CO_2 Impact of LLM

Team 1

Héloïse Roméo | Natalia Gérard | Paul Ranc
Valentin Rech | Alexis Denneulin | Valentin Templé

Table of contents

Context

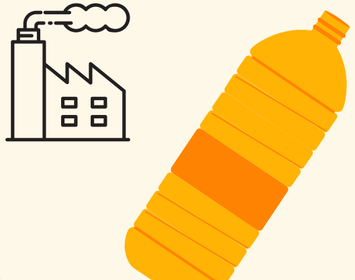
Explosion in the use of LLMs

Models such as ChatGPT, LLaMA, and Mistral are widely used to generate text, code, translate, and automate tasks.

Each request seems “immaterial,” but consumes energy and emits CO_2.

A hidden environmental cost

- LLM inference requires high computing power (GPUs, servers), → therefore high electricity consumption.
- Electricity consumption → CO_2 emissions, depending on the energy mix (coal, gas, renewable, etc.).
- Users and businesses have virtually no visibility into the environmental impact of their queries.



Google

aws

OpenAI

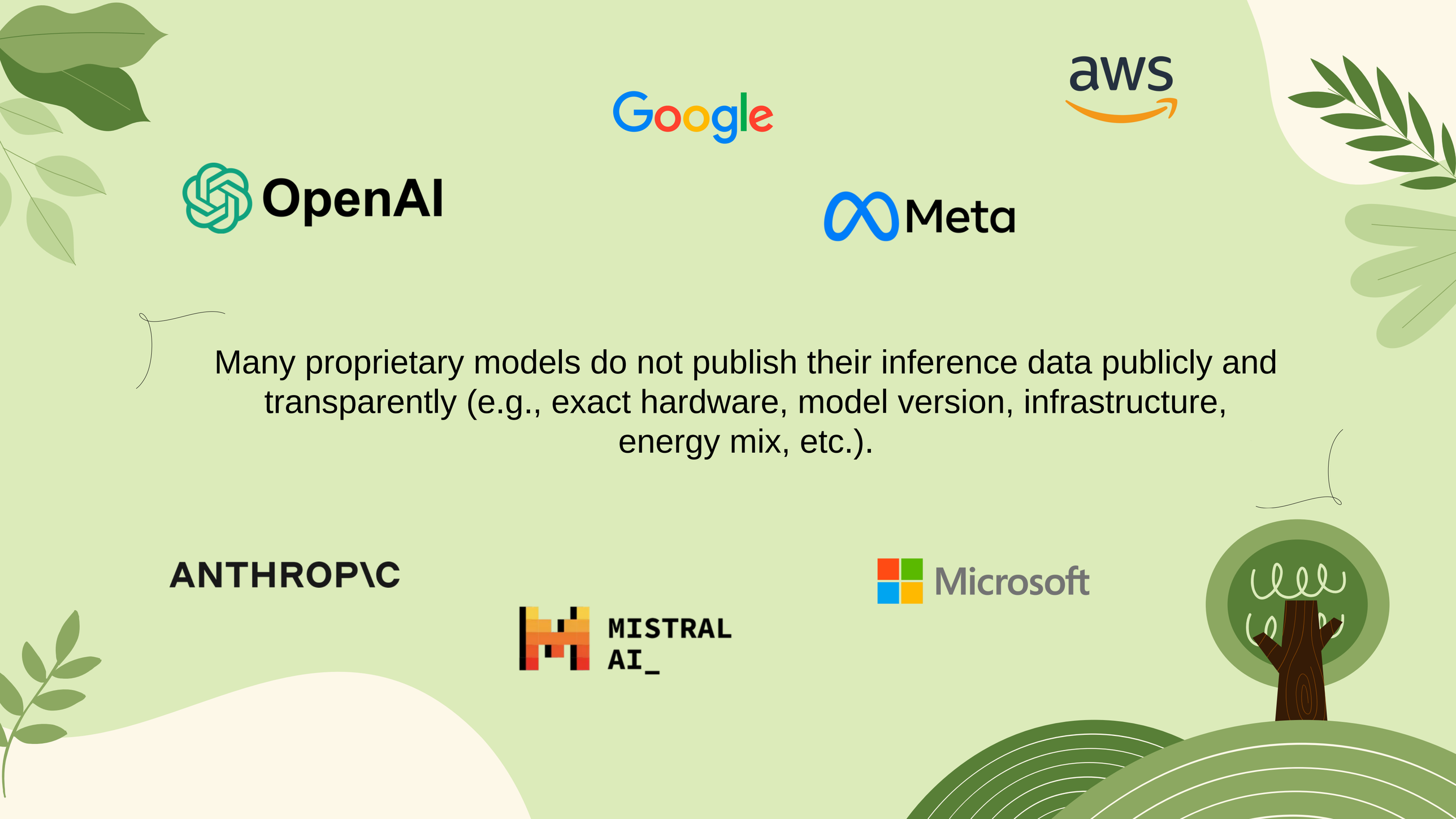
Meta

Many proprietary models do not publish their inference data publicly and transparently (e.g., exact hardware, model version, infrastructure, energy mix, etc.).

