

HEALTH AI

Project Documentation

1.INTRODUCTION

- Team id: NM2025TMID08123
- Project title :HEALTH AI : INTELLIGENT HEALTHCARE ASSISTANT
- Team leader : CHIRISTINA NIVASHINI V
- Team member : HEMADHARSHINI G
- Team member : KAVIYA A
- Team member : KOWSALYA B
- Team member : MAHESHWARI B

2.PROJECT OVERVIEW

- Purpose :

The primary purpose of health AI projects is to revolutionize healthcare by improving diagnosis, treatment, and patient outcomes, while also increasing efficiency and reducing costs through automation and advanced data analysis. Key goals include enhancing diagnostic accuracy and speed, enabling personalized treatment plans, optimizing resource allocation, detecting and monitoring diseases, and improving administrative and operational efficiency by automating routine tasks

3.TYPES OF AI IN HEALTHCARE

AI is an umbrella term covering a variety of distinct but interrelated processes. Some of the most common forms of AI used within healthcare include the following:

- **Machine learning (ML):** Algorithms are trained using data sets, such as health records, to create models capable of performing tasks such as categorising information or predicting outcomes.
- **Deep learning:** This subset of machine learning involves greater volumes of data, training times, and layers of ML algorithms to produce neural networks capable of performing more complex tasks.
- **Natural language processing (NLP):** NLP involves using ML to understand human language, whether verbal or written. In healthcare, NLP is used to interpret documentation, notes, reports, and published research.
- **Robotic process automation (RPA):** This involves using AI in computer programs to automate administrative and clinical workflows. Some healthcare organisations use RPA to improve the patient experience and the daily function of their facilities

4. AI APPLICATIONS IN HEALTHCARE

- **Healthcare analytics:** ML algorithms are trained using historical data to produce insights, improve decision-making, and optimise health outcomes.
- **Precision medicine:** AI is used to produce personalised treatment plans for patients and consider factors such as their medical history, environmental factors, lifestyles, and genetic makeup.
- **Predict diseases and illness:** Using predictive models, healthcare professionals can determine the likelihood that someone might develop a particular condition or contract a disease.
- **Interpret tests and diagnose diseases:** ML models can be trained using common medical scans, like MRIs or X-rays, to interpret and diagnose conditions such as cancerous lesions.

5.FEATURES:

- As with many other industries, AI is poised to change the healthcare landscape over the coming years.
- In addition to improving health facility operations, patient diagnoses, treatment plan development, and overall health outcomes, AI is also expected to help develop and discover new medical cures.
- For instance, NITI Aayog, the public policy think tank of the Indian government, has been exploring the use of AI to detect diabetes early and is now focusing on employing AI as a screening tool in eye care.
- The use of AI in healthcare is expected to grow significantly over the next decade.
- According to Grand View Research, the global market for AI in healthcare is projected to expand at a CAGR of 38.5 percent between 2024 and 2030 .
- By 2025, India's investment in AI healthcare is forecasted to reach 11.78 billion USD, potentially resulting in a 1 trillion USD boost to the economy by 2035 .

6.AI JOBS IN HEALTHCARE

Both AI and healthcare are growing fields that are projected to have a big impact in the coming decade. It's little surprise that AI-oriented positions are becoming increasingly common within healthcare.

If you're interested in pursuing one of these careers, below are some of the positions you should consider exploring:

*Health informatics specialist

*Machine learning engineer

*Data scientist

*AI engineering

7.LIMITATION

- Disadvantages of AI in healthcare include high costs, potential for bias and misdiagnosis, significant data privacy and security risks, lack of the critical human touch, challenges with accountability and regulation, risk of over-reliance by healthcare professionals, and the potential for job displacement. Additionally, AI systems may not account for social variables, creating treatment disparities, and ensuring patient trust can be difficult.

