CMPT 353 - Fall 2022

OSM, Photos, and Tours

Riya Roopesh 301360795
Tushrima Kelshikar 30357928

Introduction

For our project we have analyzed the following questions and came to a conclusion by handling data.

1. Once the user uploads a photo, where are the closest accommodations[Hotels] ? What are the 10 places to visit nearby?
2. I feel like there are some parts of the city with more chain restaurants (e.g. McDonand's or White Spot franchises, not independently-owned places): is that true? Is there some way to find the chain places automatically and visualize their density relative to non-chains?
3. If we collect a set of geotagged pictures can we map out the path the user took and determine the approximate distance covered by the user at the end of the tour?
4. If a user who is a tourist wants to try out different cuisine food, what cuisine is around him?

Methodology

Gather Data

We have used the data provided by the professor, which is saved as “osm.gz”, this folder contains “amentities-vancouver.json-gz”, the dataset file which the professor has cleaned from “planet.osm”. This dataset was collected from the community-provided map data, OpenStreetMap. OpenStreetMap provides data of various types in XML form.

For this project we have collected more data from the British Colombia's official Data Catalogue. We included two more files ie: hellobc-accommodation-listing-20221205.csv and hellobc-attractions-listing-20221205.csv, one containing more information on accommodations and the other containing more information on the attractions and activities in the Vancouver area. These files were read and written into a new JSON file called "hotel_data.json" and "tourism_data" respectively.

Once we were able to collect more data, we started off by cleaning the new data, and transforming it to standardize our data in a similar format. Then we appended the new data to the existing file called "amenities-vancouver.json" and named it "combined_data.json".

We used this combined data in order to analyze and answer our questions mentioned in the section above.

We have also collected our own images in order to test our implementations. The collection of images are saved in a .csv file once the user inputs all the images. We work on this "picture.csv" file to plot out a path taken by the tourist.

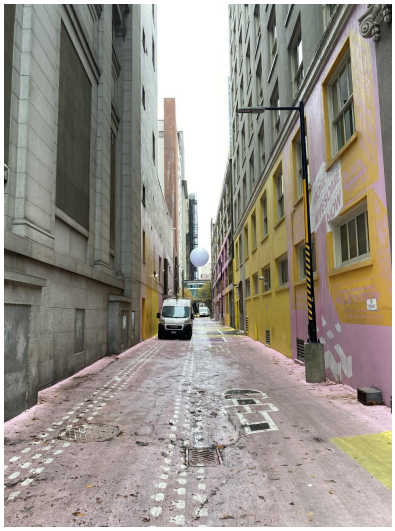
Clean Data

In order to combine our new found data to the existing data we had to carry out a few data cleaning tasks. In the tourisim_data.json file we removed those columns that were not relevant to us. And created a new column called tag containing the website information of the particular attraction. We also made sure to filter our data to contain it to the Greater Vancouver area. We carried out the same steps in our 'hotel_data.json'. Once this was done we were able to combine all three datasets ie: "amenities-vancouver.json", "hotel.json" and "tourism.json" to create "combined_data.json".

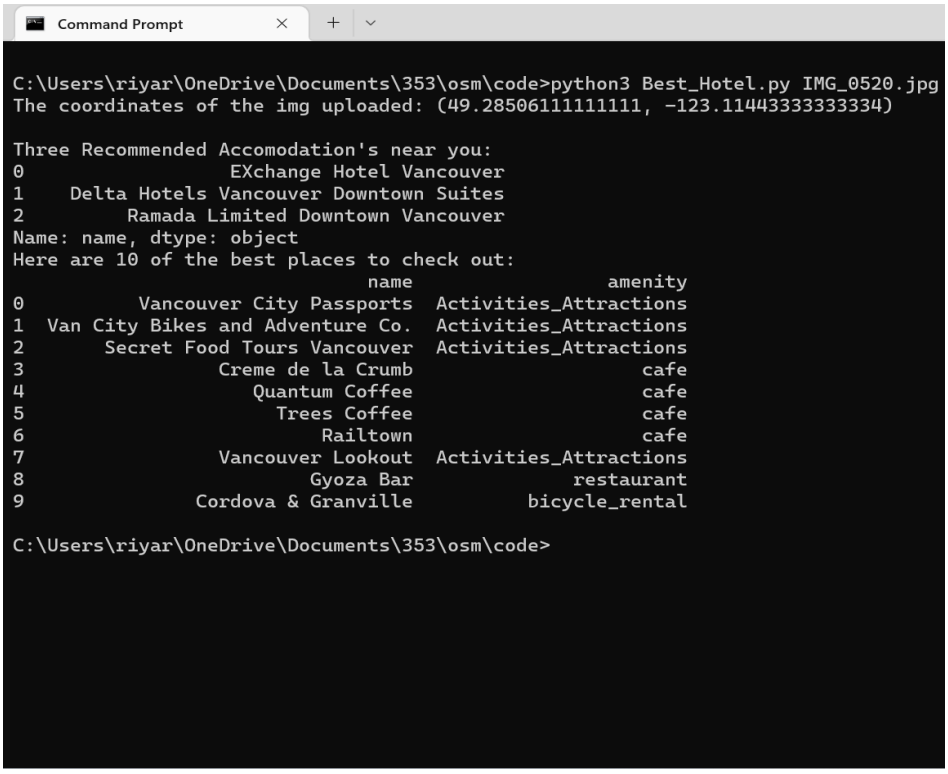
We then got rid of the column = “timestamp” and dropped those rows where the column=‘name’ is NULL.

