Project A: Twitter Dataset Analysis

INFO-I 535 BIG DATA

Project A

Data Science Residential

Fall 2017

Indiana University

Bloominton, IN, USA

Jinju(Hellen) Jiang
2000276677
heljiang@iu.edu
Oct 21, 2017

Contents

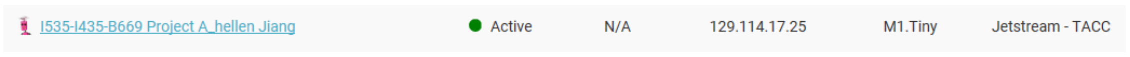
1. Abstract

In this project I used 10000 user profiles dataset which was created and cleaned by researchers at the University of Illinois in May 2011. By invoking tools to clean data and import it into the MongoDB database that is running on Jetstream VM, I manually executed data pipleline and identified locations and displayed them visually on google map.

2. Introduction

2.1 Virtual Machine in Jetstream

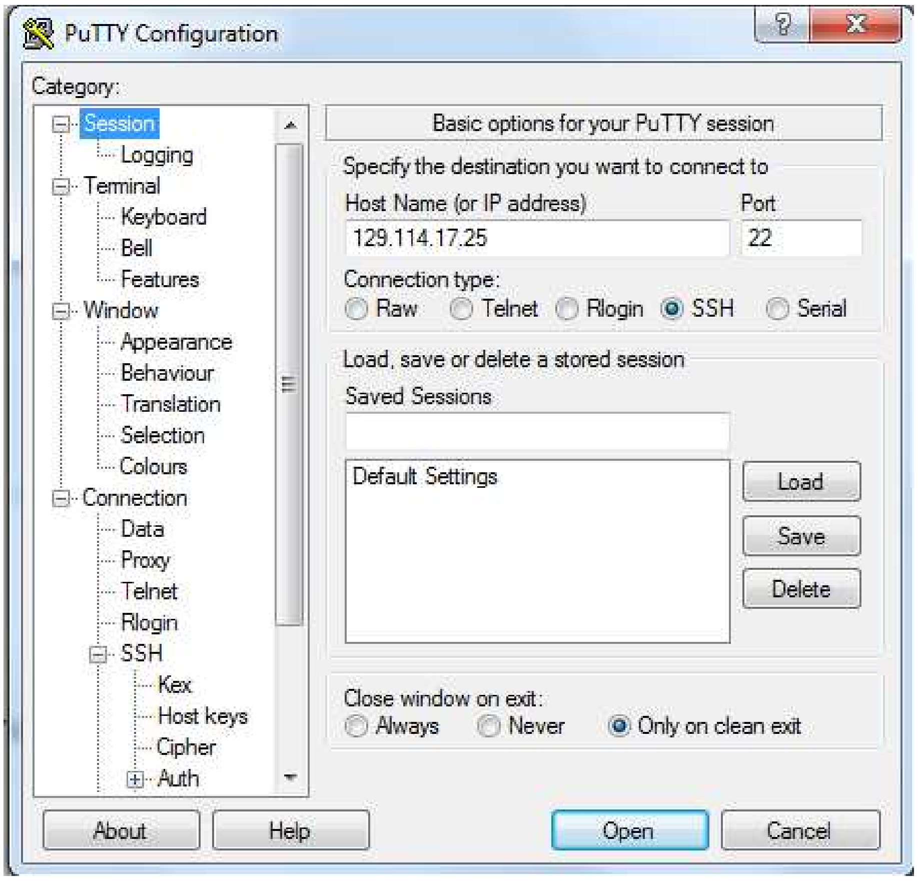
1. Tiny Machine in Jetstream



I535-I435-B669 Project A_hellen Jiang	● Active	N/A	129.114.17.25	M1.Tiny	Jetstream - TACC

2. Setup VM

❖ Connect via PuTTY by typing IP address and upload private key. Then type username and password.