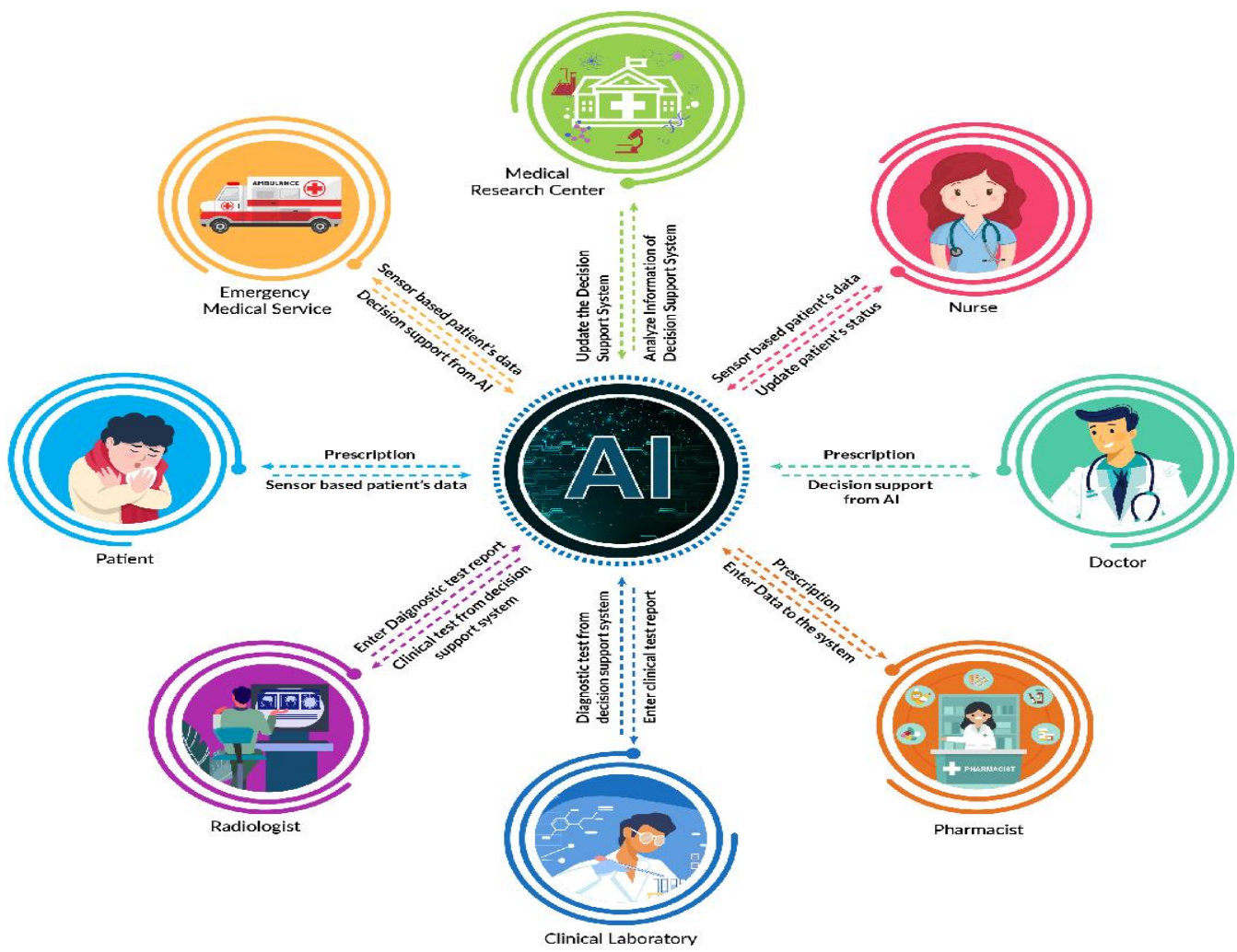
8.NECESSARY IN HEALTHCARE

- Yes, AI is necessary in healthcare to improve diagnostic accuracy, personalize treatments, streamline administrative tasks, reduce costs, and enhance overall patient care and outcomes.
- By automating processes, analyzing vast datasets, and identifying patterns, AI assists healthcare professionals in making faster, more informed decisions, leading to more efficient resource allocation and a higher quality of care, ultimately making healthcare more accessible, effective, and sustainable.


9.ARCHITECTURE


1) A smart hospital is a digital interconnected healthcare ecosystem with quick and secure data exchange between medical devices and IT systems.


2) Smart hospitals can streamline clinical and administrative workflows, enhance operational efficiency, and improve patient experiences and healing processing.




10.HUGGING FACE PAGE IS ON OPEN




**Hugging Face**





Hugging Face is way more fun with friends and colleagues!  [Join an organization](#)


Dismiss this message

 **ibm-granite / granite-3.2-2b-instruct**

 like 49  Follow  IBM Granite 2.36k

 Text Generation

 Transformers


 Safetensors


granite


language


granite-3.2


conversational


 arxiv:0000.00000



 License: apache-2.0


 Train


 Deploy

 Use this model

 Model card

 Files  xet

 Community 7

 A newer version of this model is available: [ibm-granite/granite-3.3-2b-instruct](#)

Granite-3.2-2B-Instruct

Model Summary: Granite-3.2-2B-Instruct is an 2-billion-parameter, long-context AI model fine-tuned for thinking capabilities. Built on top of [Granite-3.1-2B-Instruct](#), it has been trained using a mix of permissively licensed open-source datasets and internally generated synthetic data designed for reasoning tasks. The model allows controllability of its thinking capability, ensuring it is applied only when required.

- Developers:** Granite Team, IBM
- Website:** [Granite Docs](#)
- Release Date:** February 26th, 2025
- License:** [Apache 2.0](#)


Supported Languages: English, German, Spanish, French, Japanese, Portuguese, Arabic, Czech, Italian, Korean, Dutch, and Chinese. However, users may finetune this Granite model for languages beyond these 12 languages.



Intended Use: This model is designed to handle general instruction-following tasks and can be integrated into AI assistants across various domains, including business applications.

Capabilities


- Thinking**
- Summarization
- Text classification
- Text extraction
- ...


Downloads last month
20,105





 **Safetensors** 

Model size 2.53B params


Tensor type BF16  Chat template


 Files info

 **Inference Providers** NEW


 Text Generation


This model isn't deployed by any Inference Provider.

