rating**
* **Number of Votes**
* **Genre(s)**
* **Director(s)**
* **Top 3 Cast Members**
* **Box Office Gross** (where available)
* **Runtime**

**(**Scrap more data ^-^**)**

**Your Tasks**

1. **Data Extraction**
   * Scrape the target IMDb page(s) using Python (e.g., requests + BeautifulSoup), capturing all required fields.
   * Persist the raw output to CSV or JSON for reproducibility.
2. **Data Ingestion & Cleaning**
   * Load raw data into Pandas DataFrames.
   * Normalize data types (convert ratings to float, years to integer, gross to numeric).
   * Handle missing or inconsistent entries (e.g., absent box office data).
3. **Data Modeling & Transformation**
   * Transform your DataFrame(s) to represent one-to-many relationships:
     + Movies ⇄ Genres
     + Movies ⇄ Directors
     + Movies ⇄ Actors
   * Use Pandas techniques (such as explode, merge, and groupby) to create clean, relational DataFrames.
4. **KPIs**

You have to create valuable KPIs by yourselves.

1. **Reporting & Visualization**
   * Compile your findings in a Jupyter Notebook: display each KPI as a well-formatted DataFrame (e.g., via df.style or simple tabular output).
   * Create at visualizations (bar charts, line plots, pie charts) to illustrate your key insights.
   * Accompany each chart and table with a brief narrative interpretation of what the result means for FilmFlix’s strategy.
2. **Automation**
   * Implement a simple scheduling mechanism (e.g., Python’s schedule library or a cron job) to refresh your data and recompute KPIs on a weekly basis.

**Evaluation Criteria**

* **Critical Thinking:** Relevance and potential business impact of chosen KPIs.
* **Technical Execution:** Robustness of scraping logic, correctness of data cleaning, and efficiency of DataFrame operations.
* **Clarity:** Organization of code and documentation; readability of tables and visuals; strength of your narrative.
* **Creativity:** Introduction of additional insights or novel metrics beyond the minimum requirements.
* **Reproducibility:** Ease with which a reviewer can follow your README to reproduce all results.

This project mirrors real-world analytic workflows—combining Python programming with business acumen—to produce decision-ready insights. Good luck all !