SAT 5114 AI IN HEALTHCARE

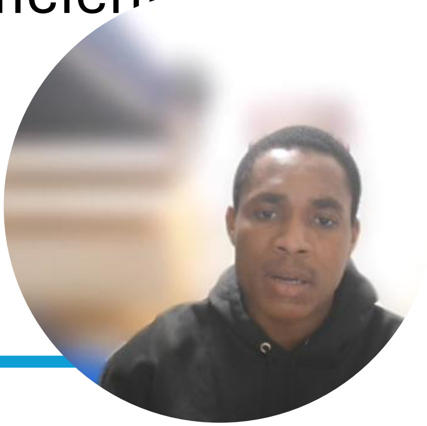
Final Project: AI-powered Telemedicine and Remote Patient Monitoring model using LLM

Presenter: Gideon Owusu

Date: 4/18/2025

Problem Statement

- Existing telemedicine have limited patient engagement and real-time artificial intelligence for documentation and clinical decision support. This leads to delayed diagnosis and significant gaps in providing efficient healthcare



Literature Review

1. The Impact of multimodal LLM on healthcare's future

The current LLM in medicine and healthcare is unimodal and can accept and process text queries. Considering the multimodal nature of Medicine and healthcare, multimodal LLMS that can handle texts and analyze images, videos, and sounds are ideal and would represent a significant achievement in the field of medicine (Mesko, 2023). The researcher further explains the hypothetical benefits of multimodal LLMS accepting patient text queries, analyzing their images of lesions and x-rays. Moreover, such an LLMS can potentially analyze vocal, lung, and heart sounds for abnormalities and generate comprehensive information and recommendations about the patient

2. Applying Object Detection and Large Language Model to Establish a Smart Telemedicine Diagnosis System with Chatbot: A Case Study of Pressure Injuries Diagnosis System.

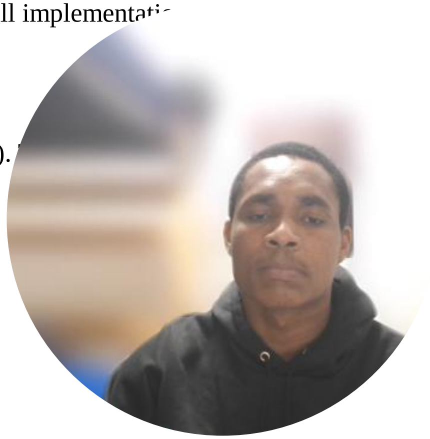
Assessing patients and pressure wounds without physical examination is a major challenge confronting Telemedicine (Chen et al., 2024). The researchers leveraged an auxiliary diagnosis system using YOLOv7 and LLM for enhanced detection and classification of pressure wounds and diagnosis. This approach generated an average F1 score of 0.9238, providing remote real-time diagnosis and staging of wounds.

3. Transformer Models in Healthcare: A Survey and Thematic Analysis of Potentials, Shortcomings, and Risks

Large language models have significant benefits in the healthcare sector, such as optimized processes, improved operational efficiency, and efficient clinical documentation (Denecke et al., 2024). Despite these benefits, the study uncovered potential challenges, such as privacy challenges, model development bias, and auditability issues hindering the full implementati of LLM in healthcare.

4. Transforming virtual healthcare: The potential of ChatGPT-4omni in telemedicine

The inception of ChatGPT-4o is a significant achievement in virtual healthcare and telemedicine, allowing users to upload medical images (Mohamad-Hani et al., 2024). argue that such milestones would improve remote patient care and access to personalized treatment plans.

