Shalini Ruppa Subramanian
Apr 1, 2016

OpenStreetMap Project
Data Wrangling with MongoDB

https://mapzen.com/data/metro-extracts/
San Jose, California is used. San Jose osm data was downloaded. I chose this region since I live around here.

https://www.openstreetmap.org/relation/112143

1. Problems encountered

After downloading the san jose california osm file, the fields address, street names, city names and postal code were audited using audit.py.

In most of the cases, the address fields were “addr:attribute” format where the attribute is city, postcode or street name. In a few cases, the entire address was captured in the “address” field and transforming it to a similar structure as “addr:attribute” was skipped.

There were abbreviations in street names like Rd., Blvd, Dr, and they were fixed using a mapping dictionary.

The postal codes in San Jose is a five letter zip code. Sometimes they are in the format (“95014-0548”, CA 94805). In the first case, the post code after the hyphen was removed and in the second case, the country name (‘CA’) was removed. These were the most common errors and they were fixed.

The city names was audited and spelling errors, inconsistency of letter cases were found. (eg. Campbelll instead of Campbell, santa Clara, Santa Clara). They were fixed using a mapping dictionary.

The data was cleaned up using shape_element function in data.py file. As each line is read from the osm file, shape element function is called and each string is converted to a dictionary.

2. Data Overview

This section contains basic statistics about the dataset and the MongoDB queries run to gather them. Some of the queries were done in mongo shell and the aggregate operators were done using pymongo. The queries are also listed in query.py file.

```
# File Sizes
san-jose_california.osm……271 MB
san-jose_california.json...333 MB

# Number of documents
```

```
db.san_jose_sample.find().count()
1398613

# Number of nodes
db.san_jose_sample.find({"type":"node"}).count()
1240140

# Number of ways
db.san_jose_sample.find({"type":"way"}).count()
158436

# Number of distinct users
db.sanjose_sample.distinct("created.user").length
1220

# Top five users and the post counts
db.sanjose_sample.aggregate[{"$group":{"_id": "$created.user",
                                "count":{"$sum":1}}},
                        {"$sort":{"count":-1}},
                        {"$limit":5}]

[{u'_id': u'nmixter', u'count': 281595},
 {u'_id': u'mk408', u'count': 153592},
 {u'_id': u'Bike Mapper', u'count': 80786},
 {u'_id': u'samely', u'count': 77422},
 {u'_id': u'RichRico', u'count': 69927}]


# Number of users with one posts
db.sanjose.aggregate([{"$group":{"_id": "$created.user",
                        "count":{"$sum":1}}},
                        {"$group":{"_id": "$count",
                                   "num_users":{"$sum":1}}},
                        {"$sort":{"_id":1}},
                        {"$limit":1}]

[{u'_id': 1, u'num_users': 292}]
```

I chose to explore the number of schools in San Jose, given that Cupertino is one of the competitive school districts.

```
db.sanjose.find({"name":{"$regex":"[Ss]chool"}}).count()
427
db.sanjose.find({"name":{"$regex":"[Ee]lementary [Ss]chool"}}).count()
176
db.sanjose.find({"name":{"$regex":"[Mm]iddle [Ss]chool"}}).count()
43
db.sanjose.find({"name":{"$regex":"[Hh]igh [Ss]chool"}}).count()
```

```
# Top 10 amenities
db.sanjose.aggregate([{"$match" : {"amenity" : {"$ne" : None}}},
                  {"$group":{"_id": "$amenity",
                   "count":{"$sum":1}}},
                  {"$sort":{"count":-1}},
                  {"$limit":10}])
```

```
[{u'_id': u'parking', u'count': 1664},
 {u'_id': u'restaurant', u'count': 858},
 {u'_id': u'school', u'count': 543},
 {u'_id': u'fast_food', u'count': 428},
 {u'_id': u'place_of_worship', u'count': 343},
 {u'_id': u'cafe', u'count': 222},
 {u'_id': u'fuel', u'count': 220},
 {u'_id': u'bank', u'count': 174},
 {u'_id': u'bench', u'count': 170},
 {u'_id': u'toilets', u'count': 166}]
```

It is interesting to note parking is the top amenity here. As the place is quite populated, it can be relatively difficult to find parking lots.

```
# Top 10 cuisines
db.sanjose.aggregate([{"$match": {"$and":
                  [{"amenity": {"$eq": "restaurant"}}, {"cuisine": {"$ne":
            None}}]}},
                  {"$group": {"_id": "$cuisine",
                              "count": {"$sum": 1}}},
                  {"$sort": {"count": -1}},
                  {"$limit": 10}])
```

```
[{u'_id': u'mexican', u'count': 76},
 {u'_id': u'chinese', u'count': 62},
 {u'_id': u'vietnamese', u'count': 52},
 {u'_id': u'pizza', u'count': 52},
 {u'_id': u'japanese', u'count': 39},
 {u'_id': u'american', u'count': 34},
 {u'_id': u'indian', u'count': 33},
 {u'_id': u'italian', u'count': 32},
 {u'_id': u'thai', u'count': 24},
 {u'_id': u'sushi', u'count': 21}]
```

Cuisines was not available for a number of restaurant fields. Looking at the popular cuisines, it can be inferred that Mexican Americans and Asians are predominant in San Jose [5].

3. Other ideas of the datasets

In examining the cities in San Jose, I came across a city named ‘Kayseri’, which is a city in Turkey. It was somehow showing up in the open street map dataset with a country name as code ‘TR”. Similarly other not recognisable cities like Dukla, Felton, Lazkao were in the cities dataset. This could be due to the large number of users manually inputting the data. To reduce this, automation could be done.
For places with lot of data, example, San Jose dataset or the San Francisco dataset, the user could be led to choose a city name from a list of city values applicable for that area. In this way, the manual entry could be limited. To build on a predefined list of city values, code could be written such that the list is automatically populated based on the previous entries. This is assuming the previous entries are audited and cleaned regularly.

When I looking up on the schools information in the open street map, we could expand the school categories to include the type of schools. There are various type of schools like public, charter, magnet, religious schools etc. The school rating information could also be included. These information could be very useful for parents having kids and choosing schools for their children.
However, when I read the document on Open street Map completeness [7], it specified licensing issues when obtaining information such as school rating and school type from third party websites. This could be a potential issue to consider how to work around this.

References

1. Udacity Lectures and Discussion Forum
2. http://wiki.openstreetmap.org/wiki/Main_Page
3. Stackoverflow
4. MongoDB tutorial page
5. http://censusreporter.org/profiles/16000US0668000-san-jose-ca/
6. http://www.school-ratings.com/
7. http://wiki.openstreetmap.org/wiki/Completeness