A Project Report

on

Health Assistant: Your Personal Medicine Recommendation Assistant

carried out as part of the Deep Learning Lab ***project***

Submitted

by

Yuvam Kothari
<Registration No. 229309251>

Data Science and Engineering

Under the Guidance of
Dr. Neha V Sharma

MANIPAL UNIVERSITY
JAIPUR

INSPIRED BY LIFE

Abstract

The growing dependency on computer-based technology has led to the storage of vast amounts of electronic data in the healthcare industry. Health professionals and doctors face increasing challenges in analysing symptoms accurately and diagnosing illnesses at an early stage. Machine Learning (ML) technologies have proven to be valuable tools in addressing these challenges, enabling efficient and accurate healthcare solutions. This project encompasses three major components: disease prediction, drug recommendation, and Retrieval-Augmented Generation (RAG) for medical text analysis and summarization.

The disease prediction system leverages ML algorithms to analyse symptoms provided by users, predicting potential illnesses by comparing the input with a pre-processed dataset. This assists healthcare professionals in making informed decisions and offers users a preliminary understanding of their health condition. For drug recommendations, the project employs sentiment analysis and Natural Language Processing (NLP) techniques to analyse user reviews and ratings, incorporating neural network to recommend personalized medications. This approach supports clinicians in prescribing suitable medicines and enhances patient care by minimizing medical anomalies.

The third component utilizes RAG systems to process and analyse large volumes of medical text. A novel approach is proposed to address inefficiencies in traditional document chunking methods by leveraging sectional summarization, ensuring the interconnected nature of medical information is preserved. This system retrieves and summarizes relevant content from medical documents, improving accessibility and aiding in learning and decision-making processes.

By integrating ML, NLP, and RAG systems, this project demonstrates a comprehensive framework for addressing key challenges in healthcare, offering innovative solutions for disease diagnosis, personalized medication, and efficient medical text analysis. The results hold promise for advancing healthcare technologies and support future research in the field.

LIST OF TABLES

LIST OF FIGURES

Contents

Chapter 1

Introduction

1.1 Overview, Motivation, Applications, & Advantages

The **Health Assistant** is a groundbreaking AI-powered system designed to streamline healthcare processes and improve the accessibility of medical advice. By integrating advanced conversational AI and document retrieval technologies, it allows users to either input their symptoms or upload medical documents, such as prescriptions, test results, or discharge summaries, to receive personalized medicine recommendations. The system leverages state-of-the-art tools, including Google's Generative AI (Gemini-PRO) and FAISS, to deliver highly accurate and context-aware responses.

Motivation
Healthcare accessibility has been a persistent challenge across the globe, particularly in underserved or resource-scarce regions. Many individuals face significant barriers, including:

- **Lack of Expertise:** Rural areas often lack sufficient medical professionals, leading to delayed or inadequate diagnoses.
- **Overburdened Systems:** Urban healthcare facilities may be overwhelmed, causing delays in patient consultations and increasing the risk of misdiagnoses.
- **Affordability Concerns:** Many people cannot afford regular medical consultations or expensive diagnostic services.
- **Time Constraints:** Busy schedules often prevent individuals from seeking timely medical advice.

The Health Assistant aims to address these issues by providing an easily accessible, cost-effective, and efficient tool for healthcare guidance. By democratizing medical advice and making it available anytime and anywhere, this system ensures that critical healthcare services are no longer limited by geographic or financial constraints.

Applications
The Health Assistant is highly versatile, offering a wide range of practical applications across healthcare scenarios:

1. **Symptom Analysis and Drug Recommendations:** Users can input their symptoms, and

the system analyzes them to suggest possible medications and treatments. This feature empowers patients with preliminary insights, guiding them toward appropriate care.

2. **Rapid Understanding of Medical Reports:** By uploading medical documents, such as test results or prescriptions, users can receive summaries and relevant insights that help them make informed decisions.
3. **Pharmaceutical Guidance:** Pharmacists can use the system to verify drug interactions or recommend alternatives based on patient-reported symptoms.

