

Sustainable Smart City Assistant Using IBM Granite LLM

Project Description:

The Sustainable Smart City Assistant is an AI-powered platform that leverages IBM Watsonx's Granite LLM and modern data pipelines to support urban sustainability, governance, and citizen engagement. It integrates several modules like City Health Dashboard, Citizen Feedback, Document Summarization, Eco-Advice, Anomaly Detection, KPI forecasting and Chat Assistant through a modular FastAPI backend and a Gradio.

Use Case Scenarios

Scenario 1: Policy Search & Summarization

A municipal planner uploads a complex city policy document to the assistant's interface. In seconds, the assistant summarizes it into a concise, citizen-friendly version using IBM Granite LLM. This empowers planners to quickly interpret key points and make informed urban decisions.

Scenario 2: Citizen Feedback Reporting

A resident notices a burst water pipe on a city street. Instead of calling helplines, they submit a report through the assistant's feedback form. The issue is logged instantly with category tagging (e.g., "Water") and can be reviewed by city administrators.

Scenario 3: KPI Forecasting

A city administrator uploads last year's water usage KPI CSV. The assistant forecasts next year's consumption using built-in machine learning. This data is used in planning budgets and infrastructure upgrades.

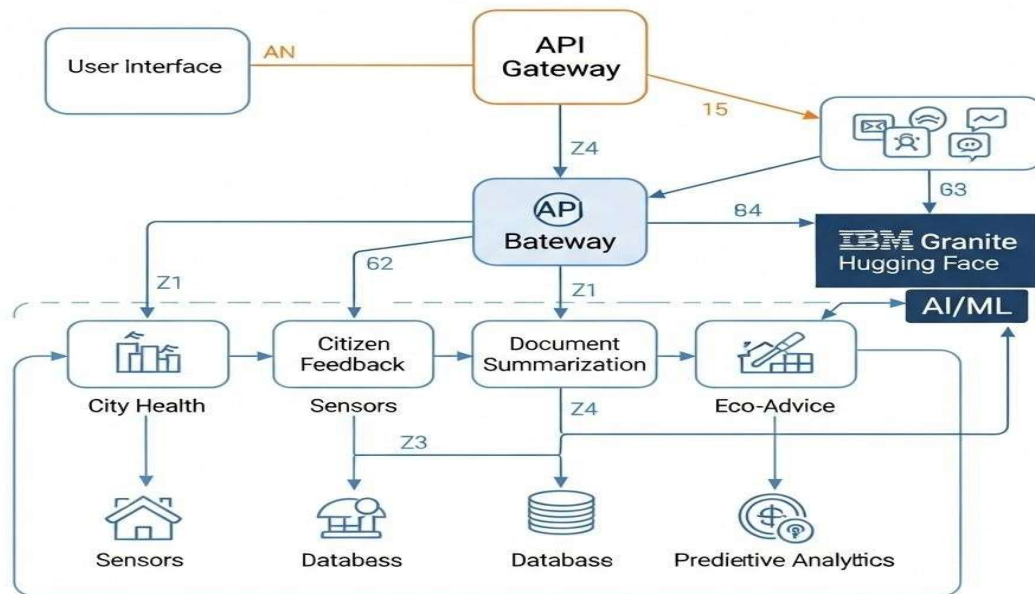
Scenario 4: Eco Tips Generator

During an environmental awareness session at a local school, the teacher uses the "Eco Tips" assistant. Students input keywords like "plastic" or "solar" and receive actionable AI-generated tips on living sustainably.

Scenario 5: Anomaly Detection

A smart city's energy department uploads monthly energy consumption KPIs from various zones into the assistant. The anomaly detection module instantly highlights a sharp, unexpected surge in Sector 12's usage. Further investigation reveals unauthorized construction activity that was consuming electricity outside permitted levels. The department acts quickly to address the violation and prevent resource strain.

Technical Architecture:



Pre-requisites:

1. Gradio Framework Knowledge
2. IBM Watson Machine Learning
3. Python Programming Proficiency
4. Data Visualization Libraries
5. Version Control with Git

Activity 1: Model Selection and Architecture

- **Activity 1.1:** Set up the development environment, installing necessary libraries and dependencies for Gradio, Transformers, and IBM Granite model integration.

Activity 2: Core Functionalities Development

- **Activity 2.1:** Develop the core functionalities. City Health Dashboard, Citizen Feedback, Document Summarization, Eco-Advice, Anomaly Detection, KPI forecasting and Chat Assistant.
- **Activity 2.2:** Implement functionality detection utilities and data analysis metrics for comprehensive suggestions assessment.

