Data Test Engineer - Technical Assignment

Overview

This task involves using Python to look at data we have created in AWS S3 buckets, performing checks on that data, and producing a report on the quality. Feel free to use any publicly available libraries and frameworks to perform the task.

Please submit your source code via a github repo or a zip file. If you use github, please do not use Adthena in the name of the repo.

Please include a README file in your solution with the following information:

- Any necessary setup instructions including everything that is needed to set up the environment, install requirements, and anything else.

Task Output

Please submit a report summary that contains your findings. This can be any text format.

Domain Knowledge:

- search_term is a search that has been performed on Google search. For example: "**car insurance**"
- device is the device used to perform the search. For example "desktop" or "mobile"
- scrape_count is the number of times that search was performed on a specific date and device.
- domain is an advertiser domain. For example "asos.com"
- ctr: the probability that someone will click an advert
- sponsored, natural, and pla appearances are the number of times a type of advert has appeared for a given search term / device / domain on a date

The Task

There will be an S3 bucket that contains two datasets described below. These are all in multi-file parquet format.

In the provided datasets described below, we have introduced some issues that break the described constraints. We expect that your tests are able to find the issues that we have introduced. You should also be able to catch any other common issues that might be present in the provided dataset.

Buckets Overview

Scrape Appearances Bucket

(**s3://adthena.data.qa.test/scrape_appearances**):

The Scrape appearances bucket contains a number of scrapes we have made for the search term on specific device and date.

Columns:
date: Date (YYYY-MM-DD).
device: String.
search_term: String
scrape_count: Int

Constraints:
Primary Key (unique): date, device, search_term.
device (non-NULL): only **mobile** and **desktop** are supported.
search_term (non-NULL): maximum 400 characters long.
scrape_count (non-NULL): > 0.

Competitor Appearances Bucket

(**s3://adthena.data.qa.test/competitor_appearances**):

The Competitor appearances bucket contains advertiser (domain) statistics from the scrapes that we have made for a given search term, on a given day and device. Each row in this dataset must have a corresponding row in the scrape appearances dataset.

Columns:
date: Date (YYYY-MM-DD).
device: String.
search_term: String
domain: String
sponsored_appearances: Int
natural_appearances: Int
pla_appearances: Int
ctr: Double

Constraints:
Primary Key (unique): date, device, search_term, domain.
device (non-NULL): only **mobile** and **desktop** are supported.
search_term (non-NULL): maximum 400 characters long.
domain (non-NULL): maximum 100 characters long.

sponsored_appearances (non-NULL): Minimum value: 0. Maximum value: the scrape_count value from scrape appearances for that search term, device and date.
natural_appearances (non-NULL): Minimum value: 0. Maximum value: unlimited.
pla_appearances (non-NULL): Minimum value: 0. Maximum value: unlimited.
ctr (optional - NULLable): Minimum value: 0. Maximum value: 1.0.

Login Credentials

We have created a bucket in S3 which you can use to read the input data from. The paths are given in the previous section. You have been assigned API access and below are the required credentials.
Access Key: AKIAWV4HZOSXKXDHGYNZ
Access Secret: sIR9COhSBIOFMTy+5kLEXqXh+K3zZ2tJtR1J/wy9

For data exploration you can use the AWS CLI (https://docs.aws.amazon.com/cli/latest/reference/s3/) with your assigned access keys or if you prefer GUI tools, Cyberduck might be an option (https://trac.cyberduck.io/wiki/help/en/howto/s3).