T.C.
SAKARYA UNIVERSITY
FACULTY OF COMPUTER AND INFORMATION SCIENCES

SWE402 - Senior Design Project

Machine Learning Retrieval-Augmented Generation System for Medical Analysis

B201202041 Yiğit SERT

Department: Software Engineering
Supervisor: Prof. Dr. Devrim AKGÜN

2024-2025 Spring Semester

Contents

List of Figures

List of Tables

Abstract

The dual challenge in modern medical AI is to achieve predictive accuracy while ensuring that the resulting insights are transparent, trustworthy, and accessible to both clinicians and patients. For this, a framework that tackles this issue by synergizing the strengths of interpretable machine learning with the advanced natural language capabilities of generative models.

This study presents an approach for personalized health recommendation generation by combining interpretable machine learning models with advanced large language models (LLMs). Decision tree-based classifiers are trained on a health dataset to predict potential health risks. From these models, human-readable if-then rules are extracted to provide transparent and understandable decision logic. These extracted rules serve as the basis for prompting multiple state-of-the-art LLMs to generate detailed and personalized health advice tailored to individual profiles. The proposed system is implemented with an interactive user interface, enabling users to input health data and receive customized recommendations. Experimental results demonstrate the effectiveness of this hybrid approach in improving recommendation quality and interpretability, highlighting the potential of integrating classical machine learning with generative AI for healthcare applications.

1 Introduction

In recent years, the proliferation of health-related data from electronic health records, wearable devices, and patient surveys has provided opportunities for personalized healthcare. This study aims to develop an interpretable machine learning framework that leverages individual health data to generate customized health recommendations. The framework combines decision tree classifiers, which are well-known for their transparency and ease of interpretation, with large language models (LLMs). The decision trees analyze health data to extract meaningful decision rules, while the LLMs translate these rules into personalized, user-friendly health advice. The scope of this research includes data preprocessing, model training, rule extraction, natural language generation of recommendations, and evaluation of the overall system's effectiveness in delivering actionable health insights.

This work focuses primarily on bridging the gap between complex machine learning outputs and end-user comprehension, which is particularly critical in health applications where interpretability and trust are essential. By integrating interpretable models with advanced language technologies, the study addresses the challenge of converting data-driven insights into practical, understandable recommendations that patients and healthcare providers can readily use.

1.1 What Are Large Language Models?

Large Language Models (LLMs) represent a significant leap in artificial intelligence, designed to comprehend and generate human-like text. At their core, LLMs are a type of deep learning model, typically built on neural network architectures like the Transformer, which excel at processing sequential data. What makes them "large" is the sheer volume of data they are trained on—billions, even trillions, of words, sentences, and documents. This extensive training enables them to learn intricate patterns, grammar, semantics, and contextual relationships within language, allowing them to perform a wide array of natural language processing (NLP) tasks with remarkable proficiency [3].

The capabilities of LLMs extend far beyond simple text generation. They can summarize lengthy documents, translate languages, answer complex questions, assist with creative writing, and even generate computer code. This versatility stems from their ability to predict the most probable next word in a sequence based on the preceding context, a process refined through vast datasets. While they don't "understand" language in the same way humans do, their statistical prowess allows them to produce coherent, contextually relevant, and often highly creative outputs, making them powerful tools for revolutionizing industries ranging from customer service and content creation to research and software development [4].

1.2 Importance of Health Data and Personalized Recommendations

The increasing digitization of healthcare has led to an explosion in the quantity and variety of available health data [5]. This data, ranging from clinical measurements to lifestyle and behavioral factors, holds the potential to revolutionize health management and disease prevention [6]. However, extracting relevant and actionable information from this vast amount of data remains a significant challenge. Traditional one-size-fits-all health advice often fails to account for individual variability, reducing its effectiveness.

Personalized health recommendations are essential for tailoring advice to an individual's unique health profile, risk factors, and lifestyle [7]. Such tailored guidance can improve patient outcomes by focusing preventive strategies and treatments where they are most needed. Personalized recommendations also empower individuals to take proactive control of their health, potentially reducing the burden on healthcare systems. The challenge lies in developing computational methods that can analyze complex data sets and communicate the results in a clear, trustworthy manner. This study addresses this need by combining interpretable machine learning with natural language generation, enabling personalized recommendations that are both data-driven and easy to understand.

1.3 Motivation

Decision trees were selected as the core predictive model due to their inherent interpretability and simplicity. Unlike many black-box machine learning methods, decision trees provide clear, rule-based outputs that can be directly understood and validated by clinicians and patients alike. This transparency is crucial in healthcare, where decisions based on model predictions must be explainable to ensure ethical use and user trust. Additionally, decision trees facilitate the extraction of decision rules, which form the foundation for generating personalized advice.

Despite their strengths, decision trees alone may produce outputs that are still too technical or complex for non-expert users. To overcome this limitation, the study integrates large language models (LLMs), advanced natural language processing tools capable of generating coherent and contextually appropriate text. LLMs translate the decision rules derived from the trees into accessible, personalized health recommendations. This combination leverages the strengths of both technologies: the rigor and clarity of decision trees with the communicative power of LLMs. The integration aims to improve both the interpretability and usability of AI-driven health recommendations.

2 Background

Interpretable and personalized health insights are essential for preventive care. Decision trees offer transparency through rule-based predictions, but these rules are often difficult for individuals to understand. Large Language Models (LLMs) help bridge this gap by transforming complex decision rules into clear, personalized recommendations. Combining decision trees with LLMs enables systems that are both accurate and user-friendly, supporting better-informed health decisions.

2.1 Decision Tree Algorithms

Decision trees create interpretable models by recursively splitting data based on feature thresholds. Advanced methods like Random Forest and XGBoost improve accuracy by combining multiple trees but reduce interpretability, highlighting the need for rule extraction techniques. An illustration that briefly summarizes the main differences and similarities between these algorithms are shown in the Figure 1

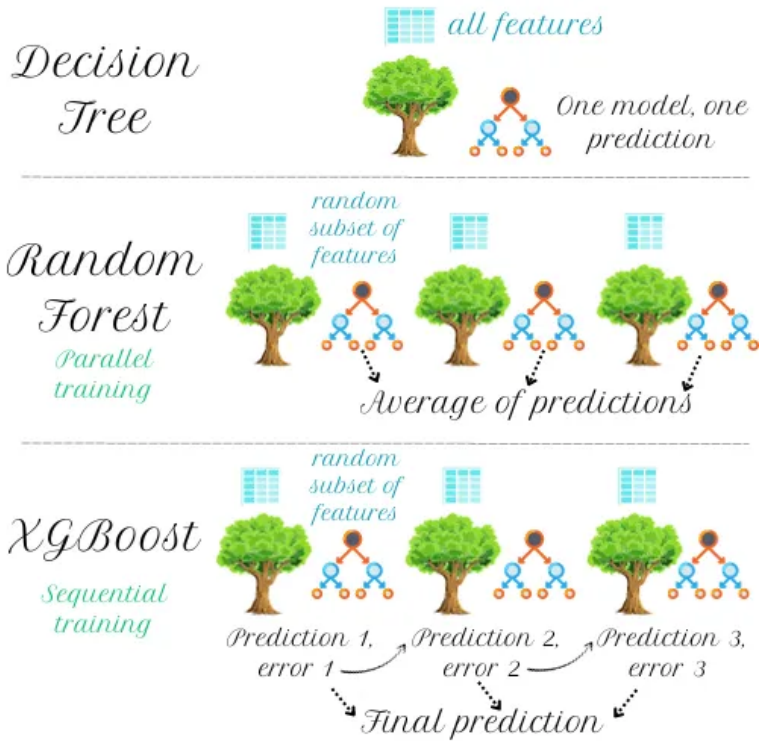

Figure 1: Comparison of Decision Tree, Random Forest, and XGBoost Algorithms [1]

Decision tree algorithms are among the most widely used methods for classification and regression due to their interpretability and intuitive structure. A basic decision tree recursively splits the dataset based on feature thresholds to create a tree-like model of

decisions. Each internal node represents a test on a feature, each branch corresponds to an outcome, and each leaf node represents a class label or prediction [8].

Random Forest enhances the performance of single decision trees by aggregating the results of multiple trees trained on different data subsets and random feature selections. This ensemble approach reduces overfitting and improves generalization but slightly sacrifices interpretability due to the number of trees involved [9].

XGBoost (Extreme Gradient Boosting) is a more recent and powerful algorithm that builds trees sequentially, where each new tree corrects the errors made by previous ones. It is known for its speed and predictive accuracy and is often used in competitive machine learning. However, like Random Forests, it is harder to interpret directly, which motivates the need for rule extraction methods [10].

2.2 Rule Extraction and Interpretability

Interpretability in machine learning refers to the degree to which a human can understand the internal mechanics of a model or the reasons behind its predictions. In healthcare, interpretability is not optional—it is essential for trust, compliance, and clinical adoption [11].

Rule extraction is a technique that transforms model behavior into human-readable if-then rules. These rules can be derived directly from decision trees or approximated from more complex models like Random Forest or XGBoost. Such extracted rules help explain how input features lead to specific predictions, allowing domain experts to validate or contest the model's logic [12].

In this study, rule extraction is used to translate the output of decision-based models into intermediate representations, which are then passed to LLMs for natural language generation. An illustration can be observed in the Figure 2

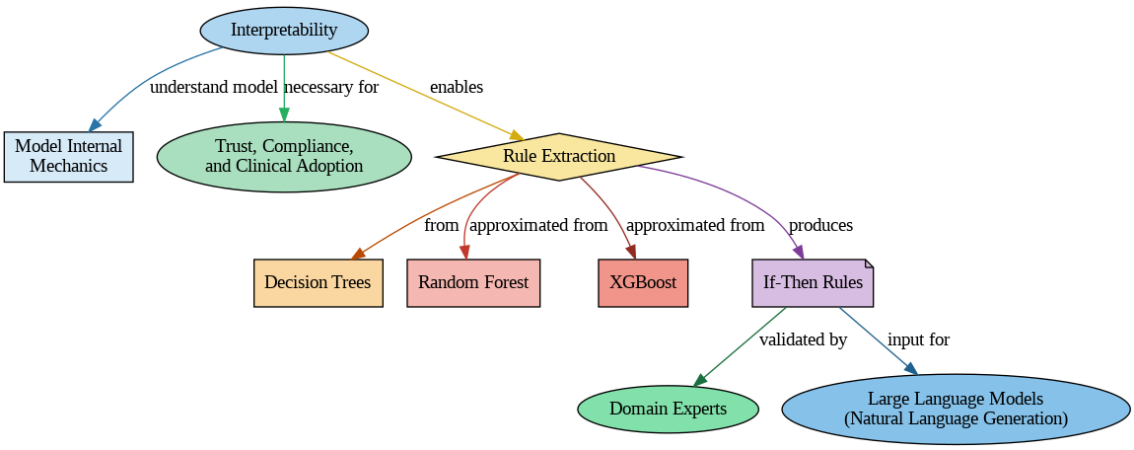

Figure 2: Conceptual Map of Interpretability and Rule Extraction

2.3 Applications of LLMs in Healthcare

Large Language Models (LLMs) such as GPT, BERT, and Gemini represent a major improvement in AI. Their ability to process and generate human-like text opens up new opportunities in healthcare communication, diagnosis support, and patient engagement [13]. Some areas are shown in the Figure 3

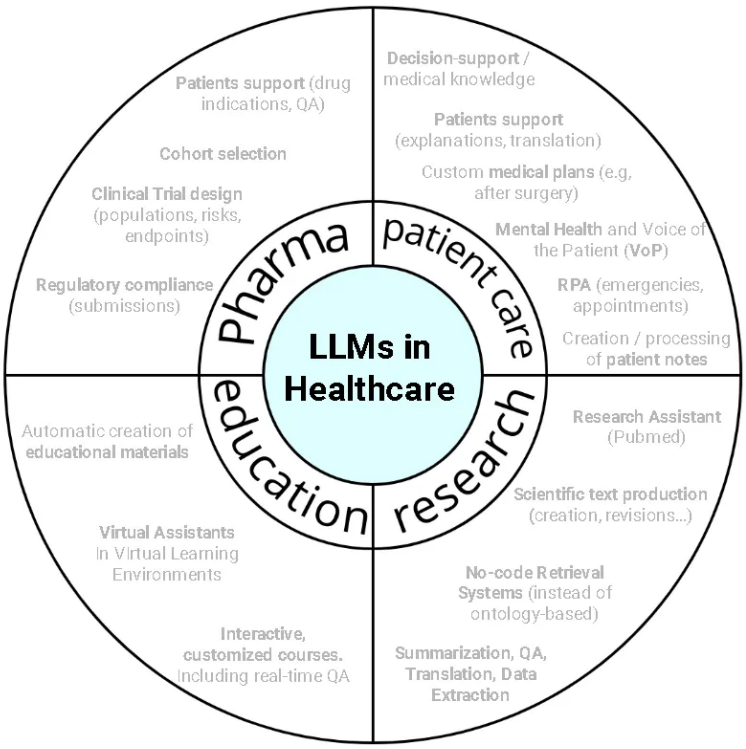

Figure 3: LLMs in Healthcare [2]

LLMs can be used to generate summaries of clinical data, assist in medical documentation, and even converse with patients to clarify health information [14]. In this study, LLMs are employed to convert structured decision rules into user-friendly, personalized health recommendations, thereby making complex machine learning outputs accessible to non-expert users.

Such applications must be carefully designed to ensure factual accuracy, safety, and alignment with clinical best practices.

2.4 Natural Language Processing (NLP) Techniques for Personalized Advice Generation

Natural Language Processing (NLP) is a subfield of AI that deals with the interaction between computers and human language. NLP techniques enable machines to understand, interpret, and generate natural language text [15].

In the context of personalized health advice, NLP enables the dynamic construction of context-aware messages tailored to an individual's health profile. Techniques like prompt engineering, template-based generation, and controlled text generation are used to ensure that the output is relevant, coherent, and easy to understand [16].

By integrating NLP with rule-based models, this study bridges the gap between raw machine outputs and meaningful, human-centric communication.