ANTHROP\C

MISTRAL AI_

Microsoft



Current issues

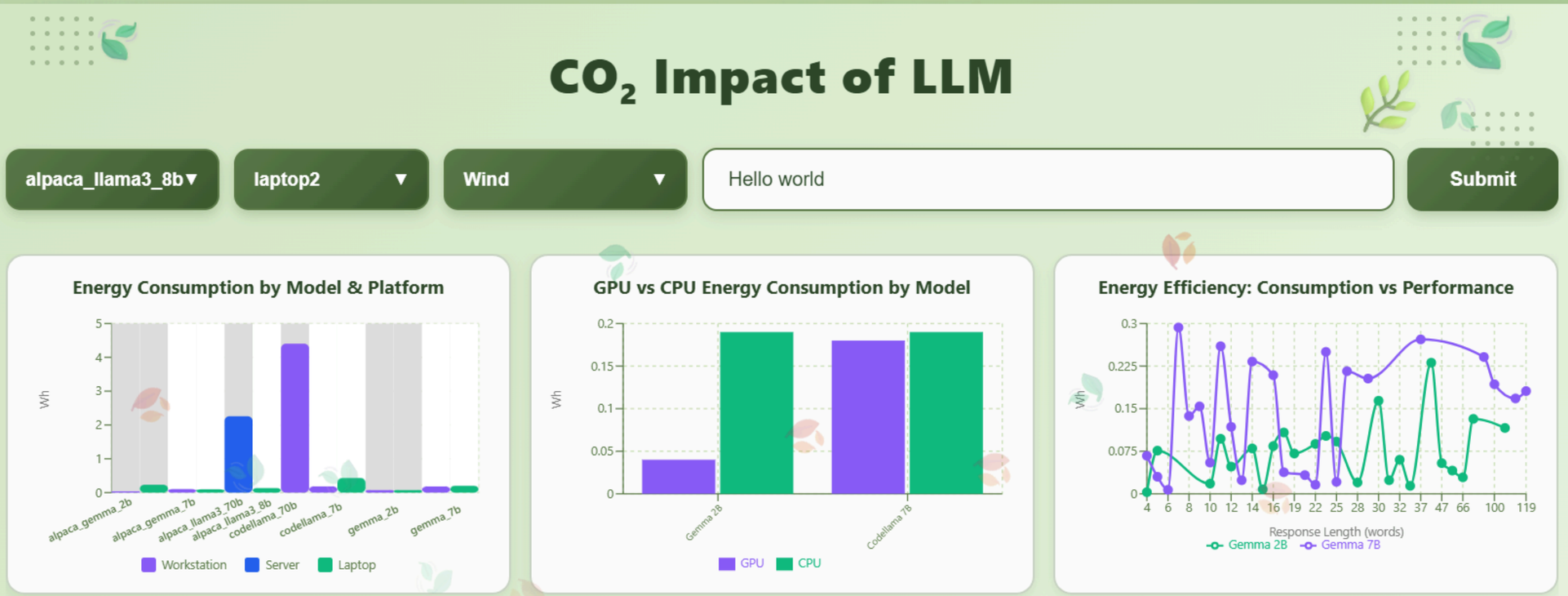
→ How much CO_2 does a single query to an LLM actually emit?

→ Does this vary depending on the model (LLaMA, Falcon, etc.), the device (laptop vs. server), or the size of the query?

→ How can this impact be made understandable and comparable to the real world (airplanes, smartphones, water, etc.)?