Advantages

The Health Assistant offers several distinct advantages that set it apart from traditional healthcare methods:

1. **Real-Time Analysis:** Unlike conventional consultations, where patients may need to wait for doctors to analyze symptoms or reports, this system provides instant feedback and recommendations, enabling faster decision-making.
2. **Minimization of Human Error:** By leveraging AI, the platform ensures consistent and accurate analysis of medical information, reducing the chances of oversight or errors that can occur in manual processes.
3. **Wide Range of Queries Supported:** From common ailments like colds and fevers to more complex issues involving laboratory reports, the system caters to a diverse array of medical queries.
4. **Improved Patient Autonomy:** Patients are empowered to take proactive steps toward their health without needing to rely solely on doctors or specialists.
5. **Scalability:** The system is highly adaptable, making it suitable for deployment in hospitals, pharmacies, and telemedicine services, as well as for individual users at home.

By combining cutting-edge technology with practical healthcare applications, the Health Assistant seeks to redefine the way medical advice is delivered and consumed, offering a future where healthcare is more accessible, affordable, and efficient for all.

1.2 Problem Statement

Healthcare decisions often require a doctor's expertise, but accessibility, affordability, and lack of real-time consultation hinder timely intervention. The problem lies in the absence of an AI-driven assistant that integrates patient input with relevant medical databases for efficient recommendations.

1.3 Objectives

1. To Create a User-Friendly Chatbot Leveraging Google Generative AI

The goal is to develop an intuitive, responsive chatbot using Google's Gemini-PRO model, capable of answering medical queries and analyzing uploaded medical PDFs, such as test reports, to provide insights and guidance. It will offer clear, accurate responses while ensuring medical data privacy and security.

2. To Retrieve Relevant Content from Medical PDFs Using FAISS
The system will extract and index content from medical PDFs using FAISS for efficient retrieval. When a user asks a question, it performs a similarity search to find the most relevant text from the uploaded documents, ensuring responses are contextually accurate.
3. To Provide Tailored Medicine Recommendations
The chatbot will analyze user-reported symptoms and available medical data to generate personalized medicine recommendations, including potential treatments and therapies, guiding users toward informed decisions and next steps for consultation.
4. To Ensure Medical Data Privacy and Security
The system will prioritize data privacy and security, adhering to best practices to ensure that sensitive medical information is handled safely, with responses generated based on factual, non-diagnostic data.
5. To Integrate Real-Time Contextual Analysis
By using the RAG framework, the chatbot will dynamically retrieve and generate responses based on the most relevant medical data available, ensuring real-time contextual accuracy and user-friendly explanations of complex medical terms.
6. To Continuously Improve Based on User Feedback
The system will refine its performance over time by analyzing user feedback, enhancing the accuracy and relevance of its responses and recommendations, ensuring continual improvement in user experience and decision-making.

1.4 Scope of the Project

This project focuses on building an AI-powered health assistant. It spans **document parsing**, **RAG implementation**, and **natural language understanding**. Future work may include multilingual support, integration with patient health records, and regulatory compliance features.

Chapter 2

Background Detail

This chapter presents a literature review of the key components of the project: disease prediction, drug recommendation, and Retrieval-Augmented Generation (RAG) for medical text analysis and summarization. The review is structured in a tabular format, providing a clear and concise comparison of existing studies, methodologies, and technologies related to each domain. This approach highlights the foundational research, identifies gaps, and establishes the relevance of the techniques employed in this project.

Table 2.1 Summary of Key Research Papers in Personalized Medicine Recommendation Systems:

S.No.	Research Paper	Major Findings	Advantages	Shortcomings	Datasets/Tools Used
1	Multimodal Reinforcement Learning for Embedding Networks and Medication Recommendation in Parkinson's Disease	Integrated multimodal embeddings for optimizing medication strategies in Parkinson's disease.	Personalized treatment; integration of diverse data types.	Limited scalability; requires extensive data preprocessing.	PPMI database, autoencoders, reinforcement learning.
2	PreGenerator: TCM Prescription Recommendation Model Based on Retrieval and Generation Method	Hybrid retrieval-generation model for generating prescriptions adhering to TCM principles.	Innovative use of templates for compatibility rules.	Limited to TCM datasets; challenges with scalability.	TCM prescription datasets, Seq2Seq, attention models.
3	A Literature Review on Medicine Recommender Systems	Ontology-based DRS leveraging structured medical knowledge.	High interpretability.	Limited applicability to cases with defined ontologies.	RDF graphs, SPARQL queries, Protégé.
4	Aspect-Based Sentiment Analysis of Drug Reviews	Aspect-based sentiment analysis of drug reviews for predicting	Captures nuanced user feedback effectively.	Data sparsity; domain dependency of models.	Drugs.com, Druglib.com, n-grams, logistic

