Sustainable Smart City Assistant Using IBM Granite LLM

Team Members

- R Shanthini
- L Praveena
- S Kamali
- I Sabasthein Dyano

1.INTRODUCTION

The Sustainable Smart City Assistant using IBM Granite LLM is a generative AI-based project designed to support urban sustainability, governance, and citizen engagement by leveraging IBM Granite models from Hugging Face. The system provides quick tools such as a City Health Dashboard, citizen feedback collection, document summarization, and eco-friendly tips, thereby enabling smarter decision-making for city management. Implemented on Google Colab for easy deployment and powered by lightweight yet efficient Granite models, the project demonstrates how AI can enhance urban living through accessibility, scalability, and real-time insights, ultimately contributing to the vision of building smarter and more sustainable cities.

2.PROJECT OVERVIEW

The Sustainable Smart City Assistant leverages IBM Granite LLM models from Hugging Face to provide AI-powered tools for sustainable urban governance, citizen engagement, and environmental awareness. It supports features such as a City Health Dashboard, citizen feedback collection, document summarization, and eco-friendly tips. The project is designed to be lightweight and runs on Google Colab with Gradio UI for easy deployment and interaction.

3.ARCHITECTURE

1. Frontend (User Interface)

Built using Gradio to provide an interactive web app.

Users can input queries, upload documents, and view dashboards.

2. Backend (Model & Processing Layer)

Powered by IBM Granite-3.2-2B-Instruct model from Hugging Face.

Handles text generation, summarization, and citizen engagement tasks.

3. Execution Environment

Deployed in Google Colab using T4 GPU for efficiency.

Uses Python libraries: transformers, torch, gradio, PyPDF2.

4. Version Control & Deployment

Source code is stored and shared via GitHub repository.

4.SETUP INSTRUCTIONS

1. Open Google Colab → create a new notebook.

2. Change runtime to T4 GPU (Runtime → Change Runtime Type → GPU).

3. Install dependencies:

!pip install transformers torch gradio PyPDF2 -q

4. Load IBM Granite model from Hugging Face (granite-3.2-2b-instruct).

5. Run the provided application code (from Drive/GitHub).

6. A Gradio link will be generated → open it to access the assistant.

5.AUTHENTICATION

Hugging Face: Requires account login and access to IBM Granite model.

Google Colab: Uses your Google account for notebook execution.

GitHub: Requires login for uploading/sharing project repository.

6.USER INTERFACE

The UI is powered by Gradio:

Input box for user queries.

File upload option for document summarization.

Interactive dashboard for city health indicators.

Generated outputs displayed in real-time in another tab.

7.TESTING

Model Testing: Queries tested for sustainability-related responses.

Functional Testing: Verified file uploads, summaries, and feedback processing.

UI Testing: Ensured Gradio interface is responsive in browser.

Performance Testing: Validated execution speed on T4 GPU.

8.KNOW ISSUES

Dependency Installation Errors in Colab if versions mismatch.

Model Loading Delay when first initializing Granite LLM.

Limited Offline Usage since app relies on Colab cloud runtime.

GPU Quotas in Colab may restrict extended usage.

9.FUTURE ENHANCEMENTS

Integration with real-time IoT data for city health monitoring.

Deployment on cloud platforms (AWS, IBM Cloud, GCP) for scalability.

