A Weekly Update on

DiabetIQ: An Intelligent Diabetes Management Application with LLM-Augmented Chatbot and ML-Based Early Risk Prediction

16 April, 2025

Group Information

Group-01 CSE299 (Section-17)

1. Saif Mohammed - 2121913042
2. Nazibul Islam Nabil - 2222456642
3. Humayra Rahman Nipa - 2121128042
4. Umme Suraia Haque Setu - 2031278642

Weekly Update Brief

- **Backend Implementation:** Implemented Backend with Flask (Python Framework) MVVM architecture.
- **ML Model Train:** Applied Machine Learning Algorithms: Logistic Regression, SVM, Decision Tree, Random Forest, KNN with Hyperparameter Tuning (Grid Search Cross Validation).
- **LLM RAG Chatbot:** Processed BADAS Guideline 2019 PDF, implemented document chunking, embeddings, and retrieval for Q&A conversation.
- **Summary Documentation:** Completed a summary documentation on *Attention in Transformers: Concepts and Code in PyTorch*

Machine Learning (Model Train)

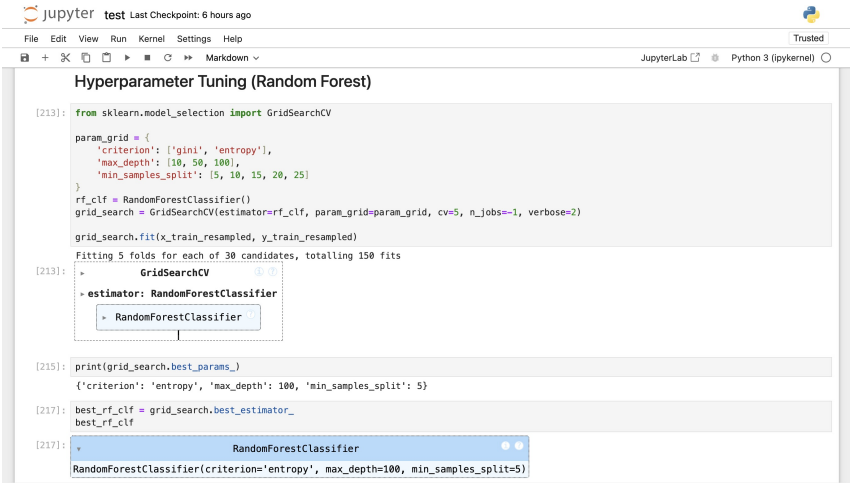


Figure: Hyperparameter Tuning (Random Forest)

Performance Metrics of ML Algorithms for Early Diabetic Prediction

Serial	Classifier	Precision	Recall	F1-Score	Accuracy	AUC
1	XGBoost	0.93	0.94	0.93	94%	0.64
2	Random Forest	0.94	0.93	0.94	93%	0.68
3	Gradient Boosting	0.94	0.92	0.93	92%	0.73
4	SVM	0.94	0.90	0.92	90%	0.78
5	Adaboost	0.94	0.88	0.91	88%	0.74
6	Naive Bayes	0.95	0.88	0.90	88%	0.81
7	Decision Tree	0.93	0.88	0.90	88%	0.71
8	Logistic Regression	0.95	0.87	0.90	87%	0.80
9	KNN	0.94	0.84	0.88	84%	0.73

Table: Sorted Performance Metrics of ML Algorithms for Early Diabetic Prediction

Fine Tune LLM RAG Q&A Chatbot

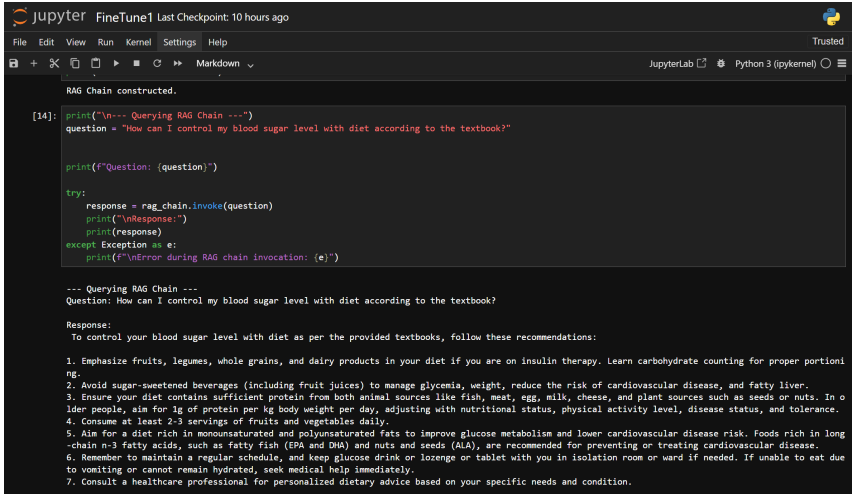


Figure: Fine Tune Chatbot

Backend Implementation: MVVM Architecture

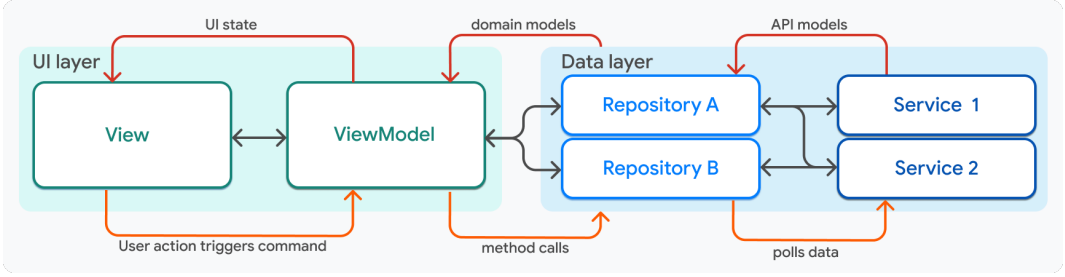
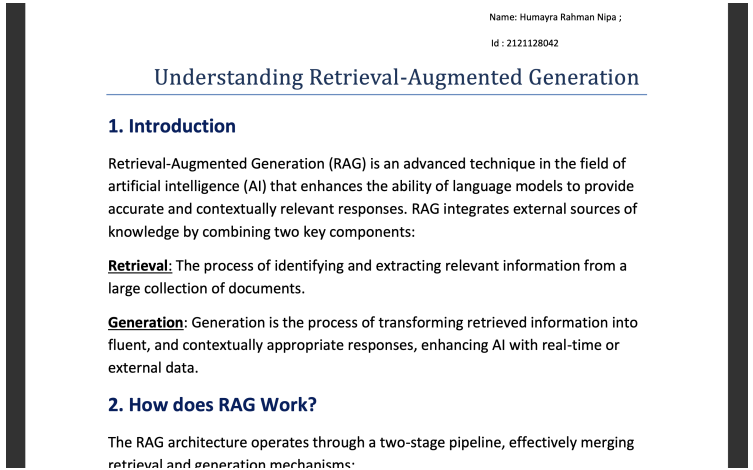


Figure: MVVM Architecture

Summary Documentation



Name: Humayra Rahman Nipa ;

Id : 2121128042

Understanding Retrieval-Augmented Generation

1. Introduction

Retrieval-Augmented Generation (RAG) is an advanced technique in the field of artificial intelligence (AI) that enhances the ability of language models to provide accurate and contextually relevant responses. RAG integrates external sources of knowledge by combining two key components:

Retrieval: The process of identifying and extracting relevant information from a large collection of documents.

Generation: Generation is the process of transforming retrieved information into fluent, and contextually appropriate responses, enhancing AI with real-time or external data.

2. How does RAG Work?

The RAG architecture operates through a two-stage pipeline, effectively merging retrieval and generation mechanisms:

Figure: Summary Documentation on RAG

Achievements

- Applied Machine Learning Algorithms for machine machine learning algorithms
- Fine-tuned basic retrieval and generation from the textbook.
- Implemented backend using MVVM Architecture (Flask)
- Summarized documentation on RAG.

Technology Stack

- **Programming Language:** Python
- **Framework:** LangChain
- **Libraries:** NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn
- **Embedding Model:** HuggingFaceEmbeddings: *intfloat/e5-small-v2*
- **Vector Database:** chromadb
- **LLM:** Ollama LLM: *mistral*
- **Document Processing:** PyPDF

Work Distribution (This Week)

- **Saif Mohammed - 2121913042**
  - ML Algorithms Application
  - ML Algorithms Evaluation
- **Nazibul Islam Nabil - 2222456642**
  - LLM Q&A Chatbot Prototype
- **Humayra Rahman Nipa - 2121128042**
  - Summary Documentation on Transformers
- **Umme Suraia Haque Setu - 2031278642**
  - Summary Documentation on Transformers

References

- A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2023.
- LangChain Engineer, "LLM Part: Learn RAG From Scratch – Python AI Tutorial," YouTube. [Online]. Available: *https://www.youtube.com/*. [Accessed: Feb. 19, 2025].
- P. Lewis, E. Denoyer, and S. Riedel, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv preprint arXiv:2005.11401*, May 2020. [Online]. Available: *https://arxiv.org/abs/2005.11401*.
- DeepLearning.AI, *Attention in Transformers: Concepts and Code in PyTorch*. [Online]. Available: `https://www.deeplearning.ai/short-courses/attention-in-transformers-concepts-and-code-in-pytorch/`. [Accessed: Mar. 11, 2025.]