Rayce Ramsay (1009734888)
Ali Shabani (1008838652)
Terry Tian (1007663468)
Grant Hamblin (1009860447)

CSC490 Assignment A2 - Data processing Pipelines - 10% of Final Grade

Due: Oct 1st, 11pm with a Github PR and on Quercus

The goal of this assignment is to start thinking about the datasets needed for your project. This includes the datasets, processing steps and integration into your application.

Part One: Aspirational Datasets (20 marks)

Dataset 1: Patient Medical History

Schema:

- patient_id (Int) primary key
- Age (int)
- Sex (Str)
- Ethnicity (Str)
- Weight_kg (Int)
- Height_cm (Int)
- Comorbidities: [Str]
- Genetic Factors (Str)

Dataset 2: Medication Dataset

Schema:

- Mediaction_id (int)
- Drug_name (str)
- Dose (int)
- Ingredient_ids [(int)]

Dataset 3: Medication Patient Exposure

Schema:

- Exposure_id (int) - primary key
- Patient_id (int) - foreign key
- Medication_id (int) - foreign key

Dataset 4: Drug to Drug Interactions

Schema:

- interaction_id (int) – (primary key)
- drug_id_1 (int) – first drug
- drug_id_2 (int) – second drug
- effect_code (str) – standardized code for the adverse effect (e.g., MedDRA)
- effect_text (str) – human-readable description of the effect
- Mechanism (str) – underlying mechanism (e.g., CYP3A4 inhibition, QT prolongation)

- severity_level (str) – severity classification
- score (int) – composite risk score (0–100, already context-aware)
- Dose_min_mg_per_day (int) – minimum daily dose where the interaction is relevant
- Dose_max_mg_per_day (int) – maximum daily dose where the interaction is relevant
- age_min_years (int)– minimum patient age where interaction applies
- Age_max_years (int) – maximum patient age where interaction applies
- Sex (str)
- Pregnancy_trimester (str) – pregnancy context (none, T1, T2, T3)
- time_to_onset_hours (int) – typical time to onset
- management_recommendation (str) – guidance on what to do if co-administered

This table functions as a master schema and can be updated when new information is acquired.

Part Two: Reality Check (30 marks)

Now review which datasets are actually available. Create a table of datasets that you could use for your project. Include public datasets, data you can scrape or data you can generate. For each dataset, include a link to the source, a description of the data and commentary on why it is relevant to your project.

Relevant Item	**Description**	**Commentary**
FDA Structured Product Labels (SPL) - Human Prescription Labels https://dailymed.nlm.nih.gov/dailymed/spl-resources-all-drug-labels.cfm	Full set of XML files for all the FDA-approved prescription drugs. It includes indications, contraindications, drug interactions, dosage, and patient safety information. Updated monthly.	We can use this dataset to extract DDI warnings and dosage recommendations for CKD patients. The data set contains drug label data and can be used for a structured knowledge base. If we need specific classes of data, we can filter those out.
Drug Bank https://go.drugbank.com/releases/latest#full	Drug database that combines chemical and pharmaceutical data with DDI information. Includes curated interactions.	This can supplement the FDA's data and is useful for narrowing down the scope to high-risk drugs used in the elderly with CKD populations. The full dataset requires an application (already done) to gain access, but there is some free data also available.
Synthea synthetic patient data https://github.com/synthetichealth/synthea	Open source synthetic dataset of EHRs, which includes demographics. Conditions, medications, lab values, etc.	Great for modelling CKD-specific patient contexts, especially for medication and demographics restrictions. Useful for testing the model and evaluating how it integrates patient context when prioritizing high-risk interactions.
Therapeutics Data Commons - DDI https://tdcommons.ai/multi_pred_tasks/ddi/	Free dataset with ~190000 DDI pairs across ~1700 drugs from DrugBank. Can be used by installing the PyTDC library	Good structured knowledge base. Lacks patient context, but is ML ready (dataframes) data extracted from Drug Bank
Mendeley Data - DDI https://data.mendeley.com/datasets/md5czfsfnd/1	Includes multiple data files for interactions, interaction types for ~110 categories	This is also extracted from DrugBank, useful because it categorizes types of interactions. So, it might be helpful for mapping risks to severity levels. Data is a bit old compared to other sources (2020)
CRESCENDDI https://github.com/elpidakon/CRESCENDDI/tree/main/data_records	A reference set that links drug pairs to clinically confirmed outcomes. Includes positive controls (harmful interactions) and negative controls (no known interaction). And single drug adverse outcomes.	Helpful for clinical relevance instead of theoretical interactions. Good for filtering out noise and focusing on meaningful risks. Lacks patient demographics but can be used alongside patient data to infer interactions.

DDInter 2.0 https://ddinter.scbdd.com/download/	Knowledge base of drug drug interactions. Includes ~300,000 DDI pairs in different categories of drugs. Includes severity levels. Also includes drug-disease interactions and drug-food interactions.	Relevant as a reference for interactions and severity. Lacks patient info but identifies high-risk interactions. Helps prioritize DDIs found in patient data.
FDA FAERS – Adverse Event Reports (U.S. FDA's spontaneous reporting data) https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html	Includes real world adverse drug event reports to FDA. includes patient demographics, drugs used, reported reactions, outcomes, and reported case details. It's available in quarterly updates.	This dataset contains a lot of useful information about patient context and outcomes. However, we need to filter this data by looking at cases where multiple drugs were used and the outcome was severe. FDA mentions there is no confirmation about drug interactions and they are merely reports. But we can combine this with DDI knowledge bases to infer adverse interactions.
Canada Vigilance Adverse Reaction Dataset https://www.canada.ca/en/health-canada/services/drugs-health-products/medeffect-canada/adverse-reaction-database/canada-vigilance-online-database-data-extract.html	Adverse reaction reports from Canada in a format similar to FAERS. Contains all reports from 1965 - 2025 in text files. Includes demographics, reactions, and seriousness.	Complements FAERS. Contains patient data so it's easy to filter for our target population.
AEOLUS(Cleaned FAERS) https://datadryad.org/dataset/doi:10.5061/dryad.8q0s4	Cleaned version of FAERS that removes duplicates and standardizes the data.	Similar to FAERS but adds structure. Real world adverse event data normalized for drug names and outcomes.
RxNorm https://www.nlm.nih.gov/research/umls/rxnorm/index.html	Standard system for clinical drugs, maintained by the U.S. National Library of Medicine. It includes normalized drug names, identifiers, and relationships between brand names	This dataset is good for normalization and instructing the LLM to generate consistent responses for identifying drugs which prevents misclassification for both training and evaluation.

Part Three: Data-processing pipelines (50 marks)

Design and implement a data processing pipeline for your project. This should include data ingestion, cleaning, transformation and data lake/warehouse design. Make sure to include:

- Data schemas
- Pipeline diagrams with the technologies you are using (open source frameworks are helpful)
- When the pipelines will run and for which use cases
- Submit code for an initial version of this pipeline
- Include next steps for features you did not implement

Pipeline Overview

Our data processing pipeline integrates multiple sources, including synthetic EHR data from Synthea, drug identifiers from RxNorm, and drug–drug interaction knowledge bases such as DDInter, into a structured, queryable system. Raw files are ingested, cleaned (e.g., standardized dates, deduplication, dose parsing), and transformed (e.g., mapping patient UUIDs to IDs, normalizing medications by RxCUI, estimating daily doses). The results are stored in a data lake as four core datasets: Patient Medical History, Medication Dataset, Patient Exposures, and a DDI Master table. Each dataset is then loaded into a database layer (e.g., DynamoDB) to support efficient retrieval by the application. Finally, a retrieval-augmented generation (RAG) component queries the database to assemble patient and drug-specific context, enabling an LLM to generate concise, prioritized, and actionable DDI alerts.

