Smart Energy Consumption Recommender using FAISS

Overview

The Smart Energy Consumption Recommender is an AI-powered system designed to analyze electricity usage patterns and provide personalized recommendations for energy saving.

By leveraging FAISS (Facebook AI Similarity Search), LLMs (Large Language Models), and RAG (Retrieval-Augmented Generation), the system can efficiently search through historical usage data and generate optimal appliance usage schedules.

Features

- Energy Usage Analysis Processes smart home energy consumption datasets.
- FAISS Search Engine Enables fast similarity searches across historical energy data.
- LLM Integration Uses GPT, Falcon, FLAN-T5, and other models for intelligent recommendations.
- RAG Implementation Combines retrieval with generative AI for context-aware results.
- Multiple Fine-Tuning Experiments Includes models fine-tuned on energy datasets.
- Recommendation Generation Suggests optimal appliance schedules and strategies to reduce consumption.

Project Structure

Smart-Energy-Consumption-Recommender-using-FAISS

- preprocessing.ipynb
- smart_home_dataset.csv
- Facebook_opt_RAG/
- Fine Tuning Using LLMs and FAISS Search/
- Fine Tuning Using LLMs, FAISS Search and RAG/
- Fine Tuning Using LLMs/
- GPT-2 with Faiss indexing/
- LICENSE
- README.md

Installation

1. Clone the repository:

```
git clone
```

https://github.com/yourusername/Smart-Energy-Consumption-Recommender-using-FAISS.git

cd Smart-Energy-Consumption-Recommender-using-FAISS

2. Install dependencies:

pip install -r requirements.txt

3. Ensure you have Python 3.8+ and a GPU-enabled environment for LLM training.

Dataset

- File: smart_home_dataset.csv
- Description: Contains smart home electricity usage data with timestamps and appliance consumption.
- Preprocessing: Run preprocessing.ipynb to clean and prepare the dataset.

Models Used

- FAISS for similarity search
- GPT-2 / DistilGPT-2
- DeepSeek
- Falcon RW 1B
- Google FLAN-T5 Large
- OPT 1.3B

How It Works

1. Data Preprocessing Cleans raw energy data.
2. FAISS Indexing Creates vector embeddings of usage patterns.
3. Query & Search Finds similar past patterns.
4. LLM Recommendation Uses retrieved patterns to generate advice.
5. Output Presents a schedule or energy-saving strategy.

Example Use Cases

- Suggests the best time to run appliances to save costs.
- Detects unusual consumption patterns.
- Recommends energy-efficient routines.

Technologies

- Python
- FAISS
- Hugging Face Transformers
- Pandas / NumPy
- Jupyter Notebooks

License