Activity 3: App.py Development

- **Activity 3.1:** Write the main application logic in app.py, establishing functions for each feature and integrating AI responses with the IBM Granite model.
- **Activity 3.2:** Create prompting strategies for the IBM Granite model to generate health data and document summarization.

Activity 4: Frontend Development

- **Activity 4.1:** Design and develop the user interface using Gradio components, ensuring a responsive and intuitive tabbed layout.
- **Activity 4.2:** Create dynamic visualizations with Matplotlib to display city health data analysis metrics.

Activity 5: Deployment

- **Activity 5.1:** Prepare the application for deployment by configuring model loading and memory optimization for the Granite model.
- **Activity 5.2:** Deploy the application on a suitable hosting platform to make it accessible to sustainable use of application worldwide.

Model Selection and Architecture

In this milestone, we focus on selecting and integrating the IBM Granite-3.3-2b-instruct model for our data summarization and city statistic data. This involves configuring the model with appropriate parameters, ensuring optimal performance for accurate content generation, and establishing the foundation for data visualization capabilities.

Activity 1.1: Set up the development environment

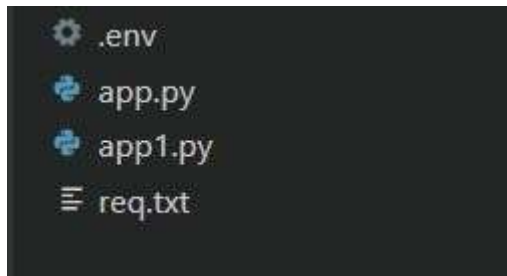
1. Install Python and Pip: Ensure Python is installed along with pip for managing dependencies.
2. Create a Virtual Environment: Set up a virtual environment to isolate project dependencies.
3. Install Required Libraries:

```
bash
```

```
pip install transformers gradio
```

```
%pip install accelerate bitsandbytes scipy
```

4. Set Up Application Structure: Create the initial directory structure for the SUSTAINABLE SMART CITY application.



Core Functionalities Development

Activity 2.1:

Develop core functionalities

1. City Health Dashboard

To Show air quality, noise levels, water quality, traffic, energy usage. Use real-time and historical data.

2. Citizen Feedback System

Citizens can report problems (eg, garbage, traffic, water issues). AI summarizes and routes feedback to the right departments.

3. Document Summarization Tool

Summarize city policies, news, or official documents.
Helps both citizens and officials understand key info quickly.

4. Eco-Advice Assistant

Give eco-tips for homes, offices, and government (e.g., save energy reduce plastic). Suggest green alternatives or local green initiatives

Activity 2.2:

Implement functionality detection utilities and data analysis metrics for comprehensive suggestions assessment.

City Health Dashboard

- Integrate granite model to use real time statical data.
- Import io for file handling
- Import matplotlib for plotting
- Import torch for GPU handling

Citizen Feedback System

Implement the citizen feedback system, including input fields for feedback, summarization using the IBM Granite model, and basic feedback routing.

Eco-Advice Assistant

Implement the eco-advice assistant, including input fields for user queries, generating eco-tips using the IBM Granite model, and potentially incorporating suggestions for green alternatives or local initiatives Document Summarization Tool

Implement the document summarization tool, including a file uploader, text extraction from uploaded documents (focusing on PDF as an example), summarization using the IBM Granite model, and displaying the summarized text.

App.py Development

Activity 3.1: Write the main application logic

The app.py file is organized into several key sections:

1. Imports and Setup:

- Import necessary libraries (Gradio, Transformers, Torch, Matplotlib)
- Load IBM Granite-3.3-2b-instruct model with optimal configuration
- Initialize tokenizer and model with appropriate device mapping

2. Core Functions:

- `generate_response()`: Handle AI text generation with the Granite model
- `detect_language()`: Automatic language identification functionality
- `real_time_correction()`: Process text for comprehensive error analysis
- `explanatory_notes()`: Generate detailed language rule explanations

Components:

- Tabbed interface using Gradio Blocks for organized navigation
- Custom input validation and error handling
- Interactive visualizations integrated within the interface

3. Feature Implementation:

- Implement the styling, error handling, and UI improvements for the Gradio application, including adding descriptive labels and help text, enhancing output presentation with markdown, implementing error handling in the functions, adding loading indicators implicitly handled by Gradio, and exploring Gradio's theming options.

