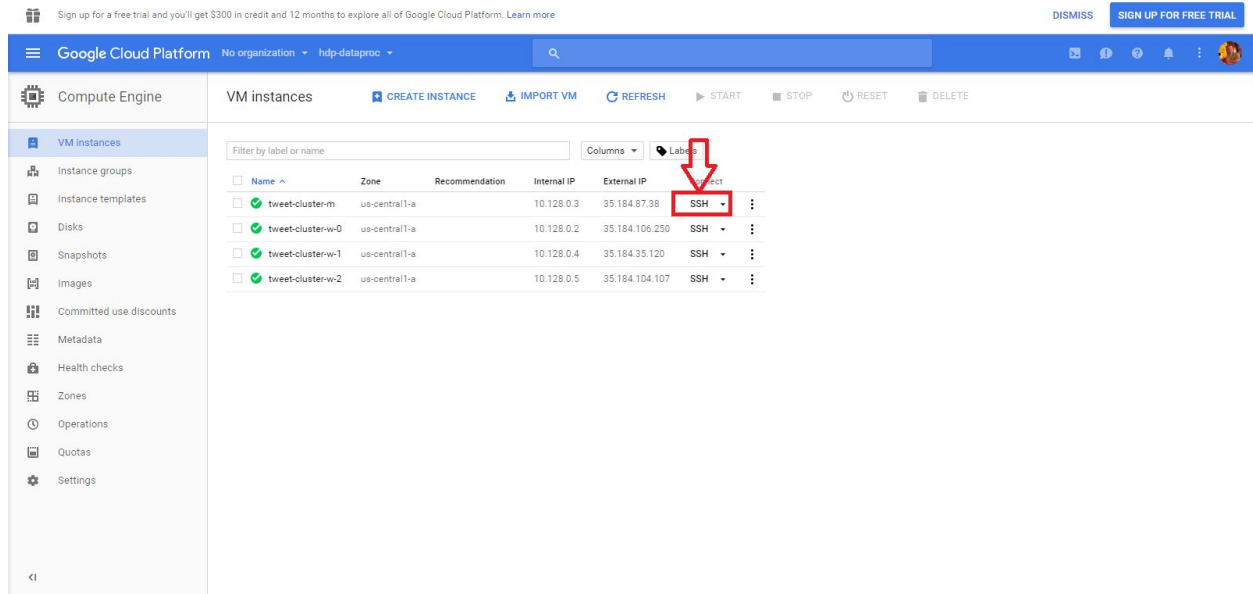


ReadMe

To use the project as a developer, you must have access to the project, located [here](#). From there, the you can access the master node by clicking on the corresponding SSH button.

