Project 3

Dataset - Twitter

Hadoop streaming mode - Python

Team Members

Anand Kumar Taridalu Subrahmanyam	M08685681
Vikas Reddy Vanteru	M08914333
Sri Harsha Jilludumudi	M09011304
Mohammed Aamer	M08802120

INDEX

1. Hourly Analysis of President's Tweets

A.

In Mapper:

Map program parses every tweet and outputs three types of keys.

Type 1: <Hour (24 hour format), 1>

Type 2: <Date (MMM DD YYYY), 1>

Type 3: <Error, 1>, <Success, 1>, <Total Tweets, 1>

In Reducer:

Reducer outputs following key-value pairs.

Type 1: <Hour, Overall # of tweets in that particular Hour>

Type 2: <Date (MMM DD YYYY), <# No. of tweets on a particular day>

Type 3: <Error, # of errors>, <Success, # of successes>, <Total Tweets, total # of tweets>

hourly_Average.py calculates expected number of tweets every hour by President and total number of days for which twitter data was parsed. It also presents number of tweets President has tweeted in each hour.

Instructions

Run the command

hadoop jar /root/hadoop-2.7.1/share/hadoop/tools/lib/hadoop-streaming-2.7.1.jar -input /data/twitter/ -output myoutput -file *.py -mapper mapTweet.py -reducer reduceTweet.py*

Get the output file from Hadoop filesystem to local filesystem as cloudresults.txt file using the below command

hadoop fs -get myoutput/part-00000 cloudresults.txt

Now run the final program *hourlyAverage.py* and capture the final output into output.csv file

./hourlyAverage.py > output.csv

Using Excel, from output.csv file, we can plot the details.

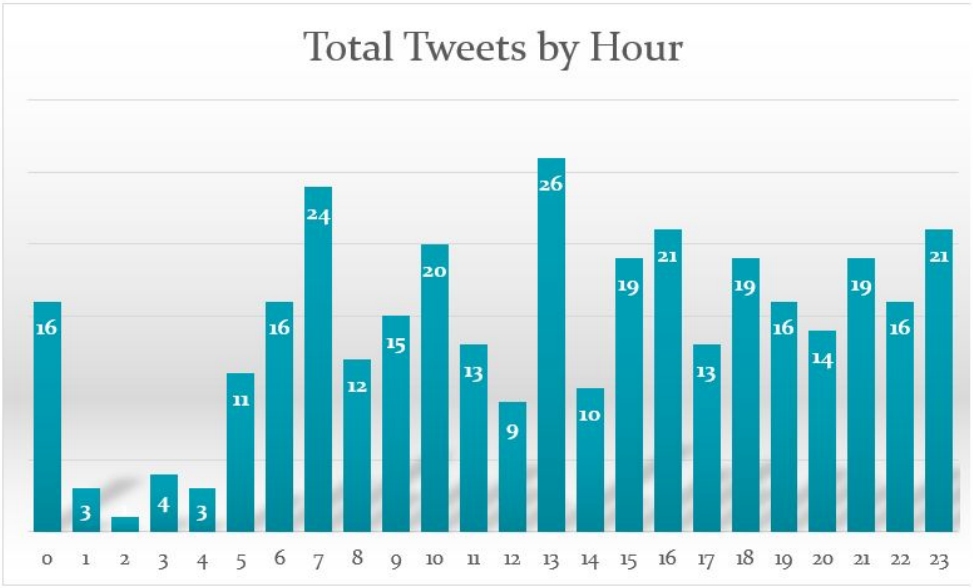
rom output.csv file, we can infer the following.

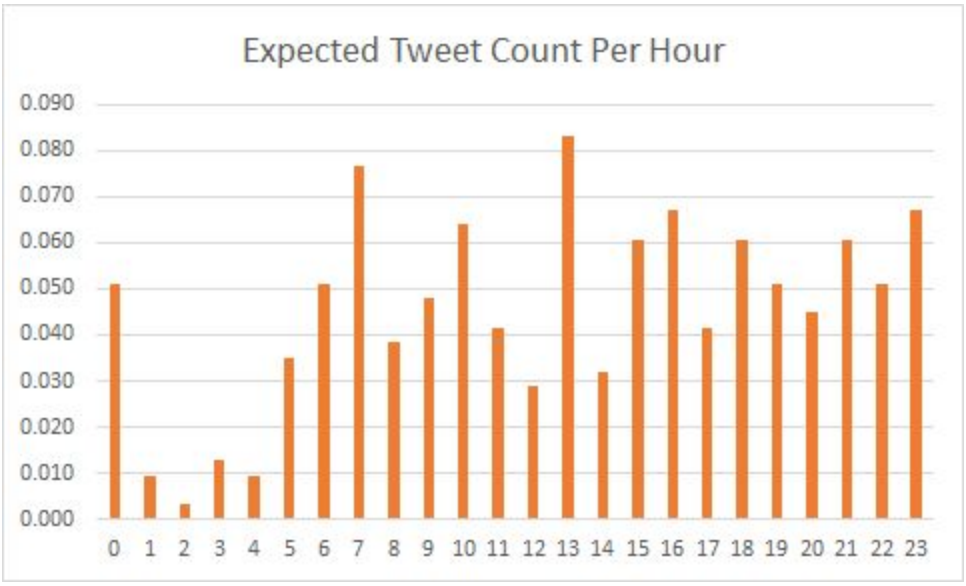
Total number of tweets the program has parsed and count of tweets that were successfully parsed and also error count saying no. of tweets that were skipped due to an error.

