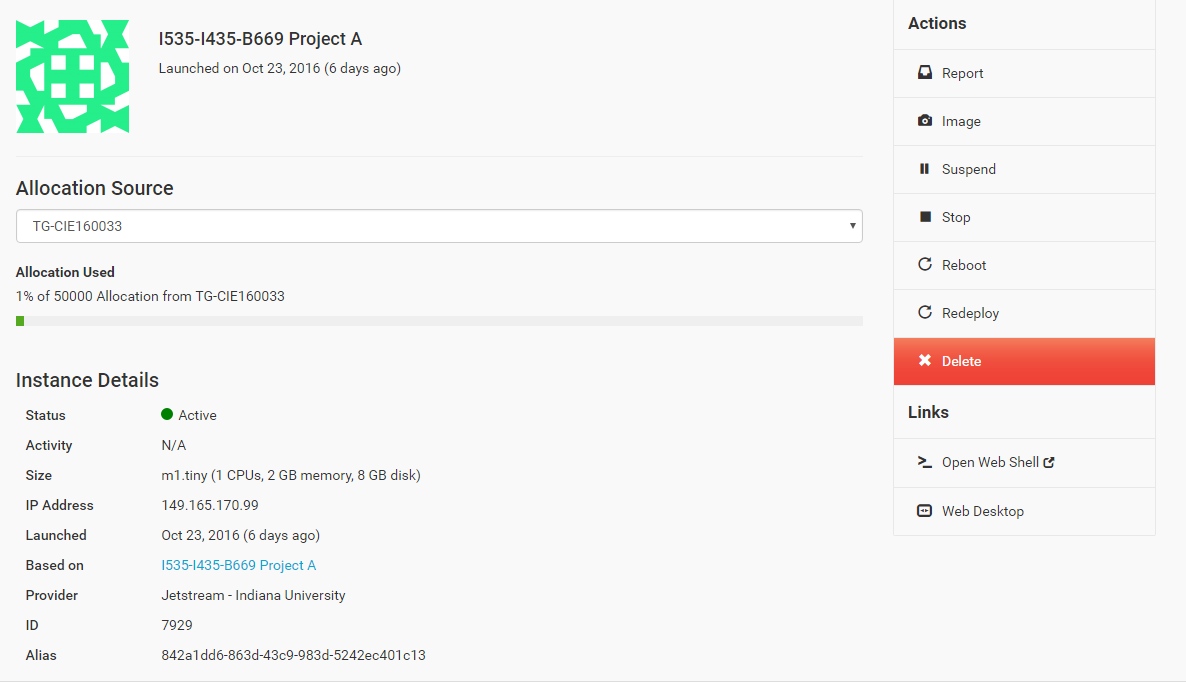
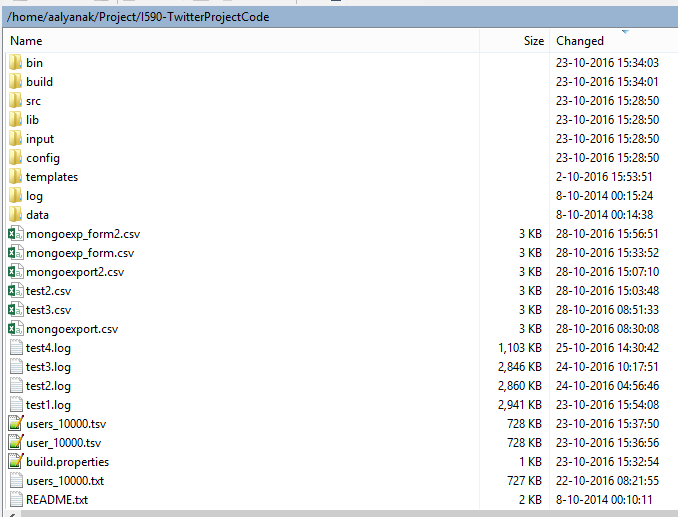
**ProjectA: Twitter Dataset Analysis**

**Setting Up Your Environment**

1. **Create and configure Virtual Machine (VM), Set up VM with Pipeline Tools and Dataset**

I started to this project with setting up my Jetstream instance. I used “*I535-I435-B669 Project A”* image.

I used Winscp for data transfer, and Putty as the web shell. I didn’t find the webshell of Jetstream very handy, didn’t prefer to work in it.