PuTTY Configuration

Category: Session

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 129.114.17.25

Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

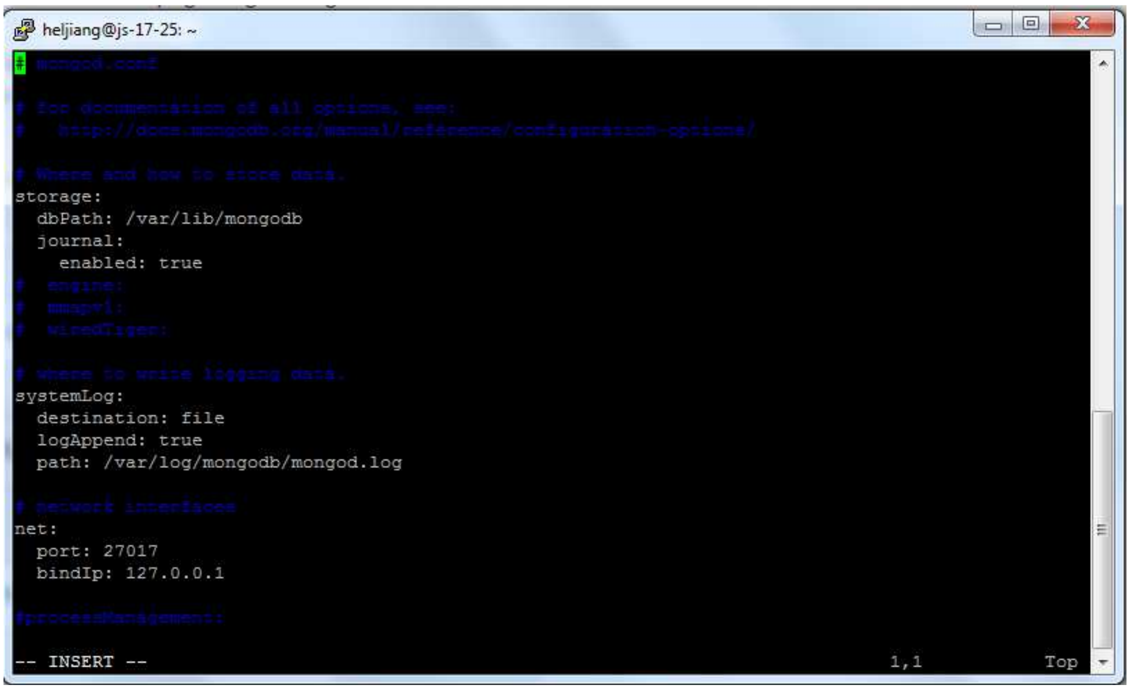
Saved Sessions

Default Settings

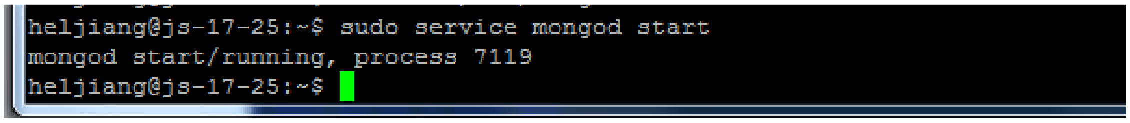
Close window on exit: Always Never [✓ Only on clean exit]

❖ Disable security in in MongoDB:

- sudo service mongod stop
- sudo vi /etc/mongod.conf
- #security:
- # authorization: enabled
- Hit Esc, then type :wq
- sudo service mongod start

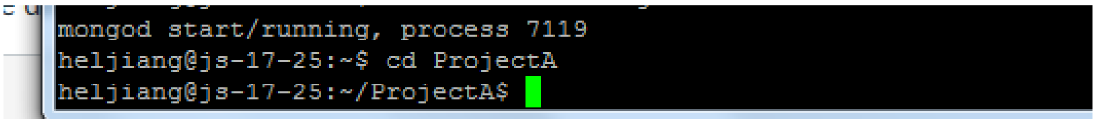


```
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: /var/lib/mongodb
  journal:
    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1

#processManagement:

-- INSERT --                                                         1,1          Top
```



```
heljiang@js-17-25:~$ sudo service mongod start
mongod start/running, process 7119
heljiang@js-17-25:~$
```

Create project directory: mkdir Project A



```
mongod start/running, process 7119
heljiang@js-17-25:~$ cd ProjectA
heljiang@js-17-25:~/ProjectA$
```

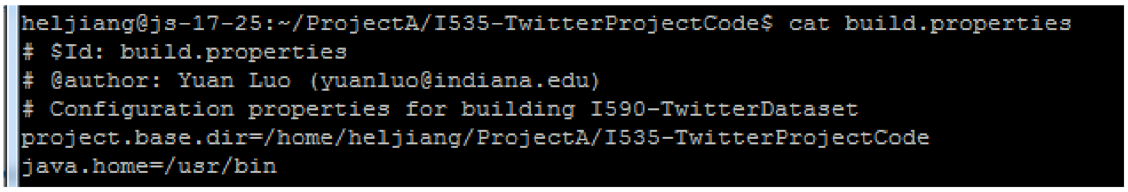
2.2 Build Pipeline Tools in VM

Step 1 - Extract the tools from their zipped and tarred package:

```
tar -zxf I535-TwitterProjectCode.tar.gz

cd I535-TwitterProjectCode
```

Step 2 - Modify configuration file via VI editor

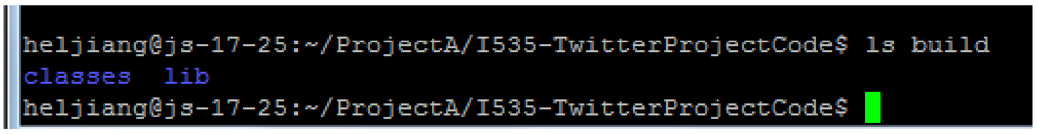
```
# $Id: build.properties

# @author: Yuan Luo

# Configuration properties for building I535-TwitterProjectCode

project.base.dir=/home/username/Project/I535-TwitterProjectCode

java.home=/usr/bin
```



```
heljiang@js-17-25:~/ProjectA/I535-TwitterProjectCode$ cat build.properties
# $Id: build.properties
# @author: Yuan Luo (yuanluo@indiana.edu)
# Configuration properties for building I590-TwitterDataset
project.base.dir=/home/heljiang/ProjectA/I535-TwitterProjectCode
java.home=/usr/bin
```

