Health AI – Intelligent Healthcare Assistant

1.Introduction

Project Title: Health AI – Intelligent Healthcare Assistant
Team ID: **NM2025TMID01492**
Team Leader: A.Sindhu Bhairavi– bhairavi.arjunan@gmail.com
Team Members:
1. M.Mahathi shree– mahathim962@gmail.com
2. K.Subashree – subashreek20@gmail.com
3. A.Srimathi – srimathichatresh@gmail.com

2. Project Overview

Purpose: Health AI offers intelligent and easy-to-understand healthcare assistance by leveraging the IBM Granite model. It enables features such as Patient Chat, Disease Prediction, and Treatment Plans for better medical guidance. **Features:**
– Patient Chat Interface
– Disease Prediction System
– Treatment Plan Recommendations – Integration with IBM Granite Model
– Secure Deployment in Google Colab

3. Architecture

Frontend: Gradio for interactive UI
Backend: Python (Transformers and Torch Libraries)
Model: IBM Granite Model from Hugging Face **Deployment:**
Google Colab with T4 GPU

4. Setup Instructions

Prerequisites:
- Python (3.x)
- Git
- Gradio Framework
- IBM Granite Model (via Hugging Face) - Google Colab Access

Installation Steps:
1. Clone the repository (if applicable):
git clone [Repository URL] 2. Install
required libraries: pip install
transformers torch gradio -q
3. Open Google Colab, change Runtime Type to T4 GPU.

5. Folder Structure

HealthAI/ |-- colab_notebook.ipynb # Main Google Colab Notebook |-- model_integration.py # Code for model interaction |-- utils.py # Helper functions |-- requirements.txt # Required Python packages

6. Running the Application

1. Open Google Colab: https://colab.research.google.com/
2. Set Runtime to T4 GPU
3. Run the initial cell to install dependencies:
!pip install transformers torch gradio -q
4. Upload and run the project code cells sequentially.

7. API Documentation

The project runs entirely in Google Colab without separate REST APIs. Interaction occurs through the Gradio web interface.

8. Authentication

No explicit authentication mechanism. The model runs locally in Google Colab and is accessed through Gradio's interface.

9. User Interface

- Landing Page with Gradio Interface
- Chat Window for Patient Interaction
- Prediction Output Section
- Treatment Plan Recommendations

10. Testing

- Manual Testing through the Gradio Interface -
Sample Inputs:
– Symptoms for disease prediction
– Query for treatment plans

11. Known Issues

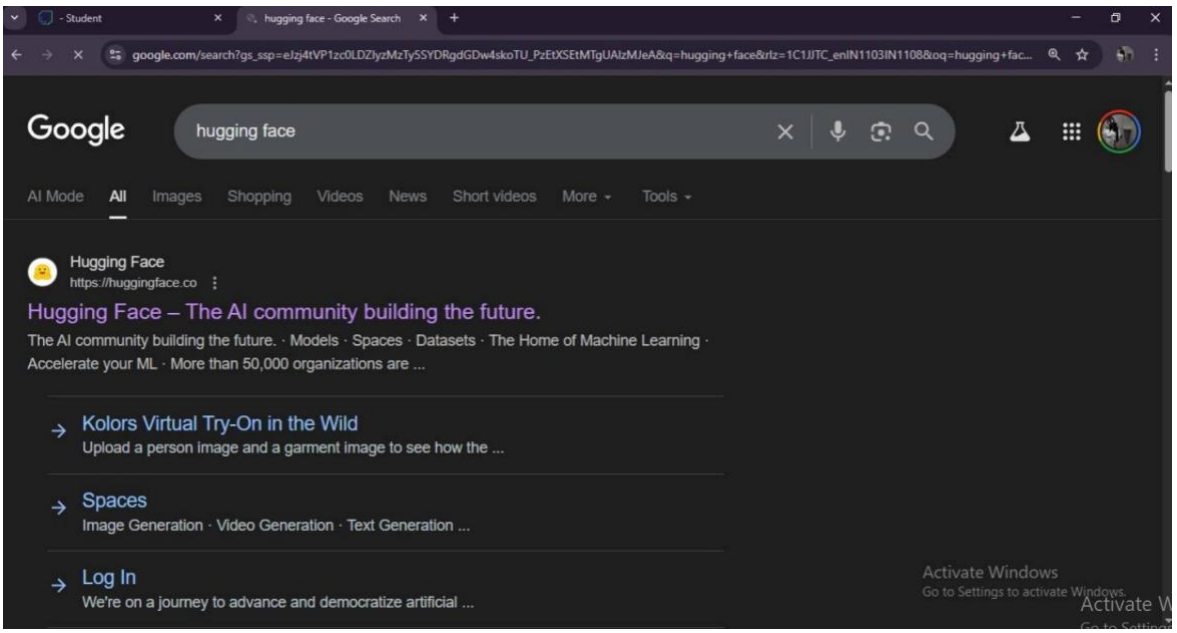
- Performance depends on Google Colab session time limits
- Occasional loading delays when model is first initialized