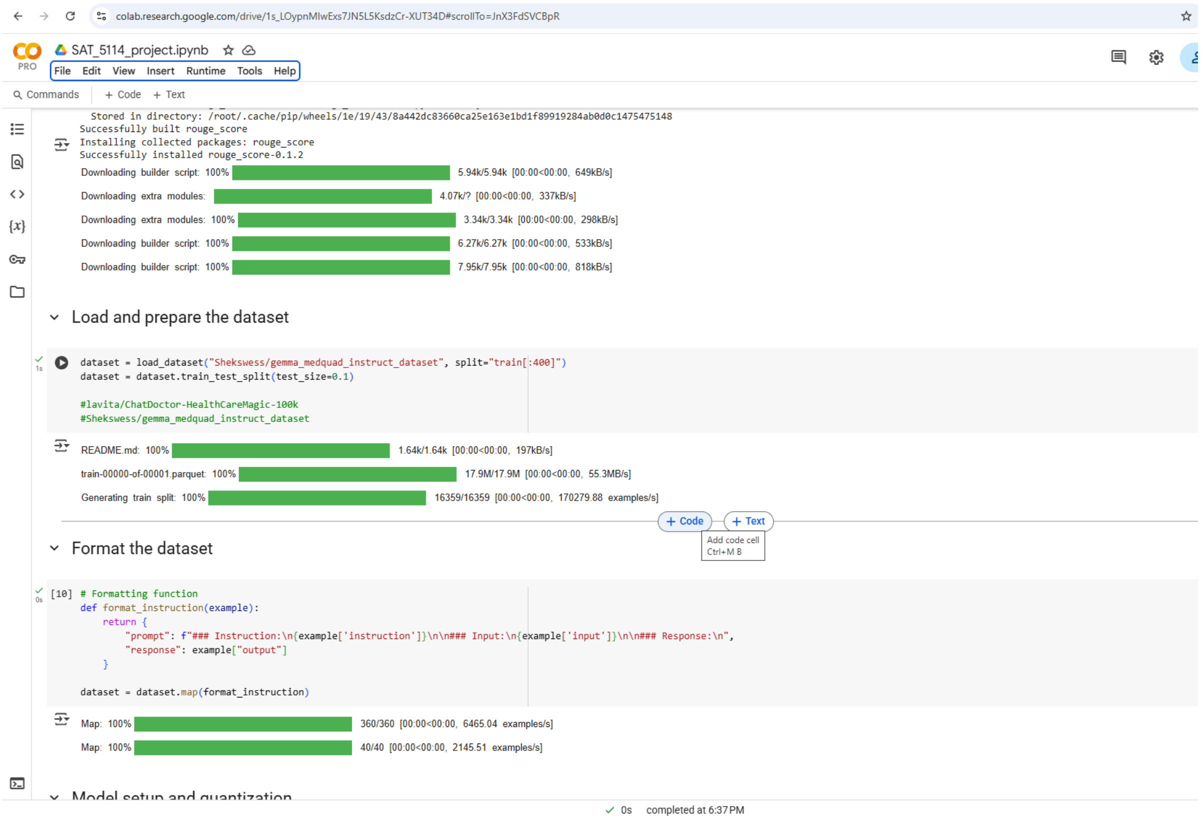


Purpose of the Code

- The purpose of the code is to develop AI-powered Telemedicine leveraging LLM to;
- Generate answers to patient queries
- Enhance patient engagement,
- Improve healthcare accessibility and
- Optimize workflows

Code Demo (Dataset and Model)

The dataset is from Hugging Face (https://huggingface.co/datasets/Shekswess/gemma_medquad_instruct_dataset)
Model: Llama 3 8b from Unsloth


colab.research.google.com/drive/1s_LOypnMIwExs7JN5L5KsdzCr-XUT34D#scrollTo=JnX3FdSVCBpR
CO PRO
SAT_5114_project.ipynb
File Edit View Insert Runtime Tools Help
Commands
+ Code + Text

```
  Stored in directory: /root/.cache/pip/wheels/1e/19/43/8a442dc83660ca25e163e1bd1f89919284ab0d0c1475475148
Successfully built rouge_score
Installing collected packages: rouge_score
Successfully installed rouge_score-0.1.2
Downloading builder script: 100%  5.94k/5.94k [00:00<00:00, 649kB/s]
Downloading extra modules:  4.07k/? [00:00<00:00, 337kB/s]
Downloading extra modules: 100%  3.34k/3.34k [00:00<00:00, 298kB/s]
Downloading builder script: 100%  6.27k/6.27k [00:00<00:00, 533kB/s]
Downloading builder script: 100%  7.95k/7.95k [00:00<00:00, 818kB/s]
```

Load and prepare the dataset

```
dataset = load_dataset("Shekswess/gemma_medquad_instruct_dataset", split="train[:400]")
dataset = dataset.train_test_split(test_size=0.1)

#lavita/ChatDoctor-HealthCareMagic-100k
#Shekswess/gemma_medquad_instruct_dataset
```

```
README.md: 100%  1.64k/1.64k [00:00<00:00, 197kB/s]
train-00000-of-00001.parquet: 100%  17.9M/17.9M [00:00<00:00, 55.3MB/s]
Generating train split: 100%  16359/16359 [00:00<00:00, 170279.88 examples/s]
```

