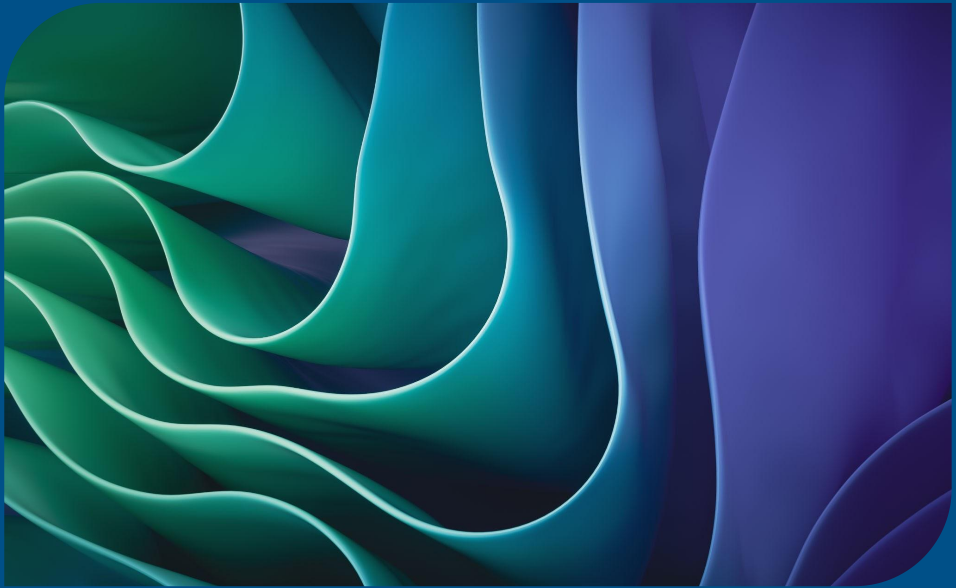
AI Presentation

Nadia Jeylani, Nate Agbemadon, Charlie Tronrud, Colton Kayser



Project Overview

- Games: We created two strategic games with sustainability themes. An unbeatable tic tac toe AI that represents nature battling pollution, and a CSP-inspired puzzle where the player is tasked with placing wells to ensure that every house has access on the grid.
- For the ML Models: Our goal was to identify country-years with non hydro renewables (Wind, Solar, Geothermal, Tidal and Wave) using indicators such as energy, land use, emissions, internet, etc.Also, group nations into four different policy archetypes.
- AI Gui: Worked off original Tic Tac Toe interface. Used various prompts to generate game. Implementing Trivia, Music, appealing Visuals, and overall advancements.

Game 1: Tic Tac Toe

Play a classic game of Tic Tac Toe against an almost unbeatable AI agent that represents nature, while the you represent pollution.

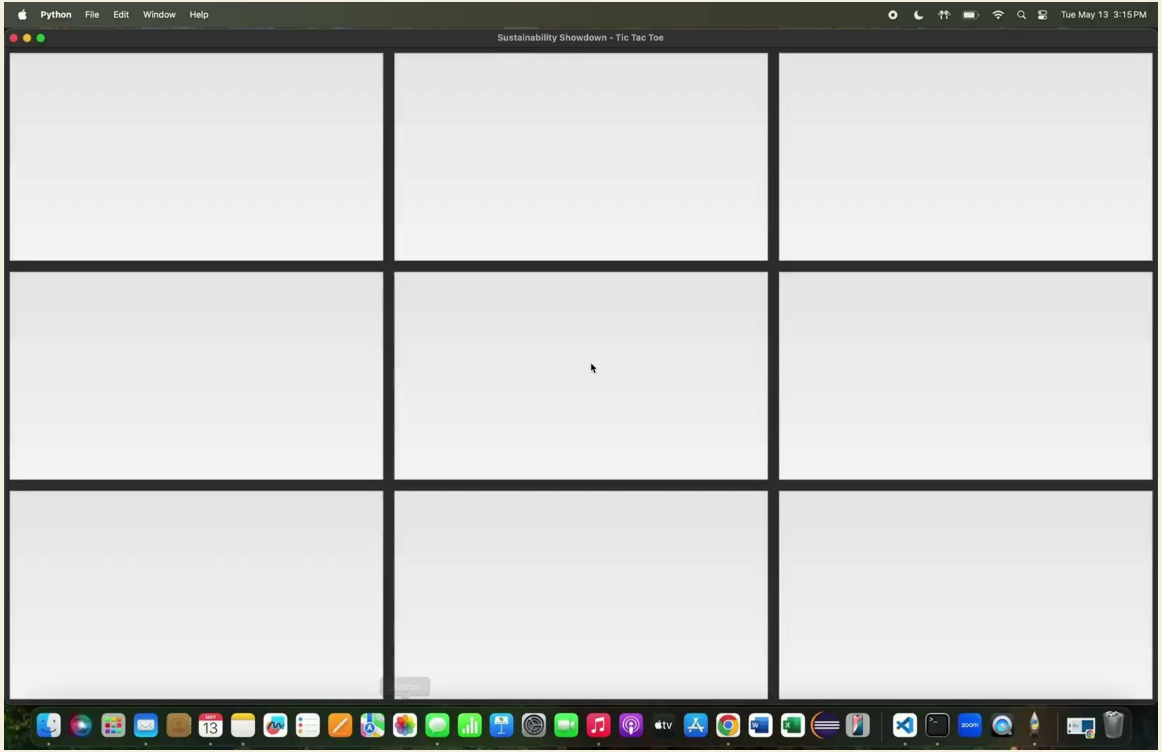
- Sustainability themes: A turn based version of Tic Tac Toe where the player (🏭) battles an AI opponent (🌳) representing nature.