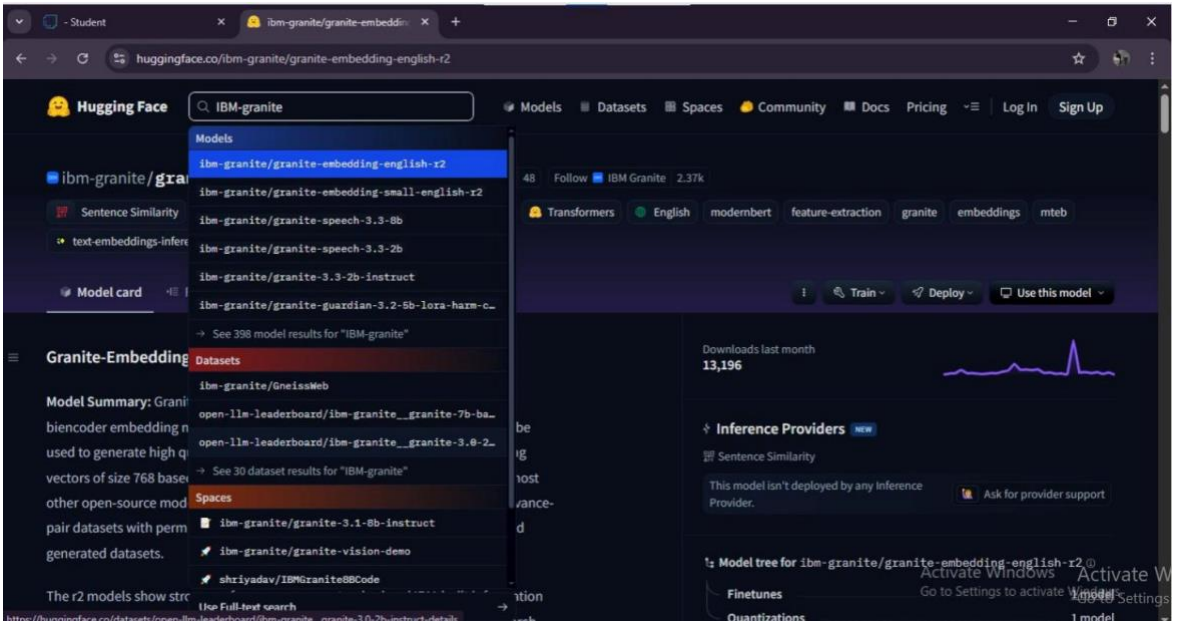
12. Future Enhancements

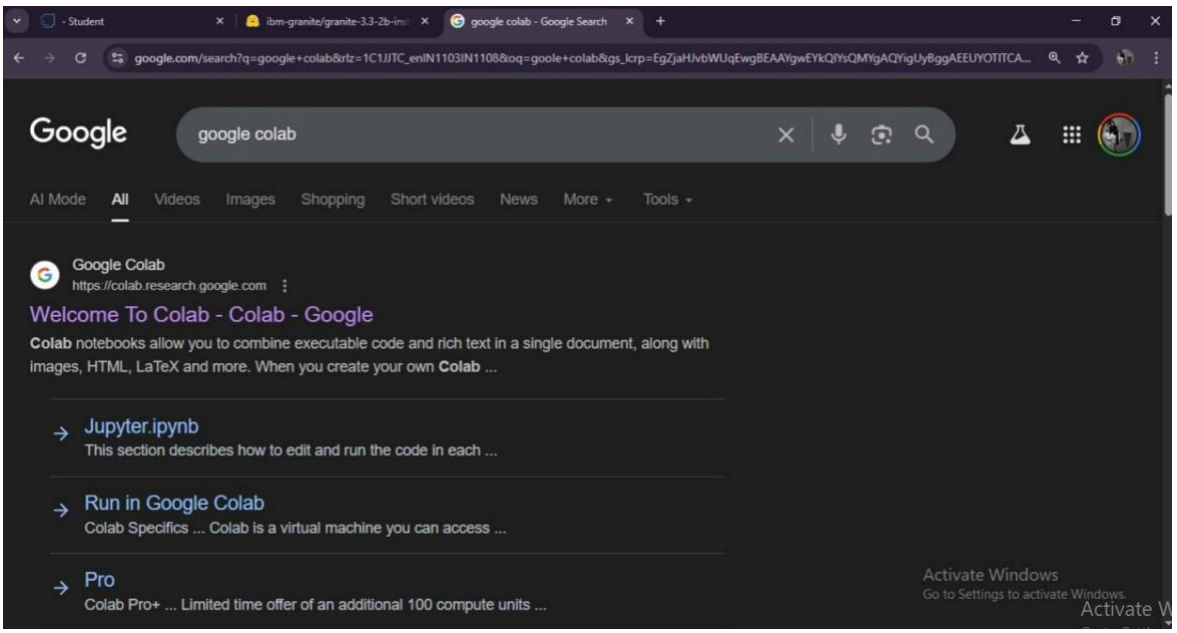
- Add Authentication System for User Tracking
- Build a Dedicated Web Interface
- Expand Database for Disease & Treatment Knowledge
- Deploy on Cloud Infrastructure for Scalability

