PROJECT REPORT

HADOOP PROJECT (P3)

Amarnath Kothapalli
Atmanand Citigori
Sai Siddhardh Reddy
Yashmin Singla

Name: SAI SIDDARDHA REDDY THATIPARTHY

Q1: Hourly analysis of President Ono's tweets from the given twitter data.

Data set taken: Twitter

Feature used: Hadoop streaming mode on python platform

The problem requires the analysis of the twitter data and output the number of tweets made by the user @PrezOno during each hour of the day from a data provided. The data is present in the form of json strings which needs to be parsed to obtain the output.

For this question we are interested in the 'created_at' field in the json data corresponding to each tweet.

Mapper: Mapper parses every tweet and searches for the user 'screen_name' value '@PrezOno'. This is done using the json library and each line is parsed using json.load( ) function. It then calculates the number of tweets made by Prez during each hour present in that data block. It the outputs the hour and number of tweets.

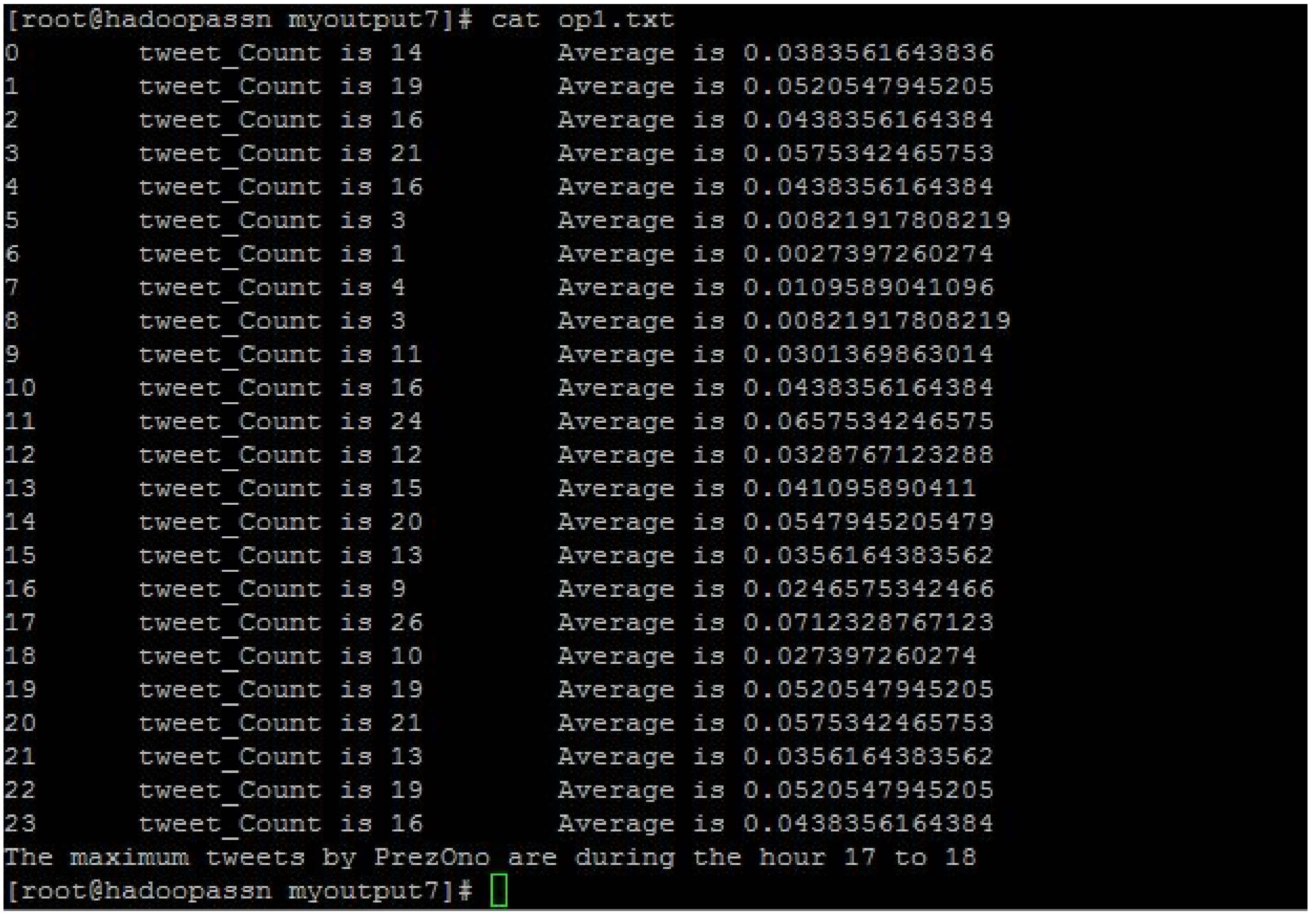
Sample Output:

```
<hour>    count:<number_of_tweets>
   0      count:0
   1      count:0
   2      count:0
   3      count:0
   4      count:0
   5      count:2
   6      count:0
   7      count:1
   8      count:0
   9      count:0
   10     count:0
   11     count:0
   12     count:0
   13     count:0
   14     count:0
   15     count:0
   16     count:0
   17     count:0
   18     count:5
   19     count:0
   20     count:0
   21     count:0
   22     count:0
```

