CMPT 353 Project:
OSM, Photos, and Tours
Simon Fraser University

Team Member

Erwin Bai	301262630
Hansen Han	301326746

Overview Of The Project

For this project, we mainly used the data provided from Open Street Map Wiki. Open Street Map contains data(Secondary Data) from the entire world where we could have walked by or seen without even noticing. We can solve critical consumer questions for tourist with the provided longitude and latitude in the OSM Wiki dataset, Airbnb Listing for Vancouver dataset(Secondary Data), and a collection of user's photo. All of these data requires cleaning, extracting, and user input for a desired answer of the question below.

1.Taking the current location, what are some amenity nearby?

2. What is the best Airbnb location with user's price range and amenity nearby?

3. Using a set of tour/location photo taken by the user, what is the total distance they traveled and location they should have seen nearby?.

Methodologies

Gathering Data

The main data set is being provided by Professor Baker who "turned the monolithic XML file" into a more reasonable in JSON format. It can be downloaded from the ProjectTour .

In order to recommend the best hotel based on user desired input, we want a large dataset on Airbnb/Hotel for the tourist to stay. From "Inside Airbnb Adding data to the debate"(See Reference), we were able to find curial Vancouver Airbnb Listings.csv as our second dataset.

To find geographic information in photographs, meaning exif data in JPEG images, we had to take a set of photos from SFU Burnaby campus to Production Way which have hidden Exif information

Cleaning Data

For our first dataset amenities-vancouver.json.gz, we are cleaning out all the invalid entry that have zero use for the user. We dropped all the data that have the name null for the name column since it could be unsafe for a tourist to attend to. The second step is to clean out all the non-interesting amenity such as bench or toilets that provides zero meaning for user to search for. The third step is to drop the column " tag" which is too hard for user to read. From here we can solve a mini question for the user. The fourth step is being decided by user. Any specific name they enter will result in other data being filtered out, but in return, will find a best Airbnb suited for the user. Finally, we created a dataframe with the all the unique place by groupby('amenity').count() for vancouver.

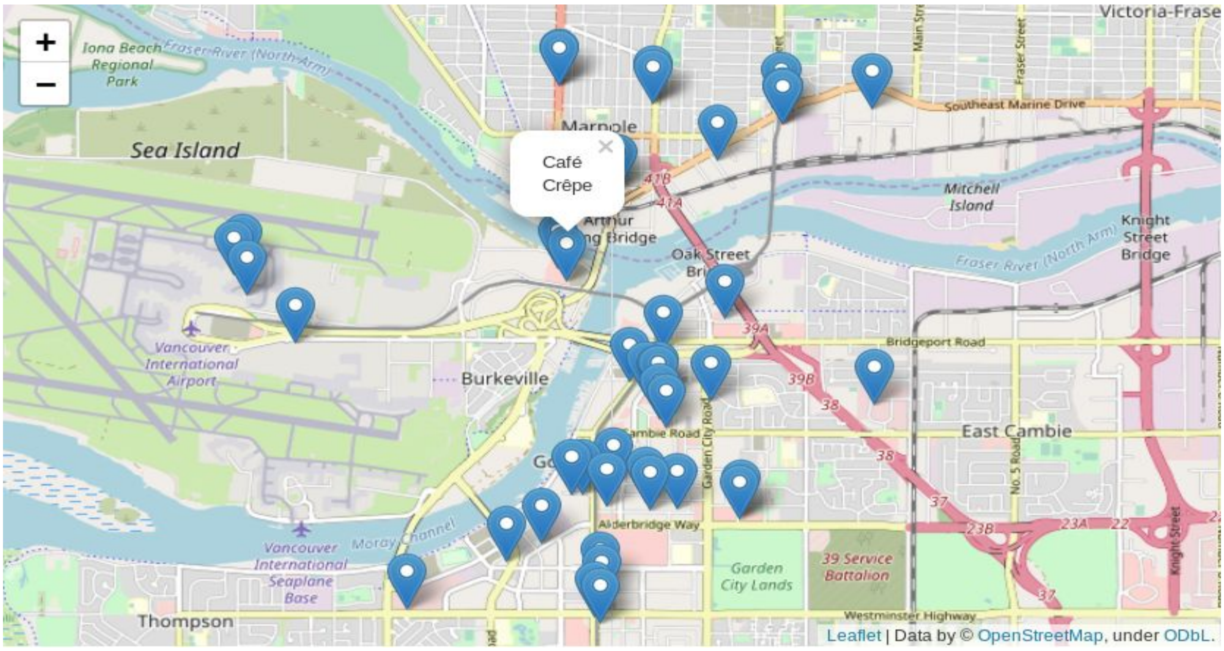
For our second dataset Listings.csv of Airbnb, we deleted few rows in the csv file that have mismatched values which is preventing us to read the file properly. We continue to deleted columns of data that are irrelevant for this analysis, we mainly want to keep the coordinates, name, price and size. The final cleaning step is also for the user to decide. They can choose their own price and number of nearby amenity to select the best Airbnb.