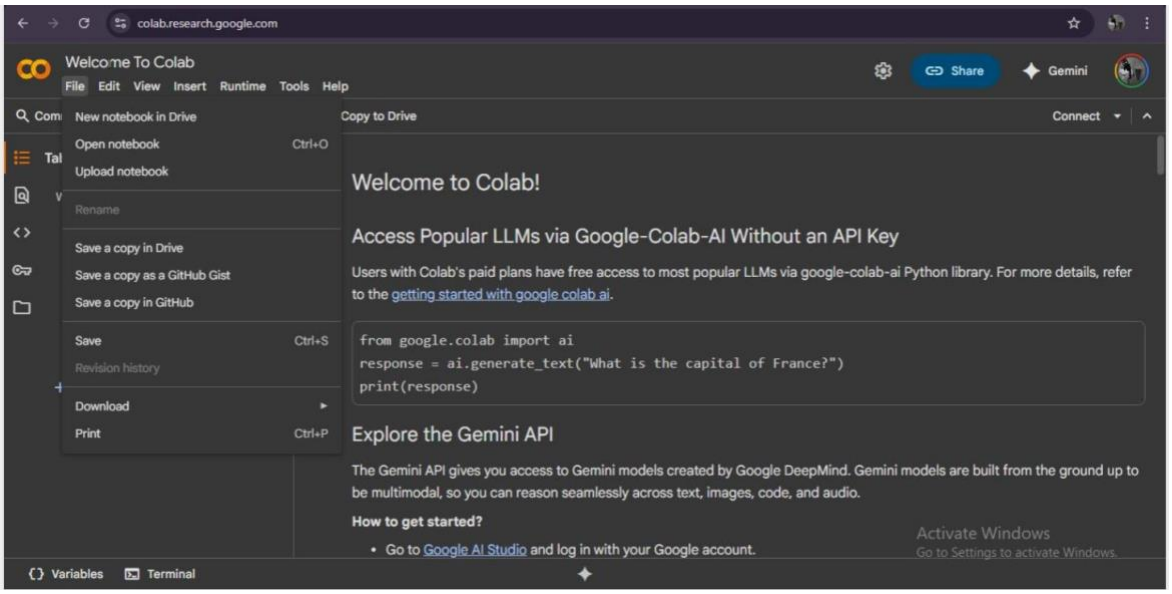
13. Screenshots or Demo

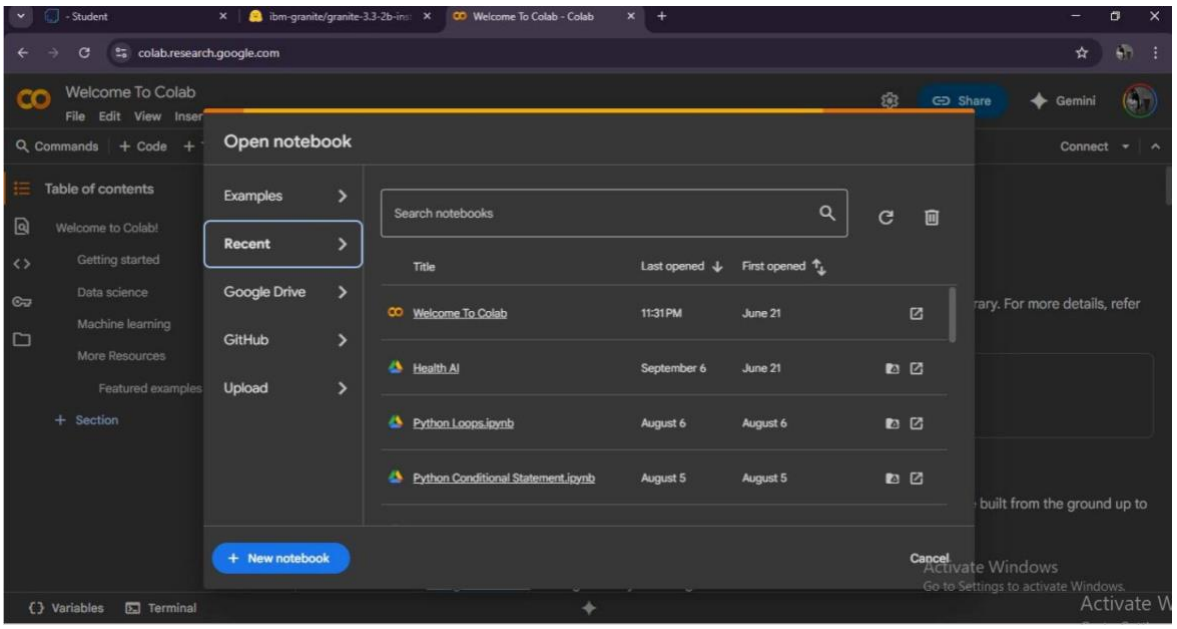
[Provide screenshots of the Gradio interface and sample outputs]

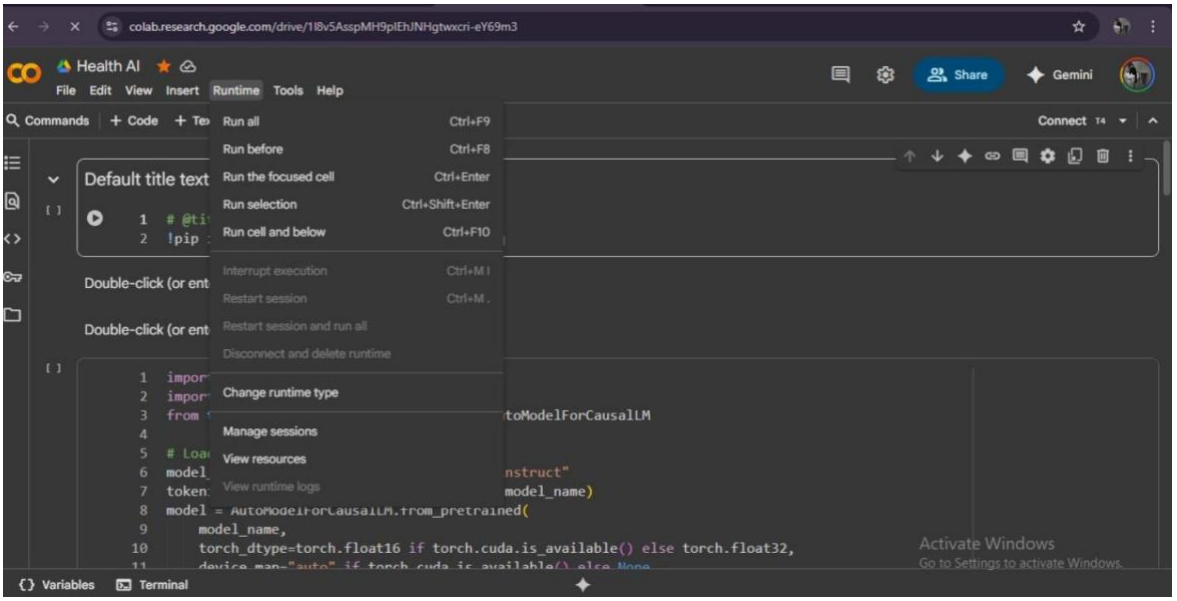

- Student
hugging face - Google Search
google.com/search?gs_ssp=eJzj4tVP1zc0LDZlyzMzTySSYDRgdGDw4skoTU_PzEtXSEtMTgUAlzMJeA&q=hugging+face&rlz=1C1JJTC_enIN1103IN1108&oq=hugging+fac...
Google
hugging face
AI Mode
All
Images
Shopping
Videos
News
Short videos
More
Tools
Hugging Face
https://huggingface.co
Hugging Face – The AI community building the future.
The AI community building the future. · Models · Spaces · Datasets · The Home of Machine Learning · Accelerate your ML · More than 50,000 organizations are ...
Kolors Virtual Try-On in the Wild
Upload a person image and a garment image to see how the ...
Spaces
Image Generation · Video Generation · Text Generation ...
Log In
We're on a journey to advance and democratize artificial ...
Activate Windows
Go to Settings to activate Windows.