		drug efficacy and side effects.			regression.
5	COGNET-AMO: Personalized Prescription Recommendation	Integrated medical order data with attention mechanisms for personalized recommendations.	High accuracy for comorbid conditions.	Computationally intensive.	MIMIC-III dataset, cross-attention mechanisms.
6	Chinese Medicine Prescription Recommendation Using GANs	Generated innovative, clinically valid TCM prescriptions using SeqGAN and CGAN architectures.	High BLEU scores for quality.	Limited adaptability to broader medical datasets.	TCM datasets, GAN models.
7	An Intelligent Disease Prediction and Drug Recommendation Prototype	Hybrid DRS combining ML, sentiment analysis, and probabilistic models for medication recommendation.	High accuracy (97–99%) with classifiers.	High computational overhead; requires extensive preprocessing.	Disease-Symptom Knowledge Database, UCI Drug Review Dataset.
8	Drug Recommendation Using Hybrid Techniques in Clinical Decision-Making	Combined collaborative filtering and domain knowledge for personalized recommendations.	Improved scalability and prediction accuracy.	Requires high-quality labeled datasets.	EMR datasets, collaborative filtering algorithms.
9	GAN-Based Traditional Medicine Framework	Developed GAN-based frameworks to generate culturally valid prescriptions with real-world applications.	Enhanced adherence to cultural and medical practices.	Requires specialized datasets for broader applications.	Specialized TCM datasets, SeqGAN, CGAN.
10	SHAPE: A Sample-Adaptive Hierarchical Prediction Network for Medication Recommendation	Proposed hierarchical patient representation using intra-visit and inter-visit encoders.	Effective in handling multimorbidity; adaptive to variable visit lengths.	High computational demands.	MIMIC-III dataset, hierarchical encoders, curriculum learning.
11	KGDNet:	Combined	Superior DDI	Dependency on	MIMIC-IV,

	Knowledge Graph-Driven Medicine Recommendation System	GNNs with transformers for longitudinal EHR data analysis and DDI reduction.	control; high precision in predictions.	curated ontologies and external knowledge.	DrugBank, ICD ontology, attention mechanisms.
12	GAN-TCM: Generative Adversarial Networks for Traditional Chinese Medicine	Used GAN frameworks to generate prescriptions adhering to TCM principles.	High cultural relevance; accurate and innovative prescriptions.	Limited scalability to modern medical datasets.	Specialized TCM datasets, BLEU, Word2Vec.

Chapter 3

System Development

This chapter provides a comprehensive overview of the development process for three key components: the disease prediction prototype, the drug recommendation system based on patient reviews, and the Retrieval-Augmented Generation (RAG) system for medical text analysis and summarization. It details the collection and preprocessing of datasets, the diverse methodologies and algorithms employed, and the tools and software utilized throughout the implementation.

3.1 Disease Prediction

Disease Prediction is a Machine Learning based system which primarily works according to the symptoms given by a user. The disease is predicted using algorithms and comparison of the datasets with the symptoms provided by the user. It can help healthcare professionals: Identify illnesses early, Research signs and symptoms, and Resolve health issues quickly.

3.1.1 Dataset Gathering

Two Disease-Symptoms knowledge datasets were collected from Kaggle which offers columns containing diseases and their symptoms. Dataset One provides a comprehensive collection of disease names and their associated symptoms, encoded in a one-hot manner. Each row in the dataset represents a single instance, such as a patient or case study, while each column represents a symptom or disease. Symptoms and diseases are encoded using binary values (0 or 1), where a value of 1 indicates the presence of the symptom or disease, and 0 indicates its absence. This dataset contains 400 symptoms and 133 distinct diseases. Dataset Two also provides a comprehensive collection of disease names and their associated symptoms. Each instance contains the disease name and symptoms are written in subsequent columns. This dataset contains 133 symptoms and 41 distinct diseases. For the purpose of making a final medicine prediction, both the dataset containing the symptoms and the diseases are combined.