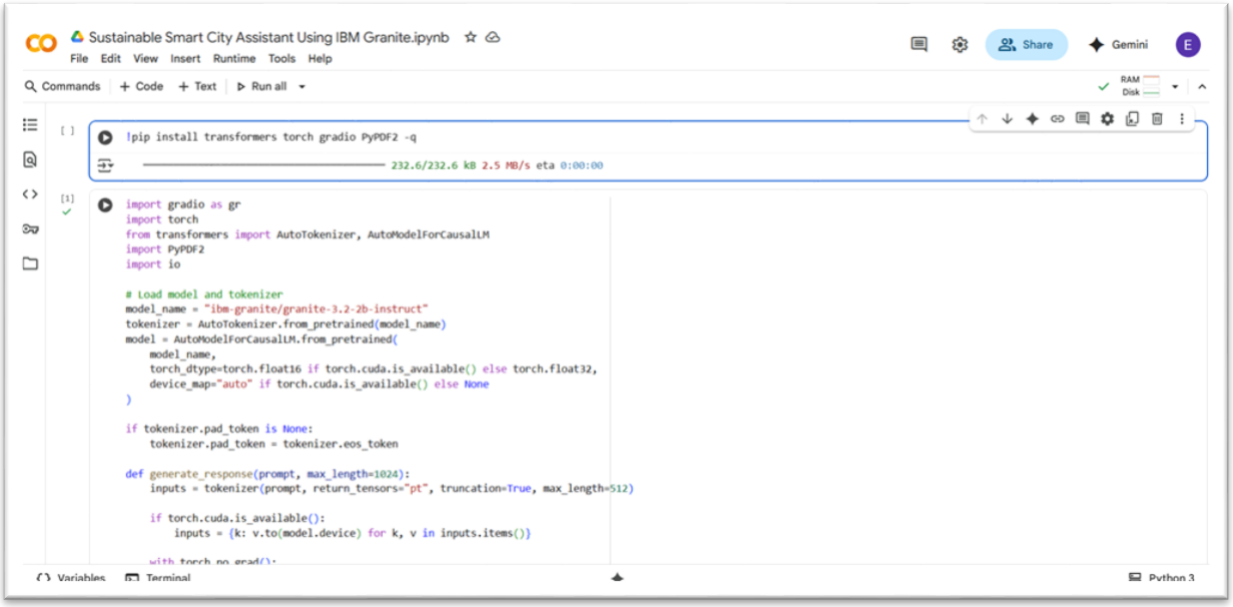
Enhanced multi-language support for diverse citizen engagement.

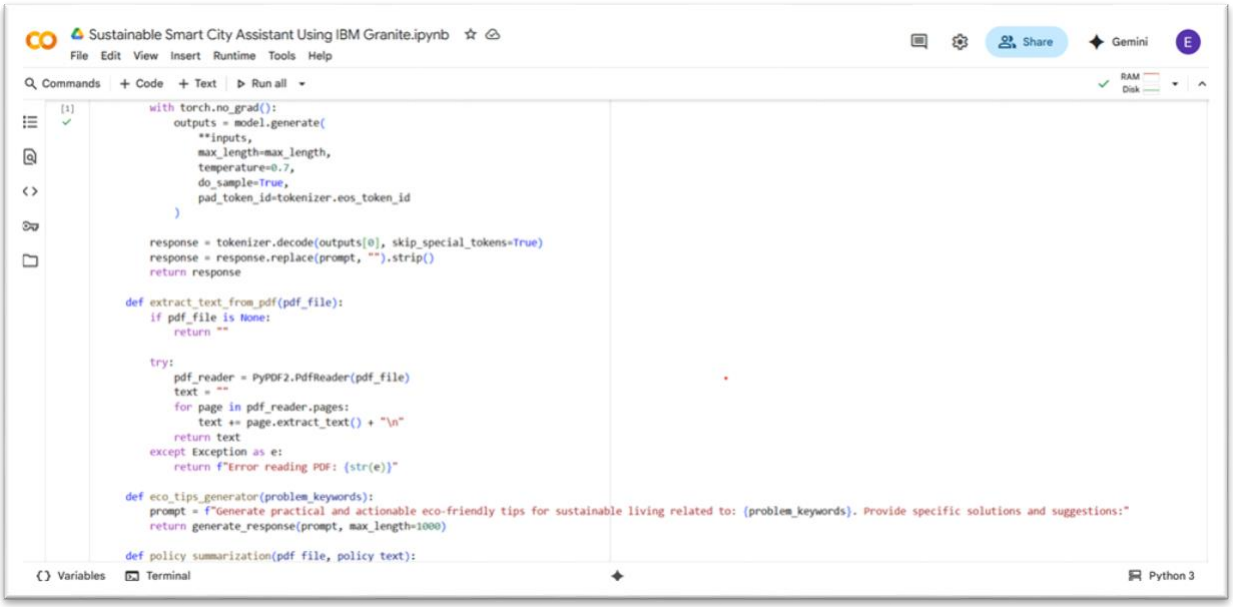
Improved UI/UX with dashboards (charts, visual analytics).

