ASSIGNMENT 5 : Android app for social distancing

- Harsh Malara

Your idea behind the app:

The app will be mainly be useful while the person is outside interacting in the world, so the app is based on social distancing during a trip.

I used three sensors : Accelerometer , Bluetooth , GPS.
The feature used can be thought of a tuple of:
(Timestamp , (GPS Lat & Long) , num of BT devices discoverable)

My idea is to use the accelerometer to detect if the person is stationary or not, and if the person is stationary : the BT discovery does not occur to reduce the battery required as the data acquired would be repetitive.

The Bluetooth discovery is occurring on a continuous basis, having a Bluetooth discovery discover devices more than a THRESHOLD number can be regarded as violating social distancing.
The score represents this understanding by considering the number of times the violation is occurring from the total number of times.

Score : 100 * (safe discoveries / total discoveries)

UI/UX design:

Keeping in mind the UI should be easy to deal with, I have included only three buttons :
1.Track -
starts the trip for the user (starts storing the feature data as “example.txt” in a local directory)
2.Stop Track -
ends the trip and the file writer is closed
3.Score -
this button can be pressed at any instant to output the score.

The hotspots detected during the journey are mapped with the timestamp & number of BT devices info.

The aim of the app interface is to make it as simple as possible, so people from all walks of life can easily use it.

Some design considerations done:
1. Reduced the frequency of BT discoveries using accelerometer info.

2. The Bluetooth adapter is turned ON before a discovery and turned

OFF after discovery is done to reduce the effective power consumption during the complete trajectory.

3. I have also filtered the BT devices to consider only MOBILE PHONES due to the fact that people may have multiple BT devices. (that part in code is commented as for testing I did not have multiple mobile phones)

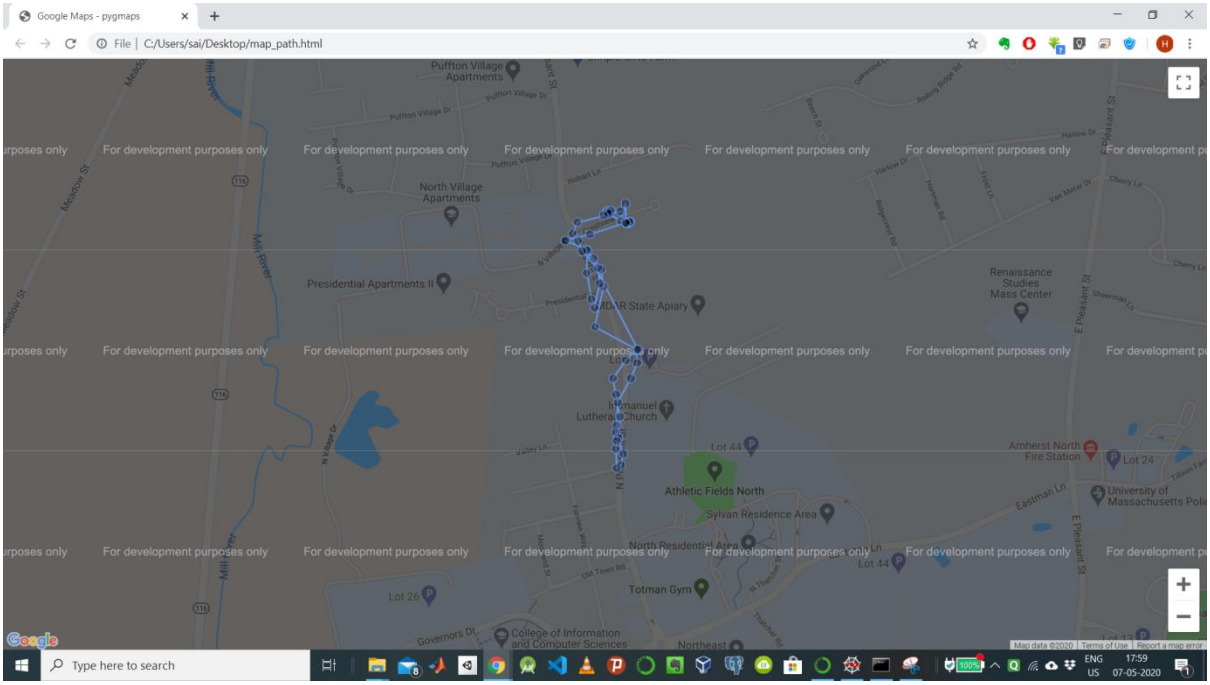
4. The data has no id for privacy reasons, also data starts being stored only when the track button is pressed
(The data stored in “example.txt” is to simulate data sent to the server)

There are two text box that are shown to show the internal data collection of the app for the TA : this would not be included in the main app.