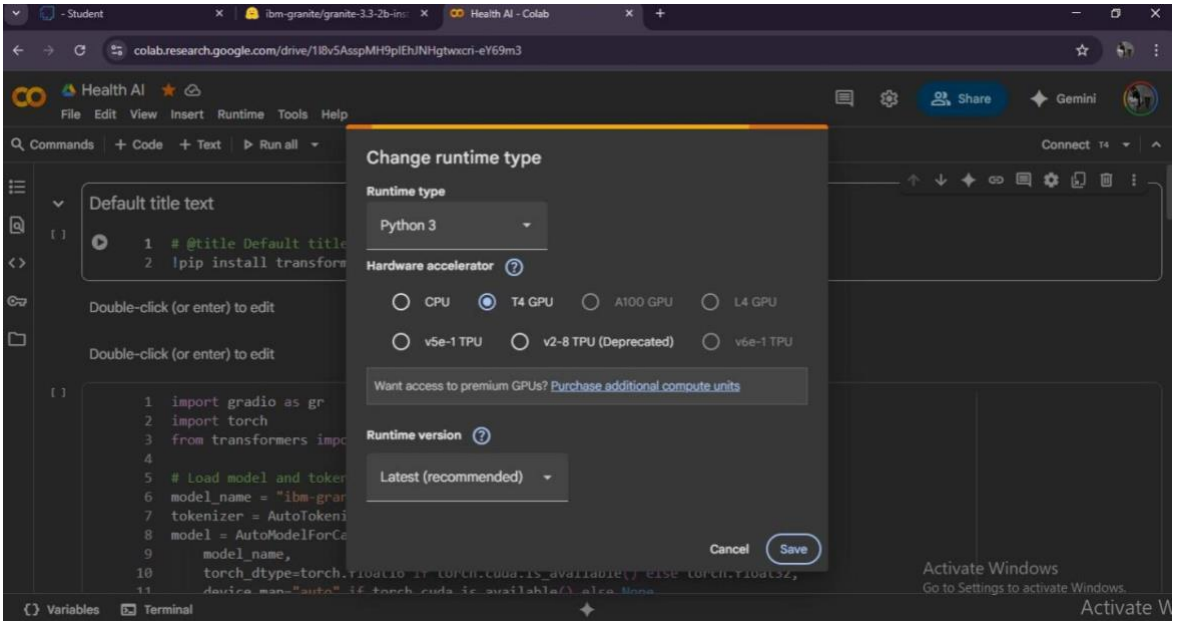

- Student
ibm-granite/granite-embeddi
huggingface.co/ibm-granite/granite-embedding-english-r2
Hugging Face
IBM-granite
Models
Datasets
Spaces
Community
Docs
Pricing
Log In
Sign Up
Models
ibm-granite/granite-embedding-english-r2
ibm-granite/granite-embedding-small-english-r2
ibm-granite/granite-speech-3.3-8b
ibm-granite/granite-speech-3.3-2b
ibm-granite/granite-3.3-2b-instruct
ibm-granite/granite-guardian-3.2-5b-lora-harm-c...
See 398 model results for "IBM-granite"
Datasets
ibm-granite/GneissWeb
open-llm-leaderboard/ibm-granite__granite-7b-ba...
open-llm-leaderboard/ibm-granite__granite-3.0-2...
See 30 dataset results for "IBM-granite"
Spaces
ibm-granite/granite-3.1-8b-instruct
ibm-granite/granite-vision-demo
shriyadav/IBMGraniteBBCode
Use Full-text search
ibm-granite/granite-embedding-english-r2
Sentence Similarity
Transformers
English
modernbert
feature-extraction
granite
embeddings
mteb
Model card
Train
Deploy
Use this model
Granite-Embedding
Model Summary: Grani
biencoder embedding n
used to generate high q
vectors of size 768 base
other open-source mod
pair datasets with perm
generated datasets.
The r2 models show stro
Downloads last month
13,196
Inference Providers
Sentence Similarity
This model isn't deployed by any Inference Provider.
Ask for provider support
Model tree for ibm-granite/granite-embedding-english-r2
Finetunes
Quantizations
1 model
Activate Windows
Go to Settings to activate Windows.