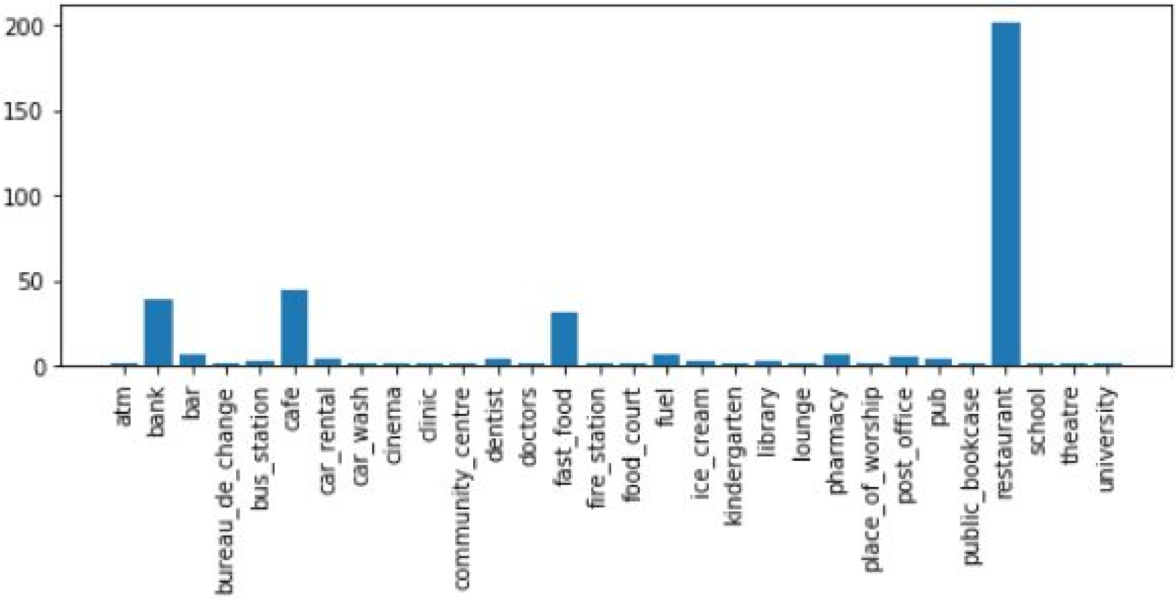
For the third dataset, we extracted our personal photo with the functions provided by Jayson DeLancey (Blog post. See Reference in ReadMe). By using get_geo_tagging(exif), we have some meaningful data such as software,model and location. By converting GPS Latitude and GPS Longitude from degrees,minutes and second to regular latitude and longitude format, we are able to filtered out the unnecessary columns and created our own csv file containing the photo's name, latitude and longitude.

Techniques

In order for the user to find nearby amenity, there are few special cases we have to considered. Our first step was for user to enter his current location and a range in meters so that we can create a **bounding box** with the user as its center. Since the earth is round, we need to **convert meter into longitude and latitude** and calculate the box size. Since now we have a box with user being at its center, we can easily display all the nearby amenity within the range of user's desire.