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Sym
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN	
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN	
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN	

Figure 3.1 Raw symptoms dataset one

	Unnamed: 0	pain chest	shortness of breath	dizziness	asthenia	fall	syncope	verti
0	0	0	1	0	0	0	1	
1	0	0	1	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	1	1	0	0	0	0	
4	0	0	1	0	0	0	0	

Figure 3.2 Raw symptoms dataset two

3.1.2 Preprocessing and Methodology

Both the datasets were clean. Before training models, text data (Symptom) is converted into numerical form using TF-IDF (Term Frequency-Inverse Document Frequency). In Natural Language Processing (NLP), understanding the relevance and significance of words within documents is crucial for many applications. For instance, symptoms like 'itching' are quite common in distinctive diseases. Such more frequently appearing symptoms in the data set would receive a lower TF-IDF score. On the other hand, less frequently appearing symptoms would receive a higher TF-IDF score, as they are of higher importance. TF-IDF assigns weights to words based on their frequency in the document (symptom description) relative to their occurrence across all documents, emphasizing distinctive terms. This transformation captures the importance of symptoms in describing diseases while reducing the influence of common but less informative terms. Following that, the data was trained by utilizing the Multinomial Probabilistic prototype, ExtraTree Classifier Model, Decision Tree Classifier Model, and Support Vector Machine Classifier Model, with various Symptoms serving as the training features and Diseases serving as the labels.

	Disease	Symptom
0	hypertensive disease	shortness_of_breath syncope sweat_sweatin...
1	diabetes	shortness_of_breath sweat_sweating_incre...
2	depression mental , depressive disorder	sleeplessness wo...
3	coronary arteriosclerosis ,coronary heart disease	pain_chest shortness_of_breath sweat_swea...
4	pneumonia	shortness_of_breath rale ...
5	failure heart congestive	shortness_of_breath rale ...
6	accident cerebrovascular	unresponsiveness ...
7	asthma	shortness_of_breath ...
8	myocardial infarction	pain_chest palpitation angina_pectoris...
9	hypercholesterolemia	shortness_of_breath syncope sweat_sweatin...

Figure 3.3 Merged symptoms dataset after preprocessing

3.2 Drug recommendation using feedback

Traditional medicine recommendation systems are designed based on rules defined by experienced physicians according to clinical guidelines. These rule-based hard coding methods can make general medical recommendations for specific diseases, but they do little to make personalized recommendations for complex patients.

With the emergence and popularization of Electronic Health Records (EHRs), the acquisition and collection of medical data have become more convenient, resulting in the accumulation of a large number of clinical data, such as vital signs, disease diagnosis prescription medicines, medical expenses, etc. At the same time, deep learning technology provides a new technical means for the mining and utilization of medical data. This enables personalized medicine recommendation, so as to make up for the defects of traditional approaches. The goal of personalized medicine recommendation is to use EHR to predict the most appropriate medicines for each patient, which can be used as a reference for doctors to prescribe. Personalized medicine recommendation model can provide better personalized treatment recommendations to doctors and patients to improve the prognosis of patients and make more effective use of medical resources.

However, the field of recommending medicines based on patient feedback remains relatively underexplored. In our study, we take a novel approach by utilizing patient reviews about the drugs they used to treat their conditions to build a recommendation model.

3.2.1 Dataset Gathering

The UCI Machine Learning Repository for Drug Review, which offers patient reviews on particular medications together with information on linked ailments and a 10-star patient rating indicating overall patient happiness, is where the dataset is acquired. Due to the fact that the two datasets in the repository (Test and Train) had the same amount of columns, they were combined for analysis and visualization. It has 213869 rows and 7 columns: ID, drug name, condition, Review, Rating, Date and Useful count. This dataset includes 916 unique Conditions (Diseases), 3671 unique Drug names, and ratings and reviews that match to the medicine names.

	uniqueID	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9	20-May-12	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8	27-Apr-10	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5	14-Dec-09	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8	3-Nov-15	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9	27-Nov-16	37

Figure 3.4 Raw drug review dataset

3.2.2 Data Preprocessing

The following procedures were used to pre-process the dataset:

1. We compared the unique number of unique IDs and the length of the train data to see if the same patient has written multiple reviews, and there weren't more than one review for one customer.

2. We checked the name of conditions and some of them contain the phrase "3 users

found this comment helpful", which seems like an error in the crawling process. Instances with such phrase were removed.

3. We checked the number of 'drugs per condition'. Considering the recommendation system, it is not feasible to recommend that when there is only one product. Therefore, we will analyze only the conditions that have at least 2 drugs per condition.

