Assignment_Arpita

Objective: To automatically map medical diagnoses to the correct **ICD-10 codes** using AI, and provide alternative suggestions and justifications for each mapping.

Approach:

- Diagnoses were provided in a Google Sheet as **lists of conditions per row**.
- **Flattened** all entries into unique individual diagnoses.
- Removed duplicates for clean, one-diagnosis-per-row mapping.
- We used **ICD-10-CM 2024 codes**, which include: ICD code and ICD description, this is version includes more detailed code , and is used widely in healthcare data systems.
- Preprocessed the data: Lower cased all text; removed special characters and extra spaces; ensured uniform formatting for better semantic matching.
- Used **all-MiniLM-L6-v2** from Hugging Face's **SentenceTransformer** library
- Converted diagnoses and ICD descriptions to **semantic embeddings**.
- For each diagnosis:
    - Compared it against all ICD-10 descriptions using **cosine similarity**.
    - Retrieved **top 3 matches**.
    - Selected the best-scoring one as the primary ICD code.
    - Stored the other 2 as alternative suggestions.
- Justified based on similarity score:
    - 0.8 = High confidence
    - 0.6–0.8 = Moderate confidence
    - < 0.6 = Low confidence → **Flagged for manual review**
- Generated a final DataFrame with:
    - Diagnosis
    - ICD-10 Code
    - ICD Description
    - Similarity Score
    - Justification
    - Alternative Suggestions
    - Needs Review (True/False)
- Exported to: icd10_mapped_output.csv

Why Sentence Transformers (all-MiniLM-L6-v2)?

- The task involves matching **short clinical phrases** (e.g., "Diabetes", "Cough") to **longer ICD-10 descriptions**.
- all-MiniLM-L6-v2 is a **lightweight, efficient transformer model** that produces high-quality **semantic sentence embeddings**.
- It allows us to compare two pieces of text **based on meaning**, not just keyword overlap.
- We chose it because:

> It is public, reliable, and works on Colab.

> It balances **accuracy** and **speed**, and is often used for semantic similarity in production.

Why Top-3 Suggestions + Justification?

- Medical language can be ambiguous (e.g., “weakness” could be neurological or general).
- To avoid incorrect one-to-one matches:
    - We include the **top-3 closest ICD codes**.
    - We flag low-confidence mappings (`< 0.6`) as **"Needs Review"**.
- Each primary mapping includes a **justification** based on the similarity score to help explain why that ICD code was selected.

Did not use LangChain or LLMs because they are good for text generation, not exactly for label matching. SentenceTransformer with cosine similarity is **faster** and **sufficient** for structured mapping tasks. This approach avoids API costs and reduces dependency on external black-box models.

Sheet link for output: icd10_mapped_output