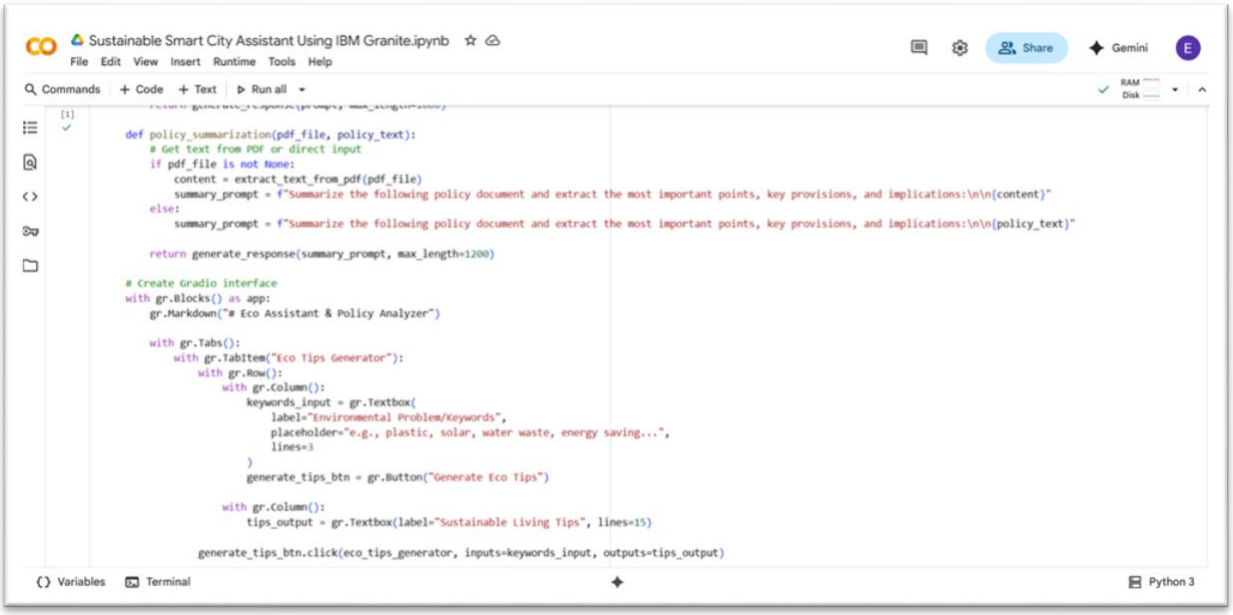
Adding voice-based interaction for accessibility.

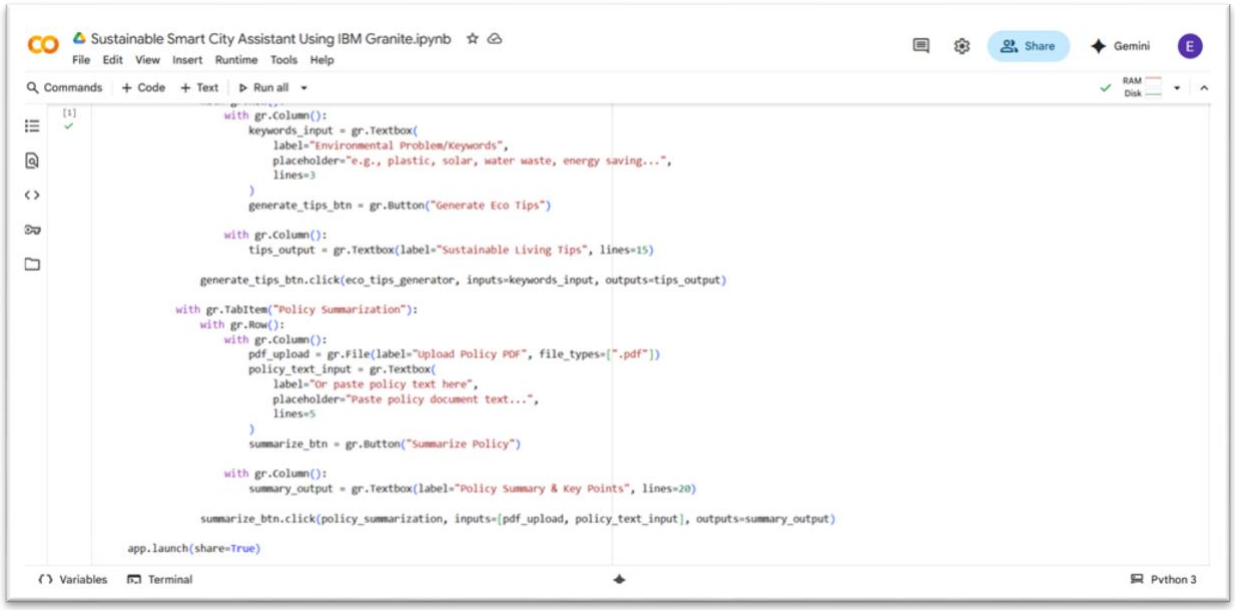
Extending use cases to waste management, traffic optimization, and energy savings.

