Wrangling data with Openstreet Map

Why I chose Pune?

1. Staying here since 3 years, hence would be easier to spot trivial errors
2. Though this would not be the case always, it's better to start with familiar set of data

Contents of the Repository

- README.md: Description of the project
- openstreet_map.py: Main Code used for cleaning/auditing data and generating csv files
- pune_india.osm.bz2: Compressed format of the complete data for Pune region (Uncompressed=300M)
- pune_india_sample.osm: Sample data
- sample.py: Code to generate sample data from huge dataset
- schema.py: Schema corresponding to the mysql schema

Know your Data

The osm file consists of 3 main tags; **nodes**, **relations**, **ways**. We will focus on ways and nodes. More about openstreet map can be found on osm wiki page

Nodes

Static points mapped using latitude and longitude geographically. For ex: McDonalds at a certain street will be mapped as a node with unique node 'id' and will have latitude and longitude to map its location geographically

```
<node id="2183530544" lat="18.5520773" lon="73.8925318" version="1"
timestamp="2013-03-03T13:05:44Z" changeset="15233276" uid="78432"
user="aveekbh">
  <tag k="name" v="Muttha Towers"/>
  <tag k="building" v="office"/>
  <tag k="entrance" v="yes"/>
  <tag k="wheelchair" v="yes"/>
  <tag k="addr:street" v="Don Bosco Marg"/>
  <tag k="addr:postcode" v="411006"/>
</node>
```

Ways

Simply put, if 2 nodes are connected by a path, then the path is nothing but a way. Ways can be either closed (starting node is the ending node) and open ways (starting and ending nodes may be geographically apart).

```
<way id="208055626" version="1" timestamp="2013-03-03T13:05:45Z"
changeset="15233276" uid="78432" user="aveekbh">
  <nd ref="2183530544"/>
  <nd ref="2183530561"/>
</way>
```

In the above samples of osm data, the `node's` `id=2183530544` is also present as a reference in the `way` tag, you can see the `nd ref=2183530544`, which refers to the node's id from node tag.

Data Wrangling

1. Street Names

Pune is a city in Maharashtra where Marathi is spoken majorly. 'Road' in English is equivalent to 'Marg' or 'Path' in Marathi.

Below are some of the occurances:

```
<tag k="addr:street" v="Maharana Pratapsinh Marg"/>
<tag k="addr:street" v="Maharana Pratapsinh Marg"/>
<tag k="addr:street" v="G. A. Kulkarni Path"/>
<tag k="addr:street" v="Pu. Bha. Bhave Path"/>
```

So, I have set a mapping dictionary to map linguistic equivalents of Road to 'Road'. In the below cases, all the entries ending with Marg or Path will now end with Road. This step is taken to just standardize the data.

```
mapping = {
    "Rd": "Road",
    "Path": "Road",
    "Marg": "Road",
    "road": "Road"
}
```

2. Postal code

There were entries where postal code seemed malformed in terms of extra spaces. So to ensure other erroneous postal codes dont bother us later, I wrote a regex which would validate the length=6, starts with 411, and if starts with 411 and has space,underscore,hypen in between, then these characters will be trimmed off and we get the postal code.

```
<tag k="addr:postcode" v="411 021"/>
<tag k="addr:postcode" v="411 046"/>
<tag k="addr:postcode" v="411 004"/>
<tag k="addr:postcode" v="411 016"/>
```

The “addr:postcode” is the standard usage and at some places the entry seems to be “addr:postal_code” which is again normalized with a postcode_mapper dictionary, where every occurance of postal_code will be replaced with standard postcode. This makes querying the database easier.

```
<tag k="postal_code" v="411001"/>
<tag k="postal_code" v="411046"/>
<tag k="postal_code" v="411027"/>
<tag k="postal_code" v="410506"/>
```

3. Phone numbers

Phone numbers can be either landaline or mobile, with and without std/isd codes. My way of standardizing this is, by just modifying the phone numbers to give only the user part i.e. exclude the ISD and STD codes, as we are aware which part of the world the data is coming from.
For ex:

```
+91-9876543210 is converted to 9876543210
```

```
+91 20 23442344 is converted to 23442344
09850985012 is converted to 9850985012
```

