Structured Perspective Generator (Vertex AI Endpoint Only)

Generates a set of structured "perspectives" (default 70) over evenly distributed bias/significance coordinates using **your deployed Vertex AI endpoint**. Public Gemini API key mode has been removed for this project.

Features

- Vertex endpoint only (enforced)
- Input specification via `input.json`
- Even distribution of `bias_x` and `significance_y` in [0,1]
- Color assignment across N colors (default 7)
- Model instructed to output strict JSON only
- Robust JSON extraction (falls back to scaffold if parse fails)
- Raw streamed text saved to `raw_model_output.txt`

Quick Start (Vertex AI)

```
python -m venv .venv
.venv\Scripts\Activate.ps1
pip install -r requirements.txt
Copy-Item .env.example .env
# Edit .env and set VERTEX_ENDPOINT only (ensure ADC auth configured)
gcloud auth application-default login
python api_request.py --input input.json --output output.json --count 70 --colors 7 --endpoint projects/<project>/locations/<region>/endpoints/<endpoint-id>
```

Environment Variables

The script loads `.env` if present.

Variable	Purpose
`VERTEX_ENDPOINT`	Required endpoint path (can be overridden with --endpoint)
`GOOGLE_APPLICATION_CREDENTIALS`	(Optional) Path to service account key if not using gcloud ADC login , Format: projects//locations//endpoints/
`DEFAULT_TEMPERATURE`	Overrides default temperature if provided
`PROMPT_SUFFIX`	Extra instruction appended to prompt (optional)

Authenticate (one of):

```
gcloud auth application-default login
# OR
$env:GOOGLE_APPLICATION_CREDENTIALS = "C:\path\to\service-account.json"
```

Input File (`input.json`)

Example already included:

```
{
  "topic": "Sustainable Urban Mobility",
  "context": "A mid-sized European city wants to reduce congestion and emissions while improving accessibility for residents and visitors.",
  "objectives": ["Lower CO2 emissions by 40% in 5 years", "Increase public transit ridership by 25%"],
  "constraints": ["Limited capital budget in first 2 years"],
  "stakeholders": ["Residents", "Local businesses", "Tourists"]
}
```

Output

`output.json` will contain an array of perspective objects:

```
[
  {
    "index": 0,
    "color": "color_1",
    "bias_x": 0.0,
    "significance_y": 0.0,
    "title": "...",
    "perspective": "...",
    "impact_score": 0.73,
    "significance_explanation": "...",
    "risks": ["..."],
    "action_hint": "..."
  }
]
```

If parsing fails, the scaffold (numeric fields only) is written instead and a warning is printed.

CLI Arguments

Flag	Default	Description
`--input`	`input.json`	Input file path
`--output`	`output.json`	Output file path

Flag	Default	Description
`--endpoint`	from `VERTEX_ENDPOINT` env	Vertex endpoint path (required if env not set)
`--model`	(deprecated)	Backwards compat; treated as endpoint if provided
`--count`	70	Number of perspectives
`--colors`	7	Number of colors
`--temperature`	0.6	Sampling temperature

Troubleshooting

- Endpoint pattern error: Ensure it matches `projects/<project>/locations/<region>/endpoints/<id>`.
- Credential errors: Run `gcloud auth application-default login` or set `GOOGLE_APPLICATION_CREDENTIALS`.
- Empty / malformed JSON: Inspect `raw_model_output.txt`.

Possible Enhancements

- Add retry/backoff logic
- Add schema validation (e.g., with `pydantic`)
- Stream partial JSON to file progressively
- Wire `PROMPT_SUFFIX` into prompt construction

PRs or suggestions welcome.