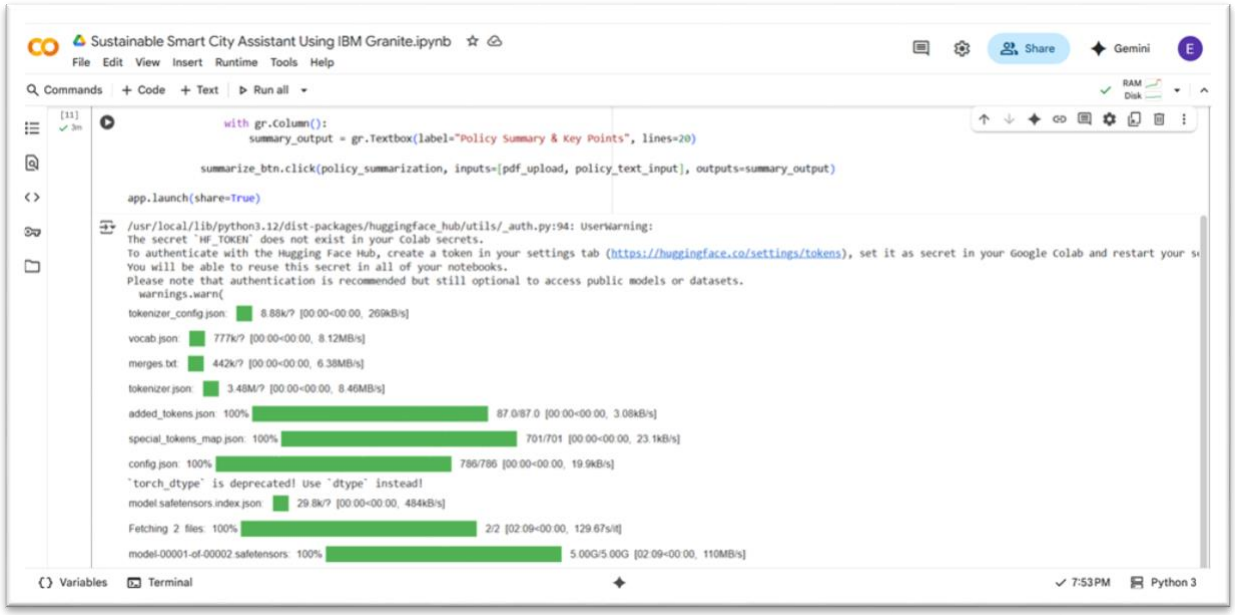
10.PROJECT SCREENSHORT

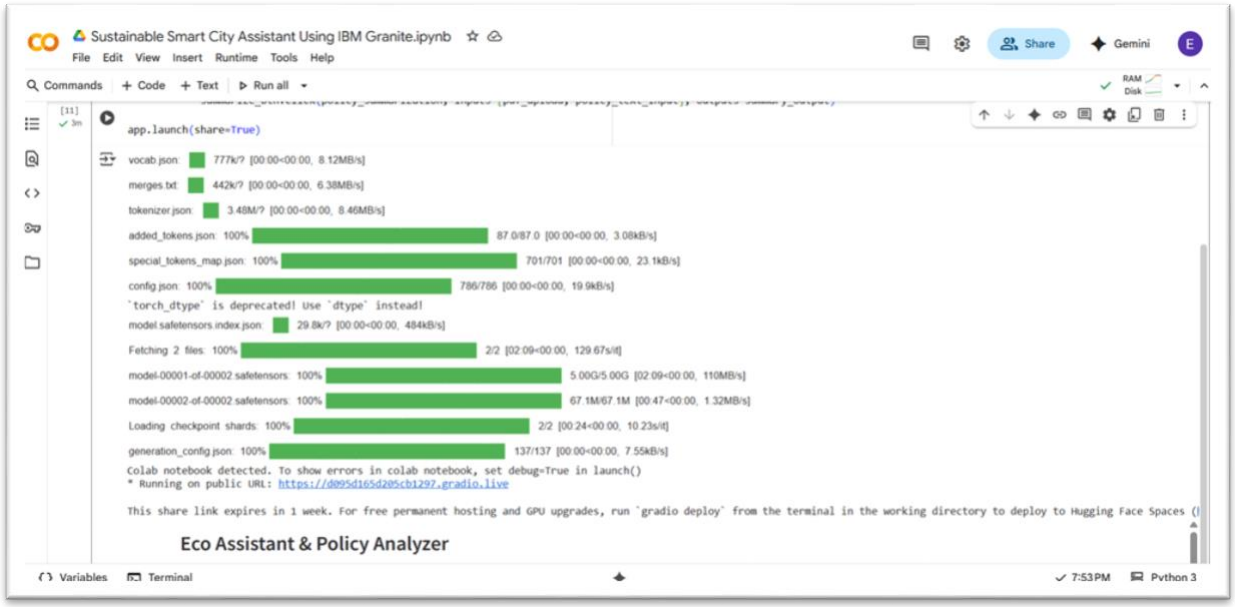


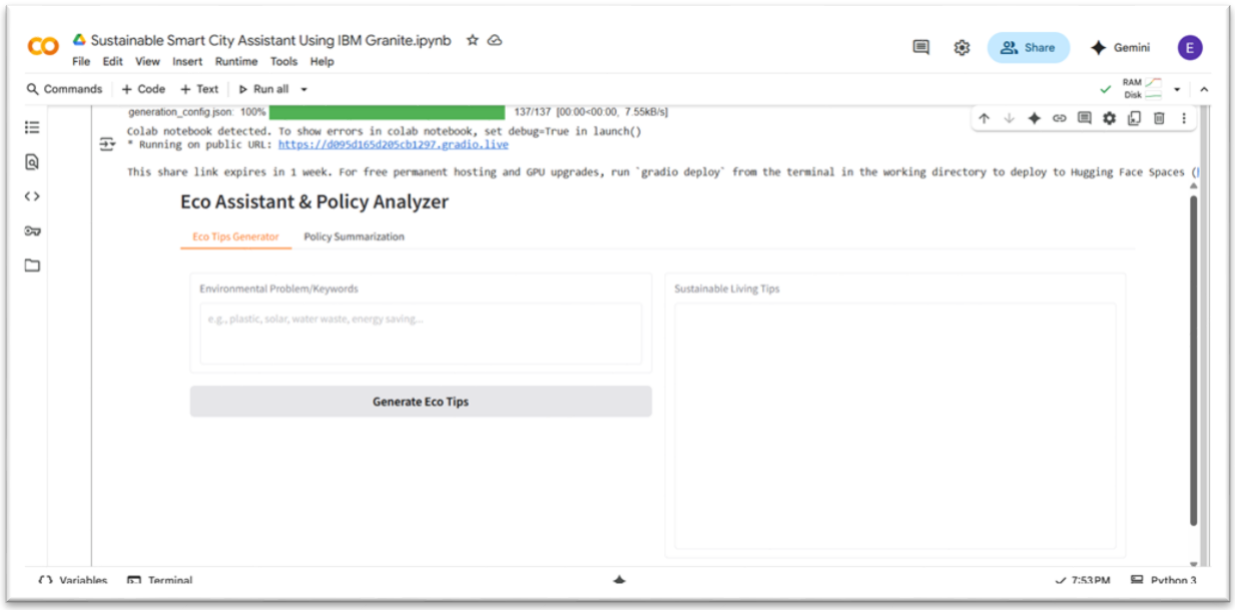

Sustainable Smart City Assistant Using IBM Granite.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM
Disk
[1]
with torch.no_grad():
outputs = model.generate(
**inputs,
max_length=max_length,
temperature=0.7,
do_sample=True,
pad_token_id=tokenizer.eos_token_id
)
response = tokenizer.decode(outputs[0], skip_special_tokens=True)
response = response.replace(prompt, "").strip()
return response
def extract_text_from_pdf(pdf_file):
if pdf_file is None:
return ""
try:
pdf_reader = PyPDF2.PdfReader(pdf_file)
text = ""
for page in pdf_reader.pages:
text += page.extract_text() + "\n"
return text
except Exception as e:
return f"Error reading PDF: {str(e)}"
def eco_tips_generator(problem_keywords):
prompt = f"Generate practical and actionable eco-friendly tips for sustainable living related to: {problem_keywords}. Provide specific solutions and suggestions:"
return generate_response(prompt, max_length=1000)
def policy summarization(pdf file, policy text):
Variables
Terminal
Python 3