**Figure 1**: The Jetsream Instance and Winscp I completed my project in.

1. **Build Pipeline Tools in VM and Running the Data Pipeline**

After I transferred project materials into instance and unzip in webshell, I changed in Putty webshell the configuration of build.properties file to:

*project.base.dir=/home/aalyanak/Project/I590-TwitterProjectCode*

*java.home=/usr/bin*

Then, I needed to disable the security authorization in */etc/mongod.conf* file. After I did that and restarted my MongoDB, I successfully built the java using the command “ant”.

**Data Pipeline Task 1: Reformat the data**

I reformatted the file with 10k users, using the command:

*./bin/reformat.sh users\_10000.txt users\_10000.tsv*

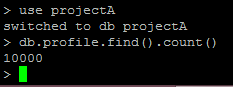
Then, I added the headers with tab seperated format.

user\_id user\_name friend\_count follower\_count status\_count favorite\_count account\_age user\_location

**Data Pipeline Task 2: Import the data into MongoDB**

When it was ready, I imported the file into MongoDB using the command:

*./bin/import\_mongodb.sh projectA profile tsv users\_10000.tsv*



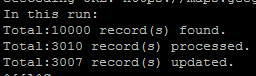
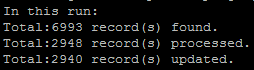
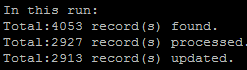
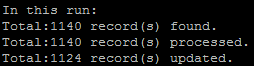
**Figure 2:** The count of records imported into MongoDB

**Data Pipeline Task 3: Query and Update the User Profile Collection**

After the records imported to MongoDB, it’s time to start geocoding. I did the geocoding 4 times in a row, using the command:

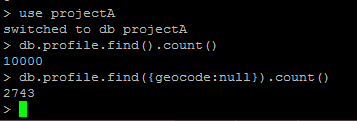
***DAY 1:*** *./bin/QueryAndUpdate.sh config/config.properties projectA profile input/query.json test1.log****DAY 2:*** *./bin/QueryAndUpdate.sh config/config.properties projectA profile input/query.json test2.log****DAY 3:*** *./bin/QueryAndUpdate.sh config/config.properties projectA profile input/query.json test3.log****DAY 4:*** *./bin/QueryAndUpdate.sh config/config.properties projectA profile input/query.json test4.log*

I was able to update 9984 records (3007 + 2940 + 2913 + 1124) in 4 days.

**Figure 3:** Number of records found/processed/updated in 4 days (*tail -f testX.log* command)

However, at the end, 2743 of my 10k records still have null geocode values. 16 of them were not able to updated, but the rest, 2727 of them were null, despite the fact that they’re all geocoded.

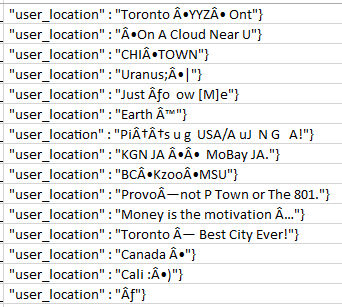


**Figure 4:** The count of Null geocode records at the end of geocoding

**QUESTION: How many locations were you able to validate (i.e., geolocate)? What is the remaining number? Give suggestions for resolving those that you were not able to resolve.**

1. I exported the 16 records and took a deep look into them. The reason for the records which are not able to get geocoded, has invalid characters in their location, and the script were not able to process them.

**SUGGESTION:** The way to avoid this and get a full 10K geolocations is to clean up the data and eliminate the invalid characters, before importing to mongoDB.



**Figure 5:** 16 user locations which are not able to get geolocated

1. The reason of having 2743 null values in geocoded user locations is either those users don’t have a real location information in their accounts or their location is left empty.

**SUGGESTION**: 1st, excluding the null values from data dump. 2nd, improving the code of exporting a data dump, to be able to differentiate a location and non-location values, and to get only the users with real locations in their accounts.



**Figure 6:** Some of the geolocated records with empty geolocations

1. Last, I’ve also found some inaccuracy in geolocated records. Some of non-location values are geolocated wrong, which are not supposed to. This is caused because some words in non-locations are actually locations in somewhere in the world. And the geocode script thought they are real locations. This is something misleading and at the end, not all the geolocations we get are completely trustworthy.

