Development of a Custom GenAI Solution for Conducting Health Data Analysis

Github link: https://github.com/aiml-hrushikesh/Development-of-a-Custom-GenAI-Solution-for-Conducting-Health-Data-Analysis.git

# 📖 Overview

This project leverages Generative AI to analyze health data and provide insights through an interactive interface. It integrates Google Gemini with Gradio to create a user-friendly application for querying health-related data.  
  
• Generate AI-Powered Insights: Uses Generative AI to answer health-related questions based on patient data.  
• Interactive Interface: A Gradio-based interface for seamless interaction.  
• Data Processing Features: Automatically handles missing data and retrieves patient-specific information.

# ✨ Features

🔍 Data Imputation: Automatically fills missing values in datasets using statistical methods.  
📂 Patient Data Retrieval: Fetches and processes patient-specific data from multiple datasets.  
🤖 Generative AI Integration: Uses Google Gemini to answer health-related questions.  
🌐 Interactive Interface: Provides a Gradio-based interface for easy interaction.

# ⚙️ Prerequisites

Ensure you have the following installed and configured:  
🐍 Python: Version 3.8 or higher  
🔑 Google Cloud API Key: With access to Gemini  
📦 Required Python Libraries: Listed in requirements.txt

# 🛠️ Installation

Follow these steps to set up the project:

1. Clone the Repository:  
 git clone <repository-url>  
 cd Development-of-a-Custom-GenAI-Solution-for-Conducting-Health-Data-Analysis

2. Create a Virtual Environment:  
 python -m venv venv  
 source venv/bin/activate # On Windows: venv\Scripts\activate

3. Install Dependencies:  
 pip install -r requirements.txt

4. Add API Key:  
 Set the environment variable: GOOGLE\_API\_KEY="your-google-api-key"

5. Ensure Datasets Are Present:  
 Place the following files in the project directory:  
 • Health\_data1.xlsm  
 • Health\_data2.xlsm

# 🚀 Usage

1. Run the Application:  
 python health\_data.py

2. Access the Gradio Interface:  
 Open the provided URL in your browser.

3. Interact with the Application:  
 Enter a Patient Number and ask a Health-Related Question.

## 💬 Example

Patient Number: 12345  
Question: "What is the average physical activity of this patient?"  
  
Output: Based on the patient's health records, the average physical activity is 5000 steps per day.

# 📂 File Descriptions

🧠 health\_data.py  
Handles the core functionality:  
• Loads health data Excel files using pandas  
• Performs missing value imputation  
• Retrieves patient-specific data  
• Sends data and user queries to Google Gemini  
• Provides an interactive Gradio interface

📜 requirements.txt  
Lists all the Python dependencies required for the project:  
• google-generativeai  
• pandas  
• scikit-learn  
• gradio  
• openpyxl  
• python-dotenv

🔒 .env  
Stores sensitive environment variables such as your GOOGLE\_API\_KEY

📊 Health\_data1.xlsm and Health\_data2.xlsm  
Excel files containing patient health data.

# 💡 Key Highlights

• Simplifies health data analysis with AI-powered insights  
• Easy-to-use interactive interface for healthcare professionals  
• Securely integrates with Google Gemini for Generative AI capabilities

**Demo:**  