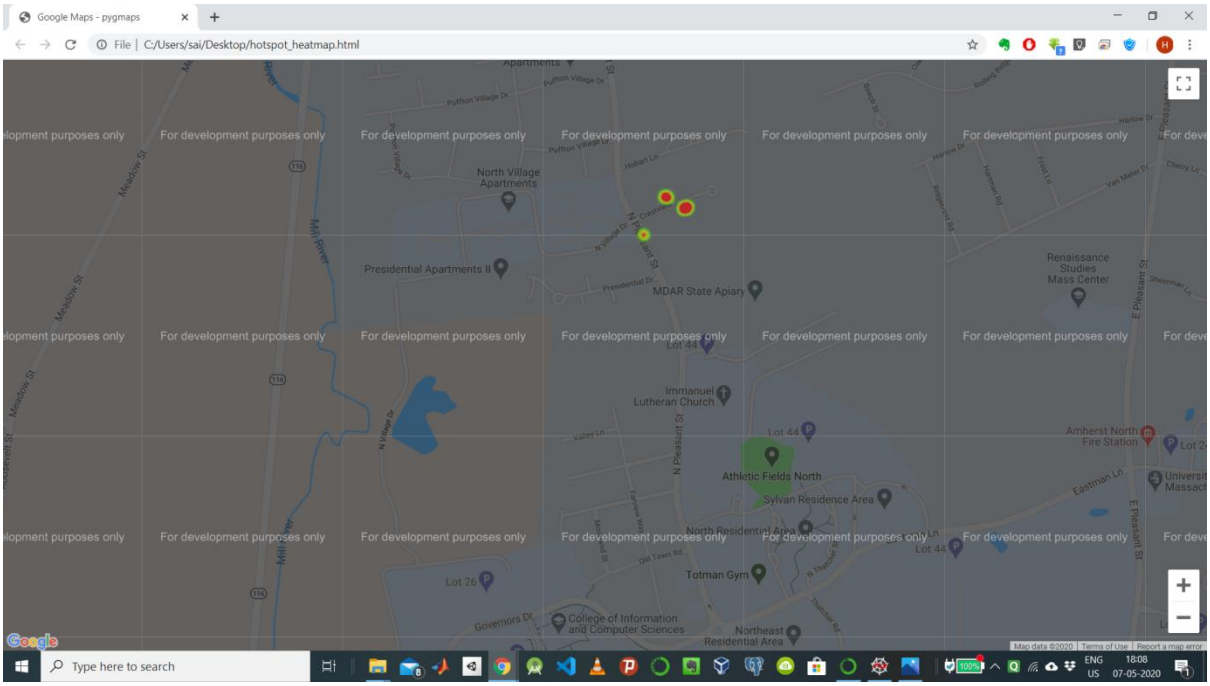
Feature Engineering (Signal Processing and/or Machine Learning):

This part was done by me with the code in “test.py” file. This was done considering that the “example.txt” was retrieved from the server. There are three main aspects to this python script :

1. It gives a visualisation like a dashboard by generating an html file (which is included on the github page). This file is to show the visualise the trajectory of the person :



2. This is on the signal processing side where the location data can be used to create a heat map identifying the hotspots where social distancing is violated, and this is also visualised by a html file with its screen shot below.



3. The problem here is creating a heat map and looking for the hotspots manually can be really time consuming and using the data from the server we can automate this process, using a clustering method DBSCAN which does not require NUM_CLUSTERS as a parameter. (The metric='haversine' is used for computing clusters in the lat,long space)

These clusters are then filtered for noisy and improper clusters and then these GPS locations are printed, with the address.
(The python library used to access address is an offline one and can be replaced with a paid online one to improve accuracy)

The cluster information detected for my collected data is -

```
Number of Computed clusters: 6

Substantial Cluster Means are 2
42.4039524 -72.528379825
[OrderedDict([('lat', '42.41037'),
              ('lon', '-72.53092'),
              ('name', 'North Amherst'),
              ('admin1', 'Massachusetts'),
              ('admin2', 'Hampshire County'),
              ('cc', 'US')])]
```

```
42.40373195 -72.5278392
[OrderedDict([('lat', '42.41037'),
              ('lon', '-72.53092'),
              ('name', 'North Amherst'),
              ('admin1', 'Massachusetts'),
              ('admin2', 'Hampshire County'),
              ('cc', 'US')])]
```

Evaluation:

Evaluation to see the correctness of the app was trivial and could be done my just manipulating with the internal variables.

But evaluation for visualisation and clustering required real data :

The data was collected for a trajectory containing -
1. My place
2. Friends place which is in front of my house
3. A long walk on the main road and back to home.

(1 & 2 were the main hotspots)

This data was collected for duration of about 20 minutes with 63 BT scan data points. This is attached in the “example.txt” file which was used for the visualisation shown in the previous part.

Differentiating your contribution:

1. Added ‘onSensorChange’ and related functions to use the accelerometer sensor data.

2. Automated the process of BT discovery by changing the way the functions were called, also added the feature of enabling and disabling BT, before & after a discovery.

Added extra functionality of detecting is the discovered device is a MOBILE PHONE or not. (Currently commented in the main code)

3. Added the functionality of buttons to start the discovery process and store the data features along the way in a text file. Added a function to output score when the button is called.

4. Added code in ‘onActivityResult’ to print the intermediate steps for the TA and also placing markers with time and number of devices information for the hotspots (based on THRESHOLD)

5. Did the processing of the data recorded using a python script to visualise the hotspots and cluster it using ML to automate the process of hotspot detection.