Data Analysis & Results

We get the image from the user, return the latitude and longitude from the get_latlong() function which uses EXIF data in JPEG images, and then use folium to map out the location of the image. Folium is a python library that is used to visualize geospatial data. In our project we are trying to point the user out to the closest accommodations based on the image coordinates. First, we added the columns of latitude and longitude of the inputted image to the dataframe of all the data. Then we drop the rows where column = ‘name’ is NULL, and reset the index.We then add a distance column that calculates the distance of the accommodations from the image location. The distance is calculated using the distance() function (haversine formula). We then also provide the user with a visualization of the map and the locations of the three recommended hotels. Along with this we also point the user out to the top 10 attractions around the area.

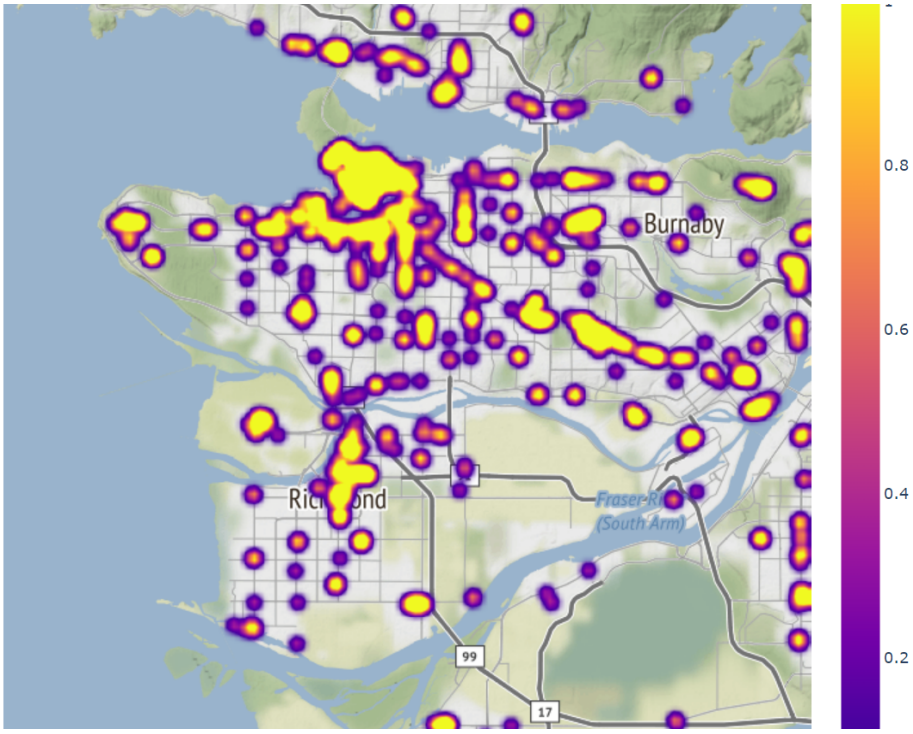