Step 3 - Build java software

```
ant
```



```
heljiang@js-17-25:~/ProjectA/I535-TwitterProjectCode$ ls build
classes  lib
heljiang@js-17-25:~/ProjectA/I535-TwitterProjectCode$
```

3. Data pipeline

3.1 Data Pipeline Diagram

Please find data pipe line in the figure 1: the first step is ETL: extracting csv data, then transformed into TSV format and added head line(user_id user_name friend_count follower_count status_count favorite_count account_age user_location), and then loaded into Mongodb. The second step is process data: identify user location by adding lat/lng geolocation information to the user profiles via QueryAndUpdate.sh. The third step is to analysis user profile, for example, analyzing validated data and invalidated data. The fourth step is to export the valid data to CSV file via Mongoexport tool. The last step is to visualization/interpret data via Google map visualization.

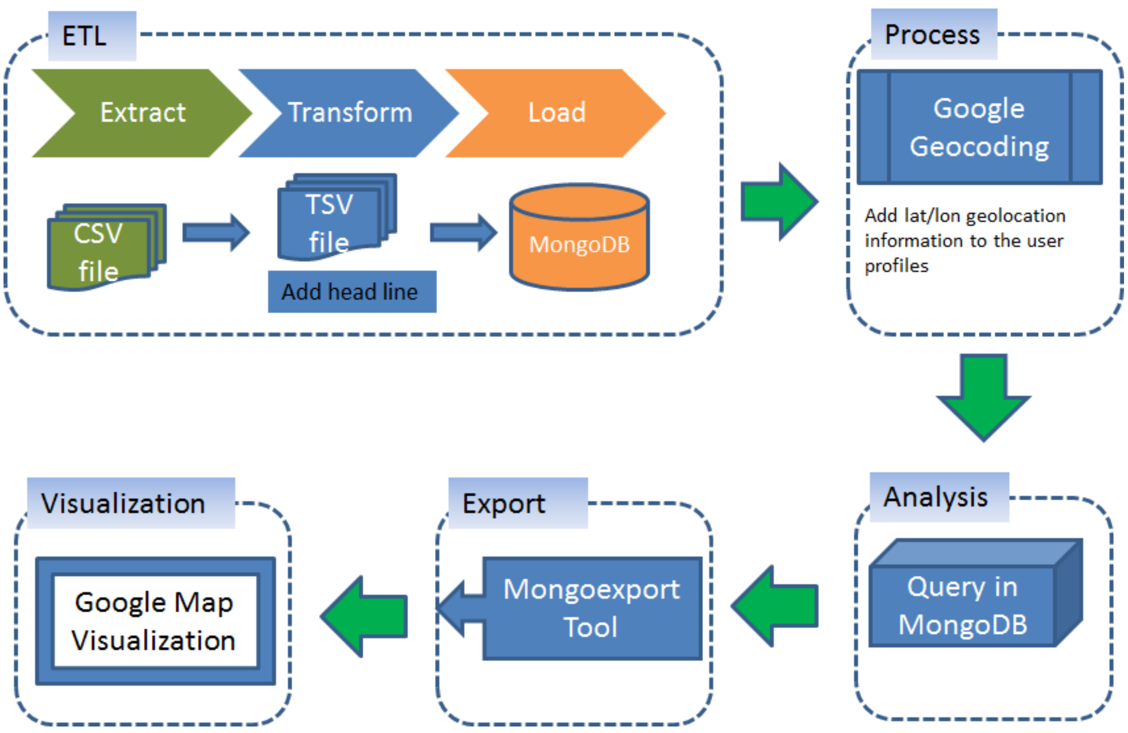


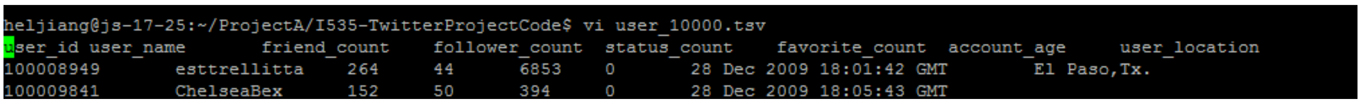
Figure 1: Data pipeline diagram

3.2 Running Data Pipeline

I. ETL

Step 1: Reformat data to TSV and add head line:

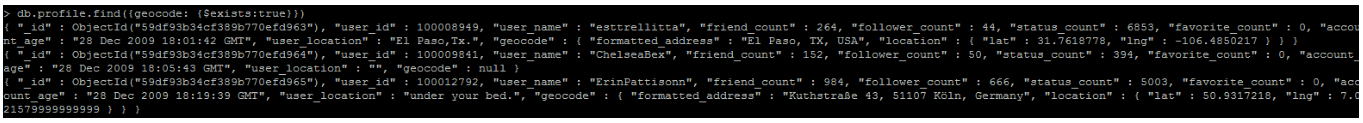
```
./bin/reformat.sh users_10000.txt user_10000.tsv
```



```
heljiang@js-17-25:~/ProjectA/I535-TwitterProjectCode$ vi user_10000.tsv
user_id user_name       friend_count    follower_count  status_count    favorite_count  account_age     user_location
100008949       esttrellitta    264     44      6853    0       28 Dec 2009 18:01:42 GMT        El Paso,Tx.
100009841       ChelseaBex      152     50      394     0       28 Dec 2009 18:05:43 GMT
```

Step 2: Load tsv data to MongoDB:

```
./bin/import_mongodb.sh projectA profile tsv user_10000.tsv
```



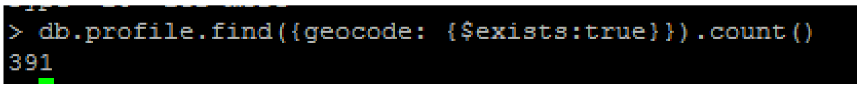
```
> db.profile.find({geocode: {$exists:true}})
{ "_id" : ObjectId("59df93b34cf389b770efd963"), "user_id" : 100008949, "user_name" : "esttrellitta", "friend_count" : 264, "follower_count" : 44, "status_count" : 6853, "favorite_count" : 0, "accou
nt_age" : "28 Dec 2009 18:01:42 GMT", "user_location" : "El Paso,Tx.", "geocode" : { "formatted_address" : "El Paso, TX, USA", "location" : { "lat" : 31.7618778, "lng" : -106.4850217 } } }
{ "_id" : ObjectId("59df93b34cf389b770efd964"), "user_id" : 100009841, "user_name" : "ChelseaBex", "friend_count" : 152, "follower_count" : 50, "status_count" : 394, "favorite_count" : 0, "account_
age" : "28 Dec 2009 18:05:43 GMT", "user_location" : "", "geocode" : null }
{ "_id" : ObjectId("59df93b34cf389b770efd965"), "user_id" : 100012792, "user_name" : "ErinPattisonn", "friend_count" : 984, "follower_count" : 666, "status_count" : 5003, "favorite_count" : 0, "acc
ount_age" : "28 Dec 2009 18:19:39 GMT", "user_location" : "under your bed.", "geocode" : { "formatted_address" : "Kuthstraße 43, 51107 Köln, Germany", "location" : { "lat" : 50.9317218, "lng" : 7.0
21579999999999 } } }
```

II. Process

Run geolocation on the user profiles in MongoDB to add lat/lon geolocation information to the user profiles:

```
./bin/QueryAndUpdate.sh config/config.properties projectA profile input/query.json test1.log
```

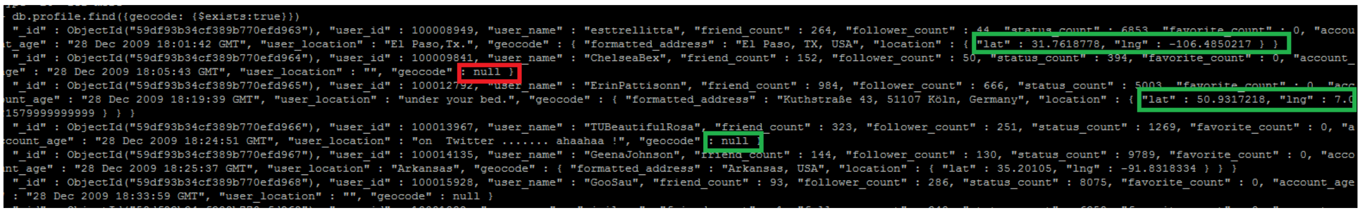
Run several times. Finally, I ran 391 user profiles:



```
> db.profile.find({geocode: {$exists:true}}).count()
391
```

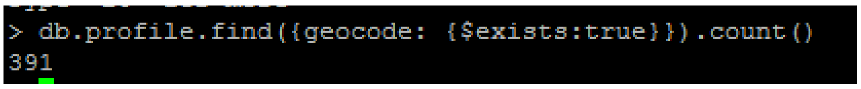
III. Analysis

After quick reviewing the geolocation data, in 391 user profiles, some geocode is valid, some is null: for those null data, it is not possible to locate them in google map. So we only can visualize valid data with correct latitude and longitude information in google map.