- Unbeatable AI Opponent: Uses Minimax with Alpha-Beta Pruning to evaluate optimal moves and prune inefficient paths, making the AI nearly impossible to beat.

- Decision Making: Demonstrates real time adversarial reasoning through efficient tree search and environmental impact modeling.

- Engaging GUI: Built with Tkinter, uses emojis to represent pollution and sustainability in a playful, intuitive interface.

Demo video


Python File Edit Window Help
Tue May 13 3:15 PM
Sustainability Showdown - Tic Tac Toe

Game 2: Water Planner

In a randomly generated neighborhood, your job is to place water reservoirs strategically to ensure each house is no more than 2 squares from water.

- CSP-Based Puzzle Game where players strategically place wells to ensure all houses have nearby access to water within 2 blocks of them.

- Randomization: Each game generates a new 5×5 grid with randomly placed houses, keeping gameplay fresh and replayable.

- Interactive Placement Logic: Players click to place or remove water sources, and a CSP-style check verifies if all houses are covered.

- AI solver: Integrates a greedy search heuristic inspired by Uniform Cost Search (UCS), prioritizing well placements that cover the most uncovered houses first—balancing efficiency and coverage to approximate an optimal solution without exhaustive backtracking.

- Built with Tkinter: Clean, intuitive GUI allows players to visualize coverage and interact with the system in real time

Demo video

ML Models

We analyzed countries using 10 indicators to predict which countries have “renewable heavy” power grids. Using a decision tree, balanced logistic regression, metric tables, heat maps and K means clustering to discover the truth.