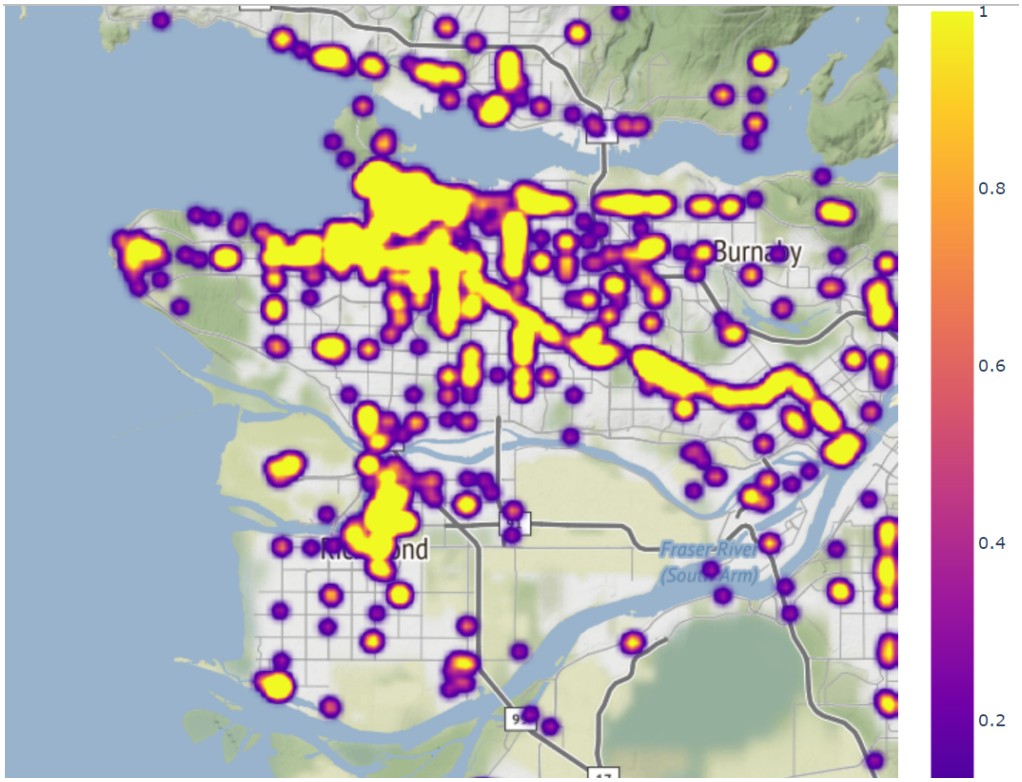
IMG_0520.jpg



We use px.density_mapbox to represent the data collected and grouped of chain restaurants versus non-chain restaurants. For the non-chain restaurants, we drop all the amenities where the

restaurant name is not duplicated. And for chain restaurants we consider cafes, fast food, restaurants, and bistros, where the name is duplicated in the dataset. This visual is of a density heat map. The first heat map represents the Chain restaurants in the city, and the second represents the non-chain restaurants. Through this visualization we are able to point out that there is a higher density of non-chain restaurants as compared to chain-restaurants.





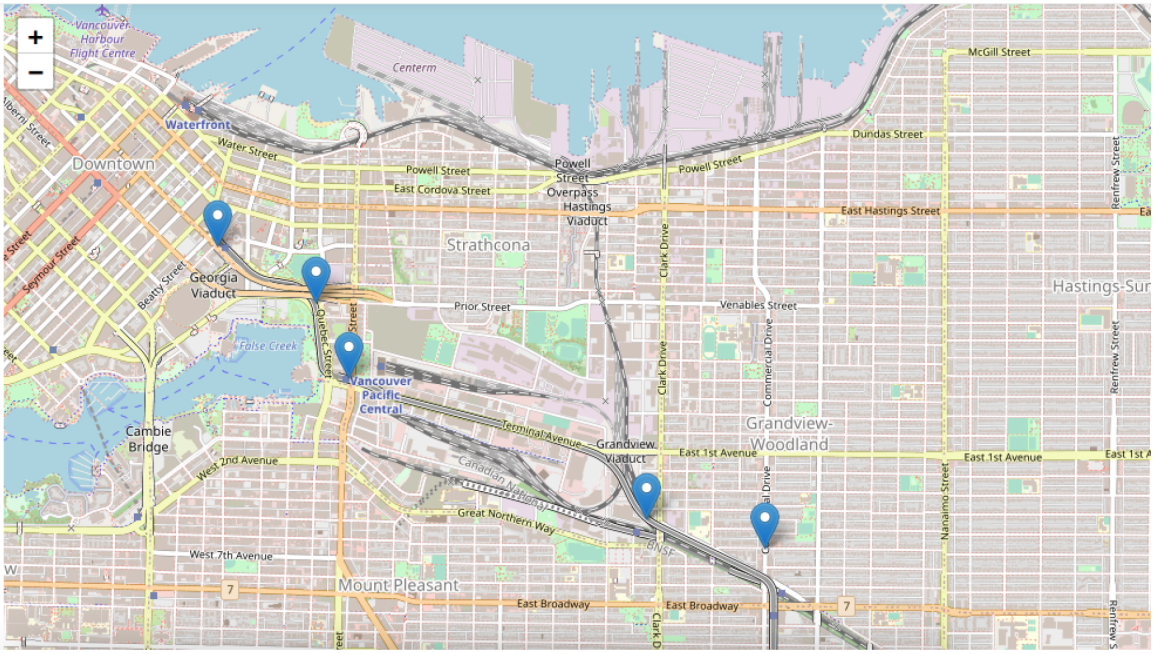
We have a dataframe with the set of all the images the user has taken, by using folium we map each location based on the latitude and longitude of the image. We do this for all the images, and using px.line_mapbox we form a path through these image locations.