23	count:0

Reducer: The reducer gets the sorted output from all the mappers and generates the total number of tweets made during each hour from the entire data provided in the HDFS file system. It also calculates the average tweets made during each hour by **assuming** that the data provided is spanned over an year and has 365 days. It also provides the hours between which @PrezOno tweeted more.

Output:



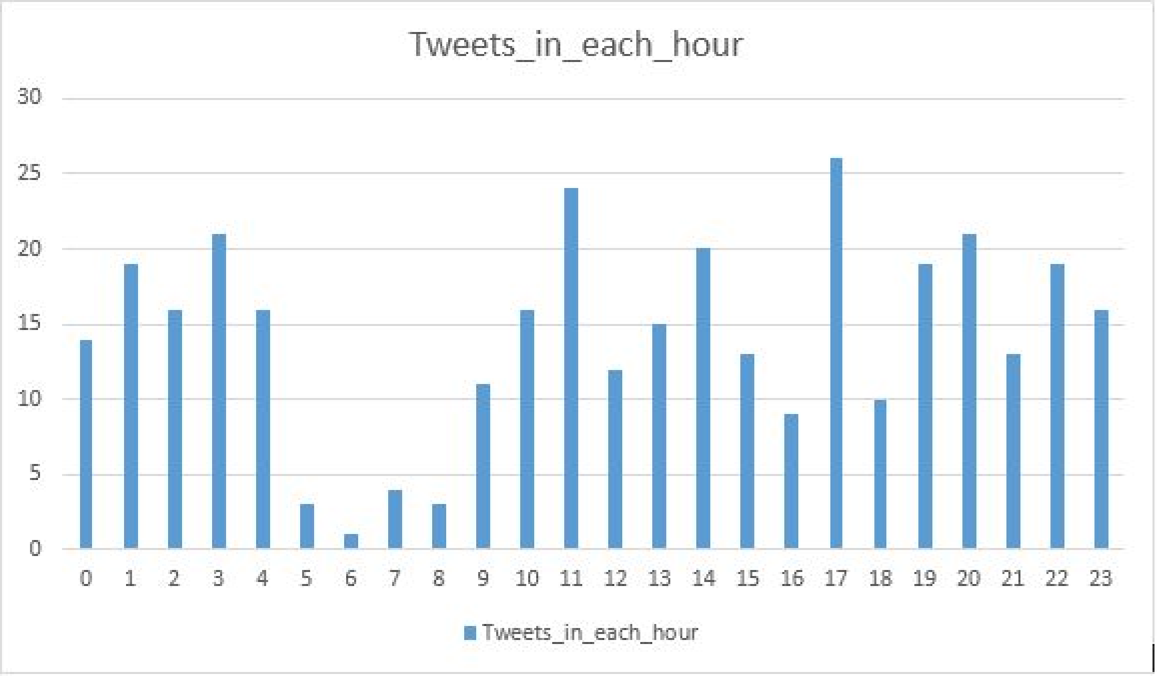
```
[root@hadoopassn myoutput7]# cat op1.txt
0       tweet_Count is 14       Average is 0.0383561643836
1       tweet_Count is 19       Average is 0.0520547945205
2       tweet_Count is 16       Average is 0.0438356164384
3       tweet_Count is 21       Average is 0.0575342465753
4       tweet_Count is 16       Average is 0.0438356164384
5       tweet_Count is 3        Average is 0.00821917808219
6       tweet_Count is 1        Average is 0.0027397260274
7       tweet_Count is 4        Average is 0.0109589041096
8       tweet_Count is 3        Average is 0.00821917808219
9       tweet_Count is 11       Average is 0.0301369863014
10      tweet_Count is 16       Average is 0.0438356164384
11      tweet_Count is 24       Average is 0.0657534246575
12      tweet_Count is 12       Average is 0.0328767123288
13      tweet_Count is 15       Average is 0.041095890411
14      tweet_Count is 20       Average is 0.0547945205479
15      tweet_Count is 13       Average is 0.0356164383562
16      tweet_Count is 9        Average is 0.0246575342466
17      tweet_Count is 26       Average is 0.0712328767123
18      tweet_Count is 10       Average is 0.027397260274
19      tweet_Count is 19       Average is 0.0520547945205
20      tweet_Count is 21       Average is 0.0575342465753
21      tweet_Count is 13       Average is 0.0356164383562
22      tweet_Count is 19       Average is 0.0520547945205
23      tweet_Count is 16       Average is 0.0438356164384
The maximum tweets by PrezOno are during the hour 17 to 18
[root@hadoopassn myoutput7]#
```