Drawing insights out of data from database

Top 10 contributors from Pune

```
mysql> SELECT u.user Name, COUNT(*) Total_Contributions FROM (SELECT user FROM
nodes UNION ALL SELECT user FROM ways) u GROUP BY 1 ORDER BY Total_Contributions
DESC LIMIT 10;
+---------------+---------------------+
| Name          | Total_Contributions |
+---------------+---------------------+
| singleton     |               98596 |
| harishvarma   |               60185 |
| jasvinderkaur |               57758 |
| sramesh       |               57663 |
| praveeng      |               56795 |
| shiva05       |               51910 |
| anushapyata   |               49537 |
| kranthikumar  |               47503 |
| harishk       |               43323 |
| saikumar      |               40371 |
+---------------+---------------------+
10 rows in set (9.71 sec)
```

Popular Cuisine

```
mysql> SELECT nt.value Cuisine, COUNT(*) as Total FROM nodes_tags nt, nodes_tags
nts WHERE nt.id = nts.id AND nts.value = 'restaurant' AND nt.tkey = 'cuisine'
GROUP BY 1 ORDER BY 2 DESC;
+--------------------------------------------------------------------------------
-----------------------------------------------------------------+-------+
| Cuisine
| Total |
+--------------------------------------------------------------------------------
-----------------------------------------------------------------+-------+
| indian
|    44 |
| vegetarian
|    13 |
| pizza
|    10 |
| regional
|     8 |
| international
|     5 |
| chinese
|     3 |
| italian
|     3 |
| barbecue
|     3 |
| burger
|     2 |
| seafood
|     1 |
| thai
|     1 |
| kebab
|     1 |
| doughnut
|     1 |
| chinese;indian
|     1 |
| indianstreetfood,_kathi_kebabs, chaat, grilled sandwiches, coffee, muffins,
brownies, eclairs, pav bhaji, pulao, biryanis, samosas, beverages, |     1 |
| regional,wraps
|     1 |
| regional,gujarati
|     1 |
| sizzlers
|     1 |
| regional,_arabic
|     1 |
| North_Indian
|     1 |
| Regional,_India,_Tandoor,_Chinese
|     1 |
| Multi-Cuisine
|     1 |
| italian,_Pizza,_Pasta,_Mexican,_Lebanese
|     1 |
+--------------------------------------------------------------------------------
-----------------------------------------------------------------+-------+
23 rows in set (0.00 sec)
```

Most followed Religion

```
mysql> SELECT nt.value Religion, COUNT(*) as Total FROM nodes_tags nt,
nodes_tags nts WHERE nt.id = nts.id AND nts.value = 'place_of_worship' AND
nt.tkey = 'religion' GROUP BY 1 ORDER BY 2 DESC LIMIT 1;
+----------+-------+
| Religion | Total |
+----------+-------+
| hindu    |    74 |
+----------+-------+
1 row in set (0.01 sec)
```

Top 10 amenities

```
mysql> SELECT value Amenity, COUNT(*) as Total FROM nodes_tags WHERE
tkey='amenity' GROUP BY 1 ORDER BY 2 DESC LIMIT 10;
+------------------+-------+
| Amenity          | Total |
+------------------+-------+
| restaurant       |   221 |
| bank             |   143 |
| atm              |   119 |
| place_of_worship |    99 |
| cafe             |    72 |
| fast_food        |    67 |
| hospital         |    42 |
| fuel             |    40 |
| school           |    36 |
| police           |    29 |
+------------------+-------+
10 rows in set (0.02 sec)
```

Number of unique users

```
mysql> SELECT COUNT(distinct(e.uid)) Distinct_Users FROM (SELECT uid FROM nodes
UNION ALL SELECT uid FROM ways) e;
+----------------+
| Distinct_Users |
+----------------+
|            550 |
+----------------+
1 row in set (0.00 sec)
```

No. of nodes

```
mysql> SELECT COUNT(*) Nodes FROM nodes;
+---------+
| Nodes   |
+---------+
| 1390911 |
+---------+
1 row in set (0.40 sec)
```

No. of ways

```
mysql> SELECT COUNT(*) Ways FROM ways;
+--------+
| Ways   |
+--------+
| 263956 |
+--------+
1 row in set (0.08 sec)
```

