Problem statement: #Track 3 Build a prototype system that utilizes LLM capabilities to flag features that require geo-specific compliance logic; turning regulatory detection from a blind spot into a traceable, auditable output.

Development Tools

1. Jupyter Notebook
2. Python
3. Workbench
4. Streamlit

APIs

1. openAI GPT4.1 mini (fine-tuned model)
2. Google Sheets API

Assets

1. Train.csv (training data for fine tuning)
2. Test.csv (validation data for fine tuning)
3. 5 regulation files (.txt)
4. Terminologies.csv

Libraries

1. streamlit
2. pandas
3. Numpy
4. os
5. torch
6. scikit-learn
7. pydantic
8. Pillow
9. gspread
10. gspread-dataframe
11. vertexai
12. openai
13. sentence_transformers
14. faiss

Additional Dataset

1. Train.csv
2. Test.csv

Data Preparation and model fine-tuning

Motivation

To improve accuracy in compliance detection and reduce ambiguous outputs, we fine-tuned ChatGPT-4.1 Mini (gpt-4.1-mini-2025-04-14). We chose this model because it offers:

- Cost efficiency: significantly cheaper than GPT-4 full, suitable for repeated inference.
- Low latency: optimized for fast responses, critical for interactive auditing in Streamlit.
- Fine-tuning support: accepts JSONL-based supervised fine-tuning, ensuring reliable structured outputs.

This trade-off allowed us to combine speed, cost-effectiveness, and structured compliance reasoning in one solution.

Dataset

- **Trained Token**: 932,790 tokens
- **Sources**:
  - Regulatory compliance feature descriptions (PF, CDS, DRT, etc.).
  - Explicit labels for `"Yes"`, `"No"` decisions with regulatory citations.
- **Data generated via AI based on the sample data provided:**
  - 80/20 Training & Validation Split
  - Labelling for violation and corresponding violation was checked and edited manually after feature generation to ensure it met requirements
- **Format**: JSONL with structured dialogues:
  - Each row contains a series of messages that simulate a conversation between the **user**, the **assistant**, and the **tool**.

User Message:

- The user provides the **input** feature description and regulation.
- This is the feature that the assistant needs to evaluate (whether it violates a specific regulation or not).

-
  ```
  {"role": "user", "content": "Evaluate whether the feature violates the specified regulation. ..."}
  ```

Assistant's Tool Call:

- The assistant calls a **tool** (e.g., a function to evaluate if the feature violates the regulation).
- This is a simulation of the **assistant asking the model to process** the input and call the function to determine if the feature violates the regulation.

-
  ```
  {"role": "assistant", "tool_calls": [{"id": "call_1", "type": "function", "function": {"name": "evaluate_regulation_violation", "arguments": "{...}"}}]}
  ```

Tool's Response:

- The tool processes the input and provides a **decision** on whether the feature violates the regulation, the **reasoning** behind it, and any **related articles** (if applicable).

-
  ```
  {"role": "tool", "tool_call_id": "call_1", "content": "{\"violates\": true, \"reasoning\": \"This feature violates compliance standards...\", \"related_articles\": [\"Transparency of user decisions\"]}"}
  ```

Assistant's Summary:

- After receiving the response from the tool, the assistant provides a **human-readable summary** of the result for the user. This message typically paraphrases the output of the tool call.

-
  ```
  {"role": "assistant", "content": "Decision: Violation of EU DSA. Rationale: Automatic restrictions without user notice or complaint/appeal contravene transparency and redress expectations."}
  ```

Training Setup

- **Base model**: gpt-4.1-mini-2025-04-14
- **Method**: Supervised fine-tuning
- **Hyperparameters**:
  - Epochs = 3
  - Batch size = 1
  - Learning rate multiplier = 2
  - Random seed = 123

Training Metrics

- Train loss: 0.000
- Validation loss: 0.000
- Full valid loss: 0.000

Safety Alignment Checks
After fine-tuning, the model was automatically tested against 15 sensitive categories.
All checks passed, ensuring the fine-tuned model complies with safety standards.
Categories evaluated:

- Advice
- Biological threats
- Cybersecurity threats
- Harassment / threatening
- Hate
- Hate / threatening
- Highly sensitive
- Illicit
- Propaganda
- Self-harm / instructions
- Self-harm / intent
- Sensitive
- Sexual
- Sexual / minors
- Violence

This guarantees that the fine-tuned BandAI model is safe for deployment in public-facing applications.