4. There are a few missing values as well in conditions. Looking at the percentage, which is less than 1, we will drop those instances.

5. In reviews, there were some words with errors like didn&# 039;t for didn't, and also html strings like \ r \ n, and the parts that express emotions in parentheses such as (very unusual for him) and (a good thing) and words in capital letters like MUCH. We will delete these parts in preprocessing as well.

6. Looking at the distribution of useful count, the difference between minimum and maximum is high. In addition, the deviation is huge. The reason for this is that the more drugs people look for, the more people read the review no matter their contents are good or bad, which makes the useful count very high. So when we create the model, we will normalize it by conditions, considering people's accessibility.

3.2.3 Model Training

While analysing the data, we saw 4-gram classifies emotions much better than other grams. Therefore, we will use 4-grams to build deep learning model. Next, we will classify ratings 1 ~ 5 as negative, and 6 ~ 10 as positive, and create a new feature 'sentiment' for it. A deep learning model was trained by using preprocessed review as feature and sentiment as label.

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 200)	1,000,200
batch_normalization (BatchNormalization)	(None, 200)	800
activation (Activation)	(None, 200)	0
dropout (Dropout)	(None, 200)	0
dense_1 (Dense)	(None, 200)	40,200
batch_normalization_1 (BatchNormalization)	(None, 200)	800
activation_1 (Activation)	(None, 200)	0
dropout_1 (Dropout)	(None, 200)	0
dense_2 (Dense)	(None, 100)	20,100
dense_3 (Dense)	(None, 1)	101

Total params: 1,062,201 (4.05 MB)
Trainable params: 1,061,401 (4.05 MB)
Non-trainable params: 800 (3.12 KB)

Figure 3.5 Neural network prototype

3.3 Enhanced Medical Text Analysis and Summarization Using Retrieval-Augmented Generation (RAG)

This project leverages a Retrieval-Augmented Generation (RAG) system to process and analyze content from medical books. By combining retrieval with generative capabilities, it aims to improve how users access and interact with extensive and detailed text resources. Also, by uploading medical documents such as health reports, test results or prescriptions, users can receive summaries and relevant insights that help them make informed decisions.

3.3.1 System Architecture

- Components:
    - Input Module: Accepts symptoms or PDFs.
    - Books Trained: Medical Book, Gray's Anatomy for Students, Harrison's Principles of Internal Medicine, Oxford Handbook of Clinical Medicine, Where There Is No Doctor, Current Medical Diagnosis & Treatment, Davidson's Principles and Practice of Medicine, Harrison's Pulmonary and Critical Care Medicine
    - Retrieval Module: FAISS-based document search.
    - Conversational Module: Generates answers using Google Generative AI.
- Diagrams:
    - Flowchart: User input → Text Extraction → Embedding → Query → Response Generation.

3.3.2 Development Environment

- Hardware: High-performance CPU/GPU system with 16GB RAM.
- Software:
    - Python libraries: PyPDF2, FAISS, LangChain, Streamlit.
    - Google API services for embedding and conversational models.

3.3.3 Methodology: Algorithm/Procedures

i. Input Parsing

The first step involves parsing and extracting text from the user-uploaded medical PDF documents. Using the PyPDF2 library, the system processes each page of the PDFs to retrieve the raw text content. This content is then pre-processed to remove unnecessary formatting, special characters, or irrelevant data, ensuring that only clean, meaningful text is passed on for further analysis. Additionally, if the user inputs symptoms directly via the chat interface, their input is pre-processed to standardize and structure it for better interpretation by the system.

ii. Text Chunking

After extracting the raw text from PDFs, it is divided into smaller, more manageable chunks using the RecursiveCharacterTextSplitter from the LangChain library. This step is crucial because it helps in handling large documents that may not be efficiently processed in one go. By breaking down the text into chunks of 10,000 characters (with an overlap of 1,000 characters), the system ensures that each chunk retains enough context to be useful for the search and retrieval process. This step also aids in making the document more digestible and improves the accuracy of context retrieval when responding to user queries.

iii. Embedding and Storage

Once the text is chunked, the next step is to convert these text chunks into vector embeddings using Google's Generative AI Embedding models. These embeddings represent the semantic meaning of the text in a numerical format, allowing for efficient similarity searches. The embeddings are then stored in a FAISS (Facebook AI Similarity Search) index, which is a highly optimized library for fast similarity search. FAISS enables quick retrieval of relevant chunks of text by calculating the similarity between the user's query and the stored embeddings. This indexing process ensures that large volumes of medical data can be efficiently queried, even if the document size grows over time.