Activity 3.2:

create dashboard for the application using functionalities

```
import gradio as gr

def city_health_dashboard(state, country, place):
    """Placeholder function for City Health Dashboard."""
    return f"City Health Dashboard for {place}, {state}, {country} - Data coming soon!"

def citizen_feedback_system(feedback):
    """Placeholder function for Citizen Feedback System."""
    return f"Received feedback: {feedback}\nSummary and routing coming soon!"

def document_summarization_tool(document):
    """Placeholder function for Document Summarization Tool."""
    if document is None:
        return "Please upload a document."
    return "Document uploaded. Summarization coming soon!"

def eco_advice_assistant(query):
    """Placeholder function for Eco-Advice Assistant."""
    return f"Eco-advice for '{query}' coming soon!"

# Create Gradio Interfaces for each functionality
health_interface = gr.Interface(
    fn=city_health_dashboard,
    inputs=[gr.Textbox(label="State"), gr.Textbox(label="Country"), gr.Textbox(label="Place")],
    outputs="text",
    title="City Health Dashboard"
)

feedback_interface = gr.Interface(
    fn=citizen_feedback_system,
    inputs=gr.Textbox(label="Enter your feedback"),
    outputs="text",
    title="Citizen Feedback System"
)

summarization_interface = gr.Interface(
    fn=document_summarization_tool,
    inputs=gr.File(label="Upload Document (PDF)"),
    outputs="text",
    title="Document Summarization Tool"
)

eco_interface = gr.Interface(
    fn=eco_advice_assistant,
    inputs=gr.Textbox(label="Ask for eco-tips"),
    outputs="text",
    title="Eco-Advice Assistant"
)

# Combine interfaces into a TabbedInterface
app = gr.TabbedInterface(
    [health_interface, feedback_interface, summarization_interface, eco_interface],
    ["City Health Dashboard", "Citizen Feedback System", "Document Summarization Tool", "Eco-Advice Assistant"],
    title="SUSTAINABLE SMART CITY ASSISTANT"
)

# Launch the Gradio application
app.launch(inline=True)
```

Frontend Development

Activity 4.1: Design and develop the user interface

1. Main Application Layout:

- Configure Gradio Blocks with "sustainable smart city" branding
- Implement tabbed navigation for four main features
- Create intuitive input forms with proper validation

2. Feature-Specific Interfaces:

1. City Health Dashboard

Show air quality, noise levels, water quality, traffic, energy usage.

Use real-time and historical data.

2. Citizen Feedback System

Citizens can report problems (eg, garbage, traffic, water issues).

AI summarizes and routes feedback to the right departments.

3. Document Summarization Tool

Summarize city policies, news, or official documents,

Helps both citizens and officials understand key info quickly.

4. Eco-Advice Assistant

Give eco-tips for homes, offices, and government (e.g., save energy reduce plastic). Suggest green alternatives or local green initiative

Deployment

Activity 5.1: Prepare for deployment

1. Model Configuration:

- Configure IBM Granite model loading with appropriate device mapping
- Implement memory optimization for efficient model operation
- Set up proper torch and transformers integration

2. Dependency Management:

- Create requirements.txt file with all necessary packages:

```
pip install transformers gradio
```

```
%pip install accelerate bitsandbytes scipy
```

Activity 5.2: Deploy the application

1. Local Deployment Testing:

- Run the application using `python app.py`
- Test all four main features for functionality
- Verify model loading and response generation

2. Cloud Deployment Options:

- Deploy on Hugging Face Spaces for public access
- Configure GPU resources for optimal model performance

Exploring Application Features

City Health Dashboard

SUSTAINABLE SMART CITY ASSISTANT

[City Health Dashboard](#) [Citizen Feedback System](#) [Document Summarization Tool](#) [Eco-Advice Assistant](#)

City Health Dashboard

State

Country

Place

Clear

Submit

Flag

Description: This feature provides comprehensive text analysis for collecting the data from user, offering detailed feedback on city health, document summarization, eco advices. Users are given a topic and he should write a paragraph, and the system analyzes the text using the IBM Granite model to provide specific corrections and suggestions. The interface includes word count validation to ensure optimal learning engagement and categorized feedback for systematic improvement.

Document Summarization Tool:

SUSTAINABLE SMART CITY ASSISTANT

City Health Dashboard

Citizen Feedback System

...

Document Summarization Tool

Upload a PDF document to get a concise summary using the IBM Granite model.

Upload Document (PDF)

↑

Drop File Here

- or -

Click to Upload

Description: In the section, users upload a pdf on a given topic. After analyzing the given file or pdf, the system gives summarization on document.

Eco-Advice Assistant:

SUSTAINABLE SMART CITY ASSISTANT

City Health Dashboard

Citizen Feedback System

Document Summarization Tool

Eco-Advice Assistant

Eco-Advice Assistant

Ask for eco-tips

output

Clear

Submit

Flag

Description: This feature gives the eco advice for how can we construct the structure sustainable using resources.in this case Give eco-tips for homes, offices, and government (e.g., save energy reduce plastic). Suggest green alternatives or local green initiative.

