1. Motivation

The task was selected to address the complexities of managing sensitive medical data, a key challenge in healthcare data engineering. With data security regulations such as HIPAA, it is critical to ensure that patient information is properly anonymized to prevent unauthorized access while still allowing for meaningful analysis. The motivation behind this task is to develop skills in extracting and anonymizing sensitive data, and also to create practical solutions for visualizing and presenting medical data in a secure and accessible way.

2. Introduction about the Task

In this series of tasks, I was tasked with handling patient medical data, ensuring privacy compliance through anonymization, and performing various operations to extract, analyze, and present the data. The primary tasks included:

Task 1: Anonymizing and extracting data from a set of patient medical reports, and generating a JSON file to store the relevant data in a structured format.

Task 2: Extracting metadata from DICOM files (standard format for medical images) and categorizing them into patientrelated, imagerelated, and machinerelated information. Additionally, visualizations were created, including collages for image display.

Task 3: Developing a simple APIbacked dashboard to display the extracted and anonymized data. The dashboard provides an interactive way to view patient information, including age, BMI, clinical findings, and images.

The methods selected focus on efficient data extraction, anonymization, and visualization to ensure that sensitive data is handled appropriately while providing actionable insights through data presentation.

3. Data Extraction, Preprocessing, and Analysis

The process involved multiple steps of data extraction, transformation, and storage:

Step 1: Data Extraction from Patient Reports

Patient data was extracted from a set of medical reports. Key fields like patient ID, gestational age, BMI, and clinical findings were extracted. The patient ID was anonymized by creating unique, anonymous identifiers.

Data extraction techniques included:

Text parsing using regular expressions to extract fields like patient ID, age, BMI, and clinical findings from unstructured data (PDFs).

A custom function was created to replace patient names and IDs with pseudonyms to maintain confidentiality.

Step 2: DICOM File Extraction

DICOM files containing CT scan images were processed to extract metadata.

Metadata was categorized into:

Patientrelated: Age, gender, medical history.

Imagerelated: Modality, dimensions, resolution.

Machinerelated: Equipment details such as manufacturer, model, and acquisition settings.

Step 3: Preprocessing and Anonymization

Once the data was extracted, sensitive information (e.g., patient names, IDs) was anonymized.

Anonymization was done by generating a random identifier and mapping it to the corresponding data (e.g., replacing the patient ID with an anonymous string).

Step 4: Visualization and Reporting

A web interface was built using Flask, where users can select a patient ID from a dropdown and view associated data including BMI, age, clinical findings, and images.

Images were displayed dynamically based on the selected patient, and DICOM collages were generated to provide an overview of the images.

Flowchart of the Data Process:

1. Extract Data from Reports > 2. Anonymize Patient Information > 3. Save to JSON File >

4. Extract Metadata from DICOM > 5. Categorize Data (Patient, Image, Machine) > 6. Visualize Images and Data

PseudoCode for the Process:

Extract patient data and anonymize

```
def extract_patient_data(report):
    patient_id = extract_patient_id(report)
    anonymized_id = generate_anonymized_id(patient_id)
    patient_data = {
        'patient_id': anonymized_id,
        'age': extract_age(report),
        'BMI': extract_BMI(report),
        'findings': extract_findings(report)
```

```
    }
    return patient_data

 Extract metadata from DICOM files
def extract_dicom_metadata(dicom_file):
    metadata = {
        'patient': extract_patient_info(dicom_file),
        'image': extract_image_info(dicom_file),
        'machine': extract_machine_info(dicom_file)
    }
    return metadata
```

4. Results

The results of this task can be seen in two main areas:

1. Anonymization and Data Extraction:

The patient data was successfully anonymized, and critical information like BMI, age, and clinical findings were extracted for further analysis.

A JSON file was generated that contained the anonymized data, making it suitable for sharing and analysis while ensuring privacy compliance.

2. DICOM Metadata Extraction and Visualization:

DICOM metadata was successfully categorized into three main categories: Patient-related, Image-related, and Machine-related.

Collages were generated from the images, providing a neat and cohesive presentation for each patient. The images and metadata were successfully visualized using a Flask-based web interface.

5. Key Findings

Key takeaways from this task include:

Anonymization is Crucial: Handling sensitive patient data requires strict anonymization protocols to prevent unauthorized access and to comply with privacy regulations.

Efficient Data Extraction: The ability to extract data from both structured and unstructured sources (PDFs, DICOM files) is a vital skill for any data engineer.

Metadata Categorization: Organizing metadata into specific categories (patient, image, machine) helps in easy analysis and makes the data more structured.

Visualization: Presenting data in an intuitive and interactive way (via a web interface) enhances decisionmaking and allows for easy exploration of data.

6. Future Work

Given sufficient time, the following improvements could be implemented:

Improved Anonymization: Enhance the anonymization process to include more robust techniques, such as using encryption or more complex hashing mechanisms for patient data.

Automation of Data Extraction: Implement a more automated system for data extraction from different types of documents (e.g., handwritten reports or other image-based formats).

Real-Time Data Updates: Enable real-time updates of the visualizations and the dashboard as new data is processed and added.

Machine Learning Integration: Integrate machine learning models to analyse patterns in clinical findings, helping to predict future medical conditions based on historical data.

Scalability: Scale the solution to handle larger datasets and ensure it can accommodate multiple patients and images in parallel.