Muhammad Abuzaid
ibnsaadoosh@gmail.com

Contents

1. Trade Offs

Backend

using nodejs is not recommended, since it's best fit in i/o bound applications not heavy computations applications as it's event loop is single threaded, so our application may block the single thread as it has data manipulation on more than 200,000 records.

- Any multi-threaded language will be good for this sort of applications, so we choose python as it's recommended in the task description, and it's good when dealing with data and plots.
- We chose Flask for API, pandas for reading files, matplotlib library for plotting.

Front End

- We work with native HTML, as there is no sense to use React or any library or framework for a small task like this, it will just slow down performance.

Who is responsible for plotting the figures

- A possible approach here would be to build an API that returns data and let the front-end of the application (vanille javascript) render the data using javascript charting library (highcharts), but every time I would now like to change something I would need to change both front-end and backend.
- So it's better to plot in the backend, and return an image to display on frontend.
- We can use a database for storing csv data, if we do so we will use noSql database, as there is only one table,

so no relations required, so no need for a relational database, but we choose not to use a database at all sense we can read from csv file through pandas directly, it will be more efficient and faster.

2. API

It’s simple, we create an endpoint for each graph (We created only 3 graphs as a proof of concept, the rest is the same idea)

```
GET /api/samples_per_class

GET /api/affiliate_channel_percentage

GET /api/affiliate_channel_provider_percentage
```

3. HTML

- The <img> tag will call the api with the required plot
- Here we create three plots

```
 <div id="container" style="min-width: 310px; height: 100%; margin: 0 auto">
     <img src="http://127.0.0.1:5000/api/samples-per-class" />
     <img src="http://127.0.0.1:5000/api/affiliate_channel_percentage" />
     <img
src="http://127.0.0.1:5000/api/affiliate_channel_provider_percentage"/>
 </div>
```