2.5 Review of Related Work and Research Gaps

Several studies have explored interpretable machine learning in healthcare, focusing on rule-based models and visualizations to explain predictions. Other works have investigated the use of LLMs for summarization, chatbot systems, and clinical decision support. However, most research treats these two areas—interpretable models and LLMs—as separate domains.

Few studies combine decision tree rule extraction with LLM-driven personalized recommendation generation. Moreover, existing systems often lack interactivity, flexibility in model comparison, and robust mechanisms for evaluating output quality.

This study addresses these gaps by proposing a unified pipeline that starts with interpretable ML models, extracts actionable rules, and uses LLMs to generate tailored health advice, evaluated through both human-readable outputs and automated quality metrics.

2.5.1 Case Study: HealifyAI – LLM-based Healthcare System

This project presents a comprehensive AI-driven healthcare system that combines traditional machine learning and a large language model (LLM) to support both medical professionals and patients. The dual-module system provides disease prediction based on symptoms and a question-answering interface capable of delivering detailed, medically grounded answers [17]. Figure 4 shows the general architecture of this system.

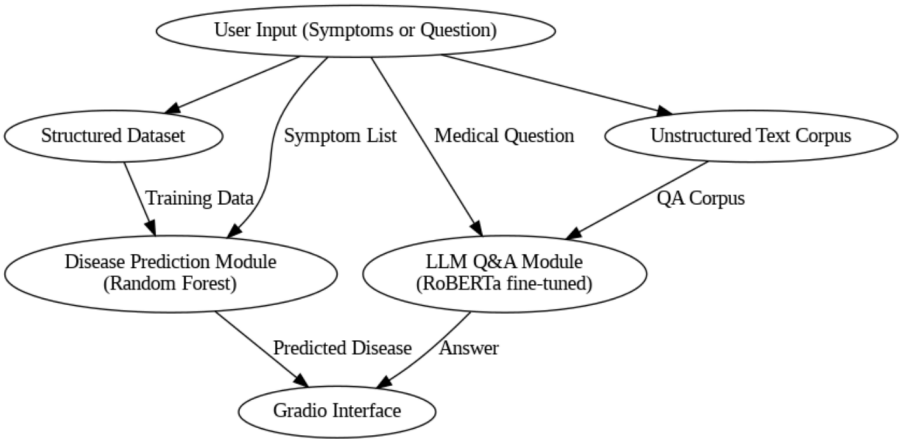

Figure 4: HealifyAI System Architecture

The main objective of HealifyAI is to create an intelligent, multi-functional medical support tool that can predict diseases from symptoms and answer complex health-related questions in natural language. The system emphasizes accuracy, interpretability, and broad coverage of diseases, ranging from common to rare conditions. By leveraging both structured datasets and unstructured medical text, HealifyAI offers a hybrid approach to enhance diagnostic support and public health literacy.

The project utilizes two core datasets:

1. **Disease Prediction Dataset**
    - **Source:** Kaggle's Disease-Symptom Knowledge Database, derived from patient records at New York Presbyterian Hospital, USA.
    - **Coverage:** Includes 135 disease categories and 400 symptoms.
    - **Data Cleaning:** Raw data contained inconsistencies in symptom names and noisy medical codes (e.g., UMLS), which were cleaned for clarity and usability.
    - **Preprocessing:** Categorical symptom data was preprocessed for machine learning input using label encoding and normalization techniques.
2. **LLM Corpus for Q&A**
    - **Composition:** A custom medical QA dataset consisting of 6,800 samples, created from scratch.
    - **Initial Collection:** Data was initially gathered through web scraping from Healthline articles.
    - **Augmentation:** Python scripts were used to simulate diverse user questioning styles, increasing the variability and naturalness of the dataset.
    - **Content:** The final dataset includes a balanced mix of condition-specific, symptom-based, and general medical knowledge queries, enabling robust performance in question answering tasks.

During the model training phase, the system was structured into two primary components: disease prediction and medical question answering. For disease prediction, a Random Forest Classifier was employed due to its interpretability and strong performance on structured, categorical data. Trained on user-provided symptom sets, the model effectively handled noisy inputs and learned to identify likely disease categories.

The question-answering component, HealifyLLM, was developed using the RoBERTa architecture, selected for its ability to manage complex language understanding tasks. The model was fine-tuned via the HuggingFace Transformers framework, using a three-stage ULMFiT-style training strategy, and achieved 98% accuracy over 12 epochs. Its performance was supported by a balanced dataset of 6,800 QA pairs spanning 135 disease types. To facilitate user interaction, the QA model was deployed with a Gradio-based interface,

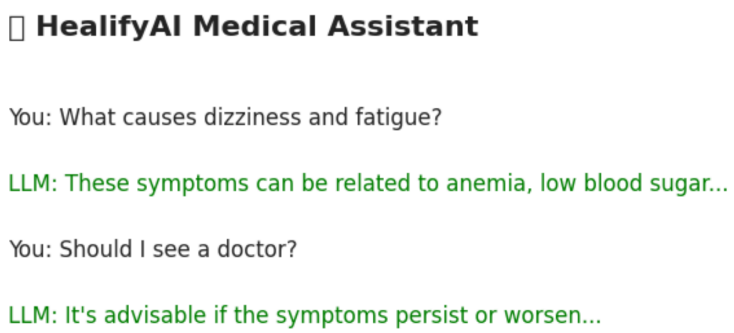

Figure 5: HealifyAI UI Mock-Up

enabling real-time, natural language responses through integration with the HuggingFace API. A mock-up is added in the Figure 5 below,

Although not based on a Retrieval-Augmented Generation (RAG) framework like MLRAG, HealifyAI is designed to emulate expert clinical reasoning by combining multi-symptom analysis with natural language explanations. The system integrates a disease prediction model that provides transparency through decision tree logic and a large language model (LLM) that delivers contextual medical advice in a user-friendly manner. Together, these components enable symptom-to-disease mapping, personalized health education, and an intuitive bridge between structured data analysis and human-comprehensible interaction, contributing to intelligent and accessible healthcare support.

2.5.2 Case Study: AI Medical Researcher MLRAG

In this project, it is shown that a comprehensive system designed to analyze health-related datasets and generate personalized recommendations using decision tree classifiers and a large language model (LLM) through Retrieval-Augmented Generation (RAG) [18].

The primary aim of this study is to develop a system that is both interpretable and adaptive to individual profiles. This system is designed to extract decision rules from health datasets, transform these rules into human-understandable language, and utilize LLMs to produce personalized recommendations. By focusing on individualized analysis rather than population-level generalizations, the proposed method contributes to the emerging field of precision public health. This field emphasizes the importance of context-aware prevention strategies tailored to specific demographic and behavioral characteristics.

This project utilizes a real-world dataset from Kaggle titled “Personal Key Indicators of Heart Disease,” which includes various lifestyle and health-related features such as BMI, smoking habits, physical activity, and sleep. The dataset primarily consists of older adults (mostly aged 60 and above) and shows that 9% of individuals have heart disease, while the majority are healthy. Key health indicators like obesity, smoking rates, and physical and mental health scores are also included.

The dataset has a balanced gender distribution (52% female, 48% male) and is of high quality with no missing or inconsistent records. Categorical variables were processed using one-hot encoding, and health outcome variables like heart disease and stroke were clearly identified as target labels. This preparation ensures the data is reliable and ready for analysis.

For each health outcome, a separate Decision Tree Classifier is trained. The feature set includes demographic variables (e.g., age, gender, ethnicity) and behavior-related variables (e.g., alcohol use, exercise, sleep quality). The depth and granularity of the decision trees are adjustable through the parameter to allow either compact or highly detailed rule extraction. A visualization is shown in the Figure 6.

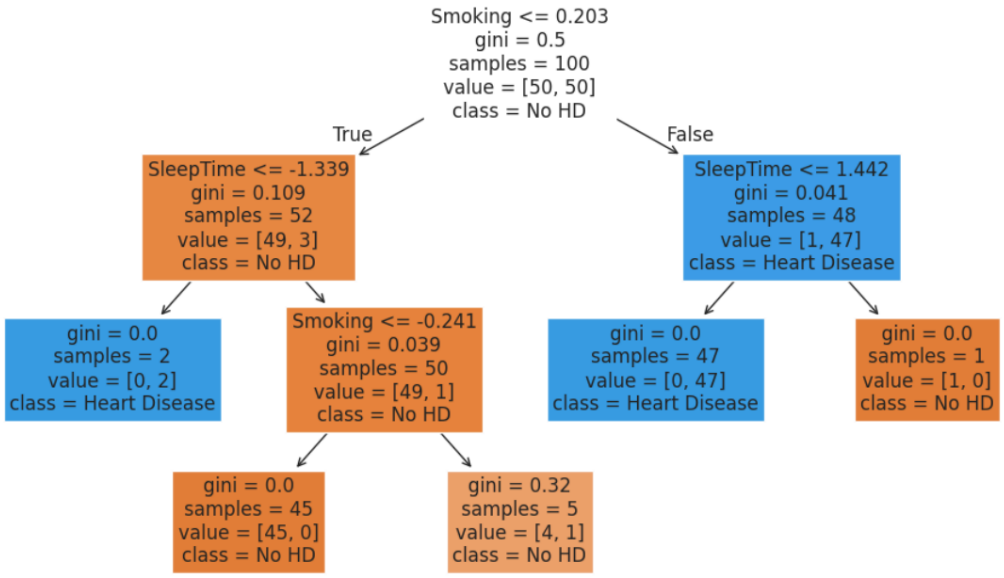

Figure 6: Example Decision Tree for Health Disease Prediction

After training, each decision tree is analyzed to extract logical rules that represent the paths from root to leaves. These rules are then converted into readable statements that explain under what conditions the risk for a health condition increases or decreases.

The extracted rules are incorporated into structured prompts, which are then fed into the Gemini 1.5 Pro model, a high-capacity LLM capable of processing long-context inputs and supporting RAG mechanisms. In each prompt, the decision rules are accompanied by synthetic or real health profiles that describe individual cases in terms of age, sex, race, BMI, lifestyle behaviors, and existing medical conditions. A mock-up for the system can be viewed in the Figure 7 below.

The LLM is instructed to perform multiple tasks: identify and rank the most critical health risk factors for the given profile, interpret model predictions in layman's terms, and provide actionable recommendations tailored to the individual's lifestyle and demographics. The prompt also includes requests for reflection on model limitations, such as potential bias or insufficient data representation.

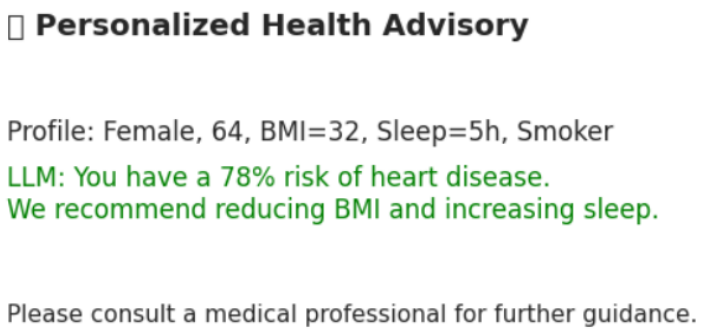

Figure 7: Personalized Recommendation Mock-Up

2.5.3 Case Study: Medical LLM – Structured Pipeline for Medical Data Interpretation

This project introduces a structured pipeline built around Llama 2, specifically optimized for interpreting, processing, and transforming medical data using large language models (LLMs). The system aims to bridge the gap between unstructured medical information—such as imaging descriptions, clinical reports, and CSV datasets—and actionable structured output. It reflects a broader trend toward using LLMs to automate and enhance medical workflows [19].

The primary goal of the Medical LLM project is to build a reliable, end-to-end pipeline that processes various types of medical data—including text, tabular data, and imaging metadata—and transforms them into structured formats such as JSON. This is achieved through prompt engineering and leveraging LLM grammar capabilities. Unlike general-purpose LLM projects, Medical LLM is tailored for healthcare-specific applications, prioritizing security, precision, and scalability. The pipeline's core is the Llama 2 model, which is adapted for medical reasoning and structured output generation through llama.cpp.

The pipeline is designed to accommodate diverse types of medical data. In the preprocessing stage, all inputs—whether textual, tabular, or imaging-based—are cleaned and sanitized to remove inconsistencies. Following this, the data is automatically classified into relevant categories such as tabular (CSV), textual (reports), or imaging (e.g., DICOM metadata) to enable specialized handling. Finally, dedicated parsing tools are used to convert these inputs into formats that LLMs can effectively understand and reason over. Sample datasets and preprocessing utilities are provided for testing and demonstration purposes.

A curated set of prompt templates is included in the project to guide the LLM when interpreting medical data. These prompts are designed to enable tasks such as interpreting medical conditions using natural language, extracting structured information, and generating JSON outputs through Llama 2's grammar functionality. In addition to prompt engineering, the Llama 2 model is fine-tuned using provided training scripts, focusing on various medical tasks such as diagnosis extraction, clinical report summarization, and

recognizing symptom patterns.

Structured output is a critical requirement for medical data processing. The system transforms LLM responses into well-defined formats such as JSON or FHIR-compatible structures. To ensure reliability, validation scripts are included to check the accuracy and consistency of these outputs. Furthermore, the pipeline is developed with strict attention to security and privacy, adhering to regulations relevant to medical data handling.

3 Proposed model

This section details the architecture and implementation of the proposed system, which integrates a suite of machine learning models with large language models (LLMs) to generate personalized health predictions and recommendations through an interactive user interface. The design adopts a modular and explainable AI approach, focusing on interpretable models for health risk assessment (with Decision Trees enabling rule extraction) and leveraging the generative capacity of LLMs for user-friendly guidance. The methodology emphasizes user interaction, model persistence, flexibility in target disease selection, and a novel LLM-based advice evaluation component.

The proposed model is composed of several interdependent components, managed through a multi-tab user interface, starting from data collection and preprocessing, proceeding through model training (or loading persisted models) and rule extraction, and culminating in personalized predictions, LLM-based advice generation, LLM-based evaluation of that advice, and multi-format report saving. The rationale behind this multi-stage pipeline is to ensure the validity of predictions, enhance user trust through interpretability, and enable non-expert users to benefit from AI-driven insights in an intuitive and actionable manner.

3.1 System Architecture and Workflow

The system follows a multi-stage, sequential workflow managed through a tabbed user interface:

1. **Data Ingestion:** Loading the health dataset.
2. **Preprocessing & Preparation:** Cleaning column names, mapping categorical and target variables, and performing one-hot encoding. Users can select the target disease (e.g., `HadDiabetes`, `HadStroke`) for subsequent steps.
3. **Data Splitting:** Dividing the dataset into training and testing sets.
4. **ML Model Training/Loading & Evaluation:** Training a suite of classifiers (Decision Tree, Logistic Regression, Random Forest, Gradient Boosting, KNN) or loading previously saved models for the selected target. Performance is evaluated using metrics like accuracy, precision, recall, F1-score, confusion matrices, and feature importances are visualized. Trained models are saved to disk using `joblib` for future use.
5. **Rule Extraction:** Extracting if-then rules from the trained Decision Tree model.
6. **Personalized Prediction:** Allowing users to input their health data via a dynamically generated form tailored to the features of the selected trained model, and receiving a risk prediction.

7. **LLM-Powered Advice Generation:** Using extracted rules and user-provided profiles to prompt multiple LLMs for personalized health advice.

8. **LLM-based Advice Evaluation:** Employing an LLM to critically evaluate the generated advice from another LLM based on predefined quality criteria.

9. **Report Saving:** Enabling users to save the combined advice and evaluation in various formats (Markdown, TXT, HTML, PDF).

The architecture is designed to be modular, allowing components to be updated or experimented with independently. This supports scalability and iterative development.

A diagram representing the system architecture is shown in Figure 8.

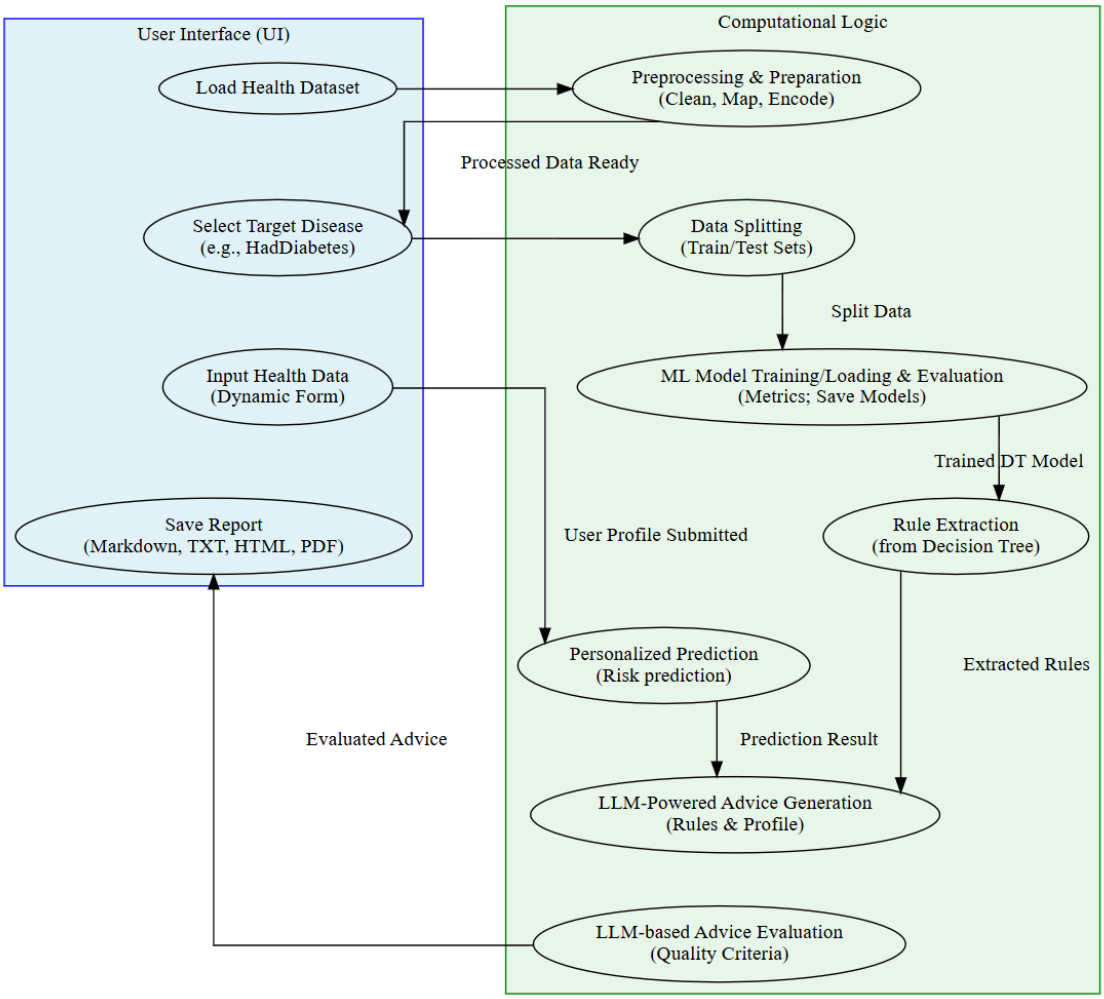

Figure 8: System Architecture Overview

3.2 Dataset Description and Preprocessing Steps

The primary dataset used in this study is the "Personal Key Indicators of Heart Disease 2022" dataset, publicly available on Kaggle, curated by Kamil Pytlak [20]. It is derived from the Behavioral Risk Factor Surveillance System (BRFSS) 2022 survey. The specific file utilized is `heart_2022_no_nans.csv`, which is a pre-cleaned version containing responses from a large number of adults in the U.S. and is focused on variables relevant to cardiovascular health and related conditions like diabetes.

A summary can be examined in the Table 1 given below.

Table 1: Summary of Indicators of Heart Disease (Top 3 Features per Category)

Dataset Summary	
Heart Disease	**BMI**
True: 27,373 (9%)	Mean: 28.3
False: 292,422 (91%)	Median: 27.3
Total: 320,000 (100%)	Std. Deviation: 6.36
Smoking	**Alcohol Drinking**
True: 131,908 (41%)	False: 298,018 (93%)
False: 187,887 (59%)	True: 21,777 (7%)
Total: 320,000 (100%)	Total: 320,000 (100%)
Stroke	**Physical Health (days)**
False: 307,726 (96%)	Mean: 3.37
True: 12,069 (4%)	Median: 2
Total: 320,000 (100%)	Std. Deviation: 7.95
Mental Health (days)	**Sex**
Mean: 3.9	Female: 52%
Median: 3	Male: 48%
Std. Deviation: 7.96	Total: 320,000 (100%)

The raw dataset undergoes several preprocessing steps managed through the UI. First, column names are cleaned by removing special characters and spaces, for example, "Had HeartAttack" becomes "HadHeartAttack". Next, target variables are mapped. The primary target variable for multi-class demonstration, HadDiabetes, is mapped from string categories such as "No" or "Yes" to numerical values (0, 1, 2, 3). Similarly, other potential binary target variables like HadStroke and HadHeartAttack are mapped from "Yes"/"No" strings to 1/0 integers. Users can also select which "Had..." column serves as the target y variable for model training and prediction. Then, categorical feature encoding is applied. Other categorical features, including State, Sex, GeneralHealth, and RaceEthnicityCategory, are identified, and one-hot encoding (pd.get_dummies with drop_first=True) converts them into a numerical format suitable for machine learning algorithms. This expands

the feature set by creating new binary columns for each category, excluding one per original feature due to drop_first. Finally, the feature set is defined: all remaining columns after separating the target variable form the feature set X. The system stores the list of these feature names, model_feature_names, which are crucial for ensuring consistency during prediction. The result of these steps is a fully numerical df_encoded DataFrame, ready for splitting and model training. A review can be examined in the Figure 9.

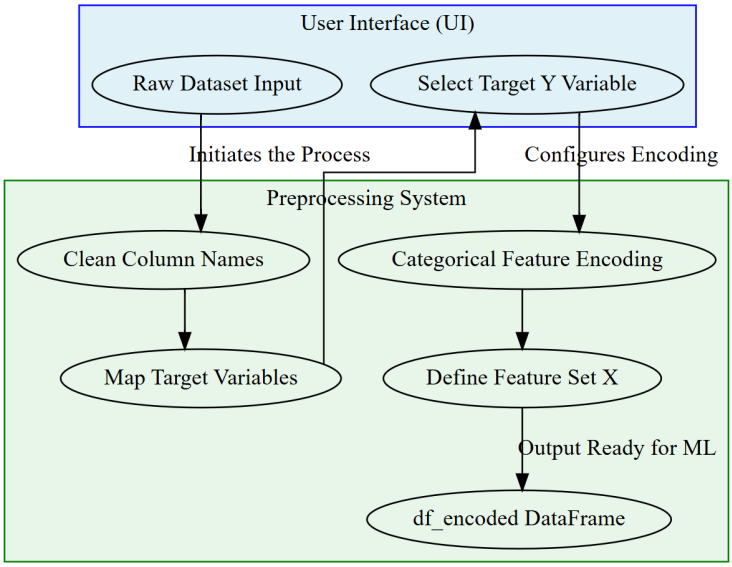

Figure 9: Preprocessing System Architecture Overview

3.3 Training, Tuning, and Persistence of Predictive Models

The system trains and evaluates a suite of machine learning models on the preprocessed and split dataset. The user selects the target disease in a prior step, and models are trained specifically for that target. The models include:

- **Decision Tree Classifier:** Chosen for its interpretability and ability to extract rules. Parameters like `max_depth=7` and `min_samples_leaf=10` are set to prevent overfitting.
- **Logistic Regression:** A robust linear model for binary or multi-class classification, often serving as a good baseline (`max_iter=1000`, `solver='liblinear'`).
- **Random Forest Classifier:** An ensemble method that aggregates predictions from multiple decision trees to improve accuracy and reduce overfitting (`n_estimators=100`, `max_depth=10`).
- **Gradient Boosting Classifier:** Another powerful ensemble technique that builds trees sequentially, with each new tree correcting errors of the previous ones (`n_estimators=100`, `max_depth=5`).

- **K-Nearest Neighbors (KNN):** A non-parametric, instance-based learning algorithm (`n_neighbors=7`).

Each model is trained on the X_{train}, y_{train} data. Hyperparameter tuning is implicitly handled by using common default or slightly adjusted parameters known to work reasonably well.

Performance is evaluated on the X_{test}, y_{test} set using standard metrics:

Accuracy measures the proportion of correctly predicted observations (both positives and negatives) to the total observations. It is a general indicator of model performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision indicates how many of the positive predictions made by the model are actually correct. It is especially important when the cost of false positives is high.

$$Precision = \frac{TP}{TP + FP}$$

Recall measures how many of the actual positive cases the model correctly identified. It is crucial when missing a positive case has a high cost (e.g., medical diagnosis).

$$Recall = \frac{TP}{TP + FN}$$

The F1-score is the harmonic mean of precision and recall. It provides a balanced measure that is especially useful for imbalanced datasets.

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives) represent the four basic outcomes. For multi-class targets, precision and recall are typically averaged (e.g., weighted average). Additionally, confusion matrices are generated, and feature importances are plotted for tree-based models.

3.4 Extraction of If-Then Rules from Decision Tree Models

From the trained (or loaded) Decision Tree model, interpretable if-then rules are extracted using the `export_text` function from `sklearn.tree`. Each path from the root of the tree to a leaf node represents a unique rule, describing a combination of feature conditions leading to a specific predicted outcome for the `current_target_disease`.

These rules are stored in a structured text format and are later used as a crucial part of the context provided to LLMs for generating personalized advice. This approach ensures that the health insights are traceable back to the model's logic, aligning with ethical AI principles and enhancing user trust.

3.5 Designing Prompt Templates

To transform the extracted symbolic rules and user-specific information into natural-language recommendations, structured prompt templates are designed. Each prompt dynamically incorporates:

1. **Task Definition:** Instructing the LLM to act as a helpful AI assistant.
2. **Target Condition:** Clearly stating the health condition the user is concerned about (e.g., `current_target_disease`).
3. **Patient Profile:** User-provided text describing their age, gender, lifestyle, habits, and any known conditions.
4. **Extracted Decision Rules:** A (potentially truncated for brevity) set of if-then rules from the Decision Tree model relevant to the target condition.
5. **Instructions for Advice Generation:** Guiding the LLM to provide empathetic, actionable advice, focus on lifestyle changes, preventative measures, and when to consult a healthcare professional.
6. **Crucial Disclaimers:** Explicitly instructing the LLM *not* to provide a diagnosis, not to interpret rules as definitive medical facts for the individual, and to strongly encourage consultation with a doctor for medical concerns.

An example prompt structure is:

```
You are a helpful AI assistant providing health-related suggestions...
**Patient Profile:**
[User-provided profile text]
**Key Decision Tree Rules (these rules helped predict risk...):**
[Extracted rules from Decision Tree model]...
Based on this information, provide empathetic and actionable advice...
**Important: Do NOT provide a diagnosis... Encourage consultation...**
```

These prompts serve as the interface between the symbolic rule logic from the ML model and the generative reasoning capabilities of the LLMs.

3.6 Integration With Multiple LLM APIs for Advice Generation

The system is integrated with several leading LLM providers and models to offer diverse perspectives in health advice generation. The current integrations include:

- **Hugging Face:** `tiiuae/falcon-7b-instruct`
- **Google Gemini:** `gemini-1.0-pro`

- **Together AI:** `mistralai/Mixtral-8x7B-Instruct-v0.1`
- **Groq Cloud:** `llama3-8b-8192` and `gemma-7b-it`

A universal `get_response` function handles communication with each provider. Users can select one or multiple LLMs to generate advice for the same profile and rule set, allowing for comparative analysis. A mock-up is shown in the Figure 10.

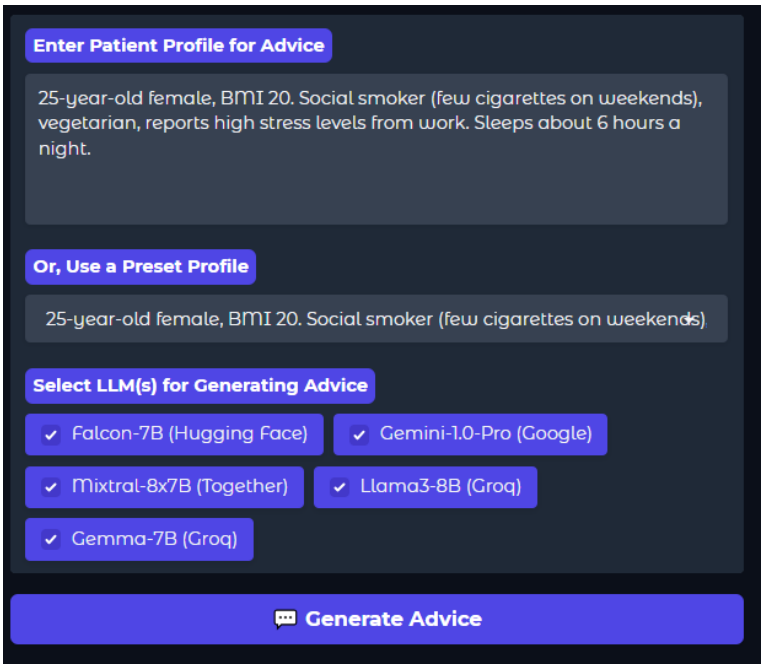

Figure 10: Multiple LLM Integration Mock-Up

3.7 LLM-based Evaluation of Generated Health Advice

A component of this system is the ability to use one LLM to critically evaluate the health advice generated by another LLM. When a user triggers this evaluation:

1. The previously generated health advice text is taken as input.
2. The system automatically calculates quantitative readability scores for the text using established formulas.
3. The user selects an ”evaluator” LLM from the available models.
4. A specific prompt is constructed for the evaluator LLM, instructing it to assess the provided advice based on predefined criteria such as: Clarity, Actionability, Safety, Avoidance of Diagnosis, Empathy, Practicality, Encouragement of Professional Consultation, and Potential Biases.

5. The evaluator LLM provides its critique in a structured format.

The primary readability metrics employed are the Flesch-Kincaid scores, which are widely used to assess the comprehensibility of English text. In the context of health communication, readability is a critical component of safety and efficacy. Complex language can create barriers to understanding, potentially leading to misinterpretation or non-adherence to advice, particularly for users with lower health literacy. We utilize two specific scores:

- **Flesch Reading Ease:** This score rates text on a 100-point scale, where higher scores indicate greater readability. A score of 60–70 is considered plain English, typically understood by individuals with an 8th to 9th grade reading level (Flesch, 1948). Our system flags scores below 60 as potentially too complex for general health advice.
- **Flesch-Kincaid Grade Level:** This score translates the 100-point scale into a U.S. school grade level, indicating the years of education required to comprehend the text (Kincaid et al., 1975). For public health information, a grade level of 8–9 is often recommended as the target to ensure broad accessibility.

As shown in Table 2, the Flesch Reading Ease Score categorizes text readability from "Very Easy" to "Very Difficult" based on numerical ranges.

Flesch Ease Score	Interpretation	Typical FK Grade Level
90–100	Very Easy	5th Grade
60–70	Plain English (Recommended for Public)	8th–9th Grade
50–60	Fairly Difficult	10th–12th Grade
30–50	Difficult	College Level
0–30	Very Difficult	University Graduate

Table 2: Flesch Reading Ease Score and Interpretation

This LLM-based critique helps identify strengths and weaknesses in the generated advice and offers users a more nuanced understanding of the AI-generated recommendations.

3.8 User Interface Implementation and Workflow

The system features a web-based user interface (UI) built using Gradio. The UI is organized into sequential tabs to guide the user through the entire pipeline.

3.8.1 Data Loading and Preparation Interface (Tabs 1 & 2)

- **Tab 1 (Load Dataset):** Loads the `heart_2022_no_nans.csv` dataset, displaying status, data preview, dimensions, and column information, as shown in the Figure 11.

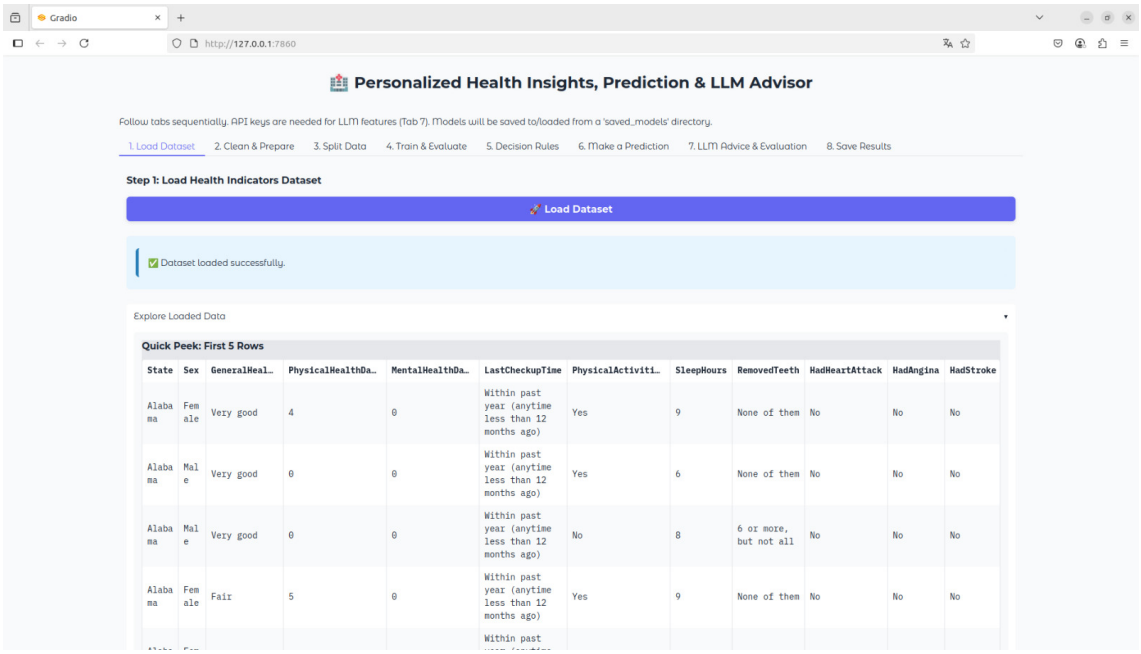

Figure 11: Load Dataset

- **Tab 2 (Clean & Prepare):** Initiates data cleaning, target variable mapping, and one-hot encoding. Users select the target disease for model training. UI shows status, processed data previews, and transformation summaries, as shown in the Figure 12.

3.8.2 Model Training and Evaluation Interface (Tabs 3 & 4)

- **Tab 3 (Split Data):** Splits data into training/testing sets for the selected target. Displays split info and target distribution visualizations, as shown in the Figure 13.
- **Tab 4 (Train & Evaluate):** Trains (or loads pre-trained) ML classifiers. Displays a training log, comparative performance metrics table, a comparison plot, feature importance plots for tree-based models, and a selectable confusion matrix, as shown in the Figure 14.

3.8.3 Dynamic Prediction Interface (Tabs 5 & 6)

- **Tab 5 (Decision Rules):** Extracts and displays if-then rules from the active Decision Tree model, as shown in the Figure 15.
- **Tab 6 (Make a Prediction):** Users select an active ML model. The UI dynamically generates an input form based on the model's features, with user-friendly elements for common inputs and textboxes for other features. Preset profiles can

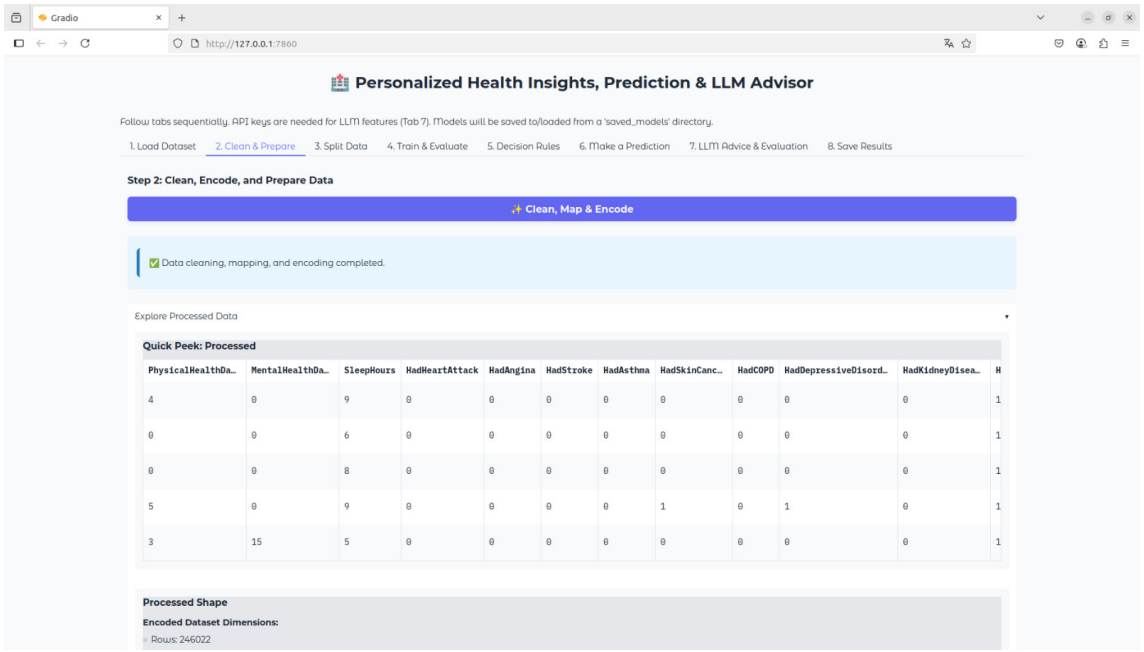

Figure 12: Clean and Prepare

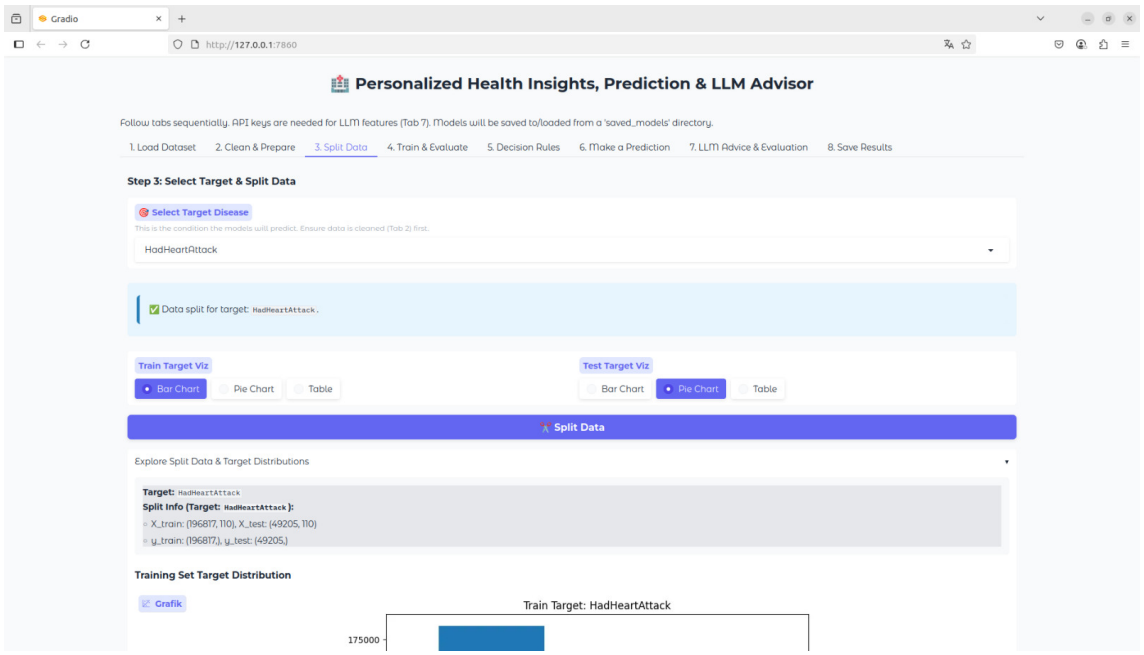

Figure 13: Split Data

auto-fill parts of the form. The system provides the model's prediction and class probabilities, as shown in the Figure 16.

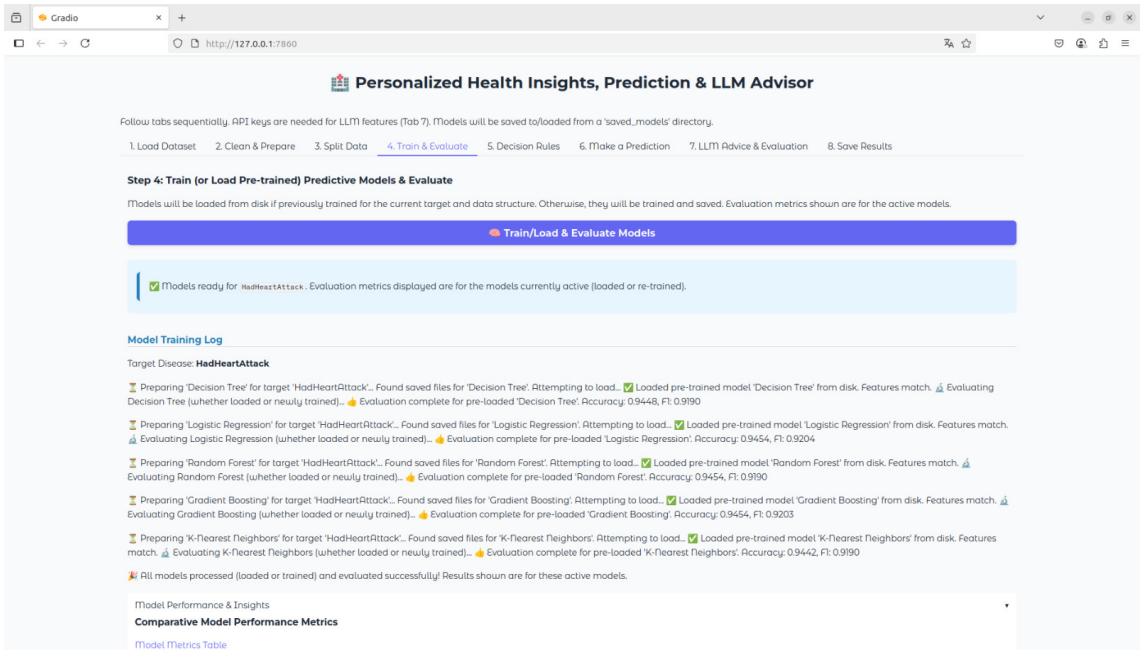

Figure 14: Train and Evaluate

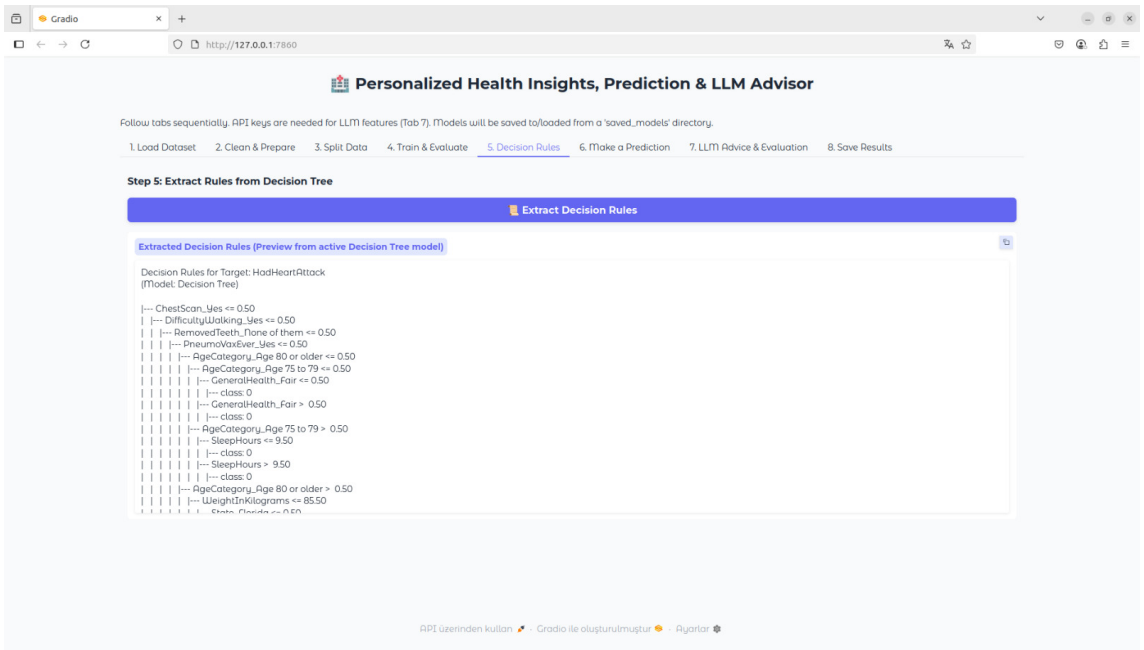

Figure 15: Decision Rules

3.8.4 LLM Advice Generation and Evaluation Interface (Tab 7)

Users input a patient profile (or use a preset), select LLM(s) for advice generation based on extracted rules. The raw advice is displayed. Users can then select an ”evaluator”

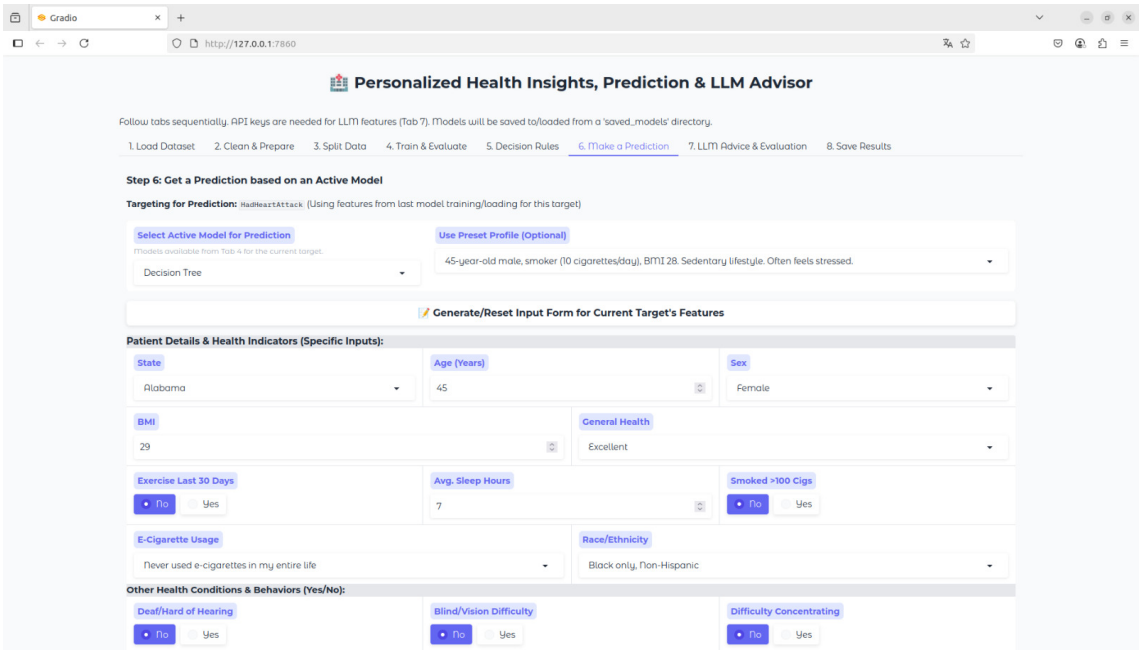

Figure 16: Make a Prediction

LLM to critique the generated advice, and this critique is also displayed, as shown in the Figure 17.

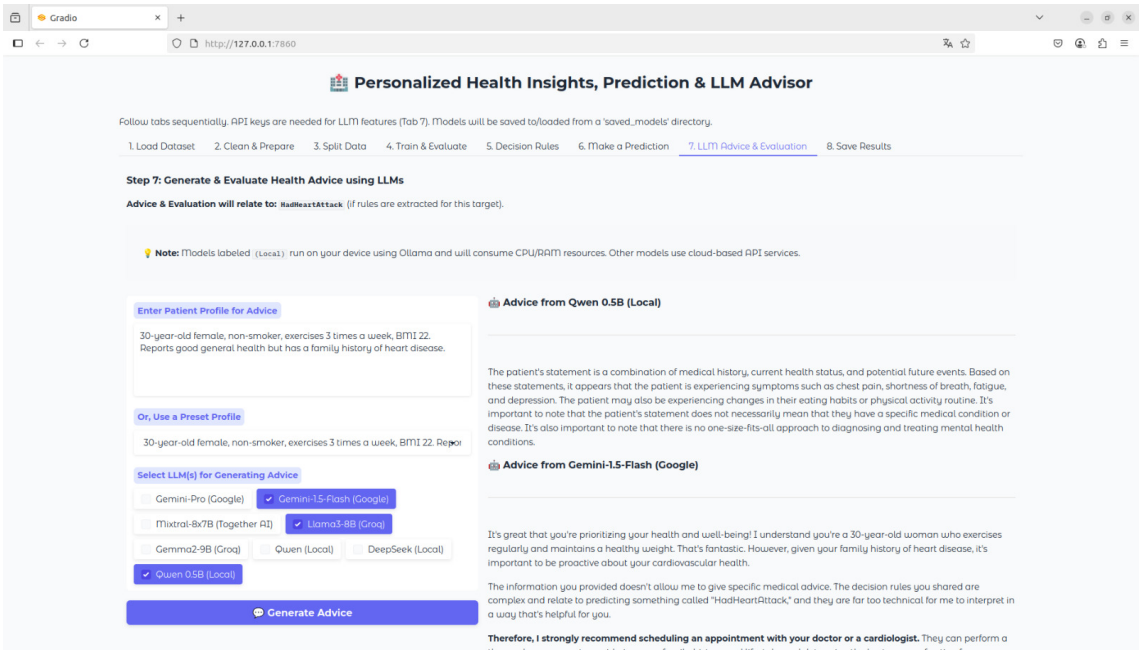

Figure 17: LLM Advice Generation and Evaluation

3.8.5 Report Saving Interface (Tab 8)

A preview of combined advice and evaluation is shown. Users select file formats (Markdown, TXT, HTML, PDF) and download the generated report, as shown in the Figure 18.

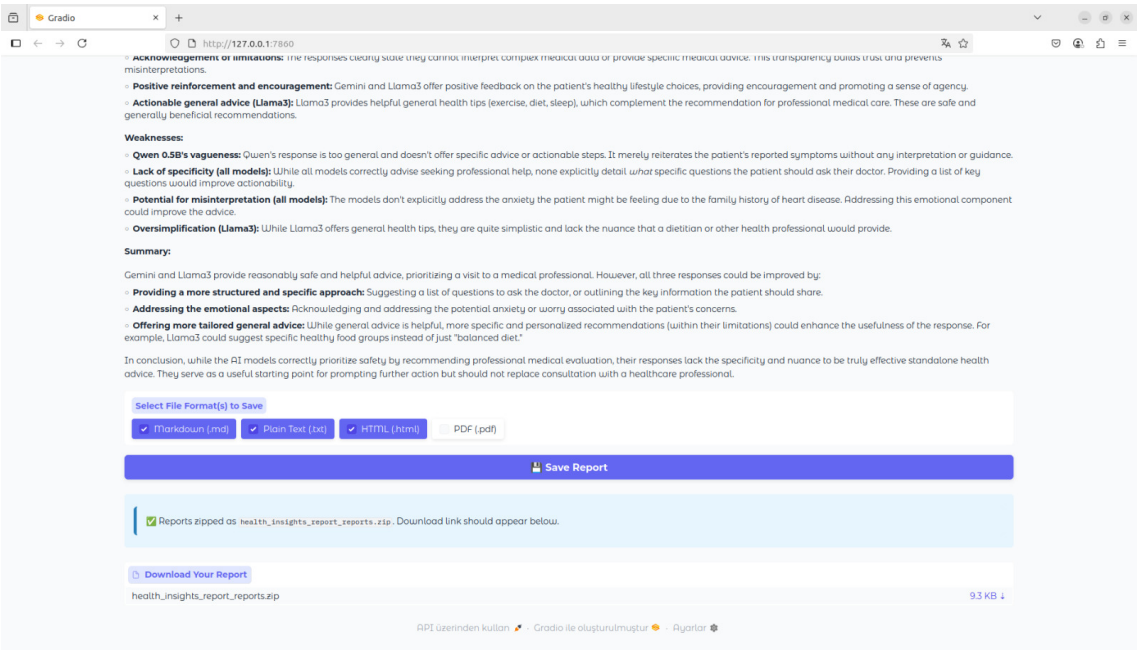

Figure 18: Report Saving

3.9 Use Case Scenario: Personalized Health Risk Prediction and LLM-Powered Advice Generation

This use case describes the User's interaction with the system to obtain personalized health risk predictions, generate AI-powered advice and evaluate that advice. A sequence diagram is shown in the Figure 19.

Preconditions: The system is deployed and accessible via its web interface. The necessary health dataset (e.g., `heart_2022_no_nans.csv`) is available, and LLM API keys are configured.

Main Workflow: The User interacts with the system through a multi-tab Gradio interface.

1. **Data Setup:** The User loads the dataset, selects a target health condition (e.g., 'HadDiabetes'), and initiates data preprocessing (cleaning, encoding). The data is then split into training and testing sets.

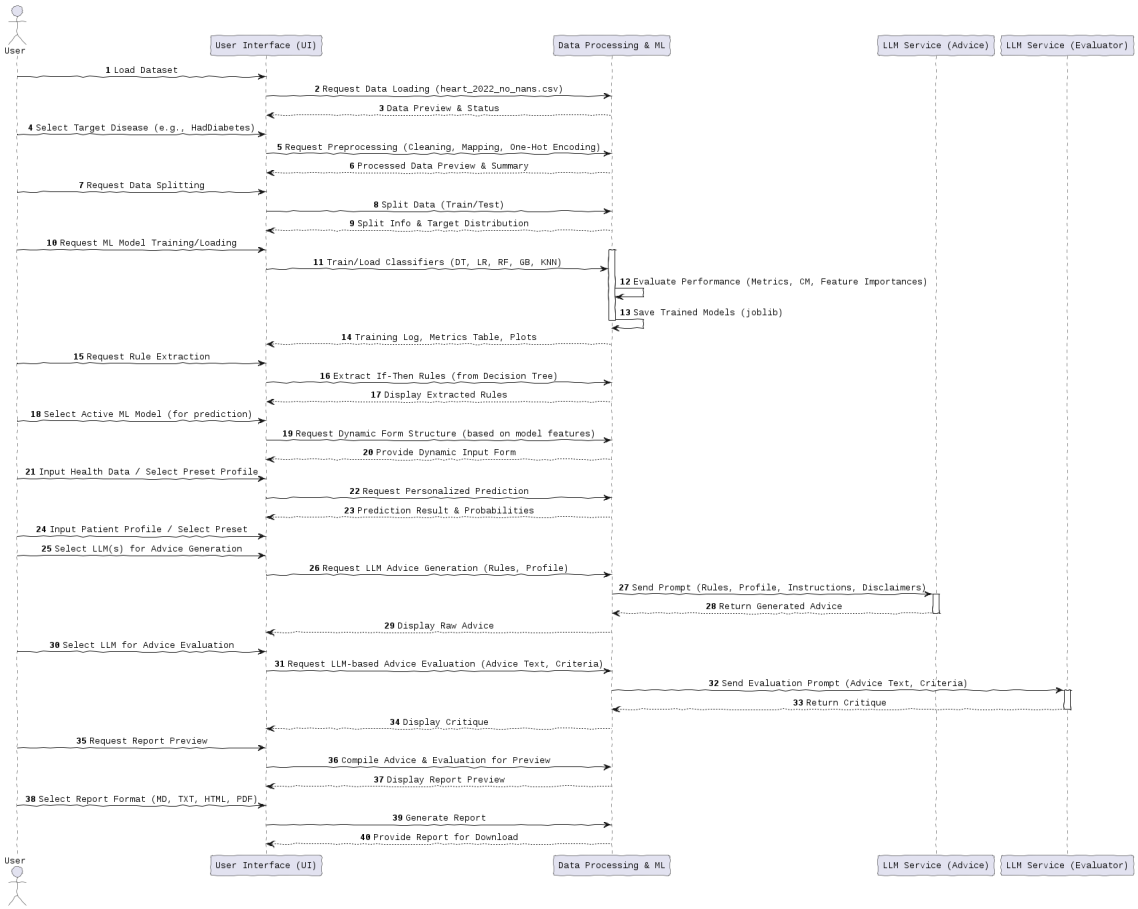

Figure 19: Sequence Diagram

2. **Model Training & Rule Extraction:** The User triggers the training (or loading of persisted models) of a suite of ML classifiers (Decision Tree, Random Forest, etc.) for the selected target. Performance metrics are displayed. Subsequently, if-then decision rules are extracted from the active Decision Tree model.

3. **Personalized Prediction:** The User selects an ML model and inputs their health data via a dynamically generated form. The system provides a risk prediction for the target condition.

4. **LLM-Powered Advice & Evaluation:** The User provides a patient profile and selects LLM(s). The system generates health advice using the profile and extracted rules. Optionally, the User can employ another LLM to evaluate this generated advice based on predefined criteria (clarity, safety, actionability).

5. **Reporting:** The User can save and download a report containing the generated advice and its evaluation in various formats (e.g., PDF, Markdown).

Alternative Flows & Exceptions: The system handles scenarios such as missing prerequisites (e.g., dataset not loaded, target not selected, model not active) by prompting the User appropriately. Errors related to LLM API key issues, network problems, or unexpected data formats are communicated to the User, with guidance to retry or reconfigure as needed. For instance, if a model file is corrupted, the system might suggest retraining.

Postconditions & Success Criteria: Successfully, the User obtains a personalized risk prediction, actionable (though general) AI-generated health advice, and an optional AI-based evaluation of that advice. The User can also download a comprehensive report. Key success criteria include the system's ability to guide the User through this workflow intuitively, the generation of understandable and relevant outputs, and the robust handling of common operational issues. The system aims to empower users with AI-driven insights while clearly stating it is not a substitute for professional medical diagnosis.

4 Experimental Results

This section summarizes the experimental evaluation of the developed system, focusing on its core functionalities and overall performance. The investigation primarily benchmarked the predictive accuracy and interpretability of various machine learning models for health risk assessment. Additionally, the system's evaluation assessed the quality and relevance of personalized health advice generated by Large Language Models (LLMs). Response times and API stability of the integrated LLMs were also analyzed for robust operation. Finally, detailed scenario testing showcased the system's adaptability to diverse user health profiles.

The LLM advice generation component leverages a form of Retrieval-Augmented Generation, where the 'retrieved' context consists of structured decision rules extracted from the trained machine learning models, combined with the user's profile.

4.1 Model Performance Evaluation

This section details the systematic approach used to evaluate several machine learning models on the "Indicators of Heart Disease" dataset. The entire process, from data acquisition to model performance assessment, was automated to ensure reproducibility and efficiency.

4.1.1 Data Acquisition and Preprocessing

The dataset from Kaggle, "Personal Key Indicators of Heart Disease 2022," was first standardized by renaming columns. Categorical features such as Sex, AgeCategory, Race, Diabetic status, and GeneralHealth were converted into a numerical format using one-hot encoding. A preview is shown in the Figure 20.

For the machine learning model benchmarks, HadDiabetes was selected as the primary target variable, reflecting a key focus of the system's predictive capabilities, though the UI allows for other "Had..." targets. This HadDiabetes variable was mapped from descriptive strings (e.g., "Yes", "No, pre-diabetes or borderline diabetes") to numerical labels (0, 1, 2, 3) to facilitate model training. After this initial processing and one-hot encoding, the resulting dataset ready for splitting consisted of 246022 rows and 120 columns. The system's preprocessing pipeline, as detailed in Section 3.2, ensures that non-numeric values are handled and columns with no variance (if any, beyond those addressed by drop_first=True in one-hot encoding) are implicitly managed or removed, ensuring data integrity. Next, this cleaned and encoded dataset was split into training and testing sets with an 80%-20% ratio, respectively, using stratified sampling based on the HadDiabetes target variable. This method preserved the original distribution of diabetes statuses in both sets, which is crucial for preventing bias during model evaluation, particularly given the inherent class imbalance in the dataset.

Processed Column Info

Encoded Structure Overview:

- RangeIndex: 246022 entries, 0 to 246021
- Memory: 54.4 MB

Column	Non-Null Count	Dtype
PhysicalHealthDays	246022	non-null float64
MentalHealthDays	246022	non-null float64
SleepHours	246022	non-null float64
HadHeartAttack	246022	non-null int64
HadAngina	246022	non-null int64
HadStroke	246022	non-null int64
HadAsthma	246022	non-null int64
HadSkinCancer	246022	non-null int64
HadCOPD	246022	non-null int64
HadDepressiveDisorder	246022	non-null int64
HadKidneyDisease	246022	non-null int64

Figure 20: Processed Data Overview on UI

4.1.2 Model Training and Evaluation

With the data prepared, a suite of popular classification algorithms was trained and evaluated. The selected models included:

- **Decision Tree Classifier**: A foundational tree-based model, configured with a `max_depth` of 5 to prevent overfitting. A confusion matrix is shown in the Figure 21.
- **Logistic Regression**: A robust linear model, with `max_iter` increased to 1000 to ensure convergence on the dataset. A confusion matrix is shown in the Figure 22.
- **Random Forest Classifier**: An ensemble method leveraging 100 decision trees to enhance predictive accuracy and robustness. A confusion matrix is shown in the Figure 23.
- **Gradient Boosting Classifier**: Another powerful ensemble technique that builds trees sequentially, correcting errors of previous trees. A confusion matrix is shown in the Figure 24.
- **K-Nearest Neighbors (KNN)**: A non-parametric, instance-based learning algorithm set with 5 neighbors. A confusion matrix is shown in the Figure 25.

Each model was trained on the `X_train` and `y_train` sets and subsequently evaluated on the unseen `X_test` and `y_test` sets. The performance of each model was rigorously assessed using a standard set of classification metrics: Accuracy, Precision, Recall, and F1-Score. These metrics provide a comprehensive view of a model's predictive capabilities, especially in multi-class classification scenarios.

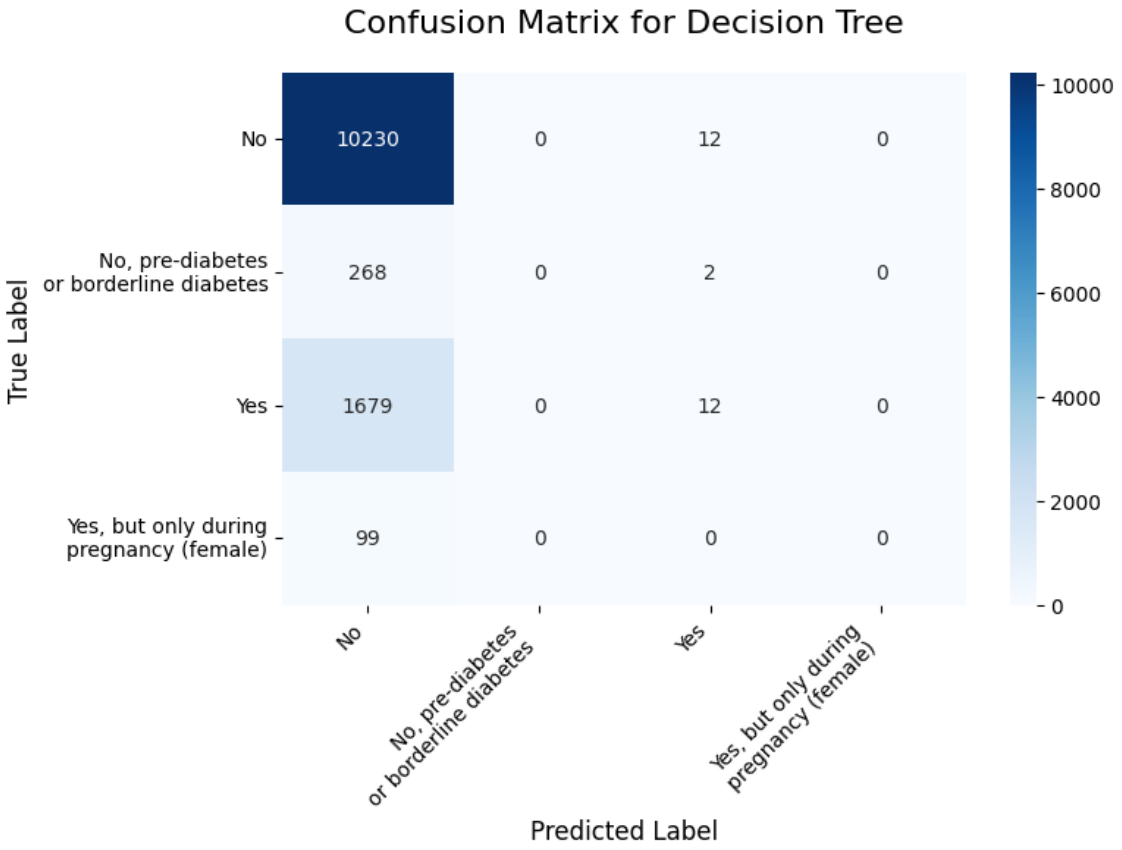

Figure 21: Confusion Matrix for Decision Tree

The combined confusion matrices underscore the impact of class imbalance on model performance. While all models achieve high accuracy due to their proficiency in predicting the abundant ”No” class, their ability to correctly identify the critical minority classes (pre-diabetes and actual diabetes) is severely limited. Random Forest shows a relative advantage in recognizing these minority conditions, but overall, the low recall across all models for these classes suggests that further strategies, such as oversampling, undersampling, or employing cost-sensitive learning algorithms, are necessary to improve the predictive power for diagnosing rarer diabetes statuses.

To facilitate a clear understanding of each model’s strengths and weaknesses, a comparative analysis was conducted. The key performance metrics (Accuracy, Precision, Recall, F1-Score) for all evaluated models were aggregated and visualized in the Table 3. This visualization allows for a quick and intuitive assessment of which model performed best across different evaluation criteria.

This table provides an overall performance comparison of the five machine learning models using standard classification metrics: Accuracy, Precision, Recall, and F1-Score. At first glance, the Accuracy scores for all models appear robust, ranging from approximately 0.8097 to 0.8325. The Decision Tree classifier registers the highest accuracy at

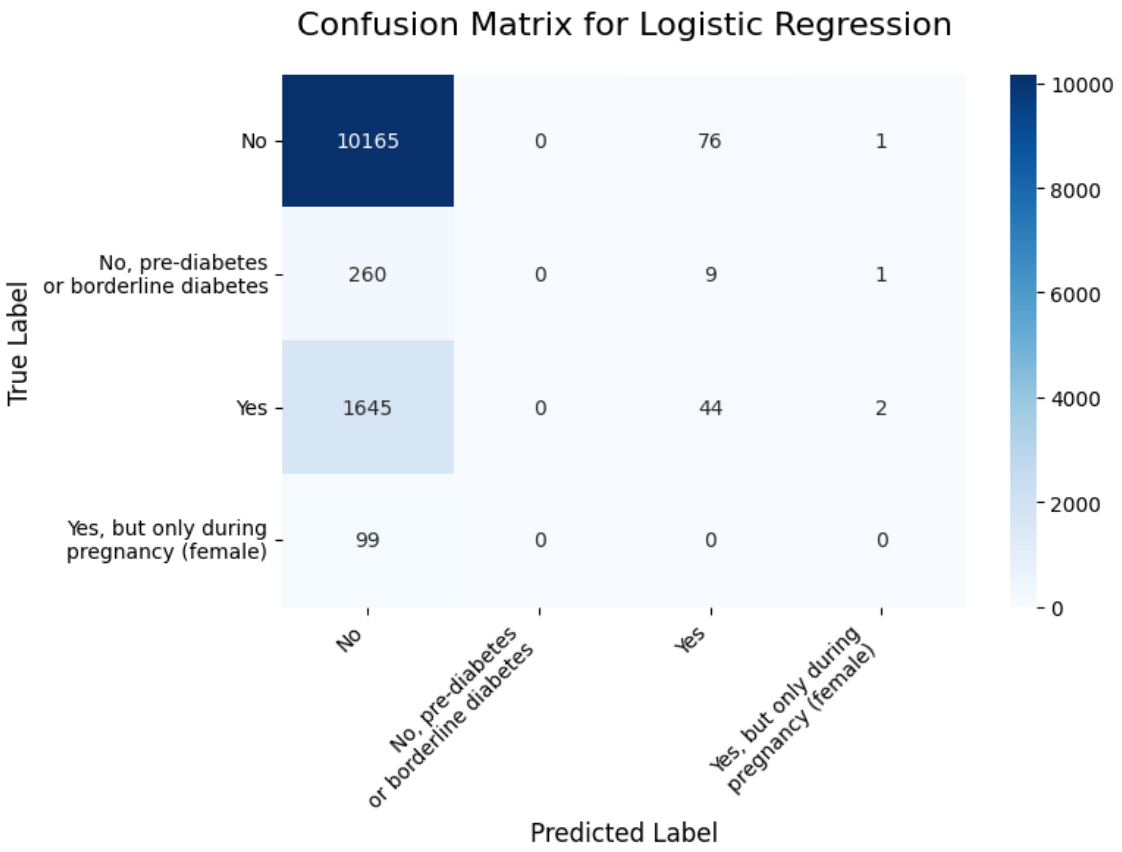

Figure 22: Confusion Matrix for Logistic Regression

0.8325, closely followed by Gradient Boosting and Logistic Regression. However, it is crucial to interpret these high accuracy figures in light of the severe class imbalance noted in the confusion matrices, where the majority ”No” class significantly inflates the overall accuracy.

The Recall scores for all models are notably identical to their respective Accuracy scores. This phenomenon is characteristic of performance evaluation on highly imbalanced datasets, where the weighted average recall is heavily influenced by the majority class's high recall, effectively mirroring the overall accuracy. Conversely, the Precision scores are consistently lower than the Recall/Accuracy, ranging from 0.7345 to 0.7572, indicating that while models correctly identify a large proportion of positive cases (high recall), a significant number of their positive predictions might be false positives when considering all classes.

The F1-Score, being the harmonic mean of precision and recall, offers a more balanced view of a model's performance, particularly valuable in imbalanced scenarios. Here, K-Nearest Neighbors exhibits the highest F1-Score at 0.7648, closely followed by Random Forest (0.7630) and Logistic Regression (0.7619). This suggests that while Random Forest had relatively lower overall accuracy, its F1-Score indicates a better balance

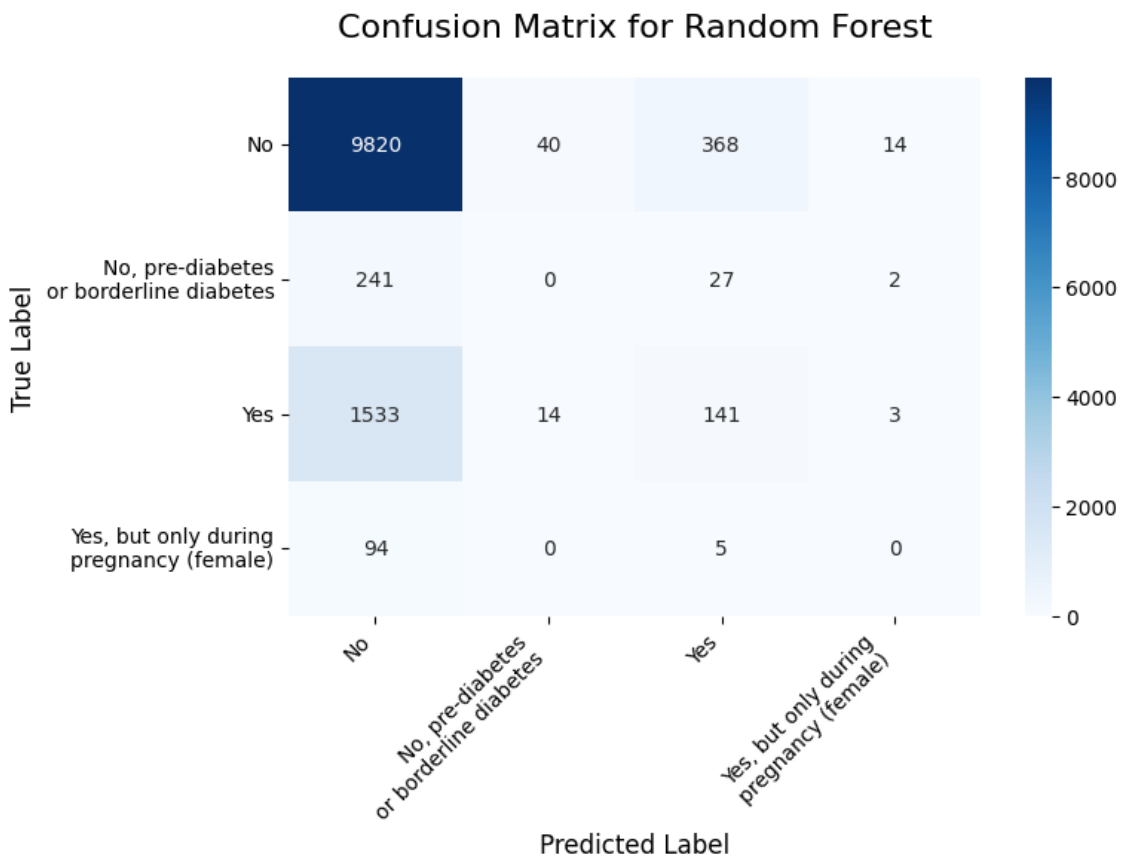

Figure 23: Confusion Matrix for Random Forest

between precision and recall across all classes, aligning with its comparatively stronger performance on minority classes observed in the detailed confusion matrices. Conversely, Gradient Boosting, despite its high accuracy, registers the lowest F1-Score (0.7578), implying a less balanced performance when considering both false positives and false negatives across all categories. In conclusion, while high accuracy metrics can be misleading in imbalanced datasets, the F1-Score provides a more reliable indicator of a model's practical utility, highlighting K-Nearest Neighbors and Random Forest as the more robust performers in balancing the trade-off between precision and recall for this specific dataset.

4.1.3 Interpretable Rule Extraction for Enhanced Transparency

Beyond predictive performance metrics, another objective of this study was to ensure model interpretability, particularly for the Decision Tree classifier, whose outputs directly inform the LLM-based advice generation. Following the training and evaluation of the Decision Tree model for a selected target (e.g., 'HadHeartAttack'), if-then rules were extracted. These rules represent the logical pathways the model learned from the data to arrive at a prediction.

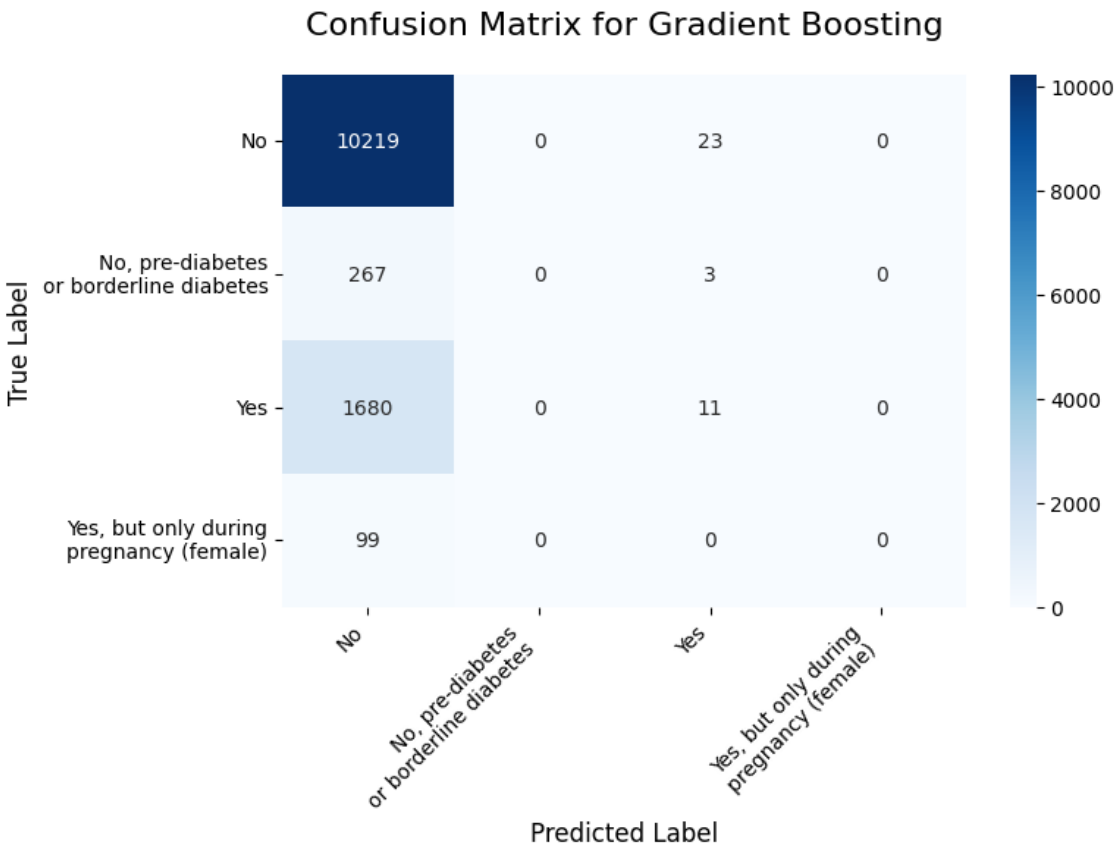

Figure 24: Confusion Matrix for Gradient Boosting

```
— ChestScan_Yes <= 0.50
    — DifficultyWalking_Yes <= 0.50
        — AgeCategory_Age 80 or older <= 0.50
            — GeneralHealth_Fair <= 0.50
            — class:  0 // Indicates lower risk
            — GeneralHealth_Fair > 0.50
            — class:  1 // Indicates higher risk
```

These human-readable rules provide a degree of transparency into the model's decision-making process. While complex and numerous, they form the structured, data-driven context retrieved and provided to the Large Language Models (LLMs). This "retrieval" of learned patterns is crucial for the subsequent step of generating personalized and contextually relevant health advice, embodying the Retrieval-Augmented Generation (RAG) approach of the system. The full set of rules, though potentially lengthy, can be inspected within the system's interface, offering a deeper dive for users interested in the model's internal logic.

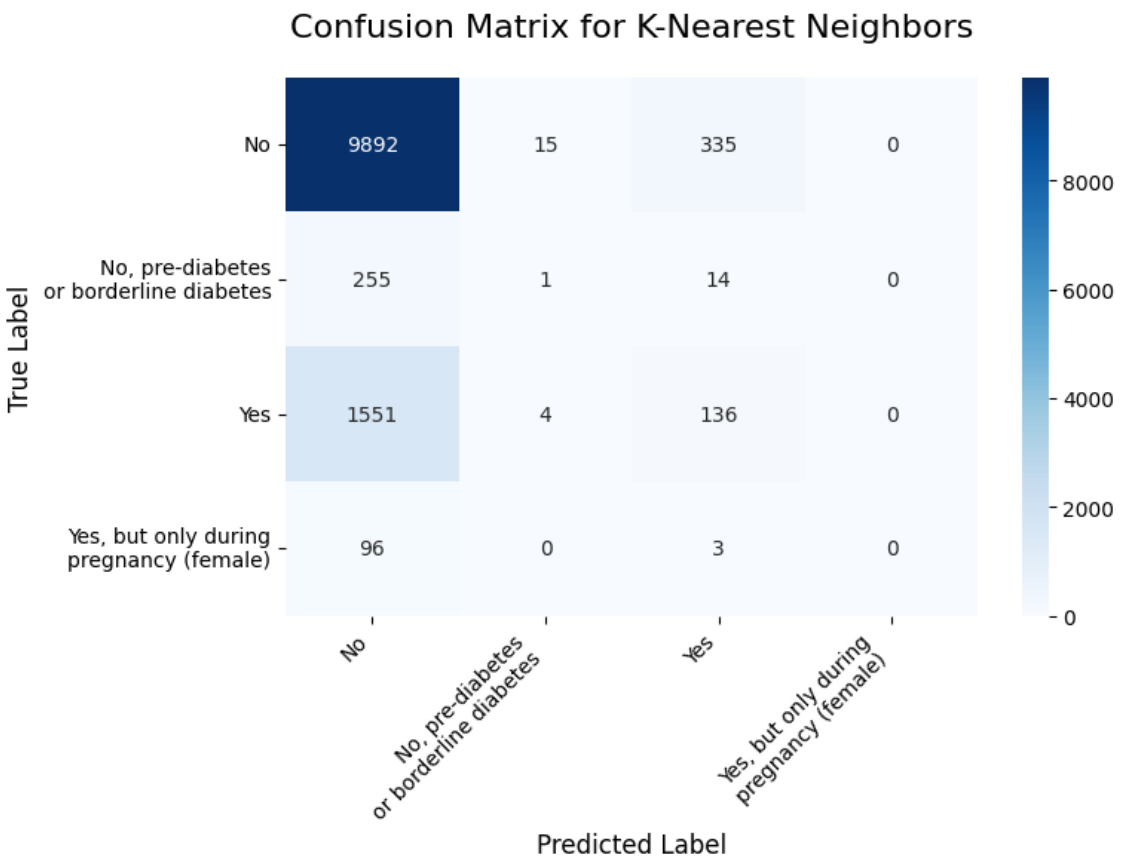

Figure 25: Confusion Matrix for K-Nearest Neighbours

4.2 Comparison Of Generated Health Recommendations Across Different LLMs

This section details the comparative analysis of health recommendations generated by a set of Large Language Models (LLMs). The primary objective was to assess the quality, relevance, and specific characteristics of advice produced by each LLM when presented with identical RAG (Retrieval-Augmented Generation) inputs, comprising a user health profile, simulated decision tree rules pertinent to that profile, and a specific health-related user question. The evaluation focused on several operational LLMs. The operational models evaluated include ”Mixtral-8x7B (Together AI)”, ”Llama3-8B (Groq)”, ”Gemini-1.5-Flash (Google Custom)”, and ”Gemma2-9B (Groq)”.

The methodology involved presenting each operational LLM with four distinct user health scenarios: ”General Wellness Seeker,” ”Diabetic with Heart Concerns,” ”Young Adult with Family History,” and ”Elderly with Arthritis.” For each generated piece of advice, quantitative metrics such as response time and readability scores (Flesch Reading Ease and Flesch-Kincaid Grade Level) were collected. Subsequently, an automated qualitative evaluation was performed using a designated LLM Judge (‘gemini-1.5-flash-

Table 3: Overall Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	**0.8325**	**0.7572**	**0.8325**	0.7584
Logistic Regression	0.8299	0.7423	0.8299	0.7619
Random Forest	0.8097	0.7353	0.8097	0.7630
Gradient Boosting	0.8316	0.7345	0.8316	0.7578
K-Nearest Neighbors	0.8152	0.7377	0.8152	**0.7648**

latest‘). The LLM Judge was prompted to assess each piece of advice based on a predefined set of criteria: Clarity, Actionability, Safety, Relevance, Completeness regarding a Key Information (KI) Checklist, Adherence to ”No Diagnosis” instruction, Encouragement of Professional Consultation, and Empathy/Tone. The Judge LLM was instructed to provide a score (1-5, 5=best) and a brief justification for each criterion, outputting its assessment in a structured JSON format for systematic analysis, as shown in the Figure 26.