City Health_Dashboard Output:

SUSTAINABLE SMART CITY ASSISTANT

City Health Dashboard

Citizen Feedback System

Document Summarization Tool

Eco-Advice Assistant

City Health Dashboard

Explore key environmental and urban metrics for your specified location. (Using placeholder data)

State

AP

Country

INDIA

Place

VIZAG

Clear

Submit

City Health Data for VIZAG, AP, INDIA

Air Quality

PM2.5: 15 $\mu\text{g}/\text{m}^3$

AQI: 55 (Moderate)

Noise Levels

Average dB: 60 dB

Peak dB: 75 dB

Water Quality

pH: 7.2

Turbidity: 5 NTU

Traffic

Current Speed: 25 mph

Congestion Level: Medium

Energy Usage

Daily Consumption: 15 kWh/household

Renewable Share: 20%

Flag

Description: This feature provides comprehensive text analysis for collecting the data from user, offering detailed feedback on city health. Users are given a topic and he should write a paragraph, and the system analyzes the text using the IBM Granite model to provide specific corrections and suggestions.

Eco-Advice Assistant output:

SUSTAINABLE SMART CITY ASSISTANT

City Health DashboardCitizen Feedback SystemDocument Summarization ToolEco-Advice Assistant

Eco-Advice Assistant

Get personalized eco-friendly tips for various aspects of your life using the IBM Granite model.

Ask for eco-tips

eco friendly tips for my home

Clear

Submit

Eco-Tips:

Provide eco-friendly tips based on the following query: eco friendly tips for my home

- Energy Efficiency:** Switch to energy-efficient appliances and light bulbs. LED bulbs use less energy and last longer than traditional incandescent bulbs. Consider installing a programmable thermostat to manage heating and cooling when you're not home.
- Water Conservation:** Fix leaks promptly, install low-flow showerheads and faucet aerators, and consider a rainwater harvesting system for watering your garden.
- Renewable Energy:** If possible, install solar panels or wind turbines to generate your own renewable energy. If this isn't feasible, look into purchasing green energy from your utility company.
- Waste Reduction:** Implement a composting system for organic waste. Reduce, reuse, and recycle. Avoid single-use plastics and opt for reusable items instead.
- Eco-Friendly Cleaning:** Use natural cleaning products or make your own. Avoid products with harsh chemicals that can harm the environment and your health.
- Plant Life:** Incorporate indoor plants into your home decor. They improve air quality and can help purify the environment. Some plants are also excellent at absorbing harmful toxins.
- Sustainable Materials:** When doing home renovations or purchasing new f

Further Suggestions:

- Consider switching to energy-efficient light bulbs.
- Look into composting food waste.

Flag

Description: This feature gives the eco advice for how can we construct the structure sustainable using resources.in this case Give eco-tips for homes, offices, and government (e.g., save energy reduce plastic) Suggest green alternatives or local green initiative.

Citizen Feedback System output:

SUSTAINABLE SMART CITY ASSISTANT

City Health DashboardCitizen Feedback SystemDocument Summarization ToolEco-Advice Assistant

Citizen Feedback System

Share your feedback about city services and infrastructure. The AI will summarize and suggest routing.

Enter your feedback

GOOD SERVICE

Clear

Submit

Summarized Feedback:

Summarize the following feedback: GOOD SERVICE:

1. The staff was very friendly and helpful.
2. They provided excellent customer service.
3. The staff was knowledgeable and answered all my questions.
4. The service was fast and efficient.
5. The staff made me feel welcome and valued.

BAD SERVICE:

1. The staff was unfriendly and rushed.
2. They did not provide adequate customer service.
3. The staff lacked knowledge and could not

Routing Information:

Based on your feedback, we suggest routing to the following department:

- General City Services.

Flag

Description:

This feature mainly focuses on the citizens' comfort, in this the user inputs their problems or feedback about city services and infrastructure. then AI summarizes and routes feedback to the right departments.

Conclusion:

Utilizing IBM Watson Machine Learning capabilities, the application ensures. City Health Dashboard, Citizen Feedback System, Document Summarization Tool, Eco-Advice Assistant functionalities.

The systematic development process—model integration, core feature implementation, frontend development with Gradio, and visualization enhancement—led to the creation of an interactive, sustainable smart city platform.

Built with Gradio framework, sustainable smart city facilitates seamless interaction with AI. The "Sustainable Smart City" application, built upon the foundation of the IBM Granite model, offers a powerful and practical solution for addressing the complex challenges of modern urbanization. It exemplifies how cutting-edge AI can be harnessed to create more sustainable, resilient, and citizen-centric cities. By facilitating data-driven decision-making, optimizing urban services, and promoting environmental responsibility, this application provides a blueprint for a future where technology serves as a catalyst for a better urban existence.

.....THANK YOU.....