iv. Query Handling

When a user submits a query, the system first processes the input to ensure it's structured and cleaned. It then performs a similarity search using FAISS to retrieve the most relevant content from the stored embeddings. The query is compared with the indexed vectors, and FAISS identifies the chunks of text that are contextually closest to the user's question. These retrieved chunks form the context that will be used to generate an accurate response. This allows the chatbot to answer questions grounded in specific, contextually relevant data from the uploaded medical documents.

v. Response Generation

Once relevant context has been retrieved, the system uses Google's Gemini-PRO model to generate a response. The model processes the question alongside the context to generate a clear and accurate answer. Gemini-PRO, a generative language model, is specifically fine-tuned to handle healthcare-related queries. It is capable of interpreting medical terminology and providing answers that are easy to understand, even when addressing complex medical conditions. The response is crafted to ensure it is not only accurate but also accessible to users, avoiding jargon and providing necessary explanations for technical terms when needed.

Chapter 4

Results and Discussion

In this chapter, we provide a detailed analysis of the performance of the various models and prototypes utilized in this project. We carefully evaluate their effectiveness in achieving the desired outcomes and compare their against each other. Additionally, we delve into the challenges encountered during the implementation, discussing the limitations and constraints that affected the project's scope and performance. These limitations include potential shortcomings in the dataset, the trade-offs made in model selection, and the areas where further improvements are required.

4.1 Disease prediction and drug recommendation prototype

The disease-symptom dataset was analyzed using various machine learning models, where symptoms were used as training features and diseases as target labels. The models were successfully trained and demonstrated high accuracy in their predictions. SVM classifier turned out to be the best model among them with an accuracy of 98%. The deep learning model for sentiment analysis of drug reviews was also developed successfully, achieving a training accuracy of 75% and a testing accuracy of 67%.

ML Model	**Accuracy**
Support Vector Machine Classifier	0.9818
Decision Tree Classifier	0.8487
ExtraTree Classifier	0.9652
Multinomial Probabilistic	0.9182