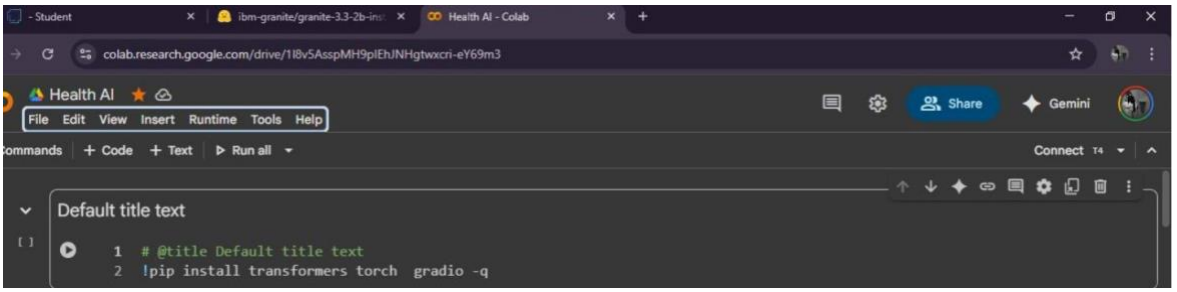

- Student
ibm-granite/granite-3.3-2b-ins
google colab - Google Search
google.com/search?q=google+colab&rlz=1C1JJTC_enIN1103IN1108&oq=goole+colab&gs_lcrp=EgZjaHJvbWUqEwgBEAAYgwEYkQIYsQMYgAQYigUyBggAEEUYOTITCA...
Google
google colab
AI Mode
All
Videos
Images
Shopping
Short videos
News
More
Tools
Google Colab
https://colab.research.google.com
Welcome To Colab - Colab - Google
Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When you create your own Colab ...
Jupyter.ipynb
This section describes how to edit and run the code in each ...
Run in Google Colab
Colab Specifics ... Colab is a virtual machine you can access ...
Pro
Colab Pro+ ... Limited time offer of an additional 100 compute units ...
Activate Windows
Go to Settings to activate Windows.

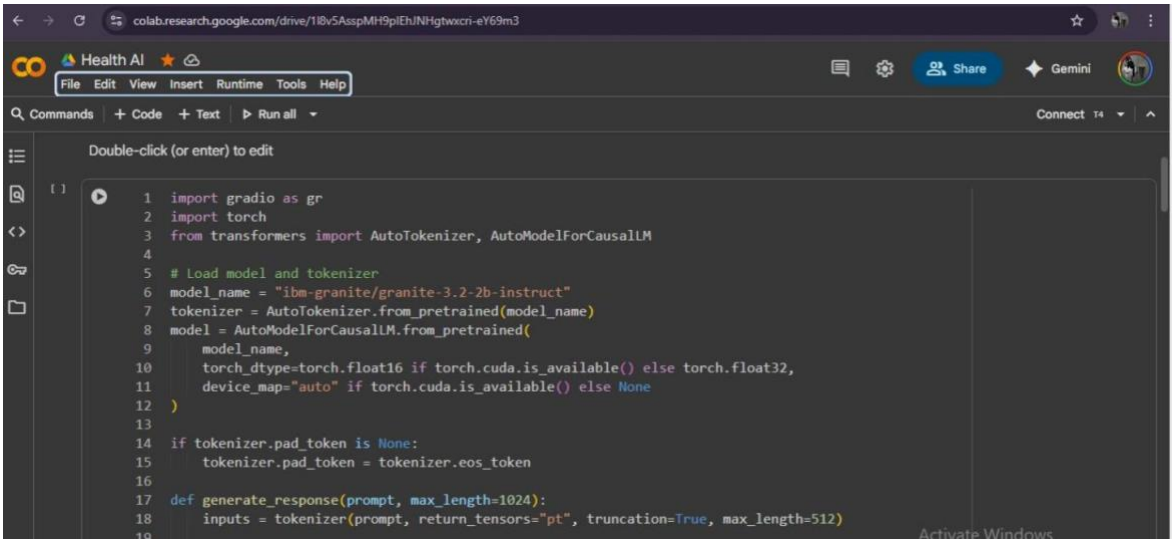

colab.research.google.com
Welcome To Colab
File
Edit
View
Insert
Runtime
Tools
Help
Share
Gemini
Connect
Copy to Drive
New notebook in Drive
Open notebook Ctrl+O
Upload notebook
Rename
Save a copy in Drive
Save a copy as a GitHub Gist
Save a copy in GitHub
Save Ctrl+S
Revision history
Download
Print Ctrl+P
Welcome to Colab!
Access Popular LLMs via Google-Colab-AI Without an API Key
Users with Colab's paid plans have free access to most popular LLMs via google-colab-ai Python library. For more details, refer to the getting started with google colab ai.
from google.colab import ai
response = ai.generate_text("What is the capital of France?")
print(response)
Explore the Gemini API
The Gemini API gives you access to Gemini models created by Google DeepMind. Gemini models are built from the ground up to be multimodal, so you can reason seamlessly across text, images, code, and audio.
How to get started?
Go to Google AI Studio and log in with your Google account.
Variables
Terminal
Activate Windows
Go to Settings to activate Windows.


Welcome To Colab - Colab
colab.research.google.com
Welcome To Colab
File Edit View
Share
Gemini
Commands
+ Code
Connect
Table of contents
Welcome to Colab!
Getting started
Data science
Machine learning
More Resources
Featured examples
+ Section
Open notebook
Examples
Recent
Google Drive
GitHub
Upload
Search notebooks
Title
Last opened
First opened
Welcome To Colab
11:31 PM
June 21
Health AI
September 6
June 21
Python Loops.ipynb
August 6
August 6
Python Conditional Statement.ipynb
August 5
August 5
+ New notebook
Cancel
Variables
Terminal
Activate Windows
Go to Settings to activate Windows.