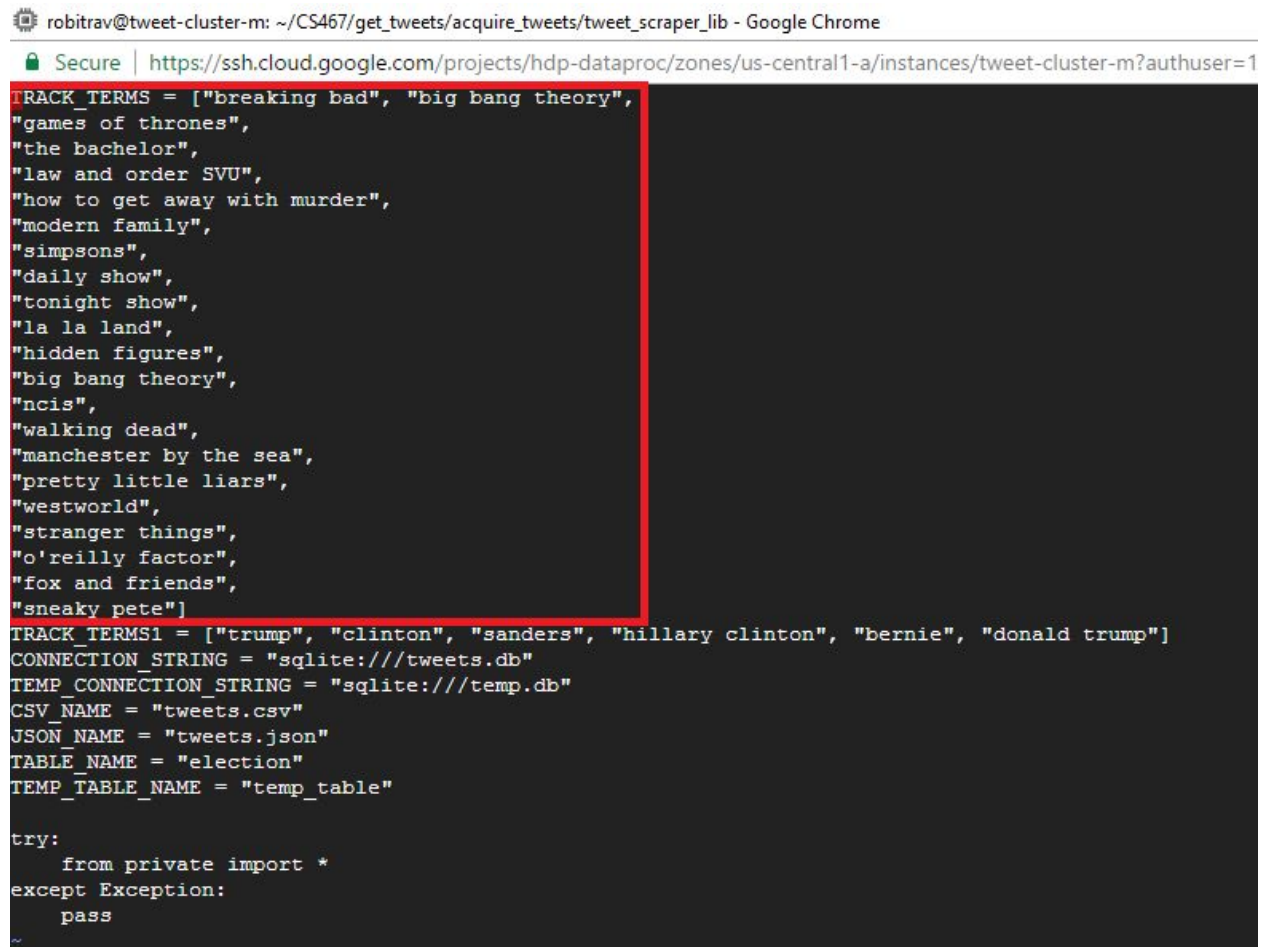


From there, you should find yourself in your own user directory (set up at the midpoint report), and have two options at: either clone the github repository using the command “git clone <https://github.com/travisrobinson2006/CS467.git>” (when commands are given to enter, they should all be given without the quotes, unless stated otherwise), which will load all the files or move into the robitrav user directory, via the commands “cd ..” followed by “cd robitrav”, again without the quotes.

From there, entering the command “ls” should show a directory labeled “CS467”, which is the main directory of our project. Move into it with the command “cd CS467”.

Once in the CS467 directory, we can start to actually use the project. To start off with, tweets will need to be collected. Toward this end, we can either screen tweets based on our original criteria, or they can screen them based on some other criteria, such as by the show “Firefly”. If you wanted to change them, navigate to the “CS467/get_tweets/acquire_tweets/tweet_scraper_lib” directory, and use the text editor of choice to edit the file “settings.py”.

The following image was generated using the vim text editor:



```
robitrav@tweet-cluster-m: ~/CS467/get_tweets/acquire_tweets/tweet_scraper_lib - Google Chrome
Secure | https://ssh.cloud.google.com/projects/hdp-dataproc/zones/us-central1-a/instances/tweet-cluster-m?authuser=1

TRACK_TERMS = ["breaking bad", "big bang theory",
"games of thrones",
"the bachelor",
"law and order SVU",
"how to get away with murder",
"modern family",
"simpsons",
"daily show",
"tonight show",
"la la land",
"hidden figures",
"big bang theory",
"ncis",
"walking dead",
"manchester by the sea",
"pretty little liars",
"westworld",
"stranger things",
"o'reilly factor",
"fox and friends",
"sneaky pete"]

TRACK_TERMS1 = ["trump", "clinton", "sanders", "hillary clinton", "bernie", "donald trump"]
CONNECTION_STRING = "sqlite:///tweets.db"
TEMP_CONNECTION_STRING = "sqlite:///temp.db"
CSV_NAME = "tweets.csv"
JSON_NAME = "tweets.json"
TABLE_NAME = "election"
TEMP_TABLE_NAME = "temp_table"

try:
    from private import *
except Exception:
    pass

~
```

The TRACK_TERMS, contained in the red box in the image, are the search terms that the program will use. To remove a term, simply delete it, along with the quotes and the comma following it. (The track terms are a Python list, and that is the syntax required for them). To add a term, such as the show “The Office,” then between the brackets of the TRACK_TERMS list add in the show you’d like, enclosed in double-quotes, and followed by a comma (for example, ““The Office”,’ would be the correct way to add “The Office” to the list). If the search term is the last in the list, there should be no comma (as seen above when “sneaky pete” is not followed by a comma).