**SUGGESTION:** Improving geolocation script to avoid from geocoding the non-location values and to be able to exclude them.



**Figure 7:** Some of the non-location values in user locations, which got geolocated

**Data Pipeline Task 4: Visualization**

The export I took from MongoDB to my project folder has these qualifications:

* Geocode must exist.
* Follower count >=10000
* Status count >= 5000
* Friend count >= 100

These 3 columns are exported to mongoexport.csv:

* Latitude of geolocation
* Longtitude of geolocation
* User name

These qualifications gave me **61** users to export. The command I’ve used to export (in one line):

*mongoexport -d projectA -c profile -q*

*"{\"geocode\":{\$exists:true,\$ne:null},\"follower\_count\":{\$gte: 10000}, \"status\_count\": {\$gte:5000}, \"friend\_count\":{\$gte:100}}"*

*--csv --fields geocode.location.lat, geocode.location.lng, user\_name -o mongoexport.csv*

The command I’ve used to reformat (in one line):

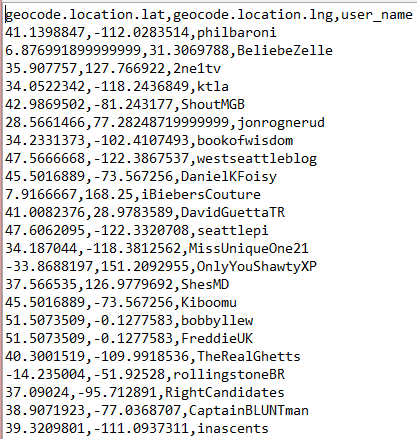
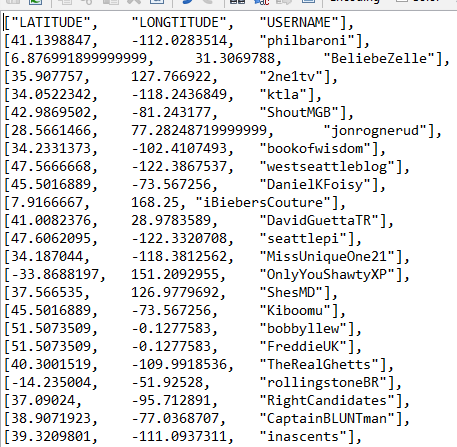
*awk 'BEGIN {print"[\"LATITUDE\",\t\"LONGTITUDE\",\t\"USERNAME\"],"}* ***(1)***

*{FS = ",";OFS = ",\t"};{$NF = "\""$NF"\"";{if (NR>1) print "["$0"],"}}'* ***(2)***

*mongoexport.csv > mongoexp\_form.csv* ***(3)***

1. Command line is adding a line with Lat, Long and name, having double quotes outside of them and square brackets at the beginning and end of row.
   1. FS is catching all commas, OFS is replacing them with comma and tab.
   2. NF is catching the last column (USERNAME), and adding doublequotes outside all rows.
   3. If clause is printing all rows inside square brackets with a comma at the end, except the first row, which contains the actual headers from DB. We already added a header row with doublequotes in step (1), so we need to delete this row.
2. Adding a new file called mongoexp\_form.csv with formatted records.

At the end, I was able to use the mongoexp\_form.csv file to use directly. It needed single or double quotes outside all header values, comma & tab seperated, and only the outside of usernames, looks like:

**Figure 8:** Unformatted and formatted export files (*mongoexport.csv* and *mongoexp\_form.csv*)

The formatted file can be copied and directly be used in the arrayToDataTable function now. Here’s an example of function using 8 sample users:

**var** data = google.visualization.arrayToDataTable([  
  
 [*"LATITUDE"*, *"LONGTITUDE"*, *"USERNAME"*],  
 [41.1398847, -112.0283514, *"philbaroni"*],  
 [6.876991899999999, 31.3069788, *"BeliebeZelle"*],  
 [35.907757, 127.766922, *"2ne1tv"*],  
 [34.0522342, -118.2436849, *"ktla"*],  
 [42.9869502, -81.243177, *"ShoutMGB"*],  
 [28.5661466, 77.28248719999999, *"jonrognerud"*],  
 [34.2331373, -102.4107493, *"bookofwisdom"*],  
 [47.5666668, -122.3867537, *"westseattleblog"*],  
   
 ]);

**NOTE:** The comma at the end of all rows, doesn’t affect the html. You don’t need to reformat the file to delete that comma.