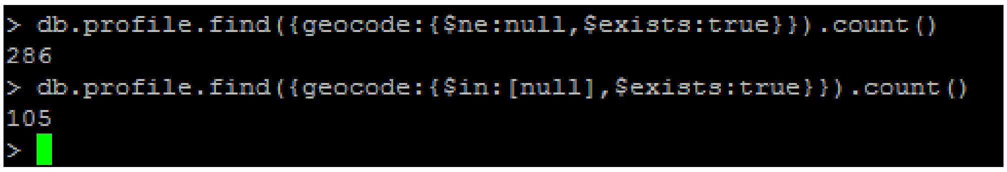
```
 db.profile.find({geocode: {$exists:true}})
 "_id" : ObjectId("59df93b34cf389b770efd963"), "user_id" : 100008949, "user_name" : "esttrellitta", "friend_count" : 264, "follower_count" : 44, "status_count" : 6853, "favorite_count" : 0, "accou
t_age" : "28 Dec 2009 18:01:42 GMT", "user_location" : "El Paso,Tx.", "geocode" : { "formatted_address" : "El Paso, TX, USA", "location" : { "lat" : 31.7618778, "lng" : -106.4850217 } }
 "_id" : ObjectId("59df93b34cf389b770efd964"), "user_id" : 100009841, "user_name" : "ChelseaBex", "friend_count" : 152, "follower_count" : 50, "status_count" : 394, "favorite_count" : 0, "account_
ge" : "28 Dec 2009 18:05:43 GMT", "user_location" : "", "geocode" : null }
 "_id" : ObjectId("59df93b34cf389b770efd965"), "user_id" : 100012792, "user_name" : "ErinPattisonn", "friend_count" : 984, "follower_count" : 666, "status_count" : 5003, "favorite_count" : 0, "acc
unt_age" : "28 Dec 2009 18:19:39 GMT", "user_location" : "under your bed.", "geocode" : { "formatted_address" : "Kuthstraße 43, 51107 Köln, Germany", "location" : { "lat" : 50.9317218, "lng" : 7.0
1579999999999 } } }
 "_id" : ObjectId("59df93b34cf389b770efd966"), "user_id" : 100013967, "user_name" : "TUBeautifulRosa", "friend_count" : 323, "follower_count" : 251, "status_count" : 1269, "favorite_count" : 0, "a
count_age" : "28 Dec 2009 18:24:51 GMT", "user_location" : "on  Twitter ....... ahaahaa !", "geocode" : null }
 "_id" : ObjectId("59df93b34cf389b770efd967"), "user_id" : 100014135, "user_name" : "GeenaJohnson", "friend_count" : 144, "follower_count" : 130, "status_count" : 9789, "favorite_count" : 0, "acco
nt_age" : "28 Dec 2009 18:25:37 GMT", "user_location" : "Arkansas", "geocode" : { "formatted_address" : "Arkansas, USA", "location" : { "lat" : 35.20105, "lng" : -91.8318334 } } }
 "_id" : ObjectId("59df93b34cf389b770efd968"), "user_id" : 100015928, "user_name" : "GooSau", "friend_count" : 93, "follower_count" : 286, "status_count" : 8075, "favorite_count" : 0, "account_age
 : "28 Dec 2009 18:33:59 GMT", "user_location" : "", "geocode" : null }
```

There are **391** data processed via geolocation tool.



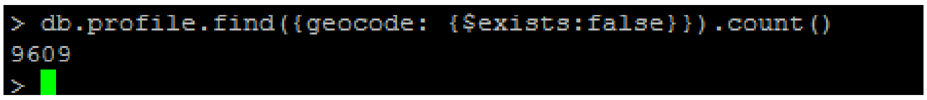
```
> db.profile.find({geocode: {$exists:true}}).count()
391
```

In those **391** dataset, there are **286** datasets with valid latitude and longitude information, other **105** dataset's geolocation is null, which means they are invalid data.



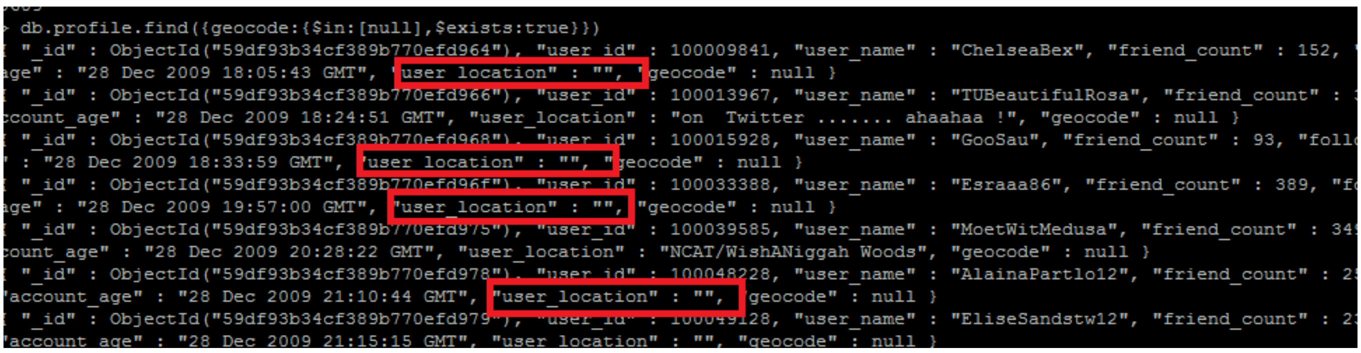
```
> db.profile.find({geocode:{$ne:null,$exists:true}}).count()
286
> db.profile.find({geocode:{$in:[null],$exists:true}}).count()
105
>
```

There are still **9609** datasets not processes with geolocation function:.

