Project 2

1 Data Description

This project involves datasets comprising biomedical and healthcare data to analyse disease patterns and cancer diagnosis characteristics. All data are sourced from publicly available medical research datasets to support healthcare analytics and pattern mining applications.

Data links:

Dataset 1: Breast Cancer Wisconsin (Diagnostic) Data Set

https://www.kaggle.com/datasets/erdemtaha/cancer-data

Dataset 2: Disease Symptom Prediction Dataset

https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset

Breast Cancer Dataset Structure

- **Patient Records:** Contains diagnostic information for 569 patients with breast cancer
- **Features:** 30 numerical features computed from digitized images of breast mass tissue
- **Target:** Binary classification (M = Malignant, B = Benign)
- **Key Features:**
    - *id:* Unique patient identifier
    - *diagnosis:* Cancer type (M/B)
    - *radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, concavity_mean, concave points_mean*: Mean values of cancer tissue characteristics
    - Additional features include standard error and worst-case values for each measurement

Disease Symptom Dataset Structure

- Disease Records: Contains information linking diseases to their associated symptoms
- Data Columns:
    - *Disease:* Name of the disease
    - *Symptom_1* to *Symptom_17:* Various symptoms associated with each disease
    - Additional files contain symptom descriptions, precautions, and severity weights
- Coverage: Multiple diseases with their symptom patterns and relationships

Biomedical Context

- **Clinical Relevance:** These datasets represent real healthcare challenges in diagnosis and symptom analysis
- **Pattern Mining Applications:** Suitable for discovering hidden relationships in medical data
- **Educational Value:** Provides hands-on experience with healthcare analytics while maintaining patient privacy

2 Task Description

2.1 Analysis of Symptom Co-occurrence Patterns

● Objective:

The goal of this task is to analyse the co-occurrence patterns of different symptoms within disease profiles. Specifically, the task aims to identify combinations of symptoms that frequently appear together in the same disease.

● Method:

Implement the Apriori algorithm to analyse the Disease Symptom dataset, identifying common combinations of symptoms that frequently co-occur within the same disease profile.

● Tips:

1. **Data Preparation for Apriori Algorithm:** Organize the data so that each disease represents a "basket." The "items" in this basket are the symptoms that appear for that disease.

2. **Analysis of Frequent Itemsets:** Analyse the frequent itemsets to determine which symptom combinations tend to co-occur within the same disease profile.

3. **Data Handling Choices:** You may choose to focus only on the presence of symptoms, treating the analysis as a binary problem (symptom present/absent for each disease).

4. **Data Cleaning**: Normalize symptom synonyms (e.g., “fever” vs “pyrexia”) before Apriori.

Ensure a **minimum number of transactions** before mining. If needed, **augment with a mirrored symptom dataset** of similar schema to increase robustness of frequent itemsets.

Code Reuse: You are encouraged to learn from online resources. However, all code must be written by you. If a specific online resource provides a key insight or a few lines of code for a standard function, you **must** cite it in a code comment immediately above the implementation. Your entire program structure and the core logic of the algorithms must be your own.

2.2 Mining Cancer Feature Patterns

● Objective:

The objective of this task is to analyse the feature sequences and patterns in cancer diagnosis data to uncover common characteristics that distinguish malignant from benign cases through sequential pattern mining.

● Steps

1. Data Preprocessing

Transform the numerical features of the breast cancer dataset into categorical sequences. Consider ranking features by their importance or value ranges to create meaningful sequential representations.

Define the **sequence semantics** explicitly: rank features per patient by **z-score (or mutual information w.r.t. diagnosis)**; group the top-k features as ordered itemsets with **max sequence length L** and **max-gap = 1** (same-order ties may form a single itemset).

2. Data Analysis:

Apply sequential pattern mining algorithms (such as GSP - Generalized Sequential Pattern algorithm) to discover patterns in how cancer features manifest in malignant vs. benign cases.

● Example

Consider transforming continuous features into ranked sequences based on their values:

- Patient A: <{high_radius}, {high_texture}, {low_smoothness}>
- Patient B: <{low_radius}, {high_compactness}, {high_concavity}>

Using sequential pattern mining, we can discover frequent patterns such as:

- <{high_radius}, {high_texture}> → Often associated with malignant cases
- <{low_radius}, {low_smoothness}> → Often associated with benign cases

This indicates common feature progression patterns in cancer diagnosis.

● Tips

- **Feature Transformation:** Convert continuous values to categorical representations (e.g., low, medium, high) based on statistical thresholds or domain knowledge.
- **Pattern Interpretation:** Focus on discovering interpretable patterns that can provide insights into cancer diagnosis.
- **Performance Considerations:** Consider the computational complexity when designing your feature transformation approach.
- The 'sequence' here is the **derived order of discretized features** (not a time series). Sequence pattern mining can be applied to ordered but **non-temporal** data.
- Use **KBinsDiscretizer** (uniform / quantile / k-means) and **report a sensitivity check** across binning strategies.

In the report, you need to document what you have done, and report the results.

Task 3: Open Advanced Tasks

- **Objective**

Define a healthcare-related application (e.g., disease prediction, cancer risk assessment, symptom-based diagnosis) based on the datasets and give a solution. The solution includes but not limited to traditional analytics, machine learning, deep learning, and LLM-related tasks. Implement your solution and provide some experimental results to show your solution works.

In the report, you need to explain the algorithm you design. You also need to document the results of your algorithm.

Weight of grading:

Task 1: 30%

Task 2: 40%

Task 3: 30%

You are expected to improve your problem solving, deep thinking, and self-learning ability through the project, which are very important skills to acquire in universities.

What to deliver:

Code: for the three tasks.

Report: The final report is up to 8 A4 pages (not necessary to write 8 pages. The page limit does not include front page).

If there is unequal contribution, include a table showing the percentage contribution and justification, supported by the email confirming all group members agree with the distribution. In the end of report, please include the individual contribution claims in the following format:

Member Name	Contribution (%)	Justification
Member name 1	p%	list of contributions to the project.

Up to 5 minutes video: In the short video, you can capture screen to do a demo of your code to show it works, and you can also highlight any other part of your report in the video.

Project are done in groups. Discussions with other students are allowed, but each group has to write your own code.

Submission Instructions

Deadline: 16 Nov.

Deliverables: Code + Report + Video Link.

Code & Report: Submit softcopy through NTUlearn.

Video: Please do not upload the video directly to NTUlearn. Instead, each group should upload their video to their own cloud storage (e.g., OneDrive, Google Drive, or other equivalent service) and generate a shareable link. The link should then be submitted via NTUlearn together with your report and code.

The TA will access your video online using the shared link. Please make sure the link is valid until grading is completed.

Grading will consider report + code + video

NOTE:

1. **MOSS:** Sharing code with your classmates is not acceptable!!! All programs will be screened using the Moss (Measure of Software Similarity.) system.
2. **You are not allowed to share your project code on the web publicly.**

TA for projects:

Liu Chen (CHEN034@e.ntu.edu.sg)

If you have questions, please email the email above. TAs can only provide some consultation for projects, but you should NOT expect TAs to help to do any part of your project.