Once the track terms have been selected, or we’ve decided to use the defaults track terms, we’re ready to collect tweets, and should navigate to the directory “CS467/get_tweets”. There, if we enter the command “ls”, we’ll see the file “run_me_to_get_tweets.py”. To run it, enter into the command line “python run_me_to_get_tweets.py” (or, if we want to run it in the background, “python run_me_to_get_tweets.py &”) We will then be greeted with the following messages (except where the time and processing files will be different):

```
/home/robitrav/CS467/get_tweets
robitrav@tweet-cluster-m:~/CS467/get_tweets$ python run_me_to_get_tweets.py
cleaning tweets...
Beginning processing for time: Y2017-M03-D15-H23
Beginning processing for file: tweets71-7-37.json
scraping tweets...
```

From here, once per hour, tweets will be dumped into a JSON file (located in CS467/get_tweets/acquire_tweets/unclean_tweets), from which they are cleaned, filtered, and saved into a separate cleaned tweets text file.

To get a sentiment score for each tweet

1. Navigate to the directory CS467/sentiment_analyzer, you will see a file named "nbAlg.py". There are two ways to run this program. If you simply enter "python nbAlg.py", the program will use "tweets_ready_for_use_final" as the input file. It will then produce an output file called "nbScores.txt" containing the following information for each tweet: the name of the show to which the tweet was referring, the state (abbreviated) that it originated from, and the tweet's sentiment score (a number between -1 and 1), tab delimited. "nbScores.txt" can be found in your current folder. Alternatively, if you want to run the algorithm on your own input file, you may enter "python nbAlg.py yourfilename". If you provide your own input file of tweets, please make sure that it is in the proper format (show name, then tweet text, then state abbreviation, tab delimited). (It should be noted here that the nbAlg program may take a couple of minutes to run)
2. You can now view your output file containing the sentiment scores: nbScores.txt.

To enter scores file into the Hadoop Distributed File System:

1. To put the output file you just created into the HDFS (distributed across the three worker nodes), type "hadoop fs -put nbScores.txt /database". If you get an error message saying that the file is already there, you can remove the existing file by entering "hadoop fs -rm -r /database", entering the command "hadoop fs -mkdir /database" and then using the above 'put' command.
2. Check that your file was transferred successfully by entering "hadoop fs -ls /database". The output file name should appear.

To create a table in Hive with your score data:

1. In the sentiment_analyzer folder, you will find a script that will create a table called nbscores filled with the data from nbScores.txt in the HDFS. To run this script, type "hive -f create_table.hql" in the command line and press enter. This may take a few seconds. The resulting table will have the following columns: content_name (name of show/movie), user_location (abbreviated state), and score (between -1 and 1).

2. Check that your table was created by typing “hive;” then enter into the command line to enter the Hive shell (the semicolon is necessary to run all commands within the Hive shell). Then type “use testdb;”. Then type “show tables;”. You should now see a table called “nbscores”. Type “quit;” to exit the Hive shell.

After loading scores into the HDFS, navigate back into the “CS467” directory. You should see there a file called “run_me_for_csv_results.sh” which we will run with the command: “sh run_for_csv_results.sh *your_choice_of_dir*”, where ‘*your_choice_of_dir*’ will create a new directory where you will save your results in. This will save a CSV file with the name “000000_0” to the directory you specify. Alternatively, you can choose not to specify a file, in which case the file will be saved into a directory called ‘here_are_results’. In the below image, the red box is the shell script that should be run, and the green box is the default value if none is specified.