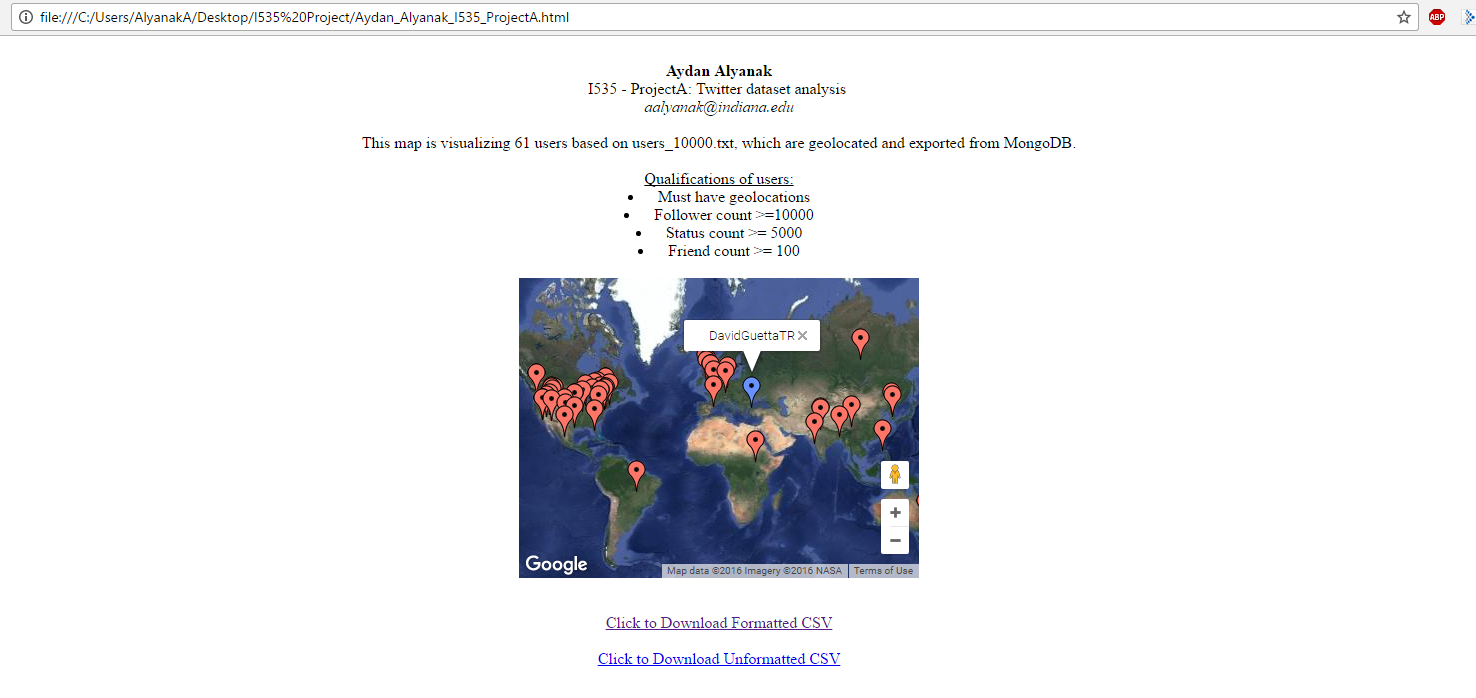
**Figure 9:** Usage of sample 8 rows of formatted data in final HTML

Finally, we can save this html code as a webpage. You can click on the pins to see usernames. The visualization of sample 8 users will look like this:



**Figure 10:** Usage of sample 8 rows of formatted data in visualization

I had 61 users in my full export. With their geolocated Latitude/Longtitude values and usernames, I’ve created my webpage with visualization. I didn’t select a specific region, I preferred to see a map with users from all around the world, having different kinds of qualifications:



**Figure 11:** Final webpage of visualization with all exported 61 users

Below, you can find the HTML (*Aydan\_Alyanak\_I535\_ProjectA.html*), formatted (*mongoexp\_form.csv*) and unformatted (*mongoexport.csv*) portions of my exported 61 users.

**RESOURCES:** The course materials, tutorials, online discussion videos, piazza discussions, the pdf of assignment, websites such as stackoverflow and google searches for script/code related consultancy.