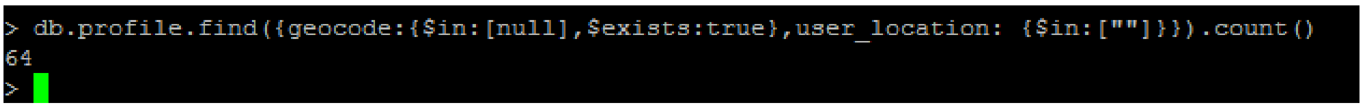


```
> db.profile.find({geocode: {$exists:false}}).count()
9609
>
```

However, in 105 invalid process data, we found many user_location is empty "", obviously for those data geolocation function could not figure out the latitude and longitude information. For example, in those **105** invalid data, there are **64** empty user_location datasets.,

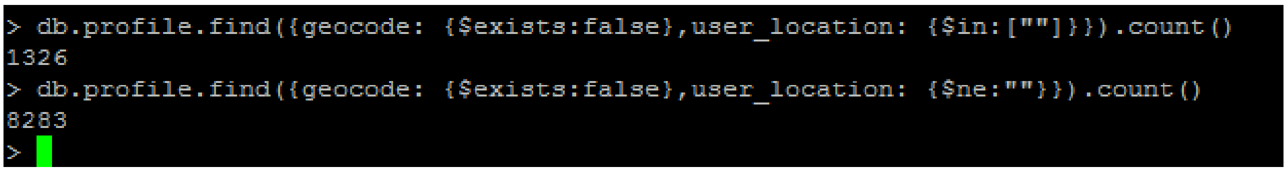


```
> db.profile.find({geocode:{$in:[null],$exists:true}})
{ "_id" : ObjectId("59df93b34cf389b770efd964"), "user_id" : 100009841, "user_name" : "ChelseaBex", "friend_count" : 152, "
age" : "28 Dec 2009 18:05:43 GMT", "user_location" : "", "geocode" : null }
{ "_id" : ObjectId("59df93b34cf389b770efd966"), "user_id" : 100013967, "user_name" : "TUBeautifulRosa", "friend_count" : 
ccount_age" : "28 Dec 2009 18:24:51 GMT", "user_location" : "on  Twitter ....... ahaahaa !", "geocode" : null }
{ "_id" : ObjectId("59df93b34cf389b770efd968"), "user_id" : 100015928, "user_name" : "GooSau", "friend_count" : 93, "foll
" : "28 Dec 2009 18:33:59 GMT", "user_location" : "", "geocode" : null }
{ "_id" : ObjectId("59df93b34cf389b770efd96f"), "user_id" : 100033388, "user_name" : "Esraaa86", "friend_count" : 389, "fo
age" : "28 Dec 2009 19:57:00 GMT", "user_location" : "", "geocode" : null }
{ "_id" : ObjectId("59df93b34cf389b770efd975"), "user_id" : 100039585, "user_name" : "MoetWitMedusa", "friend_count" : 34
ount_age" : "28 Dec 2009 20:28:22 GMT", "user_location" : "NCAT/WishANiggah Woods", "geocode" : null }
{ "_id" : ObjectId("59df93b34cf389b770efd978"), "user_id" : 100048228, "user_name" : "AlainaPartlo12", "friend_count" : 2
"account_age" : "28 Dec 2009 21:10:44 GMT", "user_location" : "", "geocode" : null }
{ "_id" : ObjectId("59df93b34cf389b770efd979"), "user_id" : 100049128, "user_name" : "EliseSandstw12", "friend_count" : 2
"account_age" : "28 Dec 2009 21:15:15 GMT", "user_location" : "", "geocode" : null }
```



```
> db.profile.find({geocode:{$in:[null],$exists:true},user_location: {$in:[""]}}).count()
64
> 
```

So for those **9609** datasets, we don't need to process all, we can exclude datasets with empty user_location(which is **1326**), so we only need to process **8283** datasets.



```
> db.profile.find({geocode: {$exists:false},user_location: {$in:[""]}}).count()
1326
> db.profile.find({geocode: {$exists:false},user_location: {$ne:""}}).count()
8283
> 
```

To process those 8283 datasets, we can use new query: querynew.json:

```
{
geocode: {$exists:false},user_location: {$ne:""}
}
```