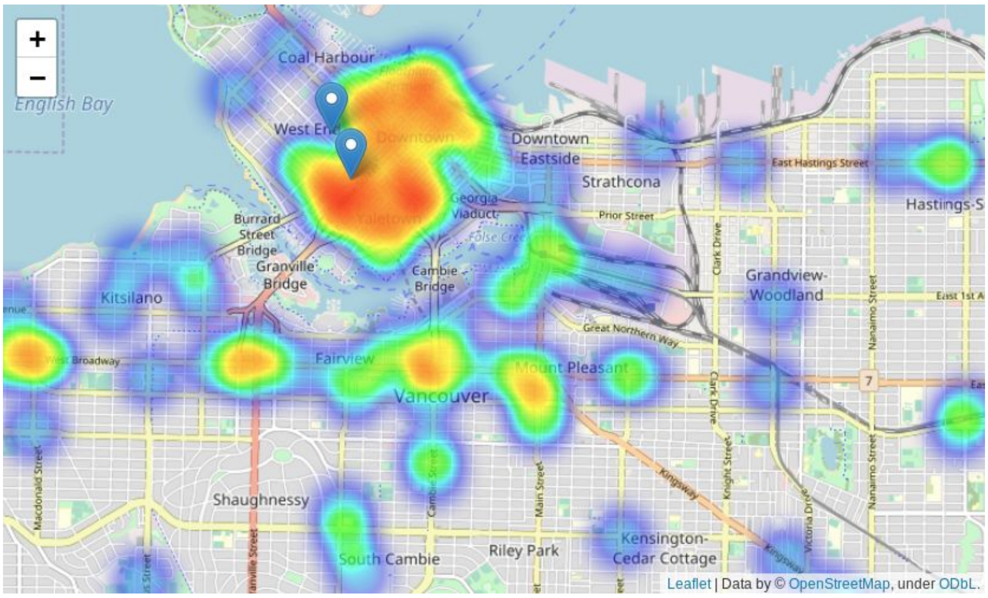
The graph(import folium) and plot below is created by lat = 49.18563055292021, lon = -123.13500846976441, to find nearby amenity within 5000 meter. User can use this to see if there are any nearby location they want to attend to.





We also approach the second question with the idea of bouding box. We allow the user to input their price range for Airbnb to filter out the Listings.csv. Then the user can input their favourite restaurant, cafe or other amenity within certain range of the hotel. By comparing each Airbnb location with user desired amenities nearby, we can recommend the best Airbnb location that satisfy the user(using bounding box).

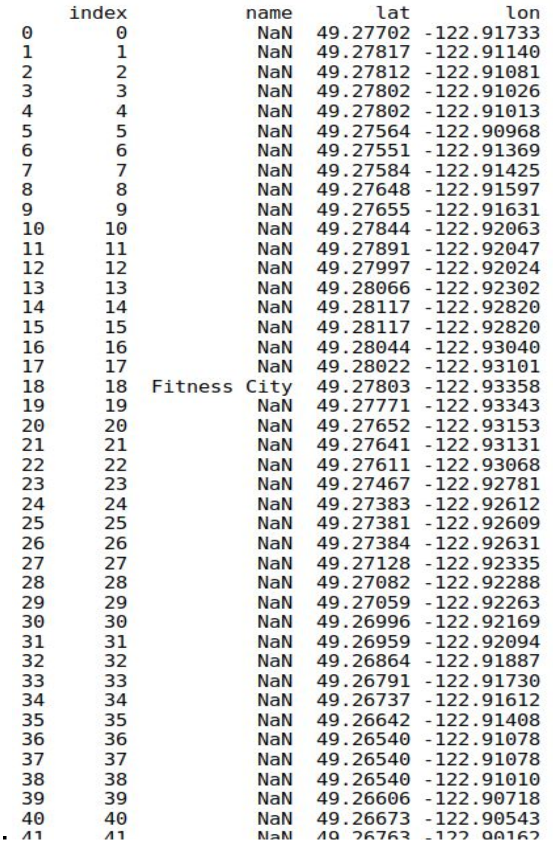
The graph below is created by max_price = 500, min_price = 61, name = KFC, Starbucks and Subway nearby. The heat map represent the density of KFC,Starbucks and Subway while the two point on the map are the Airbnb location that we recommend for the users to book.