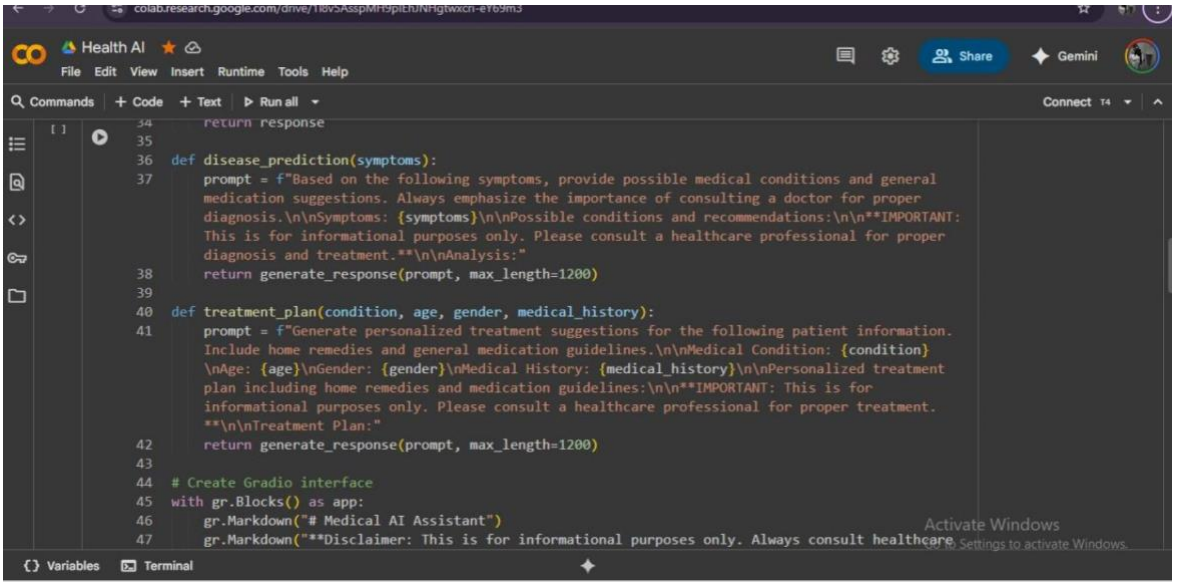

colab.research.google.com/drive/1I8v5AsspMH9pIEhJNHgtwxcri-eY69m3
Health AI
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
Connect
Run all Ctrl+F9
Run before Ctrl+F8
Run the focused cell Ctrl+Enter
Run selection Ctrl+Shift+Enter
Run cell and below Ctrl+F10
Interrupt execution Ctrl+M I
Restart session Ctrl+M .
Restart session and run all
Disconnect and delete runtime
Change runtime type
Manage sessions
View resources
View runtime logs
Default title text
Double-click (or ent
Double-click (or ent
toModelForCausalLM
nstruct"
model_name)
model_name,
torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
Variables
Terminal
Activate Windows
Go to Settings to activate Windows.


Health AI
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
Connect
Change runtime type
Runtime type
Python 3
Hardware accelerator
CPU
T4 GPU
A100 GPU
L4 GPU
v5e-1 TPU
v2-8 TPU (Deprecated)
v6e-1 TPU
Want access to premium GPUs? Purchase additional compute units
Runtime version
Latest (recommended)
Cancel
Save
Default title text
Double-click (or enter) to edit
Activate Windows
Go to Settings to activate Windows.
Variables
Terminal

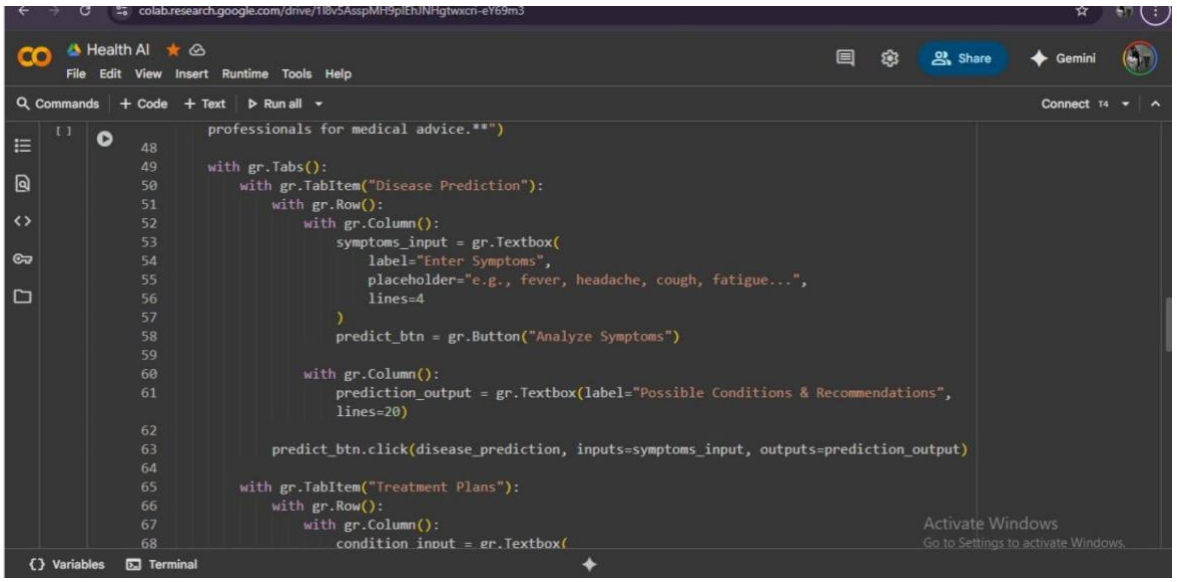

Health AI
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
Connect
Default title text
1 # @title Default title text
2 !pip install transformers torch gradio -q


Health AI
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
Connect
Double-click (or enter) to edit
1 import gradio as gr
2 import torch
3 from transformers import AutoTokenizer, AutoModelForCausalLM
4
5 # Load model and tokenizer
6 model_name = "ibm-granite/granite-3.2-2b-instruct"
7 tokenizer = AutoTokenizer.from_pretrained(model_name)
8 model = AutoModelForCausalLM.from_pretrained(
9 model_name,
10 torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
11 device_map="auto" if torch.cuda.is_available() else None
12)
13
14 if tokenizer.pad_token is None:
15 tokenizer.pad_token = tokenizer.eos_token
16
17 def generate_response(prompt, max_length=1024):
18 inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)