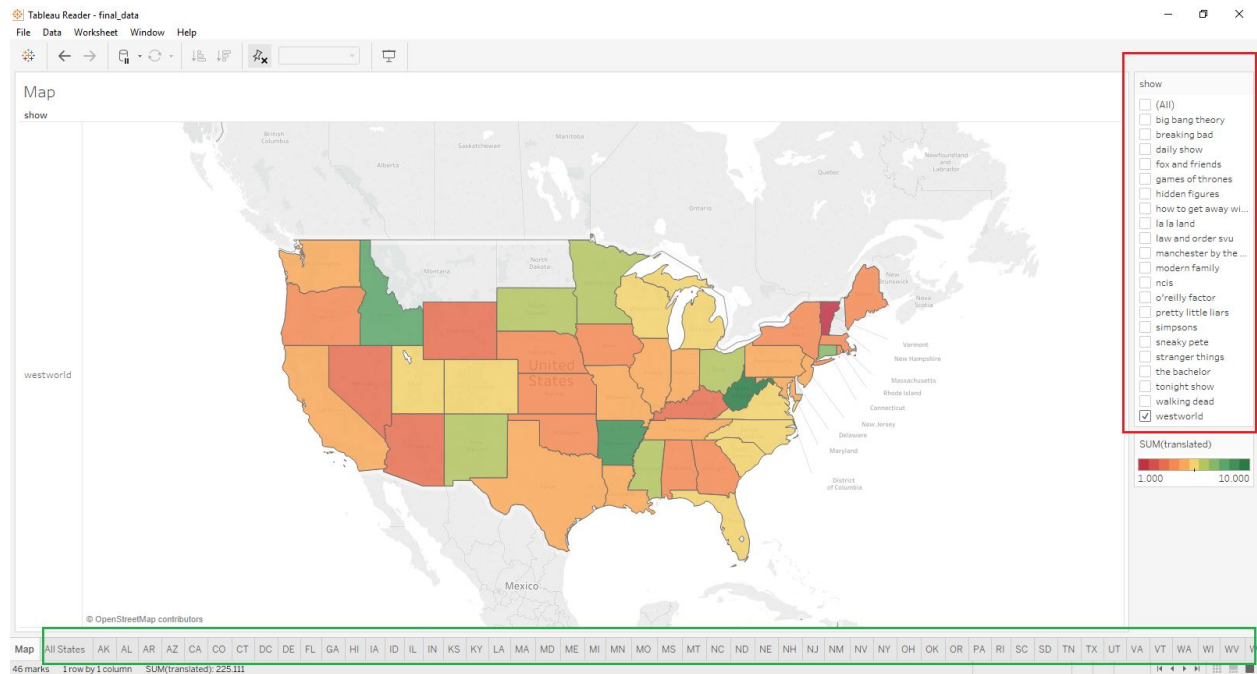
```
robitrav@tweet-cluster-m:~/CS467$ ls
agg_sentiment.sql  here_are_results  reports  sentiment_analyzer.tweets-ready-for-use
archive            hive_script.hql  run_for_csv_results.sh  website
get_tweets        README.md        sentiment_analyzer
robitrav@tweet-cluster-m:~/CS467$
```

After creating the CSV file “000000_0,” it needs to be downloaded to your local computer (click the gear icon in the top right corner, then select download file from the menu that appears) and converted to an Excel file. From here, Tableau will be capable of processing the data into images.