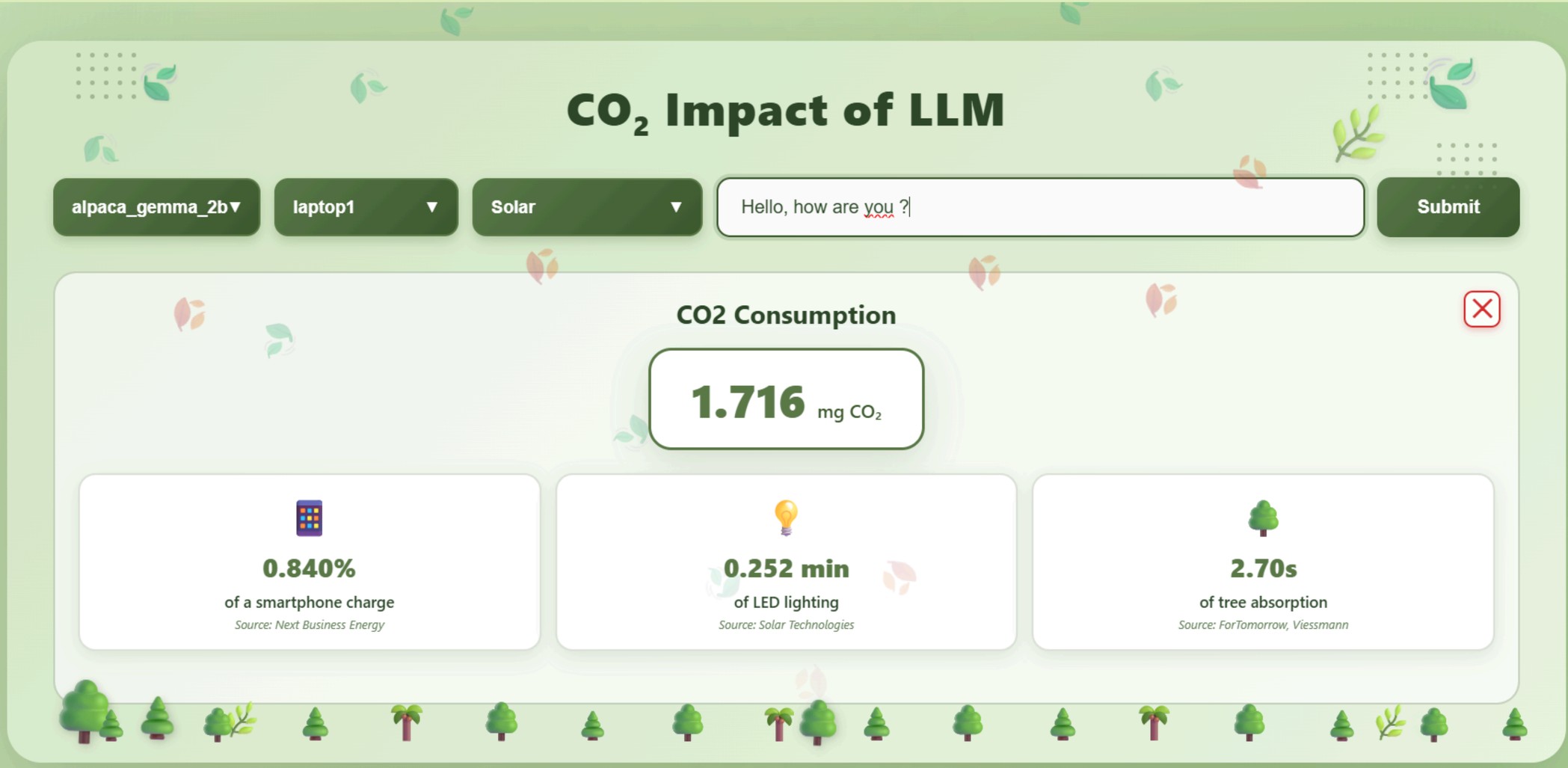
Our solution

Graphs interface with visual representations of the dashboard



Our solution

Interface with query, consumption and comparisons



Technical Approach

Data Preparation

Merged all available datasets into one unified dataset.
Created binary columns (one-hot encoding) for each LLM model and hardware type.
Kept only features that depend solely on the user prompt, not on the generated output
→ to allow prediction before inference happens.

Selected Input Features

- Prompt-related textual features :

word_count, sentence_count, avg_word_length, unique_word_count, avg_sentence_length, punctuation_count, long_word_count, verb_count, monosyllabcount

- Task type (classification of the prompt) :

task_alpaca (instruction-following), task_codefeedback (coding-related request)

- LLM model used :

model_codellama_7b, model_codellama_70b, model_gemma_2b, model_gemma_7b, model_llama3_8b, model_llama3_70b

