gojek

Take-home challenge

Machine Learning & Big Data

The goal of this exercise is to provide a realistic example of the kind of work we do and evaluate your ability to do such work. This exercise should require about 4 hours of your time.
Please reach out to us if you have any questions!
When done, please follow the submission instructions and send your submission via email.

Problem statement

We would like you to build a Web Service that provides basic analytics over Chicago Taxi Trips. The dataset can be obtained from here .
The dataset includes taxi trips in Chicago for the year 2020. There are 3,888,425 rows in the dataset and the detailed schema definition can be found in the Dataset Schema section at the end of this document.

The webservice should provide 3 endpoints as follows:

Total trips per day

```
GET /total_trips?start=<start-date>&end=<end-date>
```

Total number of trips per day in the date range, based on the pickup time of the trip.
E.g.:

```
$ curl http://localhost:8080/total_trips?start=2019-01-01&end=2019-01-31

{
  "data": [
    { "date": "2020-01-01", "total_trips": 321 },
    { "date": "2020-01-02", "total_trips": 432 },
    { "date": "2020-01-03", "total_trips": 543 },
  ]
}
```

Fare heatmap

```
GET /average_fare_heatmap?date=<date>
```

The average fare per pick up location S2 ID at level 16 for the given date, based on the pickup time of the trip.
E.g.:

```
$ curl http://localhost:8080/average_fare_heatmap?date=2019-01-01

{
  "data": [
    { "s2id": "951977d37", "fare": 13.21 },
    { "s2id": "951977d39", "fare": 4.32 },
    { "s2id": "951977d40", "fare": 5.43 },
    { "s2id": "951978321", "fare": 9.87 }
  ]
}
```

Average speed in the past 24 hours

```
GET /average_speed_24hrs?date=<date>
```

Average speed of trips that ended in the past 24 hours from the provided date. Use the most suitable data from the dataset for calculating the average speed and mention it in your documentation. Use km/h as the returned units.
E.g.:

```
$ curl http://localhost:8080/average_speed_24hrs?date=2019-01-01

{
```

```
    "data": [
      { "average_speed": 24.7 }
    ]
  }
```

Other considerations

- All dates should be in ISO 8601 YYYY-MM-DD format and date ranges are inclusive on both ends. Timezone is local to the dataset.
- Provides up to 2 decimal places if the value returned by the API is decimal.
- Please use the correct HTTP verbs and errors, and also set the response headers appropriately.
- There may be missing, bad data, or other edge cases. Use your own judgement to handle these cases and clearly state your reasoning in documentation.

Submission

Your submission should include:

- Web Service
  You can use any language, libraries and database, but be prepared to discuss the choices with us. It's also recommended to use docker-compose to launch the application together with its dependencies.
- Documentation
  Includes README.md in your submission that should contain: how to set up the application and short motivation on major design decisions. We value greatly well documented code.
- Git History
  You can do this via git-bundle. We will be taking a look at your git history, so please keep it tidy and descriptive. DO NOT share your code on a **public** repo
- Executable Scripts
  Includes **bin/setup** and **bin/run** executable scripts in your submission. **Ensure that after executing *bin/setup* and *bin/run* your application is ready to accept API requests defined in the problem statement**. Following is the requirement of each script:
  - **bin/setup**
    To download and install dependencies (including the dataset), compile the code and then run your unit test suite. Print your code coverage at the end of the test.
  - **bin/run**
    To run the application and bind to localhost. “bin/run” should accept application ports as an argument. If the port is not provided it should use port 8080. Example:

```
>  bin/run
Starting taxi analytics at port 8080

> bin/run 8081
Starting taxi analytics at port 8081
```

Your submission SHOULD NOT contain the original dataset file. Use “bin/setup” to download the dataset.

Evaluation Criteria

- Your code will need to run locally on our machines (Mac OS or Ubuntu).
- We are interested in your software design skills, so please craft the most beautiful code you can.
- You should consider this as a production system, write comprehensive tests.
- You should also consider future maintenance of the code and its architecture. Note that you will be working with us to extend this code at a later stage of the interview process.
- Runtime performance is not critical but should not be neglected, as a rule of thumb ensures all API can be completed within 300ms. Choose your algorithms and data structures wisely.

F.A.Q

Q: To design a production system, I’d need information such as the nature of the workload and the growth. What can I assume about this?

A: It’s up to you, as long as it’s documented and reasonable

Q: What sort of documentation are you looking for? A production API service would have the documentation about the endpoints generated with a tool like Swagger. Or are you looking for comments in the code only.

A: There should be at least README.md, which contains user/dev manual and design consideration. Additional documentation is a plus.

Dataset Schema

Field name	Type	Description
unique_key	STRING	Unique identifier for the trip.
taxi_id	STRING	A unique identifier for the taxi.
trip_start_timestamp	TIMESTAMP	When the trip started, rounded to the nearest 15 minutes.
trip_end_timestamp	TIMESTAMP	When the trip ended, rounded to the nearest 15 minutes.
trip_seconds	INTEGER	Time of the trip in seconds.
trip_miles	FLOAT	Distance of the trip in miles.
pickup_census_tract	INTEGER	The Census Tract where the trip began. For privacy, this Census Tract is not shown for some trips.
dropoff_census_tract	INTEGER	The Census Tract where the trip ended. For privacy, this Census Tract is not shown for some trips.
pickup_community_area	INTEGER	The Community Area where the trip began.
dropoff_community_area	INTEGER	The Community Area where the trip ended.
fare	FLOAT	The fare for the trip.

tips	FLOAT	The tip for the trip. Cash tips generally will not be recorded.
tolls	FLOAT	The tolls for the trip.
extras	FLOAT	Extra charges for the trip.
trip_total	FLOAT	Total cost of the trip, the total of the fare, tips, tolls, and extras.
payment_type	STRING	Type of payment for the trip.
company	STRING	The taxi company.
pickup_latitude	FLOAT	The latitude of the center of the pickup census tract or the community area if the census tract has been hidden for privacy.
pickup_longitude	FLOAT	The longitude of the center of the pickup census tract or the community area if the census tract has been hidden for privacy.
pickup_location	STRING	The location of the center of the pickup census tract or the community area if the census tract has been hidden for privacy.
dropoff_latitude	FLOAT	The latitude of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy.
dropoff_longitude	FLOAT	The longitude of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy.
dropoff_location	STRING	The location of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy.