Data Engineering project 2: Trendwatcher Development

Objective

Build a robust ELT pipeline that integrates data from two sources — ArXiv and HackerNews — into a functional data product. This project will involve extracting data, loading it into a data lake, and transforming it to make the end product.

Data Sources

1. **ArXiv**: Access daily updates of scientific papers using the ArXiv computer science RSS feed (https://rss.arxiv.org/rss/cs).

2. **HackerNews**: Scrape the first five pages of news articles and associated comments from HackerNews (https://news.ycombinator.com/news).

You should scrape each data source every day **three subsequent days.** This means that the submission deadline is technically three days before the final day.

For HackerNews you will need to build a very basic webscraper to do your data ingestion.

For ArXiv you will need to parse RSS feeds, which are XML files. https://feedparser.readthedocs.io/en/latest/index.html makes this very easy.

The code for both sections should be quite straightforward.

Project Components

- **Data Ingestion**: Make scripts to scrape and ingest data from both sources. For HackerNews, focus on downloading titles, links, and comments without fetching external content.
- **Data Lake Storage**: Build a data lake with tools such as Minio, Delta Lake, or another of your choice to store raw data in your bronze layer.
- **Data Processing**: Implement data deduplication and validation to clean and prepare data for analysis in your silver layer.
- **Data Analysis and Visualization**: Transform cleaned data into a 'gold' dataset and use it to perform a simple analysis, e.g., tracking mentions of specific technologies over time.

- **Dashboard**: Create a basic visualization of your analysis results using Python with Streamlit or a similar tool.
- **Orchestration:** Your pipeline needs to be orchestrated by a tool such as Airflow or Dagster and run at predetermined timelines automatically.

The data analysis section of this assignment is not the essence, this can be very rudimentary, for instance it's fine to simply count how many times 5 different topics like "large language models", "machine learning" and so on appear. Afterwards a single plot is enough for the data analysis and visualization part of the assignment.

You can do more than this if the project is interested (for instance, using natural language processing) but this is not necessary. You're free to reach out if you want more guidance.

Technology Stack

Data Processing: Choose between Polars, DuckDB, or Apache Spark.

Orchestration: Schedule and automate your pipelines using Airflow, Airbyte, or Dagster.

Containerization: Package your application using Docker to ensure reproducibility.

Final Deliverable

A PDF file including:

- Title page with names and student numbers.
- (One or more) slide outlining the chosen use case.
- (One or more) slide showing the architecture of the solution including all components. I expect a **diagram** of the architecture made with something like excalidraw or draw.io.
- (One or more) slide detailing the chosen data processing tool (polars, spark or DuckDB) and your rationale.
- (One or more) slide on data lake management
- (One or more) slide on data warehouse strategy.
- Slide with a link to the GitHub repository containing the project code and setup instructions to run your container.

The deadline for the assignment will be decided by the class through a poll on Toledo.

Evaluation

The largest part of the grade will be on the conceptual choices you made during the project. Specifically, the architecture you chose and why you chose to do it. On top of that, part of the grade will also be dedicated to your final project being runnable. Concretely, this means I will run your container and look at the final project.

- 30 % of your grade, architectural choices.
- 15 % of your grade, the code and your report.
- 15 % of your grade, the end product working or not.
- 30 % of your grade, creativity and initiative.

Creativity and initiative will be measured by looking at how you assimilated novel ideas into your solution. For instance, projects that use exactly the same tech stack and ideas as the slides will not receive a high score here (but will likely still receive a high score on the overall assignment). The goal of this criterion is to encourage trying out new things, looking on the internet for alternative architectures and so on.

This assignment counts for 2/3rd of your grade of this course. At the end you will also need to conduct a **mandatory** peer review that will contain 2 criteria, effort and contribution. The peer review will be a holistic overview of the effort and contributions of both assignments, meaning that if a group member did not participate in the first assignment, the onus is on them to rectify this in this assignment.

Recommendations

- Begin by familiarizing yourself with Docker if you have not used it before (https://docs.docker.com/get-started/).
- Consult course materials to ensure best practices in data architecture are applied.
- Take it step by step, the project might seem complicated, but each part is quite simple.