AI Engineer Home Assignment: Ad Performance Analysis Agent

Objective

Design and implement a system involving a Model Context Protocol (MCP) and an Agent to analyze ad performance data from a public BigQuery dataset.

Tools and Data

- **Data Source:** Google BigQuery public dataset: `bigquery-public-data.google_analytics_sample.ga_sessions_*` will be used as a proxy for web/app session and ad interaction data.
- **Google account:** You can use yours or create a new one just for this task.
- **Preferred Programming Language:** Python.
- **Environment:** Choose which you prefer, your local environment or a Google Colab notebook.

The Task

The assignment is divided into three parts: Data Preparation, building the model context protocol, and building an agent. Before you begin, use a google account to create a project in GCP (see this guide).

Part 1: Data Preparation and Exploration

1. Connect to the specified public BigQuery dataset: `bigquery-public-data.google_analytics_sample.ga_sessions_*`. Use this guide, in step 5 instead of looking for NYC Citi Bike Trips as suggested, search for Google Analytics Sample.
2. Use this documentation to explore the data. The following query is for you to use for the next parts, change it to suit your implementation.

   Note that this query includes all the tables in the dataset, each table in the dataset represents a single date. To access the date of the table you can use this syntext:

   ```
   SELECT DISTINCT PARSE_DATE('%Y%m%d', _TABLE_SUFFIX)
   FROM `bigquery-public-data.google_analytics_sample.ga_sessions_*`
   ```

   and to access specific month:

   ```
   `bigquery-public-data.google_analytics_sample.ga_sessions_201707*`
   ```

   This query contains all the possible dimensions and KPIs that a user can request in part 3 of the assignment.

   Download all the data in a monthly granularity.

```sql
SELECT
   -- Grouping Dimensions
   trafficSource.source AS traffic_source,
   geoNetwork.country AS user_country,
   trafficSource.medium AS medium,
   device.deviceCategory AS device_type,
   hits.page.pageTitle AS page_title,
   -- Key Metrics Aggregation
   COUNT(DISTINCT fullVisitorId) AS total_visitors,
   SUM(CASE WHEN hits.type = 'PAGE' THEN 1 ELSE 0 END) AS total_pageviews,
   AVG(totals.timeOnSite) AS avg_time_on_site_seconds,
   SUM(CASE WHEN totals.transactions >= 1 THEN 1 ELSE 0 END) AS total_conversions

FROM
   `bigquery-public-data.google_analytics_sample.ga_sessions_*`,
   UNNEST(hits) AS hits -- Unnest the hits array to analyze page-level data

WHERE
   trafficSource.source IS NOT NULL
   AND trafficSource.medium IS NOT NULL
   AND totals.visits >= 1 -- Filter out non-meaningful sessions

GROUP BY
   traffic_source,
   medium,
   device_type,
   user_country,
   page_title

   HAVING
      total_pageviews >= 20 -- Ensure enough volume per page/channel combo
```

Part 2: Building the MCP Server

Create an MCP Server in Python that exposes a "tool" (a function) to AI Clients. The server should define the following tools:

1. `get_monthly_data`: This tool should accept a month and a set of requested dimension columns and return a structured JSON response containing the calculated KPIs for these dimensions on this month (as described in the query above).
2. `get_all_data`: This tool should accept a set of requested dimension columns and return a structured JSON response containing the calculated KPIs for these dimensions on all the data.

Part 3: Building the Agent and Client Interaction

Build an AI Agent application that interacts with the MCP created in Part 2. For this part use the following flagging rules:

- Traffic rule: Avg. time on site < 120 seconds and total pageviews below 30
- Conversion rule: zero conversions and more than 250 pageviews

The agent should be able to:

1. Compare between two months what were the changes to the KPIs for all the dimensions requested by the user. Provide for each segment the percentage change in each KPI.
2. Given all the dimensions except user country, and a rule name (traffic or conversion) return all the flagged segments.
3. Return the conversion rate (total conversions / total visitors) for each user country on each device on a given month, ordered from highest to lowest.

Deliverables

Submit the following:

- Python scripts or Colab notebook containing the code and documentation.
- A brief write-up (1-2 pages) explaining:
    - The architecture of your agent and MCP.
    - Any challenges you had working on this assignment.