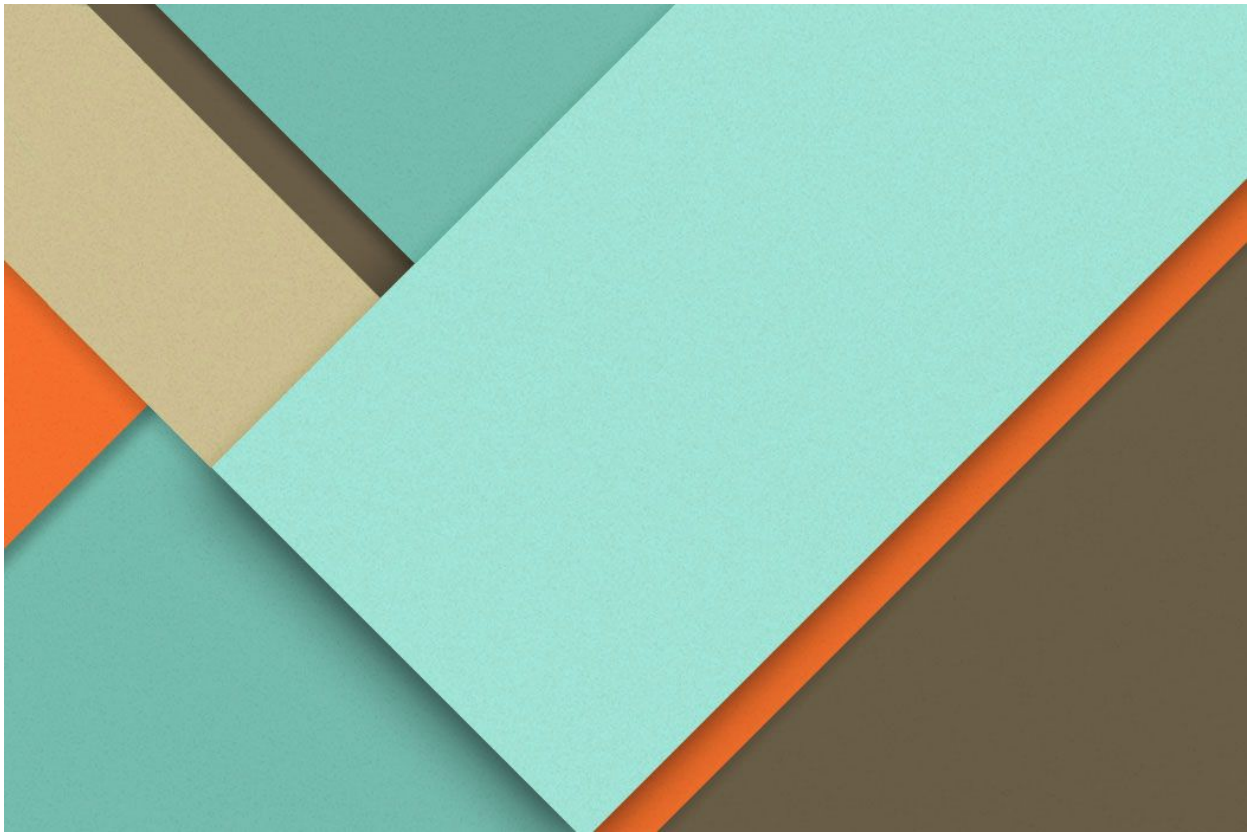

Locale Ai Report

28.04.2019

Vibhu Bhatia

839/BT/15

Overview

In this project, since I have experience in all the domains, I intended to do all the 3 tasks, but given the time constraints due to our exams coming up, I could do the front end task, performed an exploratory analysis on the data and am going to give the ideal database architecture and a very small version of API.

Goals

1. Create an mvp version of the front end using react.js
2. Performed analysis and visualization of data
3. Made a system architecture and wrote a simple python API.

Specifications

Front End

For the front end of the system the task was to make a front end application, where user can upload a csv file, and plot the coordinates in a map. I designed 2 components which seemed to do the job, one being the one where user can upload the csv file and then view the table and select the latitude and longitude columns. The 2nd component contains a map where those points are plotted and some visualisations are provided.

I used react-map-gl api which was a wrapper library made by uber on top of mapbox api due to less complexity and almost similar functions. I also planned to show the path and the distance when a person clicks on a coordinate but mapbox api doesn't provide routes in india, so that kind of proved as a dead end.

Technologies/Libraries Used : React.js, Mapbox, Bootstrap, eCharts.

Back End

I divided the Back End task into 2 tasks- api generation and Schema design. Schema design is put in here and i have made a simple api which communicates with that schema to get data.

Initially i thought of using Node.Js since it communicates with react more easily and i have a bit of experience in designing some microservices using Node.js, but i decided to stick with

python only due to flas being quick to setup and python is still an easier and powerful language.

Frameworks/Libraries used: Flask,Python pyscopg2 (PostgreSQL interface for python), numpy, pandas

Database Schema:

User :

- user_id: unique int/varchar
- name: string
- email/phone number: unique varchar/int
- area_id: int
- city_id: int

Driver :

- Driver_id: unique int/varchar
- Name : string
- Vehicle_id: int
- city_id: int
- rating : float (total_trips/cancelled_trips or some other metric)
- cancelled_trips: int
- area_id: int
- distance_travelled: float

Vehicle:

- vehicle_id: unique int/varchar
- vehicle_name: string
- tank_capacity: int
- capacity: int
- speed: int
- number_of_cars_available: int

City:

- city_id: unique int/varchar

- name: string
- population: int
- areas: either array of area codes or total number of areas
- active_drivers: int

Trip:

- trip_id: unique int/varchar
- user_id: int
- driver_id: int
- medium_of_booking: int
- travel_type_id: int (0-7: hourly rental, 8: long distance, 9: point-2-point)
- from_lat: float
- from_long: float
- to_lat: float
- to_long: float
- booking_created: date/time
- from_date: date/time
- is_cancelled: boolean

Data Science Task

This proved to be kind of limited by time constraints and a less experience in the field of price surging, so i couldn't do that but i made some visualizations of data which might help understand the data more easily. However, given some more time or if exams weren't there i might have i even read up about how surge pricing works and developed a proper system for it.