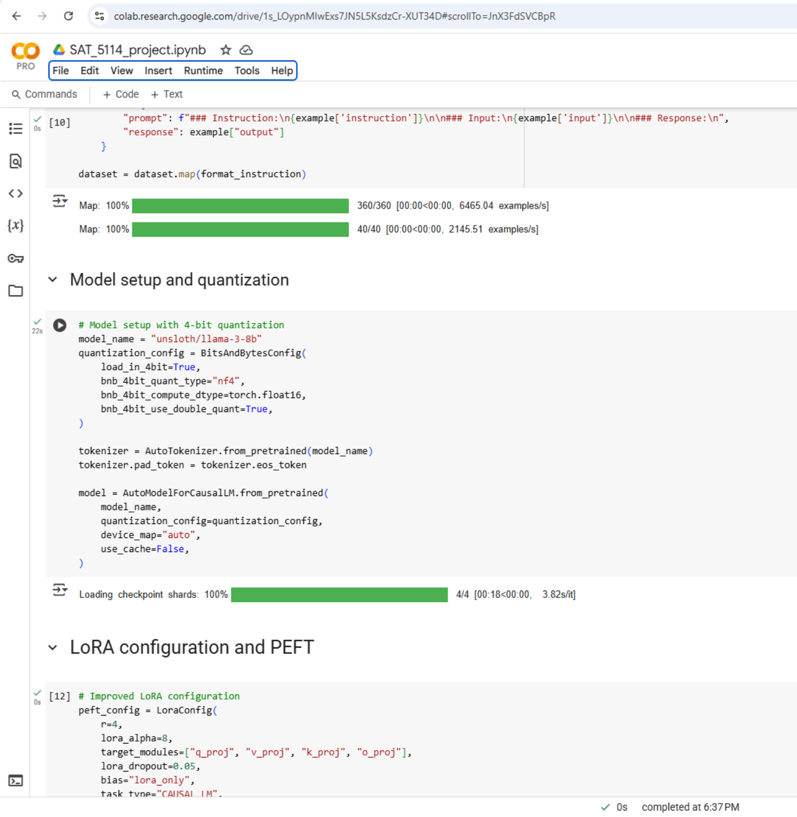
Format the dataset

```
[10] # Formatting function
     def format_instruction(example):
         return {
             "prompt": f"### Instruction:\n{example['instruction']}\n\n### Input:\n{example['input']}\n\n### Response:\n",
             "response": example["output"]
         }

     dataset = dataset.map(format_instruction)
```

```
Map: 100%  360/360 [00:00<00:00, 6465.04 examples/s]
Map: 100%  40/40 [00:00<00:00, 2145.51 examples/s]
```

Model setup and quantization

✓ 0s completed at 6:37 PM


colab.research.google.com/drive/1s_LOypnMIwExs7JN5L5KsdzCr-XUT34D#scrollTo=JnX3FdSVCBpR
CO PRO
SAT_5114_project.ipynb
File Edit View Insert Runtime Tools Help
Commands
+ Code + Text

```
[10]         "prompt": f"### Instruction:\n{example['instruction']}\n\n### Input:\n{example['input']}\n\n### Response:\n",
             "response": example["output"]
         }

     dataset = dataset.map(format_instruction)
```

```
Map: 100%  360/360 [00:00<00:00, 6465.04 examples/s]
Map: 100%  40/40 [00:00<00:00, 2145.51 examples/s]
```

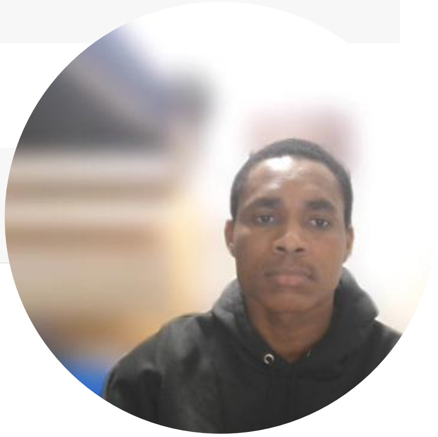
Model setup and quantization

```
# Model setup with 4-bit quantization
model_name = "unsloth/llama-3-8b"
quantization_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16,
    bnb_4bit_use_double_quant=True,
)

tokenizer = AutoTokenizer.from_pretrained(model_name)
tokenizer.pad_token = tokenizer.eos_token

model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=quantization_config,
    device_map="auto",
    use_cache=False,
)
```

```
Loading checkpoint shards: 100%  4/4 [00:18<00:00,  3.82s/it]
```

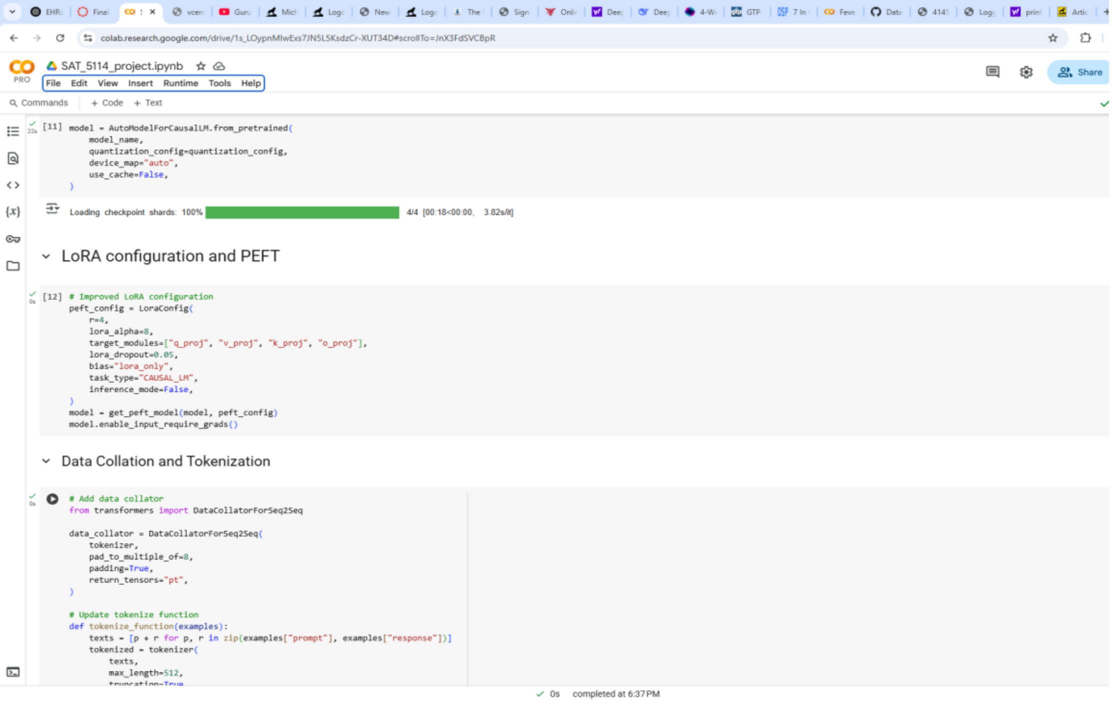
LoRA configuration and PEFT

```
[12] # Improved LoRA configuration
     peft_config = LoraConfig(
         r=4,
         lora_alpha=8,
         target_modules=["q_proj", "v_proj", "k_proj", "o_proj"],
         lora_dropout=0.05,
         bias="lora_only",
         task_type="CAUSAL_LM",
```



Code Demo- Cont’d (LoRA and Performance Metrics)

Bleu : 0.45
BertF1 score = 0.90
RougeL = 0.66



```
[11] model = AutoModelForCausalLM.from_pretrained(
         model_name,
         quantization_config=quantization_config,
         device_map="auto",
         use_cache=False,
     )
```

```
Loading checkpoint shards: 100% 4/4 [00:18<00:00,  3.82s/it]
```

