Traffic Estimation API

# Background

You’re provided with 4 CSV files containing the average numbers of opportunities per country, browser, platform name, and vertical. Where opportunity is a request that can be aligned to a value in those 4 dimensions.

For example 1 request can be an opportunity for country BG, browser Chrome, platform Mac and vertical 280000000000.

Your goal is to create an API that provides traffic estimation (expected total requests), based on user-defined targeting criteria and the historical average data from the CSV files.

# Requirements

1. Data Ingestion

Create a script or module to ingest and parse data from the 4 CSV files into a data structure that allows for efficient querying.

2. API

Create a RESTful API that allows users to query for traffic estimates based on specific targeting criteria.

**The API should accept the following parameters, all of them optional and multivalued**:

1. Browser (e.g., Chrome, Edge) - omitted when all browsers
2. Platform Name (e.g., Windows, Mac) - omitted when all platforms
3. Vertical (e.g., Finance, Entertainment) - omitted when all verticals
4. Country (e.g. BG, US) - omitted when all countries
5. Total Request Count (e.g. 100000000)

**The API should return an estimated traffic count - how many of the Total Request Count match the specified targeting criteria.**

3. Traffic Estimation Logic

Develop the logic to calculate the estimated traffic based on the provided criteria.

Use the parsed data from CSV files to derive the estimates.

4. Documentation

Provide clear and concise documentation for using the API. Include examples and instructions for input parameters and expected output.

5. Optional

Implement any additional features or enhancements that you see fit to improve the usability and functionality of the API.

# Expected Deliverables

* source code for the API
* user documentation
* any necessary configuration or setup files

# Our Evaluation Criteria

* accuracy of traffic estimation
* efficiency and performance of the API
* error handling and robustness
* quality of code and documentation