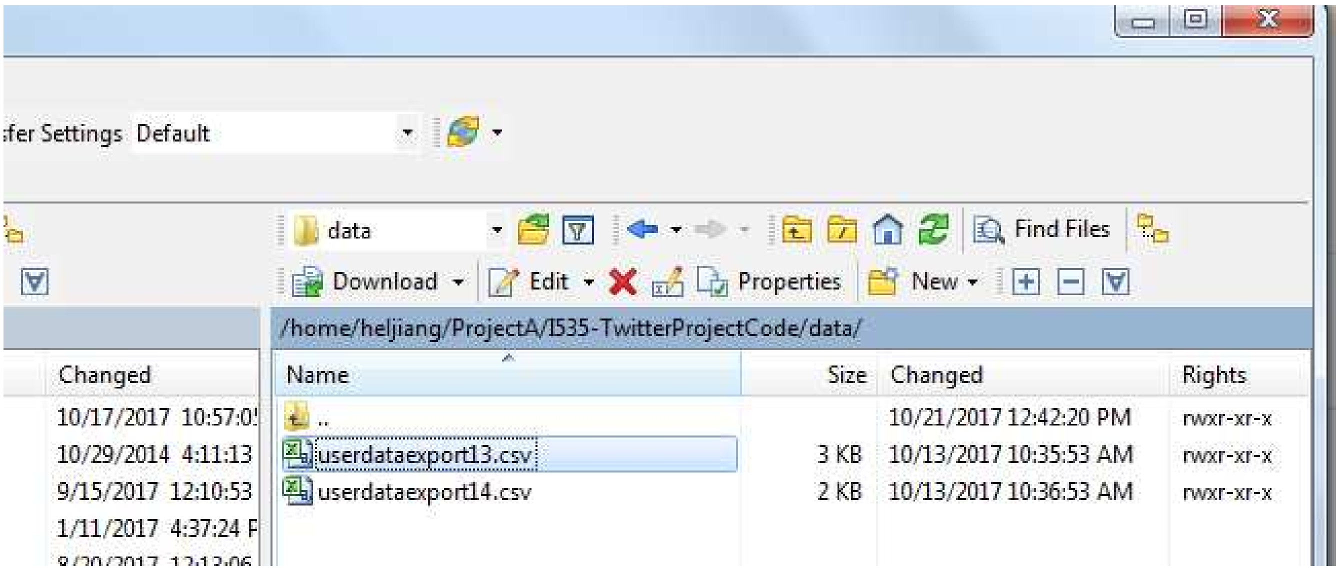
IV. Export

After data process and analysis, we can export data out from Mongodb. Normally we can export into json file or csv file. Here I exported into csv file.

In order to distinguish social people with non-social people, I used additional query(filter by friend_count): very social users(1000+), moderate social people(300,1000), non-social users(0,300). And I exported very social users and moderate social users into 2 CSV file and visualized them in google map.

userdataexport13.csv includes user profiles for moderate social people(300,1000)

userdataexport14.csv includes user profiles for very social people(1000+)



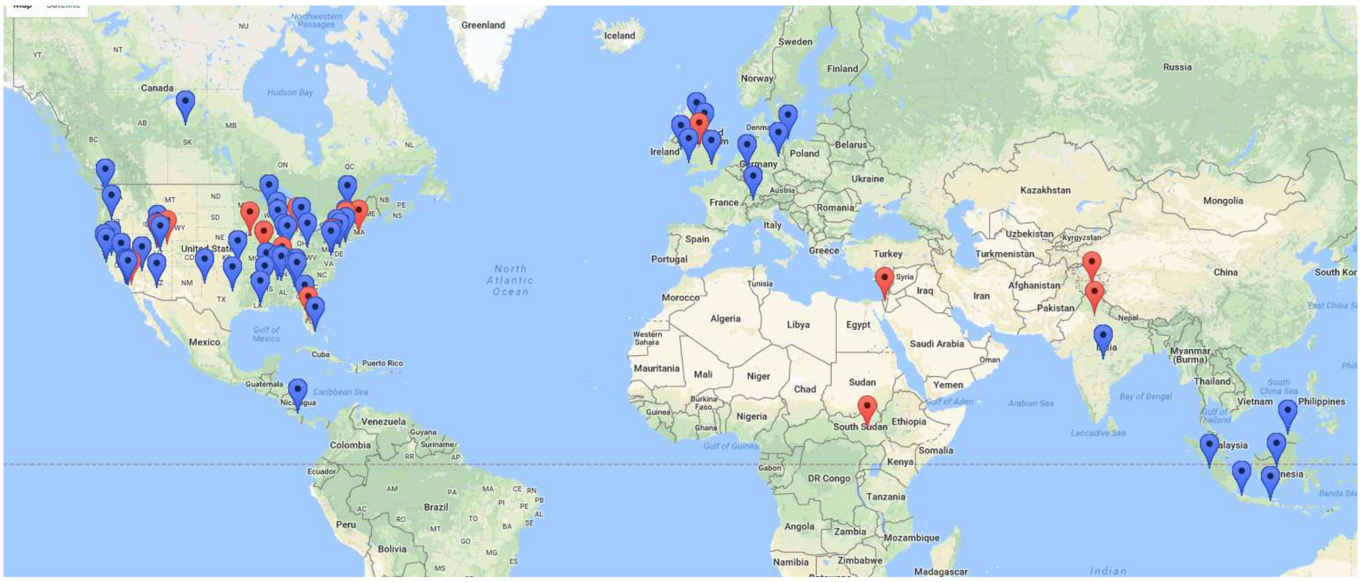
/home/heljiang/ProjectA/I535-TwitterProjectCode/data/

