William Chang Liu - 1009048852

Ryan Zhang - 1009020453

Taiyi Jin - 1009075796

Abdus Shaikh - 1007288533

CSC490 Assignment 2: Data Processing Pipelines

Part 1: Aspirational Datasets **1**
Part 2: Reality Check **3**
Part 3: Data-processing Pipelines **5**
Approach 5
Data schemas 5
Pipeline diagram with the technologies you are using 7
When the pipelines will run and for which use case 7
Next steps for unimplemented features 8

Part 1: Aspirational Datasets

An ideal dataset for our task would be a global-scale, structured electronic health record (EHR) repository designed specifically to be used for training downstream ML pipelines for various medical applications such as disease risk assessment and early prevention. Furthermore, this dataset would also adhere to differential privacy constraints in order to serve as a viable training set for synthetic data generators without leaking confidential information.

The ideal dataset would include the following:

1: Essentially Patient Demographics

- Fields: Age, sex, gender identity, ethnicity, socioeconomic status, geographic region
- Why: Demographics heavily influence most medical applications, and global representation ensures a robust sample set for generalization.

2: Diagnoses and Conditions

- Fields: Diagnosis codes/names, chronic vs acute condition, condition severity
- Why: Diagnosis information is core to our goal of creating a synthetic dataset with impactful downstream applications. Including the condition severity and acute vs chronic status will help inform decisions made downstream

3: Medications and Prescriptions

- Fields: Drug name / code, dosage, frequency, route of administration
- Why: Such medication information is crucial for modeling treatment efficacy and side effects

4: Laboratory and Test Results

- Fields: Test type, test result
- Why: Lab results provide quantitative biomarkers that can be strong early indicators of disease risk

5: Procedures and Interventions

- Fields: Procedure type, procedure reason, procedure outcomes
- Why: In conjunction with other tables, these values are useful for modelling patient health trajectories based on their medical history

7: Social and Lifestyle Factors

- Fields: Smoking status, alcohol use, physical activity levels, diet quality
- Why: These factors are strong non-biomarker indicators of a patient's health trajectory. Integrating such medical-adjacent data will provide an additional dimension to potential downstream applications

Part 2: Reality Check

Relevant Item	Description	Commentary
MIMIC-IV	Collected over a period of more than 10 years, MIMIC-IV contains electronic health record data for over 65,000 patients admitted to an ICU and over 200,000 patients admitted to the emergency department. The dataset provides broad and deep insights by including over 22 tables of different medical information per patient.	MIMIC-IV is highly aligned with our ideal dataset, It includes many aspects of our aspirational dataset such as lab results, prescriptions, diagnoses, and more. Furthermore, it is rigorously deidentified to effectively hide the identity of the individuals whose data it represents, which aligns with our goal of creating differentially private synthetic data.
eICU Collaborative Research Database	A database of over health data of over 200,000 deidentified ICU admissions across the United States. This dataset includes vital sign measurements, care plan documentation, severity of illness, diagnoses, and treatment information.	eICU provides a large sample of structured health records collected. Although all records originate from the US, they span several different hospitals and can provide us with valuable demographically diverse information. Additionally, it is also rigorously deidentified to protect the identities of the patients it was collected from. All of these reasons make it a strong candidate for our end goal.
UK Biobank	A database containing health records of over 500,000 participants spanning visits to general practitioners, hospitals, and cancer centers. The UK Biobank provides a breadth of information including cancer diagnoses, prescriptions, and medical procedures.	The UK Biobank provides structured health records from general practice, hospitals, and cancer centers, making it well-suited to model common conditions and preventative care pathways. This aligns directly with our goal of generating synthetic patient-level data for downstream medical applications. Its scale and diversity help approximate our aspirational dataset's global representation. Furthermore it

		provides a different demographic from the previous two datasets which can help generalize our synthetic data generator and push it closer to global representation

Part 3: Data-processing Pipelines

Approach

The model we will be using for our baseline, EHRDiff (Yuan, Hongyi, Songchi Zhou, and Sheng Yu. "EHRDiff: Exploring realistic EHR synthesis with diffusion models." (2023)), focuses on the MIMIC-III dataset. Our goal is to improve upon this baseline so we will only consider the MIMIC-IV dataset for this project, which is an extension and improvement of MIMIC-III. To keep comparisons consistent, we will benchmark the performance of EHRDiff on MIMIC-IV as well.