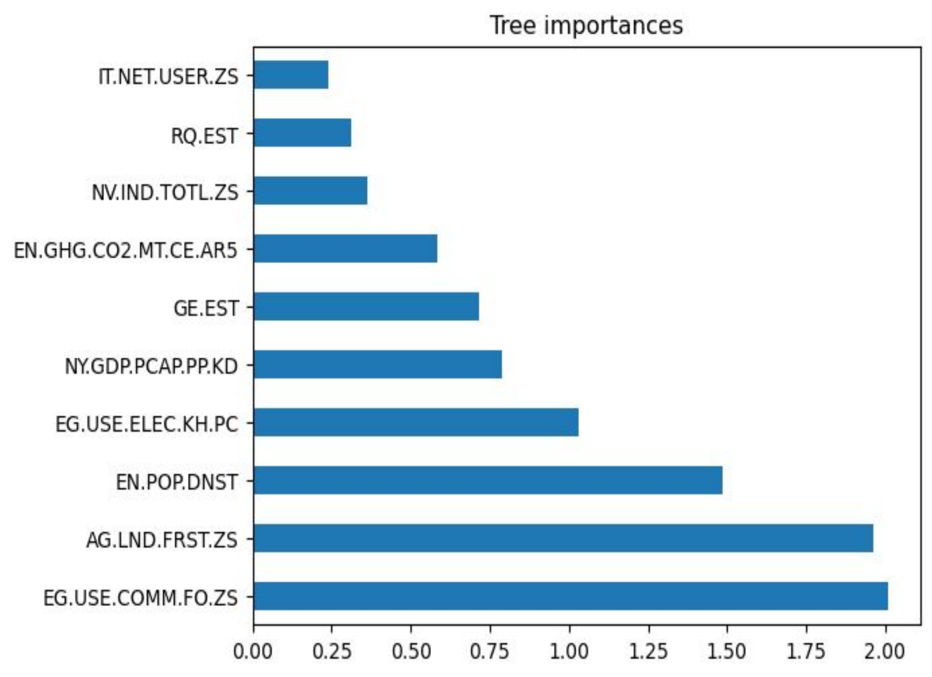
Target variable

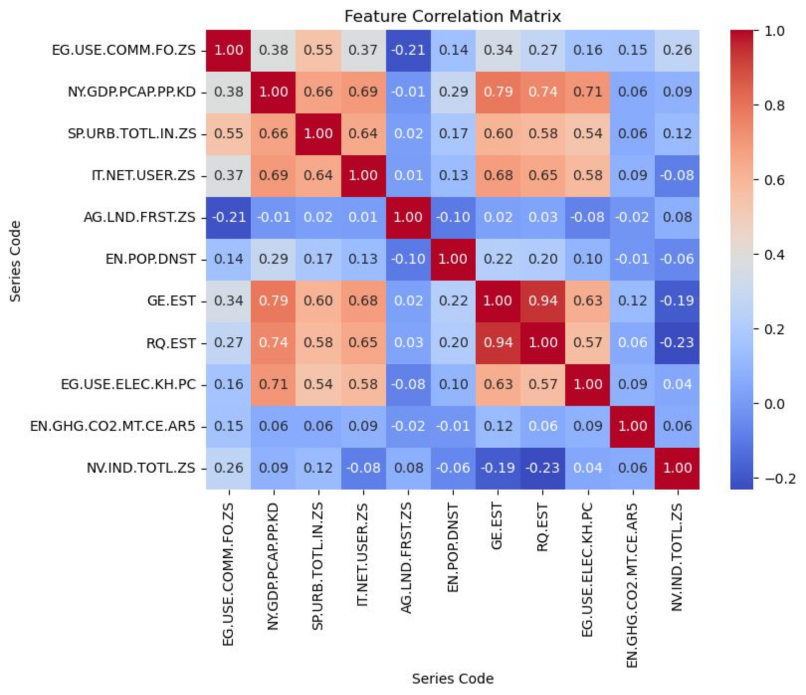
- **EN.ATM.CO2E.PC** – CO_2 emissions (metric tons per capita): used as the continuous target in our regression (or quartiled for classification).

Input features - 10 indicators (World Bank)

- EG.USE.COMM.FO.ZS – % energy from fossil fuels
- NY.GDP.PCAP.PP.KD – GDP per capita (PPP)
- SP.URB.TOTL.IN.ZS – Urban population %
- IT.NET.USER.ZS – Internet users %
- AG.LND.FRST.ZS – Forest area % land
- EN.POP.DNST – Population density
- GE.EST / RQ.EST – Gov't effectiveness & regulation quality
- EG.USE.ELEC.KH.PC – Electricity use per capita
- EN.GHG.CO2.MT.CE.AR5 – CO_2 emissions
- NV.IND.TOTL.ZS – Industry % GDP

Feature Importance and Correlations





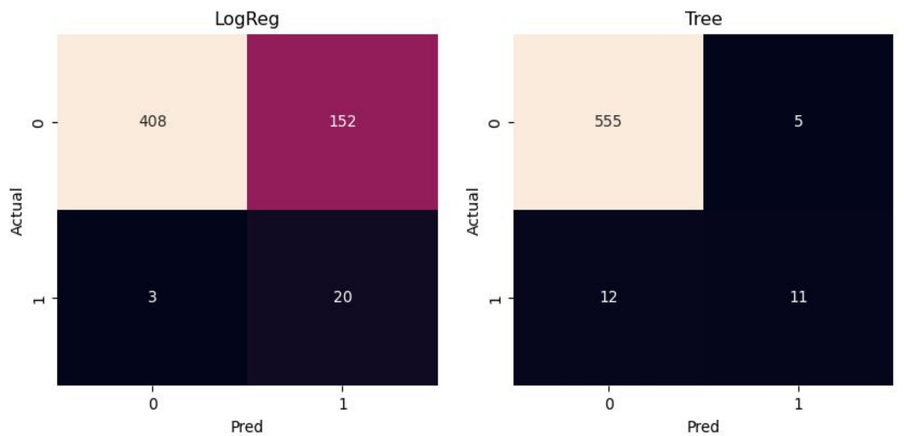
Logistic Regression, Metric and Decision Tree Findings