Sustainable Smart City Assistant Using IBM Granite.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM
Disk
[1]
def policy_summarization(pdf_file, policy_text):
# Get text from PDF or direct input
if pdf_file is not None:
content = extract_text_from_pdf(pdf_file)
summary_prompt = f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{content}"
else:
summary_prompt = f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{policy_text}"
return generate_response(summary_prompt, max_length=1200)
Create Gradio interface
with gr.Blocks() as app:
gr.Markdown("# Eco Assistant & Policy Analyzer")
with gr.Tabs():
with gr.TabItem("Eco Tips Generator"):
with gr.Row():
with gr.Column():
keywords_input = gr.Textbox(
label="Environmental Problem/Keywords",
placeholder="e.g., plastic, solar, water waste, energy saving...",
lines=3
)
generate_tips_btn = gr.Button("Generate Eco Tips")
with gr.Column():
tips_output = gr.Textbox(label="Sustainable Living Tips", lines=15)
generate_tips_btn.click(eco_tips_generator, inputs=keywords_input, outputs=tips_output)
Variables
Terminal
Python 3


Sustainable Smart City Assistant Using IBM Granite.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
RAM
Disk
[1]
with gr.Column():
keywords_input = gr.Textbox(
label="Environmental Problem/Keywords",
placeholder="e.g., plastic, solar, water waste, energy saving...",
lines=3
)
generate_tips_btn = gr.Button("Generate Eco Tips")
with gr.Column():
tips_output = gr.Textbox(label="Sustainable Living Tips", lines=15)
generate_tips_btn.click(eco_tips_generator, inputs=keywords_input, outputs=tips_output)
with gr.TabItem("Policy Summarization"):
with gr.Row():
with gr.Column():
pdf_upload = gr.File(label="Upload Policy PDF", file_types=[".pdf"])
policy_text_input = gr.Textbox(
label="Or paste policy text here",
placeholder="Paste policy document text...",
lines=5
)
summarize_btn = gr.Button("Summarize Policy")
with gr.Column():
summary_output = gr.Textbox(label="Policy Summary & Key Points", lines=20)
summarize_btn.click(policy_summarization, inputs=[pdf_upload, policy_text_input], outputs=summary_output)
app.launch(share=True)
Variables
Terminal
Python 3


Sustainable Smart City Assistant Using IBM Granite.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
RAM
Disk
[11]
3m
with gr.Column():
summary_output = gr.Textbox(label="Policy Summary & Key Points", lines=20)
summarize_btn.click(policy_summarization, inputs=[pdf_upload, policy_text_input], outputs=summary_output)
app.launch(share=True)
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
tokenizer_config.json: 8.88k/? [00:00<00:00, 269kB/s]
vocab.json: 777k/? [00:00<00:00, 8.12MB/s]
merges.txt: 442k/? [00:00<00:00, 6.38MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 8.46MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 3.08kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 23.1kB/s]
config.json: 100% 786/786 [00:00<00:00, 19.9kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 484kB/s]
Fetching 2 files: 100% 2/2 [02:09<00:00, 129.67s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [02:09<00:00, 110MB/s]
Variables
Terminal
7:53 PM
Python 3


Sustainable Smart City Assistant Using IBM Granite.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
app.launch(share=True)
vocab.json: 777k/? [00:00<00:00, 8.12MB/s]
merges.txt: 442k/? [00:00<00:00, 6.38MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 8.46MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 3.08kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 23.1kB/s]
config.json: 100% 786/786 [00:00<00:00, 19.9kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 484kB/s]
Fetching 2 files: 100% 2/2 [02:09<00:00, 129.67s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [02:09<00:00, 110MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:47<00:00, 1.32MB/s]
Loading checkpoint shards: 100% 2/2 [00:24<00:00, 10.23s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 7.55kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://d095d165d205cb1297.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
Eco Assistant & Policy Analyzer
Variables Terminal
7:53PM Python 3


Sustainable Smart City Assistant Using IBM Granite.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
generation_config.json: 100% 137/137 [00:00<00:00, 7.55kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://d095d165d205cb1297.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
Eco Assistant & Policy Analyzer
Eco Tips Generator
Policy Summarization
Environmental Problem/Keywords
e.g., plastic, solar, water waste, energy saving...
Sustainable Living Tips
Generate Eco Tips
Variables Terminal
7:53PM Python 3

11.CONCLUSION

The sustainable smart city assistant using IMB granite LLM project was successfully done by: 3rd BCA 1st Section, Elakkiya.S , Durga.P , Gloriya Jeni.A , Harini.A

Demo video link:

https://drive.google.com/file/d/1va_QIPfonIHLTrnb7IqF162YTIO3AZzR/view?usp=sharing