Name	Size	Changed	Rights
..		10/21/2017 12:42:20 PM	rwxr-xr-x
userdataexport13.csv	3 KB	10/13/2017 10:35:53 AM	rwxr-xr-x
userdataexport14.csv	2 KB	10/13/2017 10:36:53 AM	rwxr-xr-x

Changed
10/17/2017 10:57:0
10/29/2014 4:11:13
9/15/2017 12:10:53
1/11/2017 4:37:24 F

V. Visualization

Blue icons are for moderate socialized people, and red icons are for very socialized people. The majority is moderate socialized people, and they are mostly located in US. Very socialized people are minor, and they scattered everywhere, not so centeralized.



```
Here it is full html code:
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<title>google map setup</title>
```

```
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCefZle2DqxF9i51PTfoZsZoOmvWzKYhF4&sensor=true"><
/script>
<script type="text/javascript" src="js/d3.min.js" charset="utf-8"></script>
<style>

</style>
</head>

<body>
<div id="show">
</div>
<script type="text/javascript">
var doc = document,
   showDiv = doc.getElementById('show');
function lodeSupport(){
   if(navigator.geolocation){
      showDiv.style.display = 'block';
                  intMap();
   }else{
      support.innerHTML = 'Sorry your browser does not support !';
      showDiv.style.display = 'none';
   }
}

function intMap()
{
        var redIcon = "icon/red.png";
        var blueIcon = "icon/blue.png";
        var center = new google.maps.LatLng(40.7143528,-74.0059731);
        var mapOptions = {
          center: center,
          zoom: 3,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("map_canvas"),mapOptions);
        d3.csv("data/userdataexport13.csv",function(error,csvdata)
        {
                for( var i = 0; i< csvdata.length; i++ )
                {
                        var lat = csvdata[i]["geocode.location.lat"];
                        var lng = csvdata[i]["geocode.location.lng"];
                        var username = csvdata[i]["user_name"];
                        //console.log( "lat: " + lat + "\n" + "lng: " + lng + "\n" + "username: " + username );
                        var position = new google.maps.LatLng(lat, lng);
                        var marker = new google.maps.Marker({
                                icon: blueIcon,
                                position: position,
                                map: map,
                                title:username
                        });
                }
        });

        d3.csv("data/userdataexport14.csv",function(error,csvdata)
        {
                for( var i = 0; i< csvdata.length; i++ )
                {
                        var lat = csvdata[i]["geocode.location.lat"];
                        var lng = csvdata[i]["geocode.location.lng"];
                        var username = csvdata[i]["user_name"];
                        var position = new google.maps.LatLng(lat, lng);
                        var marker = new google.maps.Marker({
                                icon: redIcon,
                                position: position,
                                map: map,
                                title:username
                        });
                }
        });
}

window.addEventListener('load', lodeSupport , true);
</script>
<div id="map_canvas" style="position:absolute;left:0;top:0;width:100%;height:100%;"></div>
</body>
</html>
```

4. Further work

There are much further work to do in the future about Twitter data, for example:

Data processing on geolocation: can add more improved query to improve the geolocation processing efficiency. For example, exclude user_location empty data in the query because no need to process user_location empty data with geolocation function. Here it is one example for new query which excludes empty user_location:

```
{

geocode: {$exists:false},user_location: {$ne:""}

}
```

Data visualization: data visualization can be more easy to use , for example, can simply input some parameters to export user data(eg, friend count query, followers query, etc) and run a simple command to visualize the data in googlemap.

5. References

[1] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, Kevin Chen-Chuan Chang: Towards social user profiling: unified and discriminative influence model for inferring home locations. KDD 2012:1023-1031

[2] Twitter dataset
[3] MongoDB Reference
[4] Instructions to dump the MongoDB db
[5]Mongo Export Tool
[6]Google Maps Visualization Tutorial
[7]Google Maps JavaScript API Tutorial
[8]Google Visualization API Reference
[9] additional reference for the query criteria
[10] I535-TwitterProjectCode provided in I535 project A.
[11] Transferring Files to Jetstream
[12] users_10000.txt which contains 10,000 user profiles.