- **LogReg**: flags almost every true "renewable heavy" (recall = 0.87) but it makes many false alarms (precision = 0.12) → F1 = 0.21 (High Recall, Low Precision)

- **Tree**: far fewer false alarms (precision = 0.69) at the cost of missing some true cases (recall = 0.48) → F1 = 0.56 (Balanced Precision and Recall)

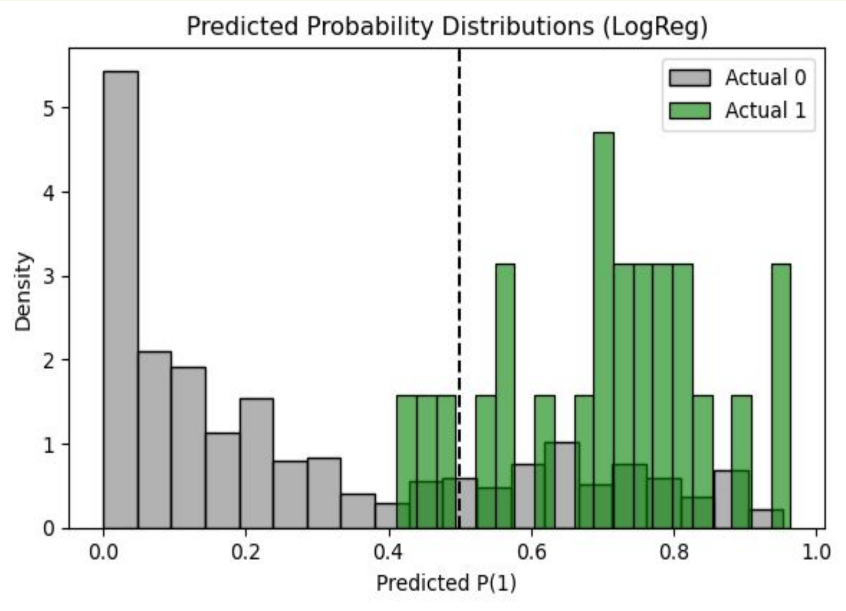
- **Overall accuracy**: 73 % vs 97 %

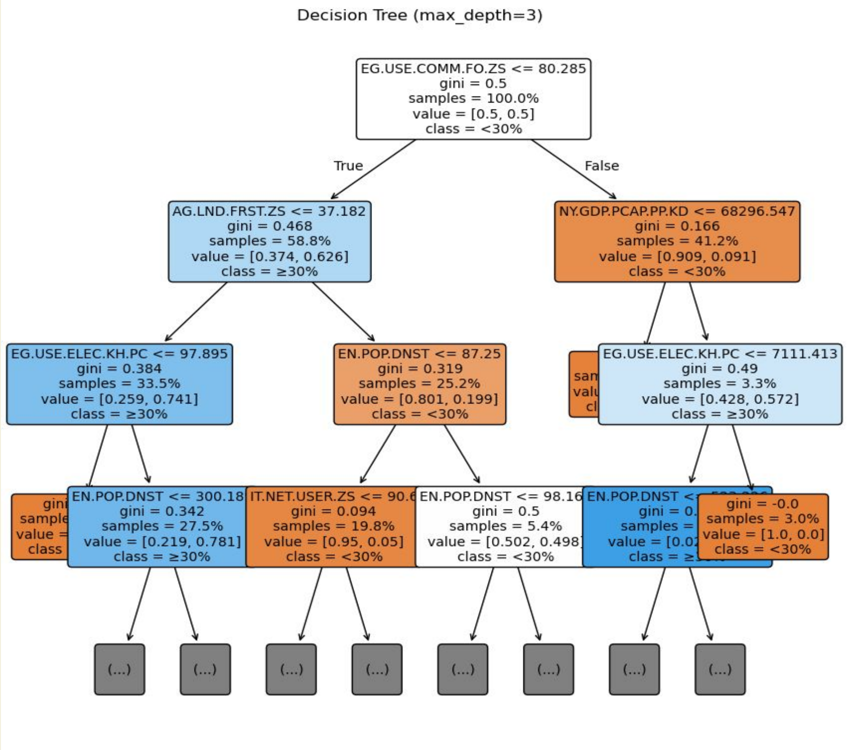
- **Takeaway**: Decision Tree delivers the best balance (highest F1) for reliably spotting renewable heavy grids