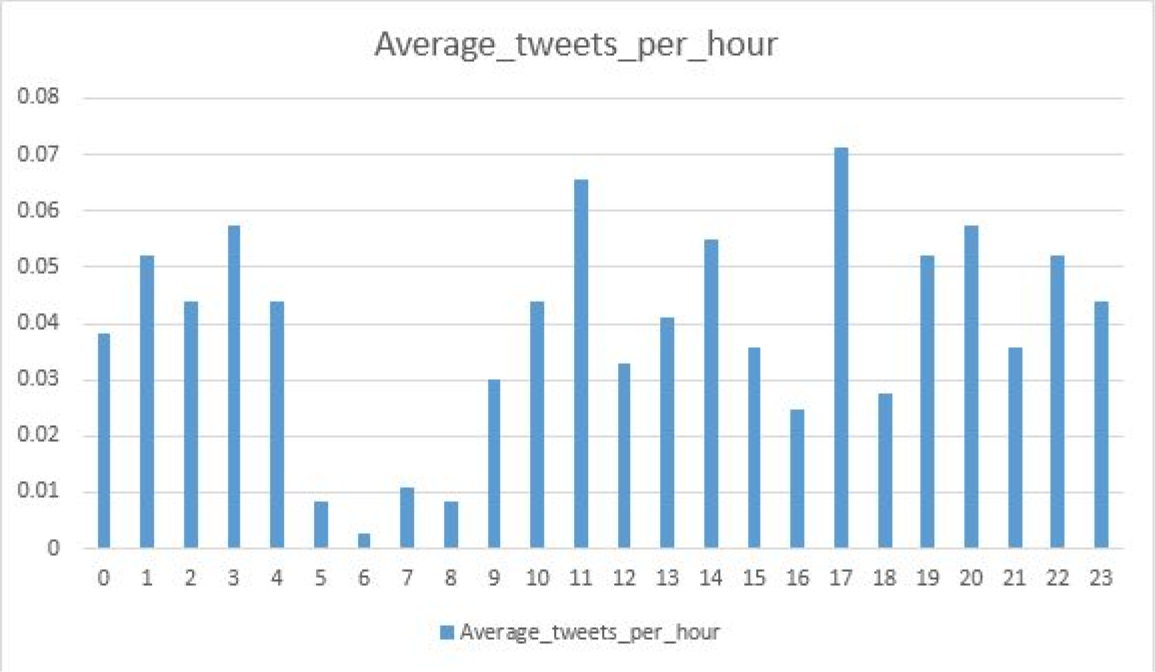
The command run for this twitter analysis is

```
hadoop jar /root/hadoop-2.7.1/share/hadoop/tools/lib/hadoop-streaming-2.7.1.jar -input
/data/twitter -output myoutput1 -file *.py -mapper q1_map.py -reducer q1_reduce.py
```