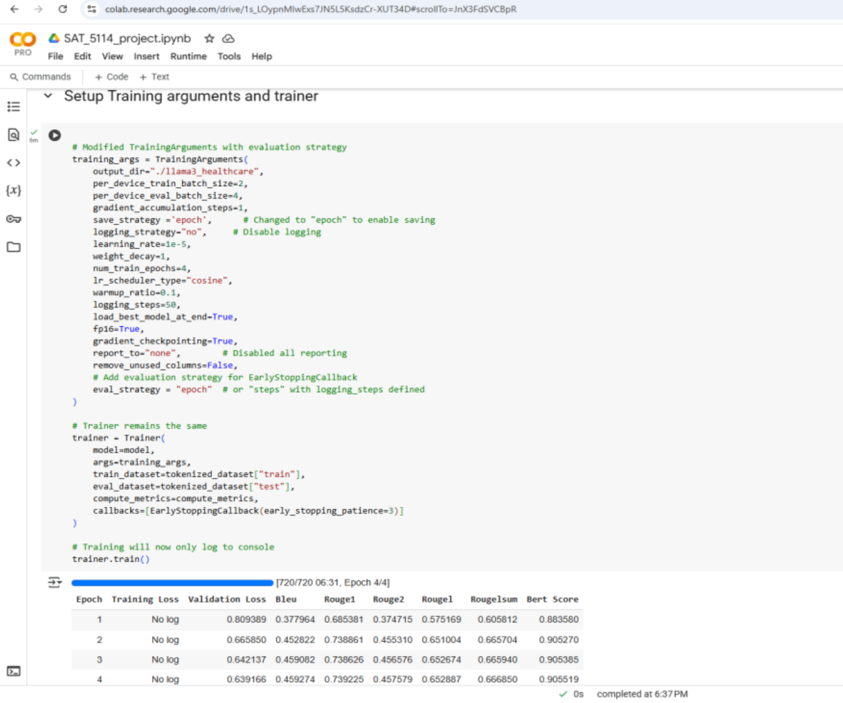
LoRA configuration and PEFT

```
[12] # Improved LoRA configuration
     peft_config = LoraConfig(
         r=4,
         lora_alpha=8,
         target_modules=["q_proj", "v_proj", "k_proj", "o_proj"],
         lora_dropout=0.05,
         bias="lora_only",
         task_type="CAUSAL_LM",
         inference_mode=False,
     )
     model = get_peft_model(model, peft_config)
     model.enable_input_require_grads()
```

Data Collation and Tokenization

```
# Add data collator
from transformers import DataCollatorForSeq2Seq

data_collator = DataCollatorForSeq2Seq(
    tokenizer,
    pad_to_multiple_of=8,
    padding=True,
    return_tensors="pt",
)

# Update tokenize function
def tokenize_function(examples):
    texts = [p + r for p, r in zip(examples["prompt"], examples["response"])]
    tokenized = tokenizer(
        texts,
        max_length=512,
        truncation=True
```

✓ 0s completed at 6:37 PM



Setup Training arguments and trainer

```
# Modified TrainingArguments with evaluation strategy
training_args = TrainingArguments(
    output_dir="./llama3_healthcare",
    per_device_train_batch_size=2,
    per_device_eval_batch_size=4,
    gradient_accumulation_steps=1,
    save_strategy ='epoch',      # Changed to "epoch" to enable saving
    logging_strategy="no",     # Disable logging
    learning_rate=1e-5,
    weight_decay=1,
    num_train_epochs=4,
    lr_scheduler_type="cosine",
    warmup_ratio=0.1,
    logging_steps=50,
    load_best_model_at_end=True,
    fp16=True,
    gradient_checkpointing=True,
    report_to="none",        # Disabled all reporting
    remove_unused_columns=False,
    # Add evaluation strategy for EarlyStoppingCallback
    eval_strategy = "epoch"  # or "steps" with logging_steps defined
)

# Trainer remains the same
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset["train"],
    eval_dataset=tokenized_dataset["test"],
    compute_metrics=compute_metrics,
    callbacks=[EarlyStoppingCallback(early_stopping_patience=3)]
)

# Training will now only log to console
trainer.train()
```