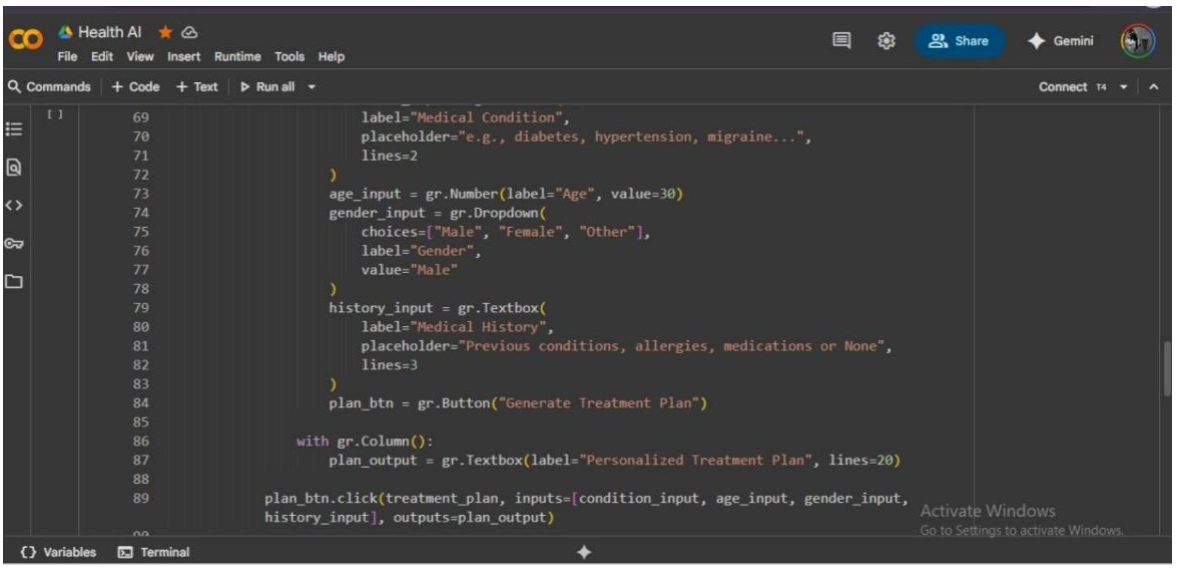
```
def generate_response(prompt, max_length=1024):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)

    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id
        )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)
    response = response.replace(prompt, "").strip()
    return response

def disease_prediction(symptoms):
    prompt = f"Based on the following symptoms, provide possible medical conditions and general
    medication suggestions. Always emphasize the importance of consulting a doctor for proper
```

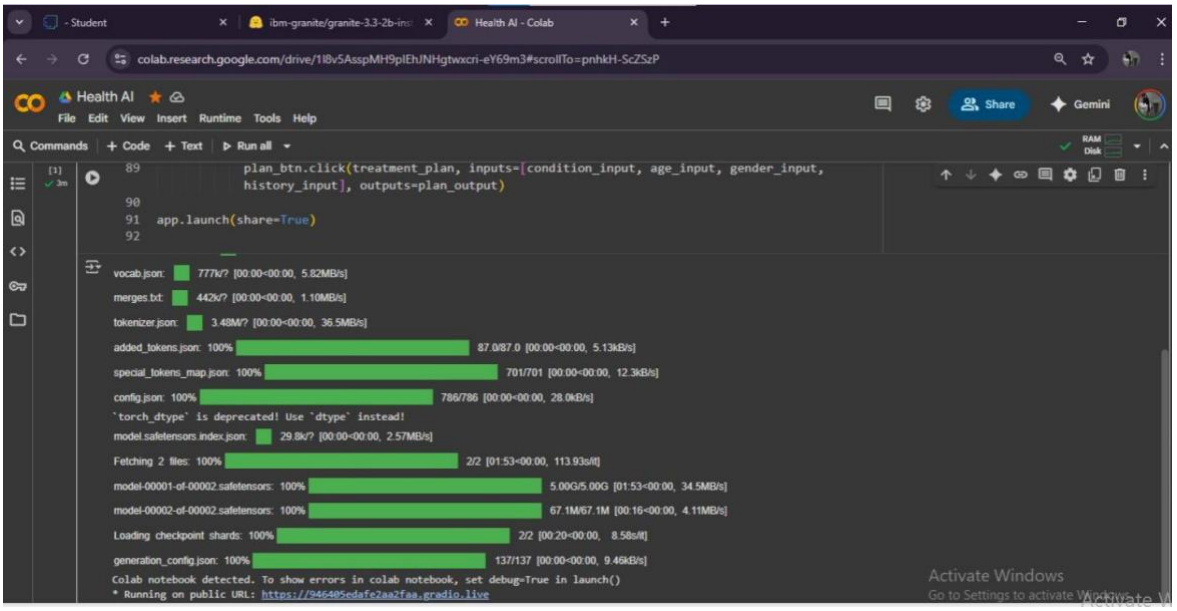
```
    return response

def disease_prediction(symptoms):
    prompt = f"Based on the following symptoms, provide possible medical conditions and general
    medication suggestions. Always emphasize the importance of consulting a doctor for proper
    diagnosis.\n\nSymptoms: {symptoms}\n\nPossible conditions and recommendations:\n\n**IMPORTANT:
    This is for informational purposes only. Please consult a healthcare professional for proper
    diagnosis and treatment.**\n\nAnalysis:"
    return generate_response(prompt, max_length=1200)

def treatment_plan(condition, age, gender, medical_history):
    prompt = f"Generate personalized treatment suggestions for the following patient information.
    Include home remedies and general medication guidelines.\n\nMedical Condition: {condition}
    \nAge: {age}\nGender: {gender}\nMedical History: {medical_history}\n\nPersonalized treatment
    plan including home remedies and medication guidelines:\n\n**IMPORTANT: This is for
    informational purposes only. Please consult a healthcare professional for proper treatment.
    **\n\nTreatment Plan:"
    return generate_response(prompt, max_length=1200)

# Create Gradio interface
with gr.Blocks() as app:
    gr.Markdown("# Medical AI Assistant")
    gr.Markdown("**Disclaimer: This is for informational purposes only. Always consult healthcare
```

```
    professionals for medical advice.**")

    with gr.Tabs():
        with gr.TabItem("Disease Prediction"):
            with gr.Row():
                with gr.Column():
                    symptoms_input = gr.Textbox(
                        label="Enter Symptoms",
                        placeholder="e.g., fever, headache, cough, fatigue...",
                        lines=4
                    )
                    predict_btn = gr.Button("Analyze Symptoms")

                with gr.Column():
                    prediction_output = gr.Textbox(label="Possible Conditions & Recommendations",
                    lines=20)

            predict_btn.click(disease_prediction, inputs=symptoms_input, outputs=prediction_output)

        with gr.TabItem("Treatment Plans"):
            with gr.Row():
                with gr.Column():
                    condition_input = gr.Textbox(
```