Number of mappers: 3213
Number of reducers: 1

The plots showing the results obtained are as follows.





Name: Yashmin Singla

Q2: Day wise analysis of President Ono’s tweets from the given twitter data.

Data set taken: Twitter

Feature used: Hadoop streaming mode on python platform

The problem requires the analysis of the twitter data and output the number of tweets made by the user @PrezOno during each day of the week from the data provided. The data is present in the form of json strings which needs to be parsed to obtain the output.

For this question we are interested in the ‘created_at’ field in the json data corresponding to each tweet.

Mapper: Mapper parses every tweet and searches for the user ‘screen_name’ value ‘@PrezOno’. This is done using the json library and each line is parsed using json.load( ) function. It then calculates the number of tweets made by Prez during each day of the week present in that data block. It the outputs the hour and number of tweets.

The day values are mapped on to integers as follows:

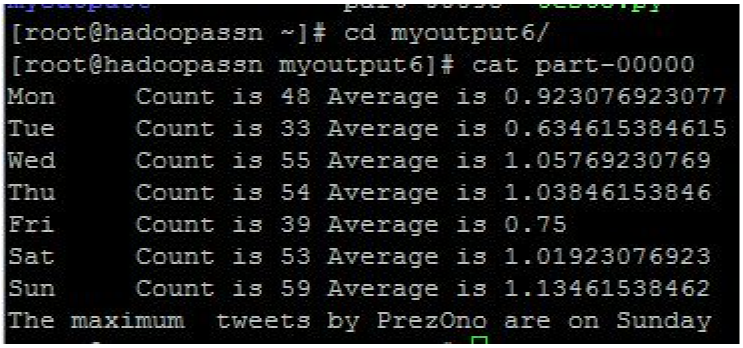
‘Mon’:0, ‘Tue’:1, ‘Wed’:2, ‘Thu’:3, ‘Fri’:4, ‘Sat’:6, ‘Sun’:7

Sample Output:

```
<day>     count:<number_of_tweets>
  0       count:0
  1       count:0
  2       count:0
  3       count:0
  4       count:0
  5       count:3
  6       count:0
```

Reducer: The reducer gets the sorted output from all the mappers and generates the total number of tweets made during each hour from the entire data provided in the HDFS file system. It also calculates the average tweets made during each day of the week by **assuming** that the data provided is spanned over an year and has 52 weeks. It also provides the day on which @PrezOno tweeted more.

Output:



```
[root@hadoopassn ~]# cd myoutput6/
[root@hadoopassn myoutput6]# cat part-00000
Mon     Count is 48 Average is 0.923076923077
Tue     Count is 33 Average is 0.634615384615
Wed     Count is 55 Average is 1.05769230769
Thu     Count is 54 Average is 1.03846153846
Fri     Count is 39 Average is 0.75
Sat     Count is 53 Average is 1.01923076923
Sun     Count is 59 Average is 1.13461538462
The maximum  tweets by PrezOno are on Sunday
```

Member name: ATMANAND CITIGORI

Q3:Comparing @PrezOno's average tweet length to that of the average of all others
Data set taken: Twitter
Feature used: Hadoop streaming mode on python platform

The given dataset is in the form of JSON strings. So once if we analyse the key value pairs of the JSON strings, it is easy to obtain the specific values of the twitter data. Firstly we used "user" value and obtained all the information regarding that specific user. "Screen_name" gives the username. "text " key gives the tweet. So to obtain the length of the tweet I used the len function.