Methodology

Our system is designed to determine whether a proposed new feature violates existing regulations. The process leverages Retrieval-Augmented Generation (RAG) and consists of two main AI components, each with distinct responsibilities:

1. Regulation Compliance Check (AI 1):
   When a query regarding a new feature is received, the system first retrieves relevant regulatory documents using an embedding-based retrieval system. The first AI model analyzes these documents in the context of the new feature and assesses whether any regulations are violated. It then outputs a decision, specifying whether a violation has occurred and, if so, which regulation is implicated.
2. Decision Validation and Consistency Check (AI 2):
   The second AI model has access to a historical database of past compliance decisions, also stored and retrieved via RAG. It receives both the relevant past records and the decision output from the first AI. By comparing the new decision against historical

outcomes, the second AI determines whether to support or challenge the initial assessment. This step helps ensure consistency and accuracy across decisions.

3. Human Oversight and Record Keeping:
   The final decision, along with supporting information, is recorded in Google Sheets and added to the historical records in real time. If human intervention is required, the output can be reviewed and modified immediately. This continuous updating allows the second AI to improve its ability to identify potential errors or inconsistencies in future decisions.
4. Monitoring and Auditing:
   For transparency and oversight, we have developed dashboards that provide a comprehensive overview of key metrics. These dashboards facilitate rapid auditing and help stakeholders monitor system performance, as detailed in the Dashboard section.

This multi-layered approach ensures robust compliance checking, ongoing self-improvement, and transparent record-keeping, thereby enhancing both the reliability and auditability of our system.

Dashboard

Creating Key Metrics

The dashboard revolves around several key metrics that are derived from the data.

1. The first is the Violation Flag, a calculated field created in Looker Studio.
   - This field assigns a value of 1 when the status_latest column indicates a violation, and 0 otherwise. This binary flag provides a clean way to aggregate violations across multiple dimensions. Using this flag, the Number of Violations (the sum of all violation flags) and the Percentage of Non-Compliant Features (the ratio of violations to total features) were defined.

2. Regional Violation Rate metric.
   - This was important to ensure that comparisons across regions were fair. Absolute violation counts tend to be misleading, since larger jurisdictions (like the European Union) naturally encompass more features and therefore more violations. To correct for this, the Regional Violation Rate divides the number of violations in a region by the total number of features from that region. This normalisation allows the chart to highlight regions with disproportionately high violation rates relative to their feature footprint, rather than just their size.

3. Percentage of Non Compliant
   - Since violation was set to 1 and non violation was set to 0, the percentage of violation (also known as non compliance) could be calculated

4. Log Volume

    - This was calculated by summing up all the distinct records in the database to determine the log volume

5. Distinct Regulations
    - Counted the number of distinct regulations that could be detected and thus, displayed in the dashboard for visibility

The dashboard was then designed to present compliance insights in an intuitive and layered way.

- The KPI Cards at the top summarise the dataset: total number of features evaluated, total violations, percentage of non-compliance, log volume, and last refreshed time. These give a quick, at-a-glance sense of compliance health.
- A pie chart of overall compliance status shows the distribution of compliant versus non-compliant features. This provides an immediate sense of the proportion of violations within the dataset.
- A bar chart of violations by regulation uses the split regulation field to highlight which laws are most frequently violated. To make this more interpretable, the metric is expressed as a percentage of total violations, showing the share of each regulation's contribution. This avoids raw counts dominating the picture and highlights which regulatory frameworks are creating the highest compliance burden.
- A geographic map of violations by region overlays violations onto a world map, providing a spatial representation of where regulatory risk is most concentrated. The normalised Regional Violation Rate metric can also be used here to prevent larger regions from skewing the results.
- A compliance status by region bar chart compares regions by the proportion of features that are compliant versus non-compliant. This chart originally showed both compliant (green) and non-compliant (red) segments, but filters were later applied to focus only on violations, making the non-compliant share more visible.
- Each of these visuals works together to provide multiple lenses: overall compliance health, regulatory hotspots, geographic risk distribution, and relative violation shares.