MIMIC-IV is a relational database consisting of 26 tables. It contains separate modules for hospital (`hosp`) and ICU (`icu`) data, with hospital containing 22 tables. For this project we will be exclusively looking to train from the hospital module, and synthetically reproduce data of this structure. The tables consist of a variety of hospital related events such as `diagnoses`, `procedures`, `prescriptions`, `microbiologyevents` and more, which all ultimately tie to a `hadm_id` (hospital admission identification) or a `subject_id` (patient identification). We will transform the data by summarizing each patient's Electronic Health Record (EHR) into a single vector.

Data schemas

For each patient's EHR, the vector representing the EHR will include "metadata" which are single vectors representing tables with one unique row per EHR, such as patient information. For tables where there may be multiple rows per EHR, such as diagnoses_icd, we will create a frequency encoding vector representing an aggregation of this table, by admissions (hadm_id). Each element of the vector will represent a unique event (diagnosis/procedure.. etc) and an integer number of occurrences in the EHR. Therefore, the EHR for each patient is formulated as a vector consisting of vectors for each relevant table in MIMIC-IV, some which represent one row of the table, and others which are frequency encoding of events recorded in the EHR.

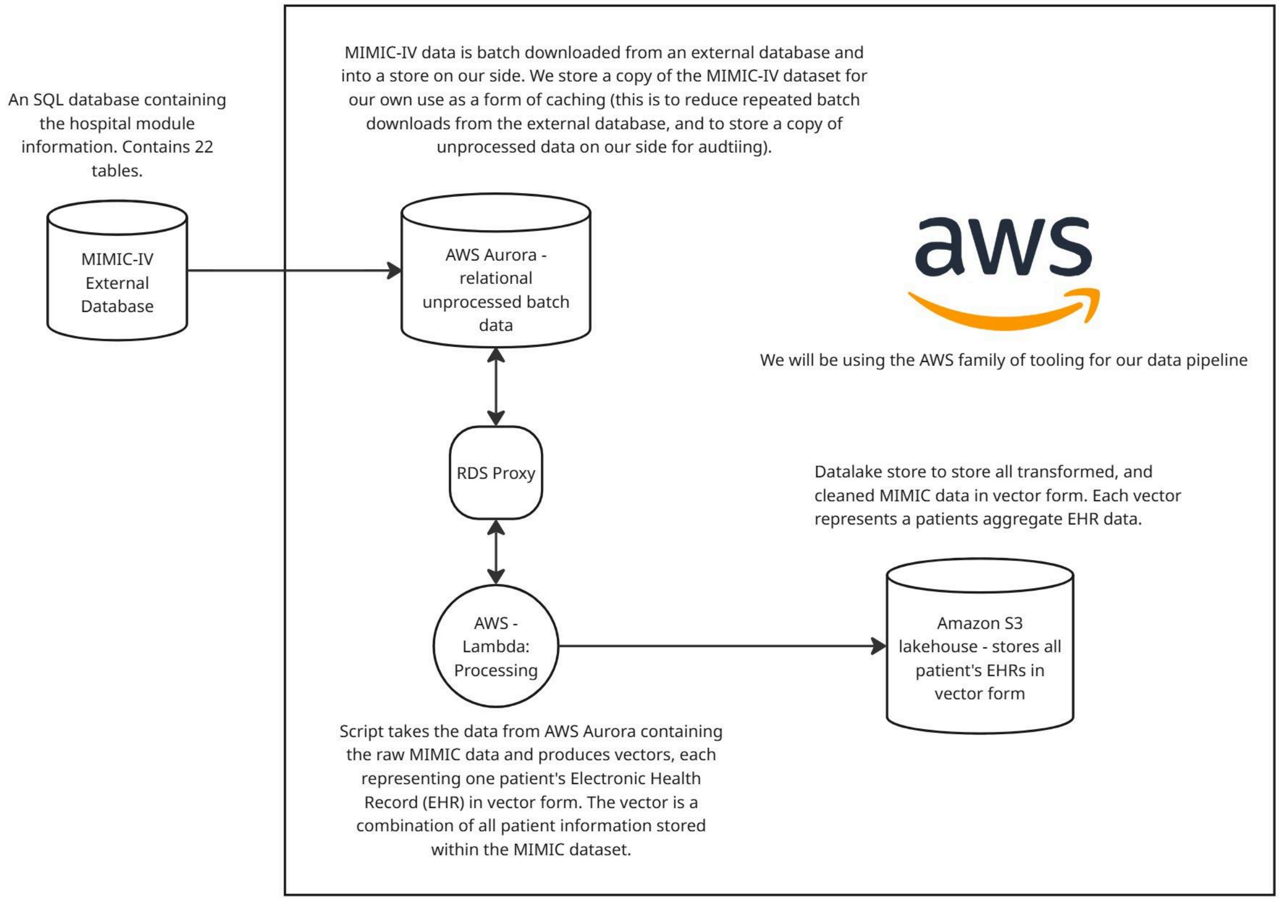
For a deeper dive into the reasoning behind omitting certain tables and attributes, consult our documentation for EHR Vector Feature Design: https://docs.google.com/document/d/1tlpzt2ApQsQ5--7vmANqSUnIjIYshp6zCpjVW-Pf1R0/edit?usp=sharing