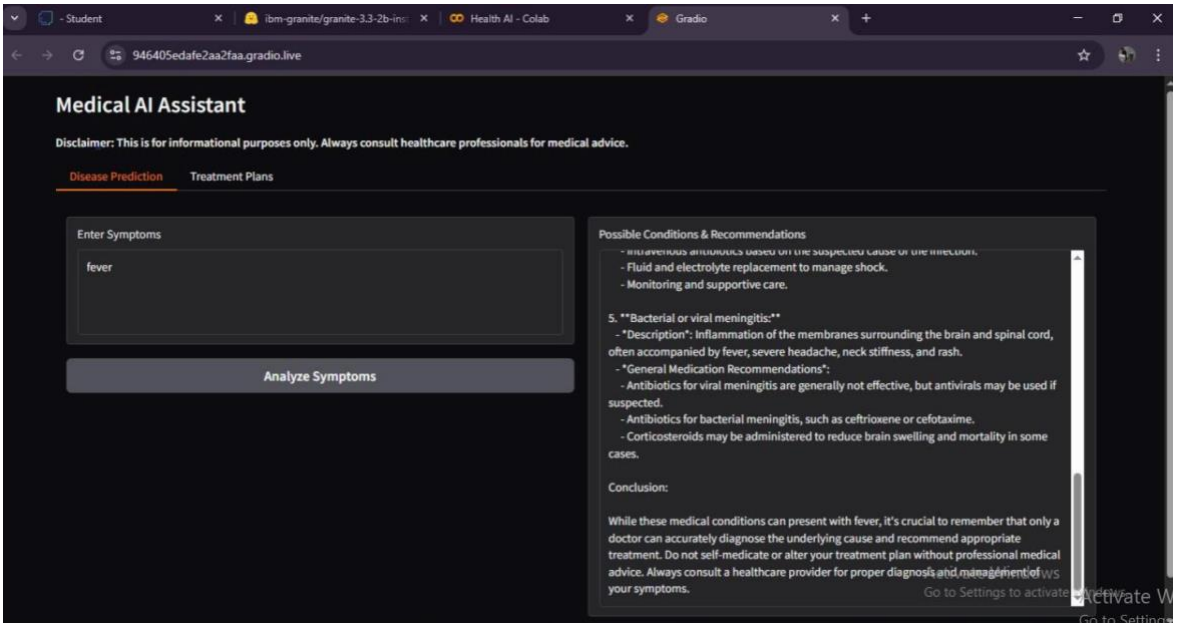


```
                    label="Medical Condition",
                    placeholder="e.g., diabetes, hypertension, migraine...",
                    lines=2
                )
                age_input = gr.Number(label="Age", value=30)
                gender_input = gr.Dropdown(
                    choices=["Male", "Female", "Other"],
                    label="Gender",
                    value="Male"
                )
                history_input = gr.Textbox(
                    label="Medical History",
                    placeholder="Previous conditions, allergies, medications or None",
                    lines=3
                )
                plan_btn = gr.Button("Generate Treatment Plan")

            with gr.Column():
                plan_output = gr.Textbox(label="Personalized Treatment Plan", lines=20)

        plan_btn.click(treatment_plan, inputs=[condition_input, age_input, gender_input,
        history_input], outputs=plan_output)
```

```
plan_btn.click(treatment_plan, inputs=[condition_input, age_input, gender_input,
history_input], outputs=plan_output)

app.launch(share=True)
```

```
vocab.json: 777k/? [00:00<00:00, 5.82MB/s]
merges.txt: 442k/? [00:00<00:00, 1.10MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 36.5MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 5.13kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 12.3kB/s]
config.json: 100% 786/786 [00:00<00:00, 28.0kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 2.57MB/s]
Fetching 2 files: 100% 2/2 [01:53<00:00, 113.93s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [01:53<00:00, 34.5MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:16<00:00, 4.11MB/s]
Loading checkpoint shards: 100% 2/2 [00:20<00:00, 8.58s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 9.46kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://94640edafe2aa2faa.gradio.live
```


- Student
ibm-granite/granite-3.3-2b-ins
Health AI - Colab
Gradio
946405edafe2aa2faa.gradio.live
Medical AI Assistant
Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.
Disease Prediction
Treatment Plans
Enter Symptoms
fever
Analyze Symptoms
Possible Conditions & Recommendations
- Fluid and electrolyte replacement to manage shock.
- Monitoring and supportive care.
5. **Bacterial or viral meningitis:**
- *Description*: Inflammation of the membranes surrounding the brain and spinal cord, often accompanied by fever, severe headache, neck stiffness, and rash.
- *General Medication Recommendations*:
- Antibiotics for viral meningitis are generally not effective, but antivirals may be used if suspected.
- Antibiotics for bacterial meningitis, such as ceftrixone or cefotaxime.
- Corticosteroids may be administered to reduce brain swelling and mortality in some cases.
Conclusion:
While these medical conditions can present with fever, it's crucial to remember that only a doctor can accurately diagnose the underlying cause and recommend appropriate treatment. Do not self-medicate or alter your treatment plan without professional medical advice. Always consult a healthcare provider for proper diagnosis and management of your symptoms.
Activate Windows
Go to Settings to activate Windows.