OpenStreetMap Data Case Study

1. MAP AREA

Kolkata, India

- https://www.openstreetmap.org/export#map=14/22.5204/88.3521

This map is of my hometown, so I'm more interested to see what database querying reveals.

2. PROBLEMS ENCOUNTERED IN THE MAP

After downloading a small sample size of the Kolkata area and running it against data_from_osm.py file, I noticed four main problems with the data, which I will discuss in the following order:

- Incorrect and inconsistent postal code ("7000 026", "700 027")
- Incorrect house number ("+09-03365001122", "15, Lake Place")
- Address in street name ("4, Gorky Terrace, kolkata- 700017", "Alipore, Kolkata", "901A, 9th Floor, Fort Knox, 6, Camac, St. Elgin, Elgin, Kolkata, West Bengal 700017")
- Inconsistent city names ("Kolkata", "kolkata", "Kolkata, West Bengal", " 700016")

Incorrect and inconsistent postal code

Postcode strings posed a problem of unnecessary spaces. This was corrected by removing these spaces.

In one postcode tag, the value was more than 6 digits. This seems to be a typo error of putting extra zeros, so extra zeros were removed.

Pandas library is employed for its useful features. The snippet below shows the different postal codes present in the data

```
n_tags_pd.value[n_tags_pd.key == 'postcode'].unique()
```

The following output is generated:

```
array(['700027', '7000 026', '700019', '700020', '700071', '700029',
       '700014', '700025', '700017', '700151', '700068', '700026',
```

```
       '700016'], dtype=object)
```

Note that the 2nd postcode is '7000 026'. It has an extra zero and a space. The function to clean such entries in the postcode is created as shown below:

```
def postcode_clean(a):
    b = a.index[a.key == 'postcode'] # finding the index where postcode entires are
present
    for i in b: #iterating through the postcode entries
        a.value[i] = a.value[i].replace(' ', '') # removing unnecessary spaces
        if len(a.value[i]) > 6:
            a.value[i] = a.value[i][0]+a.value[i][-5:] # removing extra zeros
#cleaning the n_tags_pd
postcode_clean(n_tags_pd)
n_tags_pd.value[n_tags_pd.key == 'postcode'].unique()
```

The following output is generated after cleaning:

```
array(['700071', '700019', '700045', '700027', '700020', '700026',
       '700023', '700053', '700031', '700017', '700032', '700029'],
      dtype=object)
```

Incorrect House number

House number string had street name in them. This was removed by splitting the string and storing only the house number. The street name was already present in street name field, so it was ignored.

In one house number tag, the value was a phone number. This entry was removed since phone number field had that.

The snippet below shows the node tags entries for a particular id:

```
n_tags_pd[n_tags_pd.id == '5450195823']
```

The output is:

```
id key type value
1197 5450195823 housenumber addr +09-03365001122
1198 5450195823 street addr Raja Subosh Chandra Mallik Road
1199 5450195823 amenity regular restaurant
1200 5450195823 name regular Mission Vegetarian Cafe
1201 5450195823 es name Café Misión Vegetariana
1202 5450195823 opening_hours regular Mo-Su 09:00-21:00
1203 5450195823 phone regular +09-03365001122
```

Note it has phone number in place of housenumber. Similarly, another id with problem in housenumber:

```
n_tags_pd[n_tags_pd.id == '4443890291']
```

The output is:

```
id key type value
882 4443890291 city addr Kolkata
883 4443890291 housenumber addr 15, Lake Place
884 4443890291 postcode addr 700029
885 4443890291 street addr Lake Place
886 4443890291 email regular support@integramicro.co.in
887 4443890291 name regular Integra Micro Systems (P) ltd.
888 4443890291 en name Integra Micro Systems (P) ltd.
889 4443890291 phone regular +91 033 24660363
890 4443890291 website regular https://www.integramicro.com/
```

Note "Lake Place" is in streetname as well as in housenumber. To clean such entries of housenumber the following function is created:

```
def housenumber_clean(a):
    b = a.index[a.key == 'housenumber']
    for i in b:
        a.value[i] = a.value[i].split(',')[0]
        if a.value[i][0] == '+':
            a.drop(i, axis = 0, inplace = True)
```

Address in street name

In few entries, it was observed that instead of adding address in multiple tags, it is added in street name itself. This was challenging as there was no specific format of these addresses. However, addresses did have 'Kolkata' in it. This was used to remove the city name and the following part of the string (containing state name and postcode). New tags of postcode were added where ever the value was available.