Total no. of days of twitter data parsed, average/expected number of tweets made by president are available for each hour.

Note: *UTC hours are converted to Eastern Time, by reducing time by 4 hours, this calculation is done using excel. After conversion, following plots are plotted.*

Excel Output





Analysis Inference: President tweets most at 1300 followed by 0700, 2300, 1600 and 1000 hours.

Hour	Tweet Count	Expected Number of Tweets
0	16	0.051
1	3	0.010
2	1	0.003
3	4	0.013
4	3	0.010
5	11	0.035
6	16	0.051
7	24	0.077
8	12	0.038
9	15	0.048
10	20	0.064
11	13	0.042
12	9	0.029
13	26	0.083
14	10	0.032
15	19	0.061
16	21	0.067
17	13	0.042
18	19	0.061
19	16	0.051
20	14	0.045
21	19	0.061
22	16	0.051
23	21	0.067

2. What day of the week does @PrezOno tweet the most on average? Use the same example as in #1 but for days of the week.

A. The given twitter data is in JSON format. We extract the 'user' from the JSON data and see if the 'screen_name' in 'user' to find the user_name given.

In Mapper:

All the tweets are read line by line from the standard input. If the 'screen_name' in 'user' is 'PrezOno', then we take the 'created_at' value from the JSON data to find the day on which PrezOno tweeted. The 'created_at' is in the format 'Day Month Date HH:MM:SS +0000 Year' for example, "created_at": "Wed Aug 27 13:08:45 +0000 2008"

From this data, we extract 'Day' using datetime.strptime() and strftime() methods. We then send the 'Day' and the value as '1' as the input to reducer.

Example: If PrezOno tweets on Wednesday, the output of the mapper will be (Wed, 1)

In Reducer:

For every input from mapper, we use strip() and split() methods, we separate the keys and values from the input. From mapper, key is the day on which PrezOno tweeted, so we sum up all the tweets which are tweeted on a particular day. If In the given dataset, PrezOno tweeted 150 times on sunday, we take this 150 as sunday_count.

Also, the given data is from march 2014 to february 2015. That is approximately, 52 weeks, i.e. we have 52 sundays, 52 mondays and so on.

So, if we divide the count of tweets for each day with the total number of days present(i.e. 52) we get the average number of tweets that PrezOno tweeted for each day. To find the maximum of the average we use max() method.

Finally, the average number of tweets on each day and the day on which PrezOno tweeted the most is printed out.

Output:

Sunday avg: 1.13461538462
Monday avg: 0.923076923077
Tuesday avg: 0.634615384615
Wednesday avg: 1.05769230769
Thursday avg: 1.03846153846
Friday avg: 0.75
Saturday avg: 1.01923076923
PrezOno tweet the most on Sunday1.13461538462

3. How does @PrezOno's tweet length compare to the average of all others? What is his average length? All others?

A. To answer the above question, first we need to know what the twitter data is like. The dataset is in the form of JSON strings. After getting through the schema it is known that the feature "**user**" contains the details about the twitter user and inside that the feature, "**screen_name**" contains the username, for example: "PrezOno". The tweet posted is available in the feature "**text**".
So I used these three features to solve the above problem.

In Mapper:
In the mapper I have checked for the input passed into two categories:

a. PrezOno's tweet
b. Other tweet

Depending on whose tweet it was I made the mapper return two keys PrezLength and Others Length which basically counts the length of their respective tweets and passes the length as the key's value.

Example : if it's a PrezOno tweet, output of mapper will be (PrezLength, 45)
If it is someone else's tweet, output would be (OthersLength, 32)

So the output of mapper would be either of the above two keys and their corresponding value.

In Reducer:
In the reducer I have created a list to store all the tweet lengths of PrezOno. So when the input is passed to the reducer, there are separate counters installed to keep track of the number of tweets of PrezOno and Other's tweets. Also we find the sum of all the tweet lengths using which average can be found out. Maximum and minimum PrezOno tweet lengths are found out by using the max() and min() functions on the list.

Finally, the average tweet length of both PrezOno and other's are printed out. Also the list of tweet lengths of PrezOno and the maximum and minimum of the list are printed out so as to compare them to the average length of other's tweets as asked in the question. The average tweet lengths are printed as integers so as to show the average number of characters present in a tweet.