Leaflet | Data by © OpenStreetMap, under ODbL.

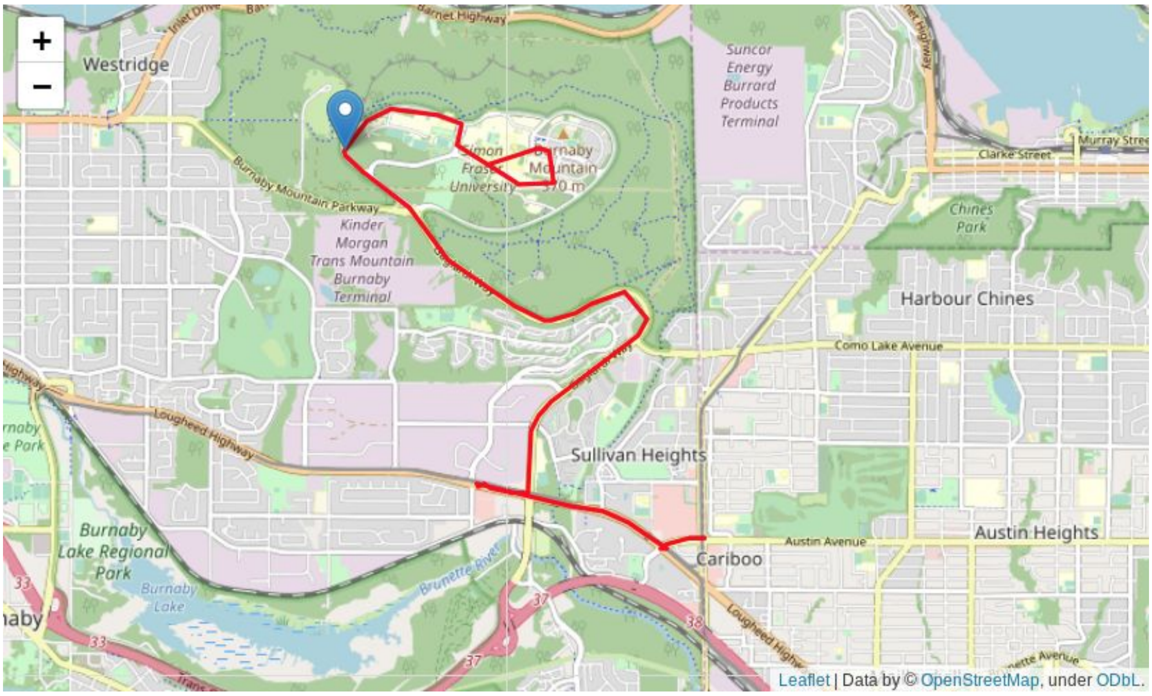
Finally, we also used the idea of bounding box for our third question. After traveling, the user can see the total distance they travel by uploading their image to extract the Exif data. This will create a line between each location of every picture that is taken. However, it is common for traveler to miss unique location that is worth going. Our program will output the unique location(things that vancouver dataset only have one of) that the user should have seen.