The following is an example of a complete EHR record as a vector with attributes/table names commented to the side:

```
[
        1, # Subject ID
        1, # Gender
        20, # Age
        0, # Date of death
        20, 12, 0, 3, …, # prescriptions table. Pre-flattened shape: (829,)
        0, 3, 2, 0, 1, 2, …, # diagnoses table. Pre-flattened shape: (1472,)
        0, 0, 1, 0, 2, 0, …, # procedures table. Pre-flattened shape: (352,)
        12, 53, 74, …, # POE table. Pre-flattened shape: (15,)
        0, 0, 2, 0, 1, …, # Services table. Pre-flattened shape: (13,)
        2, 0, 0, 1, …, # Admissions table. Pre-flattened shape: (31, )
        0, 25, 1, 0, 142, …, # OMR table. Pre-flattened shape: (14,)
        0, 0, 0, 0, …, # Hospital Events table. Pre-flattened shape: (19,)
        6, 0, 2, 0, 0, 5, …, # Pharmacy table. Pre-flattened shape: (584,)
]
```

After flattening into one single vector, the final shape is (3333,). The final number of total processed EHR vectors is 364,627. We will randomly select 300,000 EHR vectors for model training and the rest will be held for evaluation against the synthetically generated EHR vectors. This roughly fits an 80/20 training/evaluation split.

Pipeline diagram with the technologies you are using


An SQL database containing the hospital module information. Contains 22 tables.
MIMIC-IV External Database
MIMIC-IV data is batch downloaded from an external database and into a store on our side. We store a copy of the MIMIC-IV dataset for our own use as a form of caching (this is to reduce repeated batch downloads from the external database, and to store a copy of unprocessed data on our side for audtiing).
AWS Aurora - relational unprocessed batch data
aws
We will be using the AWS family of tooling for our data pipeline
RDS Proxy
Datalake store to store all transformed, and cleaned MIMIC data in vector form. Each vector represents a patients aggregate EHR data.
AWS - Lambda: Processing
Amazon S3 lakehouse - stores all patient's EHRs in vector form
Script takes the data from AWS Aurora containing the raw MIMIC data and produces vectors, each representing one patient's Electronic Health Record (EHR) in vector form. The vector is a combination of all patient information stored within the MIMIC dataset.

NOTE: bearing in mind the importance of redundancy and avoiding single points of failure, Amazon S3 has built in redundancy across availability zones in case any single data centre goes down.

For the purpose of this assignment we will be using SQLite as a temporary substitute for both our Aurora and S3 databases while we wait for activation of our AWS credit. Due to our current storage limitations we are currently only storing and working with the demo release of the MIMIC-IV database for testing purposes.

When the pipelines will run and for which use case

Since this is a research project with the goal of building upon EHRDiff (Yuan, Hongyi, Songchi Zhou, and Sheng Yu. "EHRDiff: Exploring realistic EHR synthesis with diffusion models." (2023)), we will only need to run this data pipeline one time to download the MIMIC-IV dataset locally, and then clean/transform the tables into vectors we can work with, storing these in our vector store.

Should this be applied to a business structure, for a model which can synthetically generate data to mimic an existing dataset, this pipeline would run once for the first time the dataset is received, and then on a recurring basis when the source database is modified. How often it reoccurs can be defined based on how frequently the source database is expected to be modified—we want to avoid performing this expensive pipeline repeatedly for databases which update constantly.

Next steps for unimplemented features

Our first priority is to move all data and pipelines into the cloud. This will let the system scale, support team development, and connect smoothly with downstream applications. Once in place, we will focus on handling sparsity in high-dimensional vectors, especially in the diagnoses_icd, prescriptions, and pharmacy tables. Sparsity is a concern because it can weaken representation learning, reduce model stability, and slow down inference.

The next step is to run the pipeline on the full dataset rather than the demo. This is key for checking if our preprocessing, feature engineering, and data flow work reliably at real scale, where data volume and variation are much higher.

Finally, we plan to revisit feature dimensionality reduction. The reference paper trims the long tail of rare diagnoses to focus on more frequent and informative features, but we haven't done this yet. Adding this step could reduce vector size, cut noise from infrequent codes, and improve training efficiency without losing much predictive power. We will test different thresholds for trimming to find the best balance between keeping useful information and making models easier to train.