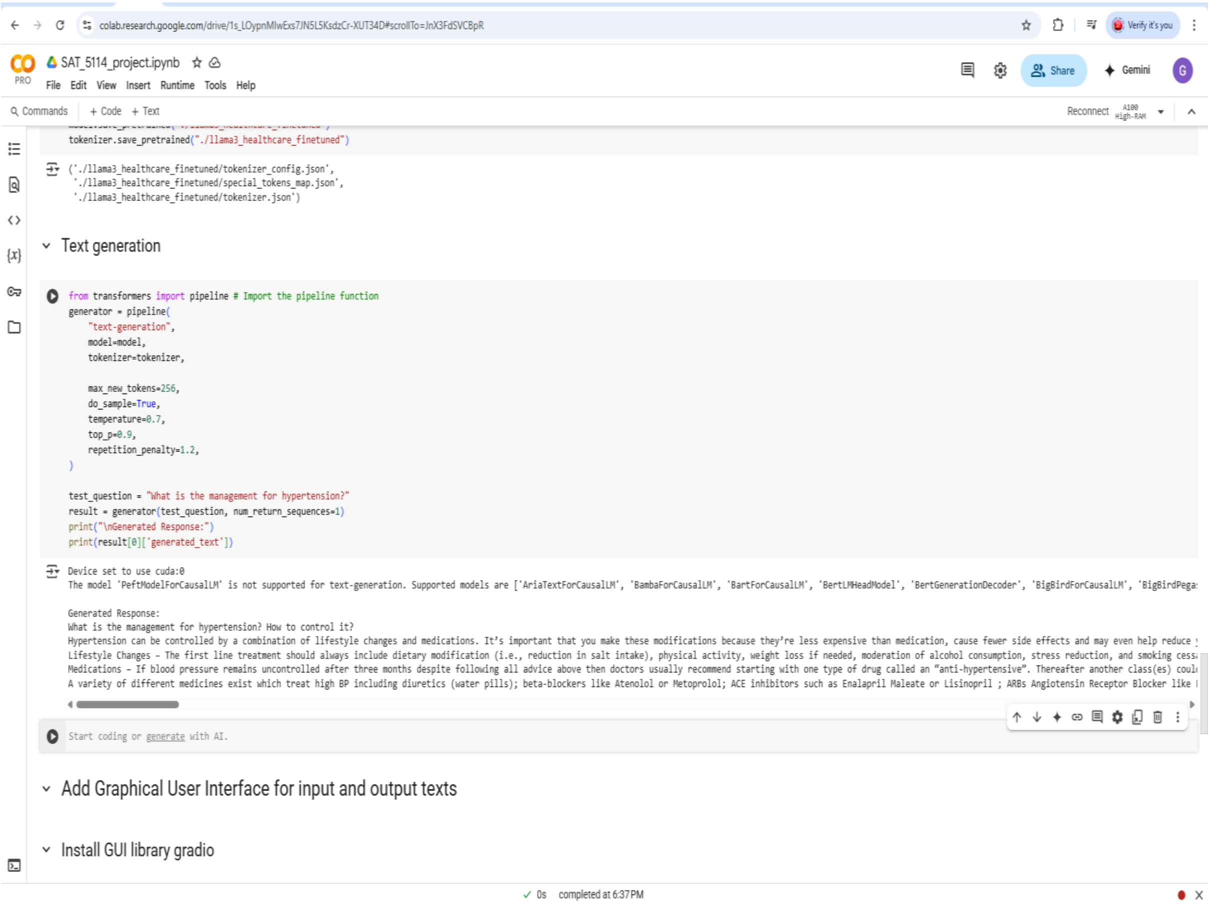
[720/720 06:31, Epoch 4/4]

Epoch	Training Loss	Validation Loss	Bleu	Rouge1	Rouge2	Rougel	Rougelsum	Bert Score
1	No log	0.809389	0.377964	0.685381	0.374715	0.575169	0.605812	0.883580
2	No log	0.665850	0.452822	0.738861	0.455310	0.651004	0.665704	0.905270
3	No log	0.642137	0.459082	0.738626	0.456576	0.652674	0.665940	0.905385
4	No log	0.639166	0.459274	0.739225	0.457579	0.652887	0.666850	0.905519

✓ 0s completed at 6:37 PM

Code Demo-Cont’d (Text Generation in GUI)

Test the model by inputting a query to generate a response
GUI using Gradio


colab.research.google.com/drive/1s_LOypnMlwExs7JN5L5KsdzCr-XUT34D#scrollTo=JnX3FdSVCBpR
SAT_5114_project.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Share
Gemini
Reconnect

```
tokenizer.save_pretrained("./llama3_healthcare_finetuned")
```

```
('./llama3_healthcare_finetuned/tokenizer_config.json',
 './llama3_healthcare_finetuned/special_tokens_map.json',
 './llama3_healthcare_finetuned/tokenizer.json')
```