 1 Ask for provider support

 **Model tree for ibm-granite/grani...**

Base model ibm-granite/granite-3...


 Finetuned ibm-granite/grani...


 Finetuned (12) this model


Adapters 2 models


Finetunes 4 models


Quantizations 36 models


 **Spaces using ibm-granite/gr...** 47


 Divyashree1326/healai


 coreyqin/ibm-granite-granite-3.2-2b-...


 tom0072012/ibm-granite-granite-3.2-...


 mohamedriazkhanm/Citizen_AI


 Thiruselvam/HealthAI


 23ucs565/smartcity


 PavithraBCA/EduTutorAI-NM2025TMI...

 V-06/aicitizen

 23ucs589/Aismartsdlc

 rathil123/smart_city123

 23ucs576/edu-ai

 avinxsh77/Health_AI + 35 Spaces

11.GOOGLE COLAB PAGE IS ON OPEN

```
Health AI
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all Connect T4

import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# Load model and tokenizer
model_name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

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 recommendations."
    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."
    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=10)

            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)

    app.launch(share=True)
```

13.GITHUB PAGE IS ON OPEN

chrish3024 / IBM-Project

Go Search Add + Open Run Close

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

IBM-Project / health_ai.py

Go to file

chrish3024 Add files via upload 09b6c27 · 3 days ago

102 lines (82 loc) · 4.45 KB

Code Blame

Raw Copy Download Edit View

```
1  # -*- coding: utf-8 -*-
2  """Health AI.ipynb
3
4  Automatically generated by Colab.
5
6  Original file is located at
7  https://colab.research.google.com/drive/17GctBXQvdf2e3FgvSZMhw-5yv3o
8  """
9
10 !pip install transformers torch gradio -q
11
12 import gradio as gr
13 import torch
14 from transformers import AutoTokenizer, AutoModelForCausalLM
15
16 # Load model and tokenizer
17 model_name = "ibm-granite/granite-3.2-2b-instruct"
18 tokenizer = AutoTokenizer.from_pretrained(model_name)
19 model = AutoModelForCausalLM.from_pretrained(
20     model_name,
21     torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
22     device_map="auto" if torch.cuda.is_available() else None
23 )
24
25 if tokenizer.pad_token is None:
26     tokenizer.pad_token = tokenizer.eos_token
27
28 def generate_response(prompt, max_length=1024):
29     inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max
30
31     if torch.cuda.is_available():
32         inputs = {k: v.to(model.device) for k, v in inputs.items()}
33
34     with torch.no_grad():
35         outputs = model.generate(
36             **inputs,
37             max_length=max_length,
38             temperature=0.7,
39             do_sample=True,
40             pad_token_id=tokenizer.eos_token_id
41         )
42
43     response = tokenizer.decode(outputs[0], skip_special_tokens=True)
44     response = response.replace(prompt, "").strip()
45     return response
46
47 def disease_prediction(symptoms):
48     prompt = f"Based on the following symptoms, provide possible medical
49     return generate_response(prompt, max_length=1200)
50
51 def treatment_plan(condition, age, gender, medical_history):
52     prompt = f"Generate personalized treatment suggestions for the follo
53     return generate_response(prompt, max_length=1200)
54
55 # Create Gradio interface
56 with gr.Blocks() as app:
57     gr.Markdown("# Medical AI Assistant")
58     gr.Markdown("**Disclaimer: This is for informational purposes only.")
59
60     with gr.Tabs():
61         with gr.TabItem("Disease Prediction"):
62             with gr.Row():
63                 with gr.Column():
64                     symptoms_input = gr.Textbox(
65                         label="Enter Symptoms",
66                         placeholder="e.g., fever, headache, cough, fatig
67                         lines=4
68                     )
69                     predict_btn = gr.Button("Analyze Symptoms")
70
71                 with gr.Column():
72                     prediction_output = gr.Textbox(label="Possible Condi
73
74             predict_btn.click(disease_prediction, inputs=symptoms_input,
75
76         with gr.TabItem("Treatment Plans"):
77             with gr.Row():
78                 with gr.Column():
79                     condition_input = gr.Textbox(
80                         label="Medical Condition",
81                         placeholder="e.g., diabetes, hypertension, migra
```

14.CONCLUSION

- The HealthAI project effectively demonstrates the potential of AI in revolutionizing healthcare assistance. By integrating IBM's Granite language model, the platform enables users to receive personalized health insights through Patient Chat, Disease Prediction, Treatment Plan Generation, and Health Analytics, making healthcare information more accessible.
- Utilizing IBM Watson Machine Learning, the application ensures accurate health question answering, detailed disease prediction, personalized treatment recommendations, and insightful health trend analysis. The structured development process—spanning model selection, core feature implementation, backend and frontend development, and deployment—led to the creation of an interactive, user-friendly platform.
- Built with Streamlit, HealthAI facilitates seamless visualization of health data and AI-generated insights, ensuring an efficient and responsive experience. This project highlights how targeted AI models and a well-structured framework can enhance healthcare accessibility. With future scalability in mind, HealthAI has the potential to expand its capabilities, incorporating more advanced diagnostics and broader medical applications.