Time taken to complete the job: 12 minutes
No. of mapper tasks: 3213
No. of reducer tasks: 1

Output:

```
[root@happysingh ~]# hadoop fs -cat  myoutput/part-00000
prez_average:104        others_average:81
PrezOno tweet lengths:
[43, 138, 100, 86, 67, 99, 101, 44, 117, 83, 140, 133, 47, 103, 78, 131, 53, 129
, 138, 128, 124, 82, 127, 140, 122, 57, 140, 140, 134, 135, 71, 133, 142, 18, 85
, 93, 135, 129, 102, 134, 126, 126, 140, 131, 100, 43, 122, 139, 93, 124, 100, 1
31, 140, 87, 104, 111, 136, 132, 140, 135, 131, 139, 108, 74, 113, 140, 140, 140
, 100, 78, 140, 75, 95, 143, 71, 98, 132, 119, 72, 95, 124, 101, 100, 140, 16, 1
35, 115, 132, 125, 96, 140, 99, 83, 131, 144, 135, 139, 138, 98, 99, 118, 89, 14
0, 128, 122, 116, 120, 67, 140, 134, 134, 123, 116, 130, 138, 94, 88, 93, 65, 10
7, 101, 100, 139, 76, 127, 87, 136, 126, 129, 26, 92, 132, 74, 42, 109, 105, 101
, 31, 106, 122, 83, 129, 111, 138, 140, 107, 94, 113, 116, 78, 104, 86, 117, 99,
 49, 95, 62, 76, 129, 133, 137, 144, 144, 138, 82, 134, 140, 42, 107, 55, 115, 1
33, 61, 109, 37, 138, 135, 138, 88, 127, 24, 138, 31, 93, 52, 34, 59, 111, 100,
129, 55, 8, 140, 76, 120, 122, 118, 20, 138, 86, 85, 30, 139, 86, 96, 136, 139,
128, 78, 90, 69, 116, 102, 140, 55, 96, 33, 71, 135, 137, 92, 132, 131, 73, 31,
138, 102, 50, 121, 87, 134, 139, 140, 67, 140, 85, 139, 74, 144, 140, 49, 122, 1
40, 63, 137, 140, 113, 31, 94, 122, 140, 136, 51, 136, 83, 140, 25, 128, 140, 14
0, 130, 132, 42, 140, 74, 140, 72, 131, 137, 75, 137, 112, 140, 58, 73, 70, 140,
 118, 59, 118, 57, 115, 60, 70, 140, 105, 140, 84, 140, 127, 101, 117, 69, 140,
130, 37, 45, 62, 86, 78, 119, 91, 91, 110, 55, 78, 110, 84, 87, 138, 118, 115, 4
9, 93, 64, 112, 102, 127, 138, 69, 40, 90, 130, 96, 103, 42, 53, 39, 122, 132, 1
40, 96, 79, 137, 97, 87, 136, 144, 140, 140, 140]
PrezOno max tweet length: 144
PrezOno min tweet length: 8
```

7. For those tweets with location information, what lat/long (or city/state) is the centroid? What was the proportion of tweets with location to those without?

A. Every tweet has two types of location information, if present -

One which shows specific latitude and longitude - “Point” coordinates and other which shows places “Polygon” - these tweets show the exact GPS location of the user.

- First type “Point” coordinates has latitude and longitude information of the location.
- Second type "Polygon" contains location of 4 lat-lon coordinates that define the general area from which the user is posting the Tweet along with the display name(city, neighborhood) of the Place and the country code corresponding to the country where the Place is located, among other fields.

The “geo” key in the root-level which contains the the “coordinates” key which represent the latitude and longitude of the location. There is also an “coordinates” key in the root-level which represent the longitude and latitude. “Coordinates” key is preferred as the “geo” is deprecated.

The “place” key in the root-level contains the location of the tweet specified by “bounding_box” which displays “Polygon” coordinates showing 4 longitude, latitude locations.

In Mapper:

All the tweets are read line by line from the standard input and the required information i.e,. locations of the tweets are obtained.

For the first type “Point”, the tweets with “geo” information were identified to get their latitude and longitude values.

If the first type is not present then second type “Place” value is checked, if it is available - the centroid of the place is calculated as “Place” tag has 4 long-lat information available.

These values along with an attribute which sends if the location data is present or not is also send to the reducer.

In Reducer:

The centroid i.e., the average of the latitudes and longitudes is calculated. Also the proportion of tweets with location to those without is calculated (tweets without Location / tweets with Location).

The time took to complete the Job(20Gb of data) : 20 minutes

Number of mapper tasks : 3213

Number of reducer tasks : 1

Output:

{"proportionOfCoordsToNoCoords": 2.0577468039835232, "centroid": [38.926748250824289, -82.26612365207464]}. The centroid points to location (approx.): 9430-10612 Bulaville Pike, Bidwell, OH 45614; Which lies 138 miles of Cincinnati.