List of zip codes

```
mysql> SELECT distinct(value) Zip_Codes FROM nodes_tags WHERE tkey like '%post
%';
+----------------+
| Zip_Codes      |
+----------------+
| 411001         |
| 411046         |
| 411027         |
| 410506         |
| 411013         |
| 411014         |
| 411040         |
| 412300         |
| 412208         |
| 412106         |
| 411004         |
| 411018         |
| 411021         |
| 411048         |
| 411009         |
| 411052         |
| 411041         |
| 411051         |
| 411028         |
| 411008         |
| 411002         |
| 412200         |
| 411007         |
| 431027         |
| 413102         |
| 411038         |
| 411033         |
| 411005         |
| 411057         |
| 411016         |
| 410500         |
| 411042         |
| 411030         |
| 411006         |
| 411011         |
| 411045         |
| 411029         |
| Paschimanagari |
| 411015         |
| 411 021        |
| 411020         |
| 412101         |
| 412105         |
| 411043         |
| 411036         |
| 411037         |
| 411 046        |
| 411060         |
+----------------+
48 rows in set (0.01 sec)
```

* There is still some cleaning to be done, as we can see that there are problematic (letters and spaces) character in postal code; though the occurrence is very less *

Top 10 places from where contributions have been done

```
mysql> SELECT value Place, COUNT(*) Total FROM (SELECT value, tkey FROM
ways_tags UNION ALL SELECT value, tkey FROM nodes_tags ) u WHERE u.tkey =
'street' GROUP BY 1 ORDER BY 2 DESC LIMIT 10;
+---------------------------+-------+
| Place                     | Total |
+---------------------------+-------+
| Karve Road                |    20 |
| Paud Road                 |    20 |
| Pashan-Sus Road           |    17 |
| Sinhagad Road             |    17 |
| NIBM Road                 |    17 |
| Kondhwa Road              |    17 |
| MIT College Road          |    13 |
| Lane Number 2             |    11 |
| Vitthal Rao Shivarkar Road |    11 |
| Gokhale Road              |     9 |
+---------------------------+-------+
10 rows in set (0.51 sec)
```

Challenges Faced

Errors

1. Error: Duplicate entry for '2147483647' for key 'PRIMARY'.
I had to change the datatype of 'id' column in nodes table from INTEGER to BIGINT as it exceeded the limit. You can refer below link for more details: http://stackoverflow.com/questions/18643648/mysql-insert-query-returns-error-1062-23000-duplicate-entry-2147483647-for

2. Error: Mysql threw unique key constraint errors when trying to dump nodes_tags.csv in table using mysql prompt LOAD cmd and skipped 3 rows when tried the same using mysqlimport.

Resolution

So, looking at the error message, one problem could have been that,
- primary key was duplicate
- the id in nodes_tags which was referred to in nodes was not present

For this,
- I first used some unix tools like 'awk' to get only the id from nodes.csv and nodes_tags.csv. Later inserted them into temporary tables without PRIMARY KEY and wrote a query to check whether data was redundant. Found out that the rows were getting skipped due to ',' being present in them and had to escape them.
- The below cmd skipped the problematic rows

```
> sudo mysqlimport --ignore-lines=1 --fields-terminated-by=',' --verbose --local -u root data_wrangling_schema /var/lib/mysql-files/nodes_tags.csv
```

- The cmd was modified to include below optional option where we mention that fields may be enclosed by '"' which was the case and it resolved the issue

```
> sudo mysqlimport --ignore-lines=1 --fields-terminated-by=',' --fields-optionally-enclosed-by='"' --verbose --local -u root data_wrangling_schema /var/lib/mysql-files/nodes_tags.csv
```

Additional Stats

- As good as 40% data comes from the top 10 contributors on the list pasted above.
- For people to contribute more to the openstreet map, there should be conventions/groups which can motivate others to contribute
- Gamification will ensure people contribute more to the OSM

Conclusion

I still feel that the data is very immature (at least for Pune). Gamification (in terms of credits and leaderboard stats) can be an important pillar which can make users contribute more to the project. Rating systems can be deployed once the data gets into a better shape, the way it is possible with Google Reviews.

References

- https://gist.github.com/carlward/54ec1c91b62a5f911c42file-sample_project-md
- http://www.thegeekstuff.com/2008/10/import-and-upload-data-to-mysql-tables-using-mysqlimport/

- https://docs.python.org/2/library/collections.htmlcollections.defaultdict
- http://www.jeannicholashould.com/tidy-data-in-python.html
- https://help.github.com/articles/basic-writing-and-formatting-syntax/
- http://www.rubycoloredglasses.com/2013/04/languages-supported-by-github-flavored-markdown/