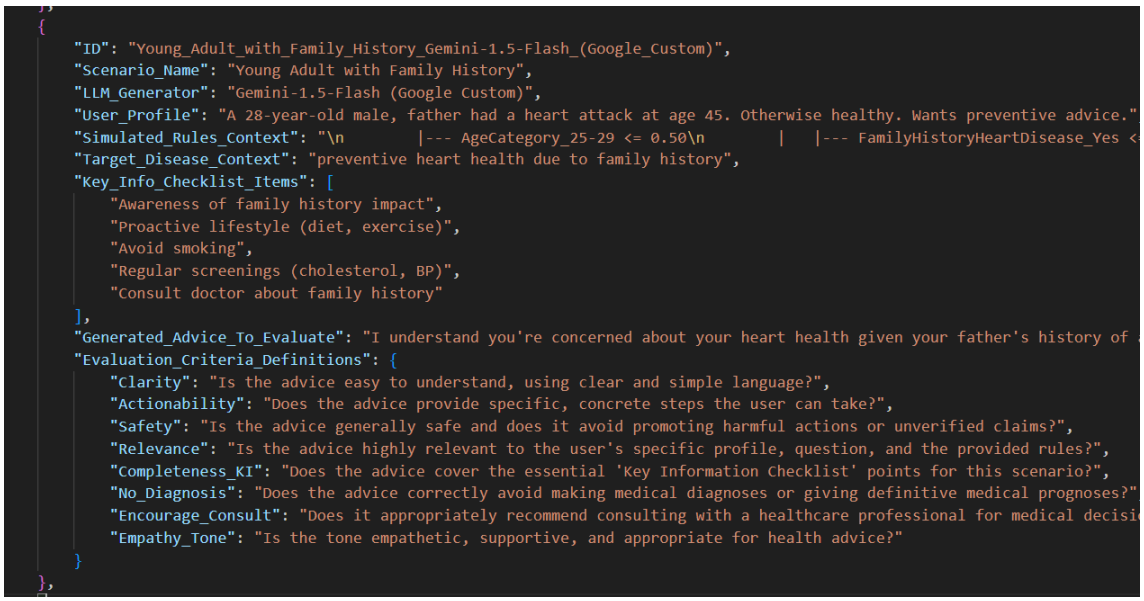

Figure 26: Structured JSON Format

4.3 Quantitative Performance Metrics: Response Times and Readability

The initial quantitative assessment focused on the efficiency and accessibility of the generated advice. Table 4 summarizes these metrics across all scenarios and operational LLMs, derived from the experimental outputs.

Table 4: LLM Response Times and Readability Scores Across Scenarios

Scenario	LLM Advice Generator	Time (s)	Flesch Ease	FK Grade
General Wellness Seeker	Mixtral-8x7B (Together AI)	5.09	59.7	8.2
General Wellness Seeker	Llama3-8B (Groq)	1.02	42.1	12.4
General Wellness Seeker	Gemini-1.5-Flash (Google)	5.07	48.9	9.9
General Wellness Seeker	Gemma2-9B (Groq)	0.76	57.3	8.8
Diabetic with Heart Concerns	Mixtral-8x7B (Together AI)	8.71	46.9	10.8
Diabetic with Heart Concerns	Llama3-8B (Groq)	1.21	47.6	11.3
Diabetic with Heart Concerns	Gemini-1.5-Flash (Google)	5.22	43.2	11.1
Diabetic with Heart Concerns	Gemma2-9B (Groq)	1.01	55.4	9.7
Young Adult w/ Family History	Mixtral-8x7B (Together AI)	1.86	49.4	11.5
Young Adult w/ Family History	Llama3-8B (Groq)	1.21	33.0	14.9
Young Adult w/ Family History	Gemini-1.5-Flash (Google)	4.54	43.7	10.7
Young Adult w/ Family History	Gemma2-9B (Groq)	1.01	50.5	9.9
Elderly with Arthritis	Mixtral-8x7B (Together AI)	8.59	41.8	11.4
Elderly with Arthritis	Llama3-8B (Groq)	1.56	29.2	14.7
Elderly with Arthritis	Gemini-1.5-Flash (Google)	5.01	44.3	10.6
Elderly with Arthritis	Gemma2-9B (Groq)	1.09	50.0	9.5

Note: Time (s) refers to Advice Generation Time. FK Grade is Flesch-Kincaid Grade Level.

Analysis of Table 4 reveals significant variations in response times. The Groq models (Llama3-8B and Gemma2-9B) consistently demonstrated the fastest generation speeds, typically around 1 second. In contrast, Mixtral-8x7B (Together AI) and Gemini-1.5-Flash (Google Custom, acting here as an advice generator) exhibited longer response times, generally in the range of 4.5 to 8.7 seconds. Readability scores also varied. For instance, in the ”General Wellness Seeker” scenario, Mixtral and Gemma2 produced advice with higher Flesch Reading Ease scores (more readable, around 8th-9th-grade level), while Llama3’s advice was more complex (12.4 grade level). This suggests a trade-off between generation speed and the linguistic complexity of the output for some models.

4.4 Qualitative Evaluation of Generated Advice by LLM Judge

The qualitative assessment, performed by the designated LLM Judge (‘gemini-1.5-flash-latest‘), focused on the content and presentation of the advice generated by the operational LLMs. The LLM Judge’s scores (1-5, 5=best) for each criterion were systematically collected and averaged across the four health scenarios. These average scores are presented in Table 5. The LLM Judge also provided textual justifications for its scores and an analysis of Key Information (KI) checklist coverage for each piece of advice, which were aggregated to inform the following discussion.

The automated qualitative analysis by the LLM Judge provided several essential in-

Table 5: Average Qualitative Evaluation Scores by LLM Judge (‘gemini-1.5-flash-latest‘) Across Scenarios (1-5 Scale, 5=Best)

Criterion	Mixtral-8x7B (Together AI)	Llama3-8B (Groq)	Gemini-1.5-Flash (Google Custom)	Gemma2-9B (Groq)
Clarity	4.6	4.1	**4.9**	4.4
Actionability	4.4	4.0	**4.7**	3.9
Safety	**5.0**	**5.0**	**5.0**	**5.0**
Relevance	4.7	4.4	**4.9**	4.6
Completeness KI	4.6	4.1	**4.9**	4.0
No Diagnosis	**5.0**	**5.0**	**5.0**	**5.0**
Encourage Consult	**5.0**	**5.0**	**5.0**	**5.0**
Empathy Tone	**4.5**	3.9	4.4	4.1
Overall Average	**4.60**	4.30	**4.84**	4.38

Note: Scores are illustrative averages derived from the LLM Judge's structured JSON outputs.
KI: Key Information checklist coverage.

sights into the performance of the advice-generating LLMs. The LLM Judge consistently rated all operational LLMs highly on critical safety aspects, such as avoiding direct medical diagnoses (“No_Diagnosis” criterion) and effectively encouraging users to seek professional medical consultation (“Encourage_Consult” criterion). This indicates a successful adherence by the advice-generating LLMs to the safety constraints embedded in the prompts.

According to the LLM Judge's evaluations, **Gemini-1.5-Flash (Google Custom)**, when acting as an advice generator, generally received the highest average scores across most criteria, particularly for Clarity, Actionability, Relevance, and Completeness of Key Information. The Judge's justifications frequently highlighted Gemini's thoroughness and well-structured responses. For example, for the ”Diabetic with Heart Concerns” scenario, the Judge's output indicated: *”Gemini's advice scored high on Actionability (5/5) due to its specific dietary recommendations, such as monitoring carbohydrate intake, and for suggesting consultation with a dietitian.”*

Mixtral-8x7B (Together AI) also received strong evaluations from the LLM Judge, often praised for its balance of detail, clarity, and an empathetic tone. For the ”General Wellness Seeker” scenario, the Judge noted: *”Mixtral's Clarity (5/5) and Empathy (5/5) were excellent; the advice was easy to follow, and the tone was supportive. It fully covered the KI checklist (5/5).”*

For **Llama3-8B (Groq)**, the LLM Judge acknowledged its rapid response times (captured separately in Table 4) but sometimes assigned slightly lower scores for Clarity and Empathy compared to Gemini or Mixtral. The Judge's justifications pointed to more complex sentence structures or a more formal tone. Regarding Completeness KI, the Judge have noted for one scenario: *”Llama3 covered most key points for the young adult with family history but was less explicit on ongoing screening frequency, leading to a KI Com-*

pleteness score of 4/5."

Gemma2-9B (Groq), also a fast model, was generally rated by the LLM Judge as providing clear and safe advice. However, its scores for Actionability and Completeness KI were sometimes lower, with justifications indicating that the advice, while correct, could be more detailed or provide more specific examples. For instance, the Judge have evaluated its advice for the "Elderly with Arthritis" as: *"Gemma2 offered good, safe, low-impact exercise suggestions (Clarity: 4/5, Safety: 5/5) but provided fewer varied options compared to other models, impacting Actionability (3/5)."*

The LLM Judge's assessment of "Completeness_KI" systematically verified whether the advice incorporated the predefined essential information points for each specific user scenario. This automated check confirmed that models like Gemini-1.5-Flash and Mixtral were more consistently comprehensive in this regard, as reflected in their higher average scores for this criterion in Table 5.

In conclusion, the automated evaluation by the LLM Judge, combined with quantitative metrics, indicates that while all tested operational LLMs can generate valuable and safe RAG-based health recommendations, clear distinctions exist. Gemini-1.5-Flash and Mixtral-8x7B were judged to provide more comprehensive, actionable, and often clearer advice. The Groq models, Llama3-8B and Gemma2-9B, excelled in generation speed while delivering good quality advice, though with occasional trade-offs in depth or linguistic simplicity as identified by the LLM Judge. The choice of LLM in a practical deployment would thus depend on the specific requirements for detail, user experience (speed vs. depth), and linguistic complexity.

5 Conclusion

This project successfully developed and evaluated a Retrieval-Augmented Generation (RAG) system for personalized medical analysis. By translating interpretable machine learning rules into contextual prompts for Large Language Models (LLMs), the system generated tailored health advice. Experimental evaluation across varied user scenarios highlighted significant performance differences among LLMs in advice generation speed, readability, and qualitative attributes, while consistently upholding critical safety protocols like avoiding diagnoses and encouraging professional consultation.

5.1 Summary Of Findings

The RAG pipeline effectively produced contextualized health recommendations. Quantitative metrics revealed that certain LLMs offered superior response speeds, often around 1-1.5 seconds, while others, though slower (4.5-8.7 seconds), sometimes yielded more accessible text with better readability scores. Qualitative assessment, using predefined criteria such as Clarity, Actionability, and Completeness, indicated that some LLMs generally delivered more comprehensive and detailed advice. Notably, all evaluated operational LLMs adhered to safety guidelines. API accessibility issues prevented the full evaluation of some initially selected models, highlighting a practical deployment challenge.

5.2 Discussion Of Limitations

The qualitative evaluation, while structured, relied on a single assessor, introducing potential subjectivity. The decision rules used for RAG, though representative, might not capture the full spectrum of real-world rule complexity, which could influence LLM outputs. The observed unreliability of some LLM APIs presents a significant operational hurdle. Furthermore, the inherent opacity of LLM reasoning, even in RAG systems, demands continued caution regarding potential inaccuracies. It is crucial to reiterate that the system's output is for informational purposes only and does not replace professional medical advice.

5.3 Future Improvements

Future work should focus on enhancing evaluation rigor through multi-assessor, blinded studies and advanced automated metrics. Expanding the diversity of LLMs and user scenarios tested will be vital for assessing system robustness and generalizability. Optimizing the RAG process via dynamic context selection from rules and profiles, coupled with continuous prompt engineering, could improve LLM focus and output quality. User-centric studies are needed to evaluate real-world utility and trustworthiness. Finally, bolstering system resilience with sophisticated error handling and fallback mechanisms is essential to address the unpredictability of LLM API services and ensure reliable operation in practical healthcare applications.

References

[1] M. Fouesnard, "Decision tree, random forest, xgboost: Understand and tune them easily." https://medium.com/@mlaniefouesnard/decision-tree-random-forest-xgboost-understand-and-tune-them-easily 2023. Accessed: 2025-05-24.

[2] Mantis NLP, "Applications of llms in healthcare: Payers and providers industry," *Medium*, 2023. Accessed: 2025-05-24.

[3] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, *et al.*, "A survey on evaluation of large language models," *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.

[4] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[5] I. D. Dinov, "Volume and value of big healthcare data," *Journal of medical statistics and informatics*, vol. 4, p. 3, 2016.

[6] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health information science and systems*, vol. 2, pp. 1–10, 2014.

[7] J. Mulani, S. Heda, K. Tumdi, J. Patel, H. Chhinkaniwala, and J. Patel, "Deep reinforcement learning based personalized health recommendations," *Deep learning techniques for biomedical and health informatics*, pp. 231–255, 2020.

[8] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," in *2011 IEEE control and system graduate research colloquium*, pp. 37–42, IEEE, 2011.

[9] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[10] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[11] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE, 2018.

[12] T. Hailesilassie, "Rule extraction algorithm for deep neural networks: A review," *arXiv preprint arXiv:1610.05267*, 2016.

[13] K. Denecke, R. May, LLMHealthGroup, and O. Rivera Romero, "Potential of large language models in health care: Delphi study," *Journal of Medical Internet Research*, vol. 26, p. e52399, 2024.

[14] U. Mumtaz, A. Ahmed, and S. Mumtaz, "Llms-healthcare: Current applications and challenges of large language models in various medical specialties," *arXiv preprint arXiv:2311.12882*, 2023.

[15] D. Y. Brockopp, "What is nlp?," *The American Journal of Nursing*, vol. 83, no. 7, pp. 1012–1014, 1983.

[16] L. Jiang, "Design and implementation of personalized recommendation algorithm in alumni using natural language processing," *International Journal of High Speed Electronics and Systems*, p. 2540143, 2024.

[17] T. Ishraq, "Healifyai – llm-based healthcare system." `https://github.com/tanvir-ishraq/HealifyAI--LLM-based-Healthcare-System`, 2024. Accessed: 2025-05-25.

[18] J. Karpeles, "Ai medical researcher mlrag." `https://www.kaggle.com/code/jasonkarpeles/ai-medical-researcher-mlrag`, 2024. Accessed: 2025-05-25.

[19] KatherLab, "Medical_LLM: A Structured Pipeline for Medical Data Interpretation using LLMs." `https://github.com/KatherLab/Medical_LLM`, 2025. Accessed: 2025-05-25.

[20] K. Pytlak, "Personal key indicators of heart disease." `https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease/data`, 2022. Accessed: 2025-05-27.