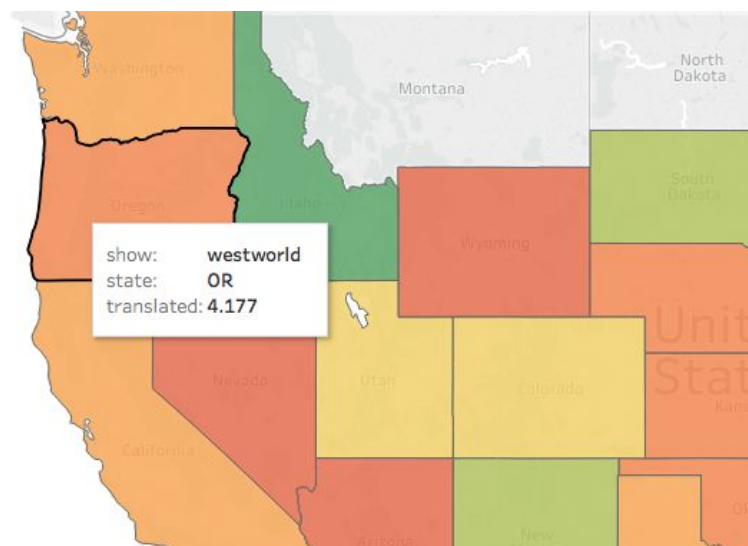
```
agg_sentiment.sql  get_tweets  README.md  run_for_csv_results.sh  sentiment_analyzer.tweets-ready-for-use
Hive  Hive_script.hql  reports  sentiment_analyzer  website
robitrav@tweet-cluster-m:~/CS467$ cd sentiment_analyzer
robitrav@tweet-cluster-m:~/CS467/sentiment_analyzer$ hadoop fs -put nbscores.txt /database
17/03/17 16:30:34 INFO gcs.GoogleHadoopFileSystemBase: GHFS version: 1.6.0-hadoop2
put: /database/nbscores.txt: File exists
robitrav@tweet-cluster-m:~/CS467/sentiment_analyzer$ hadoop fs -put nbscores.txt /database
17/03/17 16:30:57 INFO gcs.GoogleHadoopFileSystemBase: GHFS version: 1.6.0-hadoop2
put: /database/nbscores.txt: File exists
robitrav@tweet-cluster-m:~/CS467/sentiment_analyzer$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.4.1.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.10.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-2.1.0.jar/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine
(i.e. spark, tez) or using Hive 1.X releases.
hive>
hive> exit
>
robitrav@tweet-cluster-m:~/CS467/sentiment_analyzer$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.4.1.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.10.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-2.1.0.jar/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine
(i.e. spark, tez) or using Hive 1.X releases.
hive> show databases
OK
default
testdb
Time taken: 1.163 seconds, Fetched: 2 row(s)
hive> exit;
robitrav@tweet-cluster-m:~/CS467/sentiment_analyzer$ hadoop fs -ls /database
17/03/17 16:33:53 INFO gcs.GoogleHadoopFileSystemBase: GHFS version: 1.6.0-hadoop2
Found 1 item
-rw-r--r-- 2 robangkhalil hadoop 2242156 2017-03-17 13:29 /database/nbscores.txt
robitrav@tweet-cluster-m:~/CS467/sentiment_analyzer$ ls
create_table.hql  nbscores.txt  pos_set.csv  testbiohScores-2-15.txt  tweets-ready-for-use
dataset_nbscoresClean.txt  nbTest1  pos_set.txt  testbiohScores.txt  tweets_ready_for_use
__init__.py  neg_set.csv  screenshot_of_results.png  training_set.py  tweets_ready_for_use_final
metric_klss  neg_set.txt  sentiment_analyzer_lib  tweets-2-15
nbhlp.py  outbiohScores.txt  test_set.csv  tweets_ready_for_use
robitrav@tweet-cluster-m:~/CS467/sentiment_analyzer$ cd ..
robitrav@tweet-cluster-m:~/CS467$ ls
agg_sentiment.sql  get_tweets  README.md  run_for_csv_results.sh  sentiment_analyzer.tweets-ready-for-use
Hive  Hive_script.hql  reports  sentiment_analyzer  website
archive            hive_script.hql  run_for_csv_results.sh  website
get_tweets        README.md        sentiment_analyzer
robitrav@tweet-cluster-m:~/CS467$ cd ..
robitrav@tweet-cluster-m:~/CS467$ ls
agg_sentiment.sql  get_tweets  README.md  run_for_csv_results.sh  sentiment_analyzer.tweets-ready-for-use
Hive  Hive_script.hql  reports  sentiment_analyzer  website
archive            hive_script.hql  run_for_csv_results.sh  website
get_tweets        README.md        sentiment_analyzer
robitrav@tweet-cluster-m:~/CS467$
```

To view our Tableau workbook that was used to generate the images on the website:

- 1) Go to <https://www.tableau.com/products/reader> and download Tableau Reader (free). Download for your machine (either Windows or OSX).
- 2) If you have downloaded our [Github repository](#) to your local computer, navigate to CS467/data_files and find the file named “final_data.twbx”. This file is a Tableau Packaged Workbook which can be read/opened by the free Tableau Reader.
- 3) Open final_data.twbx with Tableau Reader
- 4) From here, you can visually inspect our data and maps. You’ll see map selections on the right side of the screen (red box) and graph selections on the bottom of the screen (green box).



When looking at the maps, you can mouse over each state and see stats for that particular state for a specific show:



Rohan Gokhale

Travis Robinson

Chase Hu