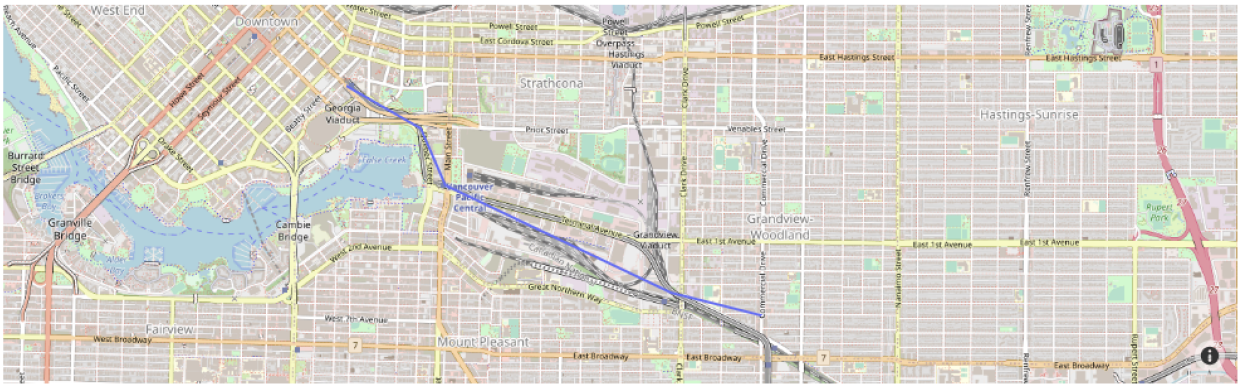
After cleaning up the data we are left with a dataset grouped by cuisines having latitude, longitude, and the user's current location. We use the distance() function based on haversine formula and find the distance of the nearest type of cuisine available to the user's location.We extract the cuisine type from all the 'tags' column, then concat this cuisine column to the main dataframe. Drop all the rows that don't have a 'cuisine', In the end, drop the 'timestamp', 'amenity', 'tags', and 'name'.

```
Recommended Cuisine:                    distance
cuisine
chocolate                0.185197
coffee_shop;italian      0.376796
international;regional   0.397938
kebab;shawarma           0.416499
noodle;rice              0.453848
oyster_bar               0.430495
russian                  0.157246
```

Lastly, we created a file with stored images taken by the user from start to the end of the day. We obtain the image coordinates as mentioned before and store it in a csv file called ‘picture.csv’. These coordinates are used to represent a path taken by the user with the help of as well as to calculate the distance covered by the user in total. Currently the user has uploaded 5 pictures, first we show the location of where each picture was taken. Then, we show the final path of the tour, whichever location the user has been to.

Out[8]:





Limitations

Since we do not have any other basis on which to choose the hotel, like ratings, we have only chosen the hotels that have been segregated by distance.

For the chain restaurants, we have only considered cafes, fast foods, restaurants, and bistros.

Project Experience Summary

Riya Roopesh

CMPT 353 - OSM, Photos, and Tours

- Worked in a 2 person group to analyze various questions raised for the data analysis.
- Gathered data from BC Data Catalogue and appended it to the original dataset.

Tushrima Kelshikar

CMPT 353 - OSM, Photos, and Tours

- Cleaned data for analyzing density heat map to compare chain versus non-chain restaurants in the vancouver region.
- Analyzed, mapped and visualized real world data from OpenStreetMap.

References

https://catalogue.data.gov.bc.ca/dataset/hellobc-accommodations-listing
https://www.fusebulbs.com/how-to-plot-coordinates-on-the-map-python/
https://www.youtube.com/watch?v=BpE9mk1FeJ4
https://www.youtube.com/watch?v=1S4JGjU8x_E
https://plotly.com/python/scattermapbox/
https://wellsr.com/python/plotting-geographical-heatmaps-with-python-folium-module/
https://blog.jovian.ai/interesting-heatmaps-using-python-folium-ee41b118a996
https://stackoverflow.com/questions/4764932/in-python-how-do-i-read-the-exif-data-for-an-image
https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points
https://georgetsilva.github.io/posts/mapping-points-with-folium/