

Background

This dataset, compiled by researcher Damien Charlotin, tracks real court cases around the world where generative AI was caught producing fake or misleading legal content, such as fabricated citations or arguments. Each record includes details like the case name, jurisdiction, date, responsible party, AI tool used, and outcome. It's an ongoing project with over 400 cases so far, offering a snapshot of how AI misuse is emerging in the legal system.

Upon observing the Kaggle page and a non-exhaustive Google search, I saw that there has not been any major public visualized data stories published on this dataset. Most articles written about the dataset echo the overarching thesis of my story: The use of AI is spreading faster than courts can regulate.

[Link to Forbes Article](#)

[Kaggle Link](#)

[Charlotin's Interactive Dataset](#)

[Link to Project Repo](#)

Imbalances and Biases on the Data

The data is likely imbalanced and incomplete in several ways:

- Reporting bias: The dataset depends on publicly available or reported cases, so many minor or unpublicized incidents may be missing. Courts or law firms might not disclose AI misuse, meaning the data underrepresents real occurrences.
- Jurisdiction bias: Most recorded cases come from English-speaking or high-profile jurisdictions, where reporting is more transparent, so it may not reflect global legal practice evenly.
- Role imbalance: Cases involving pro se litigants and lawyers are overrepresented, while corporate or government uses of AI in filings are rarely documented.
- Temporal bias: Since the dataset is updated manually, newer AI incidents might lag behind actual events.

It's a valuable starting point, but it probably overrepresents documented, Western, and publicized incidents rather than the full scope of AI misuse in law.

AI-Generated Code for Visualizations

I used a mix of Gemini and GPT-5 as Python coding assistants for my visualizations in Google Colab. They performed surprisingly well in understanding context, given I did not employ any high-level prompting techniques. The built-in Gemini in Colab sometimes struggled to evaluate the past coding cells while generating new code in a new cell, when I switched to GPT-5, it surprisingly understood the context better than the built-in Gemini at times.