Model	Precision(1)	Recall(1)	F1(1)	Accuracy	AUC
Logistic Regression	0.116279	0.869565	0.205128	0.734134	0.864829
Decision Tree	0.687500	0.478261	0.564103	0.970840	0.734666

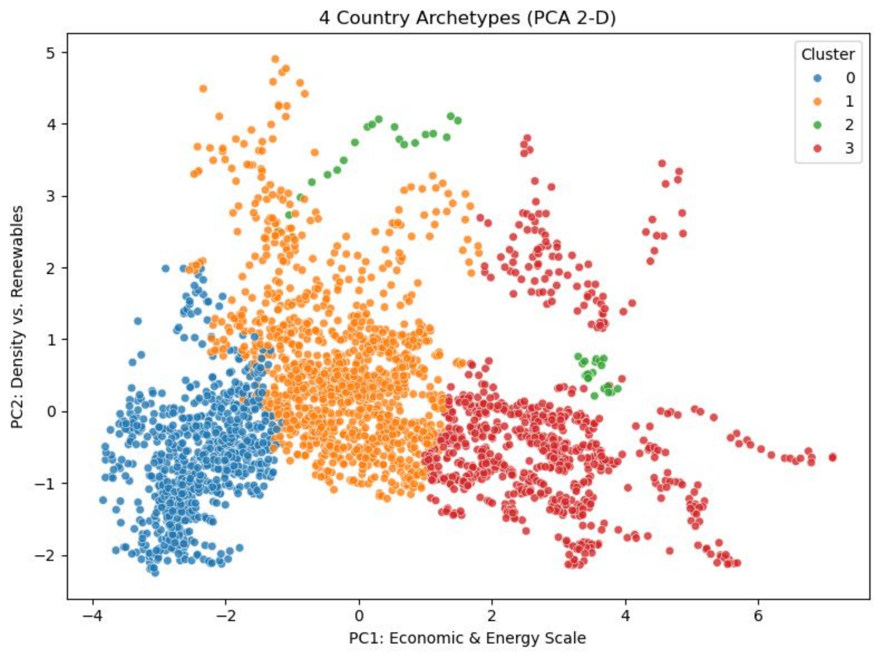
Logistic Regression and Decision Tree Interpretations



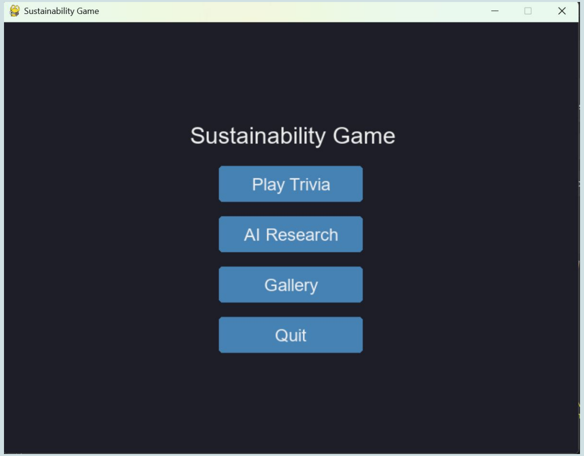


2-D PCA scatter plot of K - Means clusters (k = 4)

- **C0: Low income, biomass dependent nations,** with weak institutions.
- **C1: Emerging fossil users,** ramping up energy demand under a new developing governance.
- **C2: Industrial giants** with massive CO2 emissions.
- **C3: Dense, wealthy, tech savvy** states balancing high demand with serviceable economies.



GenAI-Driven Development Process



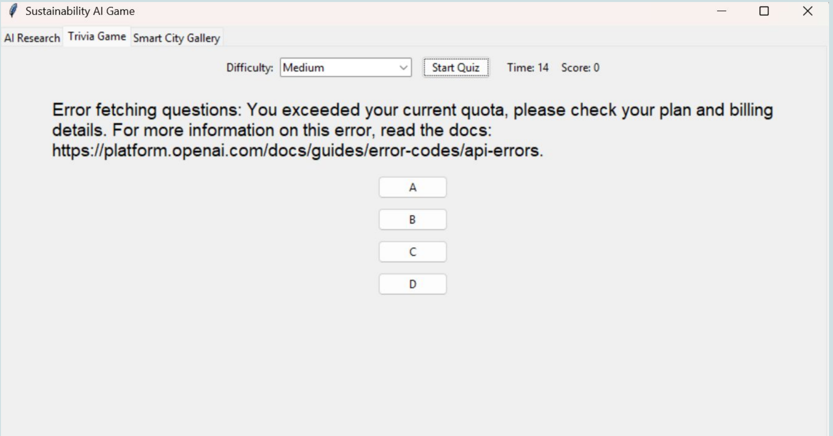
1. Project Setup

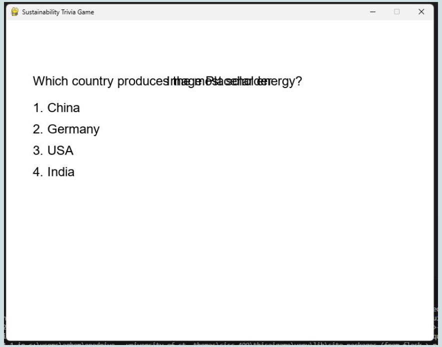
- Chosen tools:
  - **Content & Images:** OpenAI API (ChatGPT & DALL·E)
  - **UI/Engine:** Pygame (Python)
  - **Data Pipeline:** JSON files for questions, policies, prompts

2. Prompt Engineering

- Policies & Simulations:
  i. "Generate 5 bold sustainability policies with AI forecasts vs. real data"
  ii. Refine prompts to output clean JSON tables
- Trivia Questions:
  i. "Create 50 multiple-choice questions on renewable energy, waste, ecosystems"
  ii. Calibrate for 3 difficulty tiers and clear, factual explanations
- Smart City Images:
  i. "Render vertical forest towers at sunset, floating solar farms, rainwater bridges"
  ii. Add one-line captions from hypothetical eco-innovators

Using AI to generate game content speeds up development, but it demands strong prompt-crafting skills, vigilant fact-checking, and in depth asset management to turn raw AI outputs into a polished game.





Lessons Learned & Next Steps

- **Prompt Precision:** Small tweaks yield more structured AI output
- **Future steps:**
  - Add analytics for question difficulty
  - Allow user-uploaded images and captions
  - Turn into a web-deployed app for broader access

Challenges of AI-Generated Content

- **Inaccurate or Inconsistent Outputs**
  - AI sometimes invents data (e.g. “50 Mt CO_2 reduced”) that doesn’t match real sources.

BIGGEST TAKEAWAY

Human-in-the-Loop Verification

- AI outputs occasionally contained factual inaccuracies or formatting glitches
- A quick manual review step ensures question correctness and consistent JSON schemas

Learning Outcomes

- Games: We had to deepen our understanding of adversarial search, we leveraged a minimax tree with alpha-beta pruning, to make an AI opponent that is almost unbeatable, this along with the constraint satisfaction problem that we developed, created two fun and engaging games.
- For the ML Models: We had a great time having to dive in and learn further about all things distinguished classification (predicting renewable heavy vs not) from the regression itself. Also, had to learn how to apply the right metrics and models for each.
- Ai GUI: ChatGPT and Claude were by no means great at first. If we had stopped at prompt 5 we would have had no running code. The logic was never there for anything on the backend (It did look pretty though)

Challenges and Improvements

- Games: We had not had much experience creating python GUI's prior to this project, so it took some time to connect the backend logic we created to make our game operate, with the simple, yet effective interfaces that we developed for our games.
- For the ML Models: We tackled severe class imbalance and missing data. The goal we set forth was one that interested all of us, so a bit of data wrangling and we were able to figure it out. A simple improvement would be find better data but with the data we had, boosted trees could have helped in handling imbalance and the nonlinear interactions better.
- Ai GUI: The challenges remained logic based, nothing on the backend was able to configure through our first 5 prompts let alone 20. Improvements remained in running through countless times we were able to eventually convince GPT and Claude that it knew something. In adjusting some of the logic we were able to complete a running game.

Conclusion and Links

- Games: Creating the unbeatable AI opponent using minimax with alpha-beta pruning helped us appreciate how decision trees can simulate strategic thinking and optimize outcomes in real-time gameplay.
- For the ML Models: Decision Tree was far superior, balancing both precision and recall while minimizing false positives and maintaining majority of the true renewable cases. The clusters revealed four archetypes, with a very simple interpretation of each. Next steps could definitely consist of forecasting in hoping to predict which dynamics can help predict which countries will move more toward the renewable state. Link to Data here.
- Ai GUI: It took over 30 prompts to nail the backend logic, after that, it did a solid job in layering trivia, music, and visuals on the Tic-Tac-Toe UI to deliver a fully working AI-powered game.