Table 4.1 Models vs Accuracy

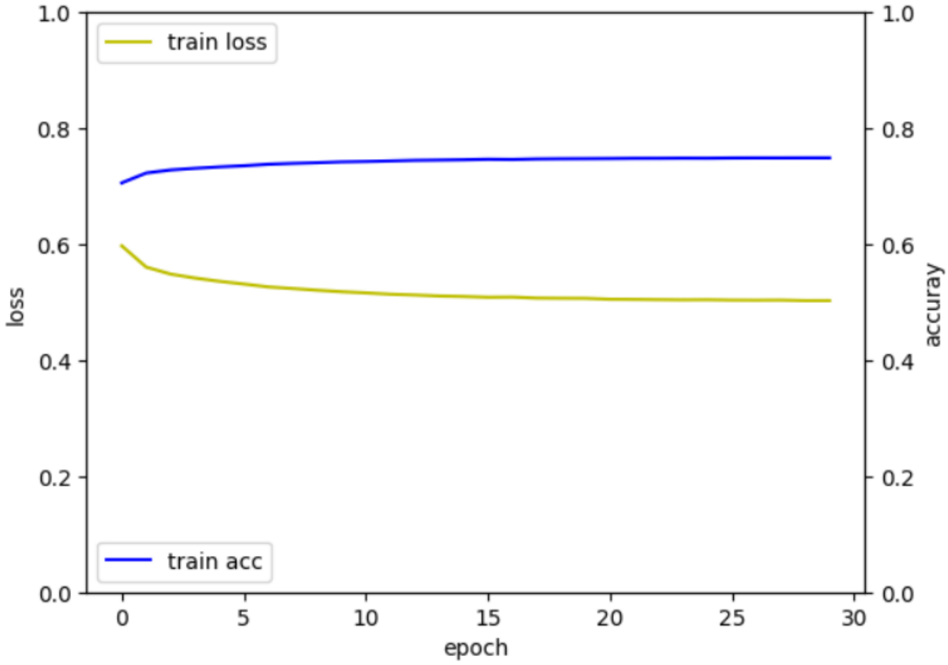

Figure 4.1 Accuracy vs epochs for deep learning model

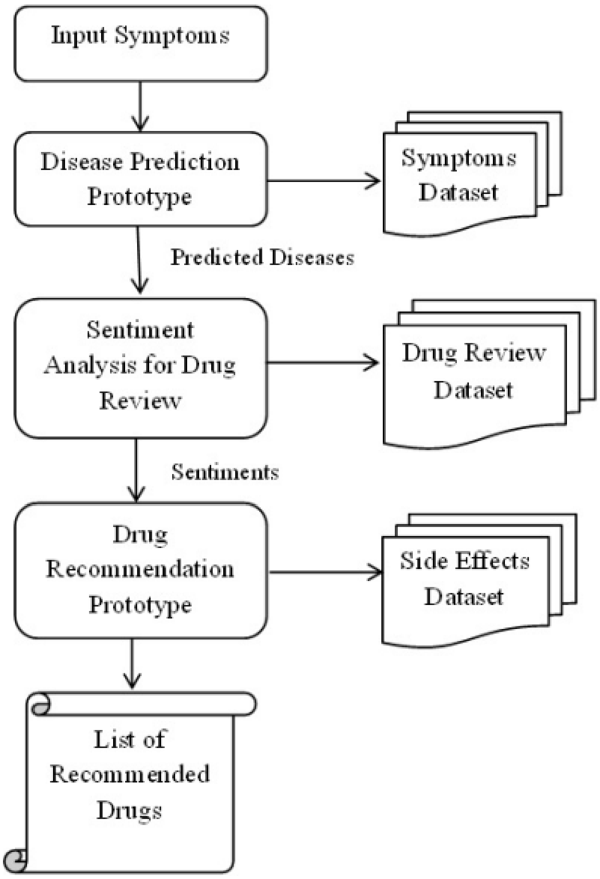

Figure 4.2 Design implementation pipeline and dataflow of the recommender prototype.

4.2 Enhanced Medical Text Analysis and Summarization Using Retrieval-Augmented Generation (RAG)

The system effectively identifies user-relevant context from uploaded documents and generates accurate, well-structured recommendations. It successfully processed eight medical books, creating a robust database of retrievable chunks, and delivered precise query results, demonstrating its capability to support learning and decision-making. User feedback highlighted significant improvements in the accessibility and usability of information. Overall, the system showcases the potential of Retrieval-Augmented Generation (RAG) models for complex text retrieval, particularly in academic and medical contexts. It is built on Streamlit for an intuitive front-end.

4.3 Limitations

i. For disease prediction prototype training data is small. To make more rows, the number of rows for each disease be multiplied. With this procedure, we have sought to produce a dataset in which all the symptoms of a specific disease are dispersed at random across various rows. This results in a dataset with several rows for a single disease and its symptoms indicated in each row differently, 0 or 1.

ii. Sentiment analysis using sentiment word dictionary has low reliability when the number of positive and negative words is small. For example, if there are 0 positive words and 1 negative word, it is classified as negative. Therefore, if the number of sentiment words is 5 or less, we could exclude the observations.

iii. To ensure the reliability of the predicted values, we normalized usefulCount and multiplied it to the predicted values. However, usefulCount may tend to be higher for older reviews as the number of cumulated site visitors increases. Therefore, we should have also considered time when normalizing usefulCount.

iv. If the emotion is positive, the reliability should be increased to the positive side, and if it is negative, the reliability should be increased toward the negative side. However, we simply multiplied the usefulCount for reliability and did not consider this part. So we should have multiplied considering the sign of usefulCount according to different kinds of emotion.

v. Performance of RAG system could degrade with extremely large documents or vague queries.

4.4 Month-wise Plan of Work

Month	Task	Status
Month 1	Literature Review & Tool Selection	Completed
Month 2	Development of Modules	Completed
Month 3	Testing and debugging	Completed
Month 4	Documentation and Finalization	Completed

Chapter 5

Conclusion

The Health Assistant represents a significant advancement in the use of AI for improving healthcare access and delivery. By building disease prediction prototype, recommending medicines using patient reviews and by integrating document analysis with conversational AI, the system can process large volumes of medical data, extract relevant insights, and provide tailored, context-aware responses to users. This helps patients and healthcare professionals make more informed decisions, while ensuring that medical knowledge is accessible and easy to understand. The use of Google's Gemini-PRO model and FAISS for context retrieval and response generation allows for efficient and accurate recommendations based on user input and uploaded medical documents. Ultimately, the Health Assistant is a step forward in leveraging AI for personalized healthcare, making it easier for users to understand their conditions and access relevant treatment information.

