GlucoSense: A RAG-LLM System for Personalized Diabetes Risk Prediction and Explanation

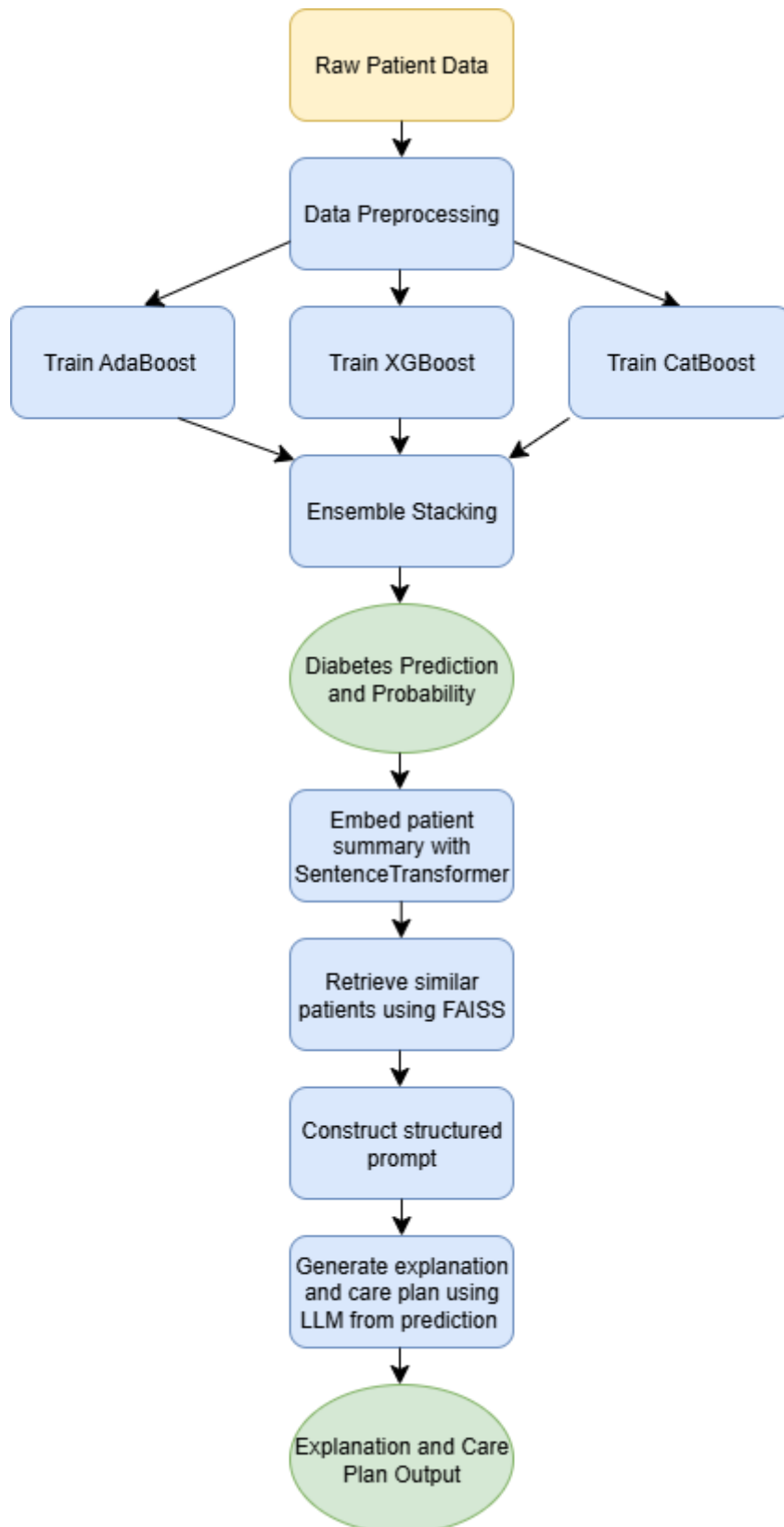|	
Software Requirements	**OS**: Windows 10 / Ubuntu 22.04 **Language**: Python 3.10 **Libraries**: scikit-learn, imbalanced learn, faiss-cpu, sentence-tranformers, xgboost, catboost, shap **Embedding model**: sentence-transformers/all-MiniLM-L6-v2
Hardware Requirements	**Processor**: Intel i5 or equivalent **RAM**: Minimum 8GB **Disk**: Minimum 20 GB free space **GPU** For LLM