The table below is done by creating a bounding box for every location that the user visited. Within a certain range, if a user did not take a photo of a nearby unique place, we will print out the location and name of that location so the user can visit next time they come to vancouver(Example is the ‘Fitness City’ nearby SFU which we did not take a photo of).



```
    index         name      lat        lon
0       0          NaN 49.27702 -122.91733
1       1          NaN 49.27817 -122.91140
2       2          NaN 49.27812 -122.91081
3       3          NaN 49.27802 -122.91026
4       4          NaN 49.27802 -122.91013
5       5          NaN 49.27564 -122.90968
6       6          NaN 49.27551 -122.91369
7       7          NaN 49.27584 -122.91425
8       8          NaN 49.27648 -122.91597
9       9          NaN 49.27655 -122.91631
10     10          NaN 49.27844 -122.92063
11     11          NaN 49.27891 -122.92047
12     12          NaN 49.27997 -122.92024
13     13          NaN 49.28066 -122.92302
14     14          NaN 49.28117 -122.92820
15     15          NaN 49.28117 -122.92820
16     16          NaN 49.28044 -122.93040
17     17          NaN 49.28022 -122.93101
18     18 Fitness City 49.27803 -122.93358
19     19          NaN 49.27771 -122.93343
20     20          NaN 49.27652 -122.93153
21     21          NaN 49.27641 -122.93131
22     22          NaN 49.27611 -122.93068
23     23          NaN 49.27467 -122.92781
24     24          NaN 49.27383 -122.92612
25     25          NaN 49.27381 -122.92609
26     26          NaN 49.27384 -122.92631
27     27          NaN 49.27128 -122.92335
28     28          NaN 49.27082 -122.92288
29     29          NaN 49.27059 -122.92263
30     30          NaN 49.26996 -122.92169
31     31          NaN 49.26959 -122.92094
32     32          NaN 49.26864 -122.91887
33     33          NaN 49.26791 -122.91730
34     34          NaN 49.26737 -122.91612
35     35          NaN 49.26642 -122.91408
36     36          NaN 49.26540 -122.91078
37     37          NaN 49.26540 -122.91078
38     38          NaN 49.26540 -122.91010
39     39          NaN 49.26606 -122.90718
40     40          NaN 49.26673 -122.90543
41     41          NaN 49.26763 -122.90162
```

The graph below is similar to what we done in “GPS Tracks: How Far Did I Walk?”(Excersie 3). By using latitude and longitude from the photo dataset, we can



output the total distance they traveled and output it as a .gpx file. (GPS visualizer to generate the graph, and the point on the graph is the Fitness City we missed)

Conclusions

With the above approach, we hope our program can recommend the best Airbnb/Hotel with user input. During their visitation, we can locate all the nearby location such as restaurant, cafe and other for user to travel to. Finally when they leave Vancouver, we hope to give generated the total distance they traveled from each location by photo and recommend what they miss during their trip.

Limitations

The first limitation is that the data set is too small to get all the information or amenity in Vancouver, like FlyOver Canada, Aquarium and science museum. So the result we got is not that perfect, like there may be more unique places along the path from SFU to lougheed, or there may be more suitable hotel for the user. If we can get a more complete database, we can do better for those problems.

The second limitation is that we design a simple user interface which can let user enter some requirement like the location of the hotel, the price range of the hotel and stores that user needed nearby the hotel, but it is really ugly and difficult to use, and we don’t have enough time to improve it, so we comment them and use a simple default for this project.

Project Experience Summary

Erwin Bai 301262630

OSM, Photo and Tours(CMPT 353) Nov 2019 to Dec 2019

- Built a team around 2 people and start to investigate and analyze the problem
- Gathered data for better analysis of desired question with teammate and written the general idea of how to implement the program
- Cleaned data for better analyzing the question
- Developed and completed the idea of bouding box for data filtering
- Generated Exif information from photo and written the data to CSV
- Caculated the total distance a user traveled
- Outputted the place that the user should have visited
- Generated data visualization such as plot and map

Zelin Han: 301309342

OSM, Photo and Tours(CMPT 353) Nov 2019 to Dec 2019

- Cleaned data for better analyzing the question
- Brainstormed the general idea for implementation of data
- Applied Folium to generate GPS map for better visualization
- Applied bounding box code to complete the question
- Accepted user input to recommand the best hotel that satisfiy their needs.
- Compared two large data set for recommanding of a Airbnb hotel
- Utilize pandas to further separete data into different groups based on amenity
- Design the report format for general audenice to read