Data Schemas

Dataset 1 - Patient Medical History

Column	Type	Notes
patient_id	int (PK)	Surrogate key (UUID → int)
Age	int	Years (calculated from birthdate)
Sex	str	As recorded in Synthea
Ethnicity	str	From Synthea ETHNICITY
Weight_kg	float	Latest observation, rounded to 0.1
Height_cm	int	Latest observation, rounded to cm
Comorbidities	[str]	List of diagnoses per patient
Genetic_Factors	str/null	Placeholder (not available in Synthea)

Dataset 2 - Medication Dataset

Column	Type	Notes
medication_id	int (PK)	Surrogate key
Drug_name	str	Preferred name from RxNorm
Dose	int/null	Primary strength (mg)
Ingredient_ids	[int]	Ingredient CUIs from RxNorm (IN)
RxCUI	int	RxNorm identifier

Dose_components_mg	[int]	Parsed dose components (mg)
Dose_detail	str	Original RxNorm SCD string

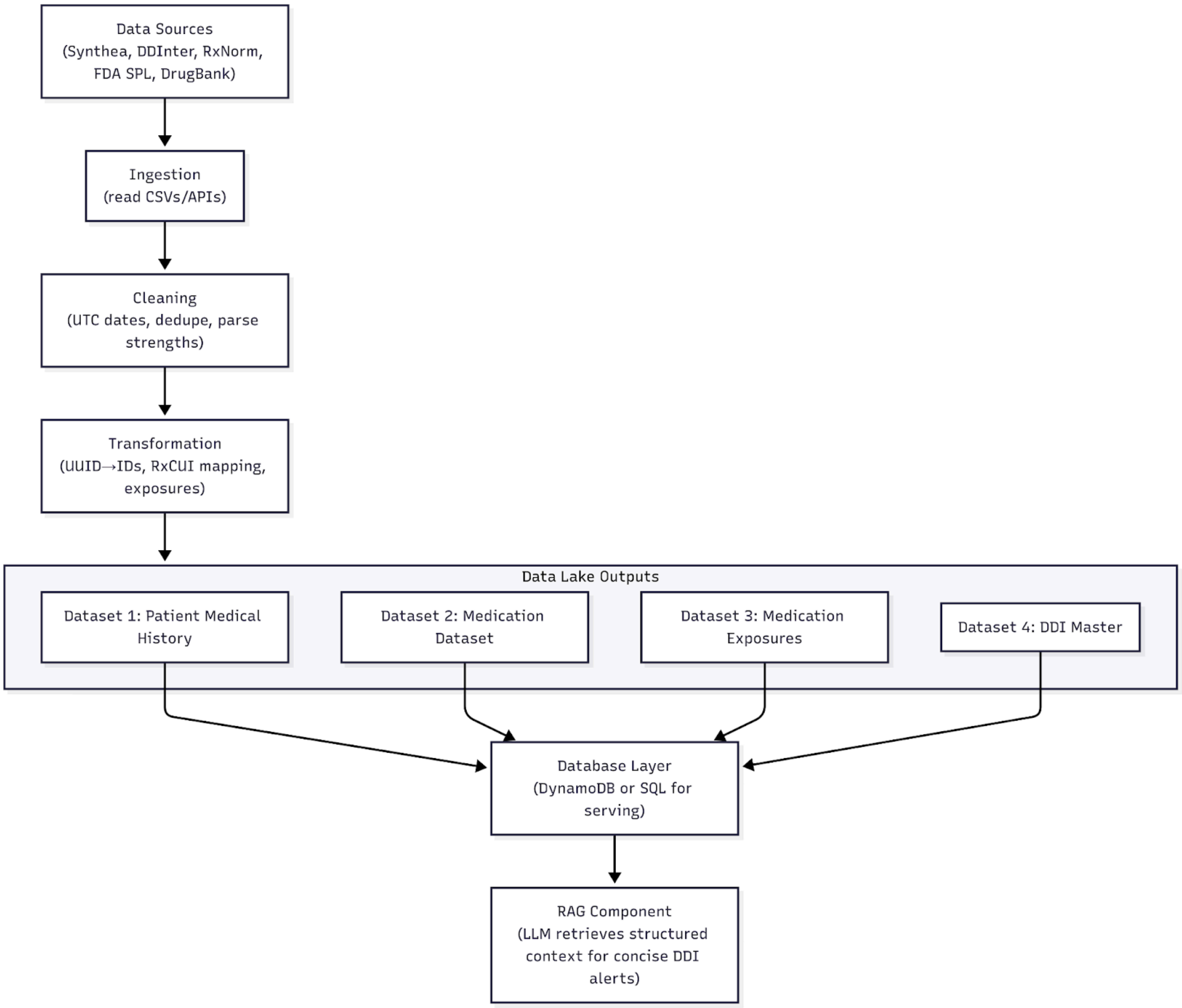
Dataset 3 - Medication Patient Exposure

Column	Type	Notes
exposure_id	int (PK)	Surrogate key
patient_id	int (FK)	Links to Patient Medical History
medication_id	int (FK)	Links to Medication Dataset
start	datetime	UTC start date
stop	datetime	UTC stop date
daily_dose_mg	float	Estimated daily dose (strength × frequency)

Dataset 4 - Drug-Drug Interactions

Column	Type	Notes
interaction_id	int (PK)	Surrogate key
drug_id_1	int (FK)	Medication dataset ID
drug_id_2	int (FK)	Medication dataset ID
rxcui_1	int	RxNorm ID
rxcui_2	int	RxNorm ID
severity_level	str	Normalized (Contraindicated/Major/Moderate/Minor/Unknown)
sources	[str]	Provenance (e.g. DDInter, RxNorm)

Diagram


Data Sources
(Synthea, DDInter, RxNorm, FDA SPL, DrugBank)
Ingestion
(read CSVs/APIs)
Cleaning
(UTC dates, dedupe, parse strengths)
Transformation
(UUID→IDs, RxCUI mapping, exposures)
Data Lake Outputs
Dataset 1: Patient Medical History
Dataset 2: Medication Dataset
Dataset 3: Medication Exposures
Dataset 4: DDI Master
Database Layer
(DynamoDB or SQL for serving)
RAG Component
(LLM retrieves structured context for concise DDI alerts)

Execution Strategy

Current prototype: The pipeline is designed to be run once end-to-end to ingest, clean, and transform data into the four datasets. Because we are working with static synthetic data (Synthea CSVs, downloaded DDInter files), the outputs are stable and do not need continuous updates.

Future extension: If we decide to use data that is updated regularly, we will consider automation for long-term consistency. This includes:

- Monthly batch update: If using FDA SPL or DrugBank data, re-run ingestion to incorporate new drug labels and interaction records.
- Quarterly batch update: Refresh adverse event reports (FAERS/Canada Vigilance), which are published on a fixed schedule.
- On-demand run: If a new patient EHR is added, rerun only the ingestion/transformation steps for that patient's medical history and exposures, while reusing the static medication/DDI reference tables.
- Long-term automation: Pipeline could be scheduled with an orchestration tool (e.g., Prefect) to ensure updates without manual execution.

Current Code Implementation

See Github or Colab link:
https://colab.research.google.com/drive/1NYHmsGP2NPMuY4FkcGNAKr3M2DyWzCuz?usp=sharing

Next Steps

Data Pipeline

1. Move collab notebook code into AWS Lambda function which is scheduled for quarterly runs.
2. Set up Cloud Database (Roles & Permissions) to contain outputs of data cleaning scripts
3. Further refine data models to contain more relevant information
    a. Real World Patient Data - Annotated by Drug Interactions
    b. Larger DDI Database

RAG Pipeline

1. Determine which LLM model to use, and where to run model inference (AWS vs Modal)
2. Create vector embeddings from data sources

Submission Instructions

Submit a pull request to the course Github repo with your assignment in a folder named a2 with your a2.pdf, include your team members names on the first page with student IDs.