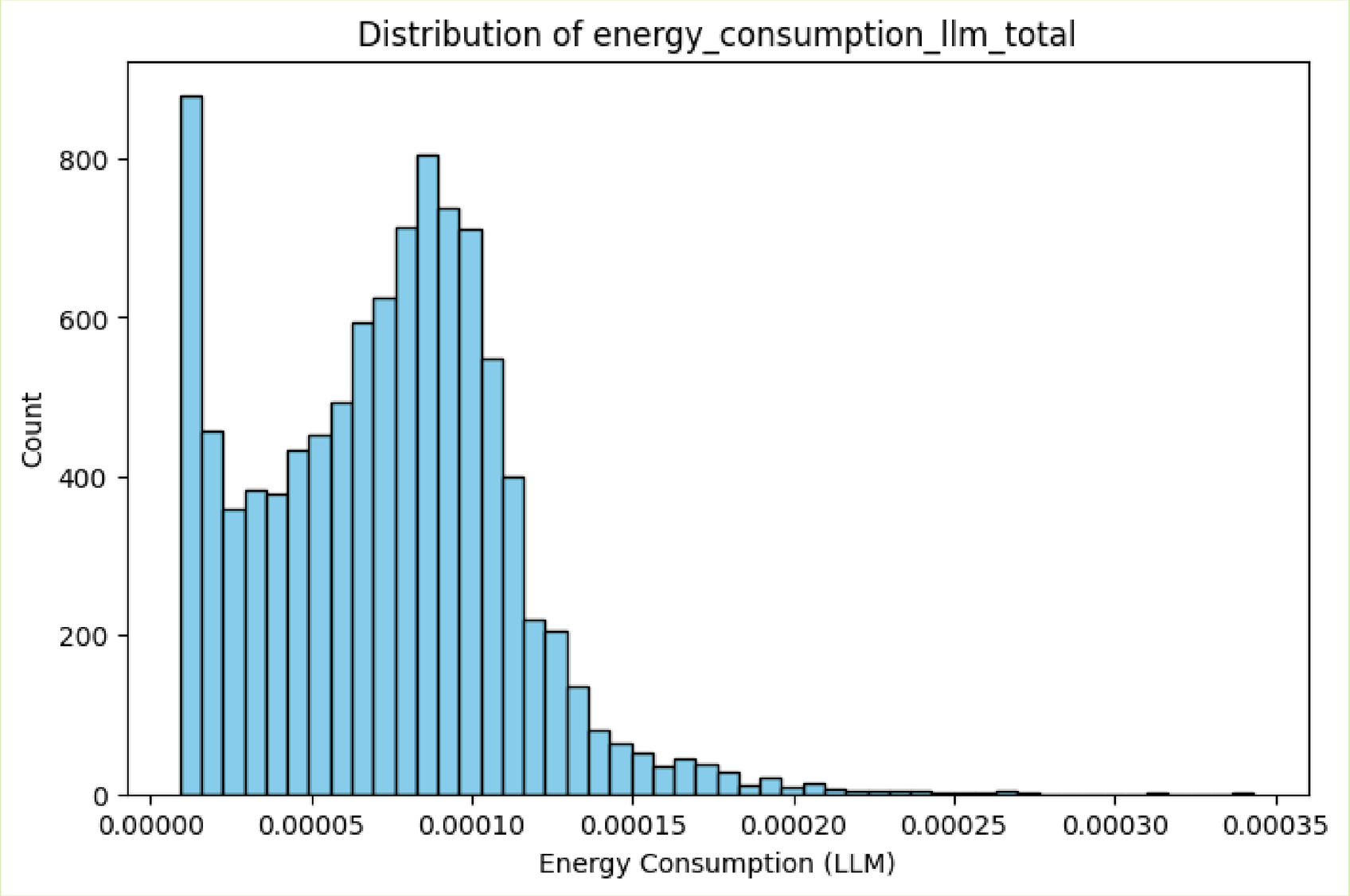
- Hardware type (execution environment) :

hardware_laptop1, hardware_laptop2, hardware_workstation, hardware_server

Model Training

Machine Learning model used: Random Forest Regressor
Trained on the entire dataset with these selected features.
Target variable predicted: Energy consumption per request (Wh) → later converted to CO_2 emissions (gCO_2e).

Technical approach



Our solution

A simple interface to make the invisible carbon impact of each AI query visible

A solution designed with the end user in mind 

No technical knowledge required (GPU, tokens, energy, etc.)

Minimalist UX interface

A clear, simple, and powerful output 

CO_2 score generated by the request

Immediate visualization + interpretation with comparison cases

Graphs with visualisation of the dashboard used for training of the platform

Why is this significant ? 🎯

-Makes an invisible subject → visible
-Transforms technical data into concrete & memorable comparisons
-Enables everyone to become aware in two clicks and adopt more responsible usage
-Accessible to individuals, students, businesses → democratization of sustainable AI

Our vision for the future

Our goal is not to stop AI but it’s to make it sustainable.

Smarter and greener AI models
Use our results to optimize models: reduce energy consumption without compromising performance.
Create a “Green Ranking” of the most efficient LLMs (performance vs. CO_2).

Integration into real-world tools
Provide a public API: integrable into VS Code, Hugging Face, websites, enterprise dashboards, etc.
Enable companies to track their AI emissions in real time.

Personalized recommendations for eco-friendly AI
Offer users:
→ “This model generates the same response, but consumes 3× less energy.”
→ “Use this hardware or cloud region to reduce the impact by 40%.”

Thank you for your attention