Abstract

This project aims to develop an intelligent, hybrid healthcare framework that combines structured machine learning models, semantic similarity search, and large language model (LLM) reasoning to predict diabetes risk and provide personalized care insights.

GlucoSense leverages advanced ensemble models (AdaBoost + XGBoost with CatBoost meta-learner) for accurate risk prediction, and uses SentenceTransformer-based semantic embeddings with FAISS indexing to retrieve historically similar patient cases. This not only enhances transparency and trust, but also allows contextual, analogical reasoning.

In the extended version of the system (to be added), an LLM wrapper (e.g., GPT-4 or Mistral) will interpret model outputs and retrieved cases to explain *why* a particular prediction was made, and suggest a care plan tailored to the patient's clinical context.

Methodology

```
              ┌──────────────────┐
              │ Raw Patient Data │
              └──────────────────┘
                       │
                       ▼
              ┌──────────────────┐
              │ Data Preprocessing│
              └──────────────────┘
            ╱          │          ╲
           ▼           ▼           ▼
┌──────────────┐ ┌──────────────┐ ┌──────────────┐
│ Train AdaBoost│ │ Train XGBoost│ │ Train CatBoost│
└──────────────┘ └──────────────┘ └──────────────┘
           ╲          │          ╱
            ▼          ▼         ▼
              ┌──────────────────┐
              │ Ensemble Stacking│
              └──────────────────┘
                       │
                       ▼
              ╭──────────────────╮
              │ Diabetes Prediction│
              │   and Probability │
              ╰──────────────────╯
                       │
                       ▼
              ┌──────────────────┐
              │ Embed patient    │
              │ summary with     │
              │ SentenceTransformer│
              └──────────────────┘
                       │
                       ▼
              ┌──────────────────┐
              │ Retrieve similar │
              │ patients using FAISS│
              └──────────────────┘
                       │
                       ▼
              ┌──────────────────┐
              │ Construct structured│
              │      prompt       │
              └──────────────────┘
                       │
                       ▼
              ┌──────────────────┐
              │ Generate explanation│
              │ and care plan using│
              │ LLM from prediction│
              └──────────────────┘
                       │
                       ▼
              ╭──────────────────╮
              │ Explanation and Care│
              │    Plan Output    │
              ╰──────────────────╯
```

Functional Components of the Project

1. Diabetes Prediction Engine

- Trained on structured features using an ensemble of:
    - AdaBoost Classifier
    - XGBoost Classifier
    - CatBoost Classifier (as meta-learner)
- Preprocessing includes scaling, feature selection, and SMOTE oversampling.
- Returns: Prediction (diabetic/non-diabetic) and probability score.

2. Semantic Similarity Engine (FAISS + SentenceTransformer)

- Converts patient data into descriptive medical summaries.
- Embeds them using MiniLM sentence transformer (all-MiniLM-L6-v2).
- Stores embeddings in a FAISS index for efficient similarity search.
- On prediction, retrieves top-k similar historical cases from the same predicted class.
- Returns: Table of similar patients with key vitals (age, BMI, glucose, etc.).

3. LLM Wrapper (Personalized Explanation + Care Plan Generator)

- Uses retrieved similar cases and prediction output as prompt context.
- Feeds into GPT-based or open-source LLM.
- Prompt includes:
    - Predicted label and probability.
    - Patient vitals.
    - Top-5 similar case summaries.
- Returns:
    - Human-readable explanation of why the prediction was made.
    - Personalized care advice:
        - Lifestyle adjustments.
        - Medical check-ups.
        - Risk mitigation steps.