Text generation

```
from transformers import pipeline # Import the pipeline function
generator = pipeline(
    "text-generation",
    model=model,
    tokenizer=tokenizer,

    max_new_tokens=256,
    do_sample=True,
    temperature=0.7,
    top_p=0.9,
    repetition_penalty=1.2,
)

test_question = "What is the management for hypertension?"
result = generator(test_question, num_return_sequences=1)
print("\nGenerated Response:")
print(result[0]['generated_text'])
```

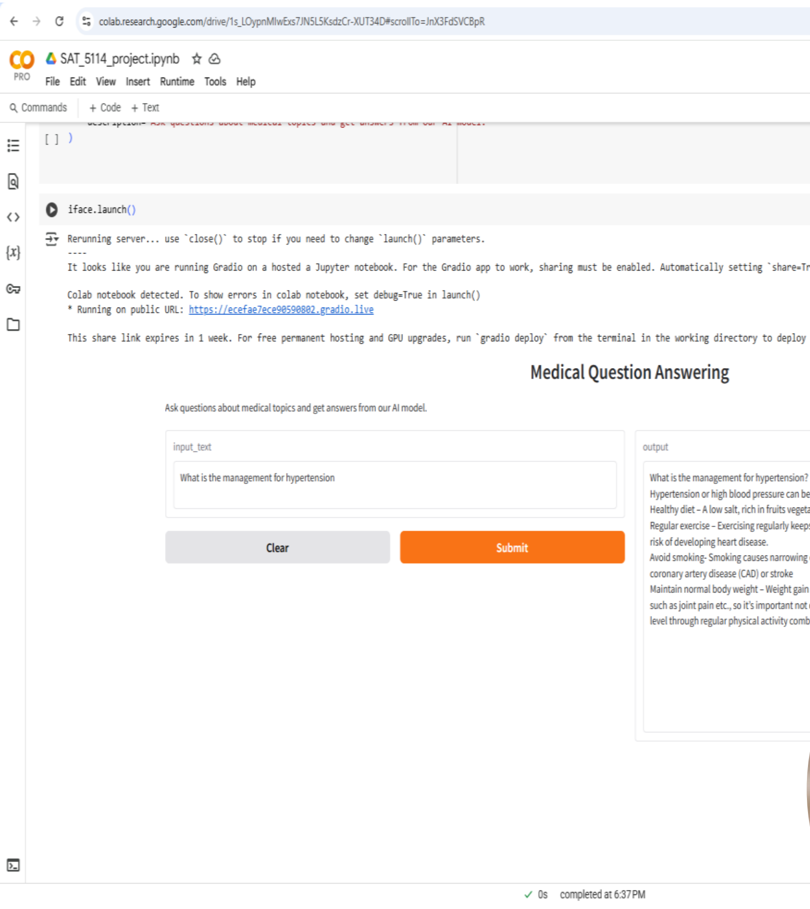
```
Device set to use cuda:0
The model 'PeftModelForCausalLM' is not supported for text-generation. Supported models are ['AriaTextForCausalLM', 'BambaForCausalLM', 'BartForCausalLM', 'BertLMHeadModel', 'BertGenerationDecoder', 'BigBirdForCausalLM', 'BigBirdPega

Generated Response:
What is the management for hypertension? How to control it?
Hypertension can be controlled by a combination of lifestyle changes and medications. It’s important that you make these modifications because they’re less expensive than medication, cause fewer side effects and may even help reduce y
Lifestyle Changes – The first line treatment should always include dietary modification (i.e., reduction in salt intake), physical activity, weight loss if needed, moderation of alcohol consumption, stress reduction, and smoking cess
Medications – If blood pressure remains uncontrolled after three months despite following all advice above then doctors usually recommend starting with one type of drug called an “anti-hypertensive”. Thereafter another class(es) coul
A variety of different medicines exist which treat high BP including diuretics (water pills); beta-blockers like Atenolol or Metoprolol; ACE inhibitors such as Enalapril Maleate or Lisinopril ; ARBs Angiotensin Receptor Blocker like I
```

Start coding or generate with AI.