5.1 Future Work

The development of the Health Assistant doesn't end here. There are several key areas where this project can evolve and expand to further improve healthcare access and usability:

Integration with Patient Health Records One of the major next steps is to integrate the Health Assistant with patient health records (EHRs). By securely linking with EHR systems, the chatbot can access real-time, personalized health data, such as previous diagnoses, medications, test results, and more. This integration would enable the chatbot to provide even more personalized and accurate recommendations, taking into account the patient's entire medical history. It could also facilitate better communication between patients and healthcare providers, ensuring that patients receive guidance tailored to their unique health profile.

Support for Non-English Medical Documents Another important area for future expansion is the ability to process and understand non-English medical documents. By incorporating multilingual NLP models and translation capabilities, the Health Assistant can support a broader user base, including non-English speaking patients. This will enable the system to process medical documents in a variety of languages and provide responses in the user's preferred language. In a global healthcare context, such multilingual support will make healthcare more accessible to diverse populations, ensuring equitable access to healthcare information.

Enhanced Data Privacy and Compliance with Healthcare Regulations Ensuring data privacy and regulatory compliance is critical when dealing with sensitive healthcare information. The future development of the Health Assistant will focus on enhancing data security protocols, ensuring that patient data is stored and processed in compliance with healthcare regulations such as HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation). This will include implementing stronger encryption, secure data storage solutions, and strict access controls to ensure that sensitive information remains confidential and is only accessible to authorized individuals. Additionally, compliance with these regulations will also extend to ensuring that the AI system adheres to medical ethics and does not make diagnoses or provide unauthorized medical advice.

References

[1] Fanglin Zhu, Lizhen Cui, Yonghui Xu, Zhe Qu, and Zhiqi Shen. "A Survey of Personalized Medicine Recommendation."

[2] SUVENDU KUMAR NAYAK. "An Intelligent Disease Prediction and Drug Recommendation Prototype by Using Multiple Approaches of Machine Learning Algorithms."

[3] CHUITIAN RONG, XUEYAN LI, XUEMEI SUN, and HUABO SUN. "Chinese Medicine Prescription Recommendation Using Generative Adversarial Network."

[4] FENG GAO, NA ZHAO, YAO CHEN, JIANJUN GUO, YONGQING WANG, HONGYU YUAN, and SHAN LU. "COGNET-AMO: Personalized Prescription Recommendation using Attention over Medical Order Information."

[5] HONGBUM KIM, CHANYOUNG PARK, JONG HOON KIM, SEJEONG JANG, and HYO KYUNG LEE. "Multimodal Reinforcement Learning for Embedding Networks and Medication Recommendation in Parkinson's Disease."

[6] ZIJUAN ZHAO, XUETING REN, KAI SONG, YAN QIANG, JUANJUAN ZHAO, JUNLONG ZHANG, and PENG HAN. "PreGenerator: TCM Prescription Recommendation Model Based on Retrieval and Generation Method."

[7] Marieke M. van Buchem, Olaf M. Neve, Ilse M. J. Kant, Ewout W. Steyerberg, Hileen Boosman, and Erik F. Hensen. "Analyzing Patient Experiences Using Natural Language Processing: Development and Validation of the Artificial Intelligence Patient Reported Experience Measure (AI-PREM)."

[8] Rajat Mishra and S. Shridevi. "Knowledge Graph Driven Medicine Recommendation System Using Graph Neural Networks on Longitudinal Medical Records."

[9] Sicen Liu, Xiaolong Wang, Jingcheng Du, Yongshuai Hou, Xianbing Zhao, Hui Xu, Hui Wang, Yang Xiang, and Buzhou Tang. "SHAPE: A Sample-Adaptive

Hierarchical Prediction Network for Medication Recommendation."
[10] Felix Gräßer and Surya Kallumadi. "Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning."
[11] Benjamin Stark, Constanze Knahl, Mert Aydin, Karim Elish, Mamata Garanayak, Sangram Keshari Swain, Sandeep Kumar Panda, and Deepthi Godavarthi. "A Literature Review on Medicine Recommender Systems."