Since there was no specific format in the initial part of the string, floor number, house number remained a part of street name in the final database.

Inconsistent city name

The problem was in few city tags value was in lowercase while in rest was having first letter in uppercase. This was corrected by changing first letter to uppercase for all.

In one city tag, the value had "Kolkata" along with state name "West Bengal". The sate name was removed to maintain the consistency of the tag value. In one city tag, the value had postcode "700016". This was replaced by "Kolkata".

3. DATA OVERVIEW

This section contains basic statistics about the dataset, the SQL queries used to gather them.

3.1 FILE SIZES

```
charlotte.osm ......... 53 MB
charlotte.db .......... 27 MB
nodes.csv ............. 20 MB
nodes_tags.csv ........ 0.063 MB
ways.csv .............. 3 MB
ways_tags.csv ......... 2 MB
ways_nodes.cv ......... 7 MB
```

3.2 NUMBER OF NODES

```
sqlite> SELECT COUNT(*) FROM nodes;
```

249619

3.3 NUMBER OF WAYS

```
sqlite> SELECT COUNT(*) FROM ways;
```

49216

3.4 NUMBER OF UNIQUE USERS

```
sqlite> SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
```

321

3.5 TOP 10 CONTRIBUTING USERS

```
sqlite> SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
ORDER BY num DESC
LIMIT 10;
```

hareesh11,62187

udaykanth,40069

Rondon237,37996

thrinath,32464

vamshiN,31688

harishk,9205

saikumar,8565

vikramsingh,8108

Apreethi,7658

venkatkotha,6318

3.6 NUMBER OF USERS APPEARING ONLY ONCE (HAVING 1 POST)

```
sqlite> SELECT COUNT(*)
FROM
    (SELECT e.user, COUNT(*) as num
     FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
     GROUP BY e.user
     HAVING num=1)  u;
```

82

4. ADDITIONAL IDEAS

This section contains some additional ideas about the data and few suggestions.

4.1 CONTRIBUTOR STATISTICS AND SUGGESTIONS

The contributions of users seem incredibly skewed. Here are some user percentage statistics:

- Top user contribution percentage: (“hareesh11”) 24.9%
- Combined top 5 users' contribution: 81.9%
- Combined Top 10 users contribution: 97.8%
- Combined number of users making up only 100 posts or less: 270 (about 84% of all (321) users)

To increase contribution from users of OpenStreetMap, user data could be displayed more prominently like weekly leaderboard per city or State, highlighting some users who have contributed the most edits to the map. Another way to engage users could be

by giving goodies like t-shirts, coffee mugs etc. to users who reach a milestone such as submission of 1000 or 10,000 edits.

4.2 TOP 10 APPEARING AMENITIES

```
sqlite> SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10;
```

restaurant,37
atm,18
fast_food,14
fuel,13
bank,11
cafe,11
cinema,9
hospital,9
place_of_worship,9
college,7

4.3 MAX SPEED

The city has a speed limit of 40KMPH for most parts of the it.

```
sqlite> SELECT value as num, COUNT(*)
FROM ways_tags
WHERE key='maxspeed'
GROUP BY value
ORDER BY num DESC;
```

60,4
40,293
30,14
20,1

5. CONCLUSION

After this review of the data I believe that the Kolkata area has been already well cleaned. There were only few values which I have to clean for the purposes of this exercise. It also revealed that the city has the max speed limit of 40KMPH for most part of it.