Add Graphical User Interface for input and output texts

Install GUI library gradio

✓ 0s completed at 6:37 PM


colab.research.google.com/drive/1s_LOypnMlwExs7JN5L5KsdzCr-XUT34D#scrollTo=JnX3FdSVCBpR
SAT_5114_project.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Share
Gemini
Reconnect

```
iface.launch()
```

```
Rerunning server... use `close()` to stop if you need to change `launch()` parameters.
----
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio app to work, sharing must be enabled. Automatically setting `share=True` (you can turn this off by setting `share=False` in `launch()` explicitly).

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://ecefae7ece90590802.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (https://huggingface.co/spaces)
```

Medical Question Answering

Ask questions about medical topics and get answers from our AI model.

input_text

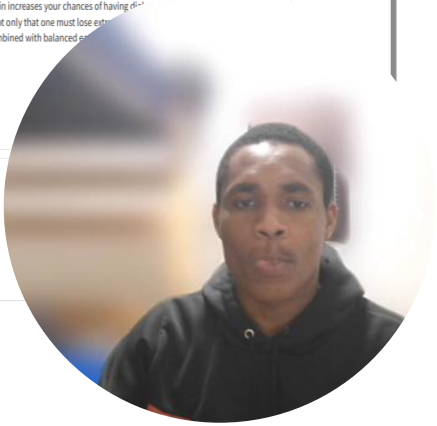
What is the management for hypertension

Clear

Submit

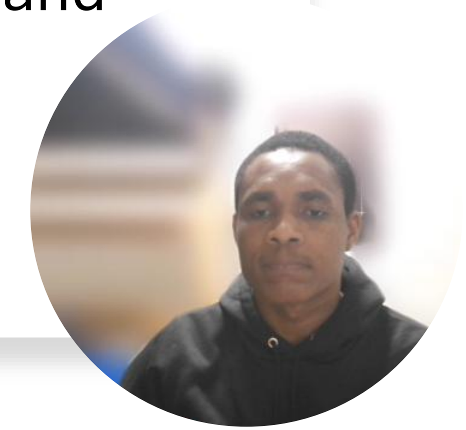
output

What is the management for hypertension?
Hypertension or high blood pressure can be controlled and managed by following a healthy lifestyle. This includes:
Healthy diet – A low salt, rich in fruits vegetables and nuts diet helps to control your blood pressure levels.
Regular exercise – Exercising regularly keeps you fit and controls your weight which has been linked with increased risk of developing heart disease.
Avoid smoking- Smoking causes narrowing of arteries leading to development of cardiovascular diseases like coronary artery disease (CAD) or stroke
Maintain normal body weight – Weight gain increases your chances of having di
such as joint pain etc., so it’s important not only that one must lose ext
level through regular physical activity combined with balanced e



Conclusion

- This model performed well regarding generating responses to queries; however, it was trained on a significantly small portion of the dataset to conserve computer power which affected its performance
- It can be upgraded to accept queries in different formats, like audio and video queries



References

1. Meskó B

The Impact of Multimodal Large Language Models on Health Care’s Future

J Med Internet Res 2023;25:e52865

URL: https://www.jmir.org/2023/1/e52865

DOI: 10.2196/52865

2. Applying Object Detection and Large Language Model to Establish a Smart Telemedicine Diagnosis System with Chatbot: A Case Study of Pressure Injuries Diagnosis System

Chun-Chia Chen, Chia-Jung Wei, Tsung-Yu Tseng, Ming-Chuan Chiu, and Chi-Chang Chang

Telemedicine and e-Health 2024 30:6, e1705-e1712. https://www.liebertpub.com/action/showCitFormats?doi=10.1089%2Ftmj.2023.0715

3. Denecke, K., May, R. & Rivera-Romero, O. Transformer Models in Healthcare: A Survey and Thematic Analysis of Potentials, Shortcomings and Risks. *J Med Syst* 48, (2024). https://doi.org/10.1007/s10916-024-02043-5

4. Mohamad-Hani, T., Jamal, A., Khalid, A., Fadi, A., Ibraheem, A., Malki, K. H., . . . Al-Eyadhy, A. (2024). Transforming virtual healthcare: The potentials of telemedicine. *Cureus*, *16*(5) doi: https://doi.org/10.7759/cureus.61377