Mapper:

There are a total of 3213 mappers. I have categorised the screen names into two parts, PrezOno and others. The command "json.loads(line)" reads each line and checks whether if there is a above mentioned keys. So if screenname is prezono then it calculates the length and updates to the the original length of Ono's tweets, and number of tweets ie...count of the tweets are increased if there is any tweet by Ono. Similarly all others (except Ono) tweet lengths and their counts are appended to the variables respectively.

So there are a total of 6426 outputs from all the mappers. Ie.. one is the Ono tweet and one is the all_other tweet.

Output of mapper when run on part-03197 :

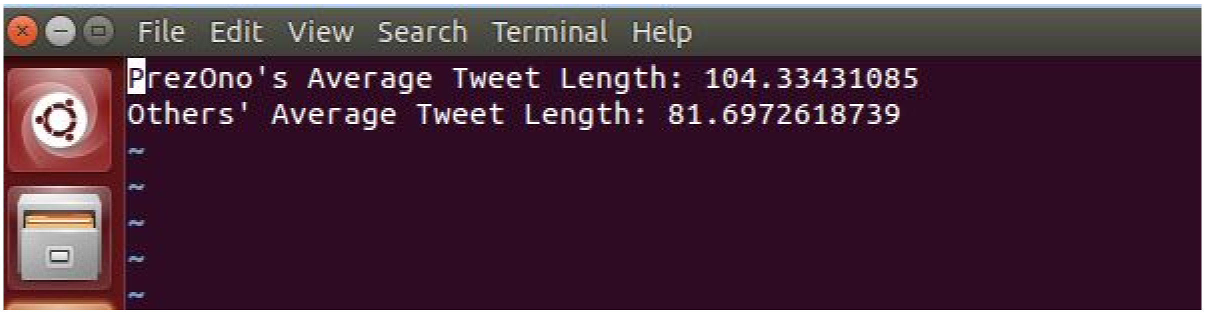
Person	length	count
PrezOno	0	0
Other	21162	290

Reducer:

The output of the mapper is sent to the reducer. The key for the reducer is either PrezOno or Other. The length and count of 'PrezOno' output from all the mappers are added in the reducer, and the length and count of the other key ie 'Other' is also added. The avg of all the tweets is given by

avg length = total length of all the tweets / number of tweets.

Screenshot of the final output is :



```
PrezOno's Average Tweet Length: 104.33431085
Others' Average Tweet Length: 81.6972618739
~
~
~
~
~
```

The command run for this twitter analysis is

hadoop jar /root/hadoop-2.7.1/share/hadoop/tools/lib/hadoop-streaming-2.7.1.jar -input /data/twitter -output myoutputfinal -file *.py -mapper q3_map.py -reducer q3_reduce.py

Number of mappers: 3213
Number of reducers: 1

Name: AMARNATH KOTHAPALLI

Q5: Analysis to find which twitter user tweeted most and comparing their average tweet lengths.

Data set taken:Twitter

Feature used:Hadoop streaming mode on python platform

The given dataset is in the form of JSON strings. So we need to parse the key value pairs of the JSON strings, to obtain the specific values of the twitter data. We use the user 'screen_name' to extract username of the user who made the tweet. The 'text' field is the actual tweet made.

Mapper:

The command "json.loads(line)" parses each line and extracts the username and the length of the tweet made by the user. The mapper then gives the username, tweet_length, 1 as output

Output of mapper when run on part-03197 :

Person	length	count
MoussaJuuf	94	1
DannngCuh	34	1
_jameira	37	1
Its_Nylaaa	144	2
AlmightyPronto	52	1

REDUCER:

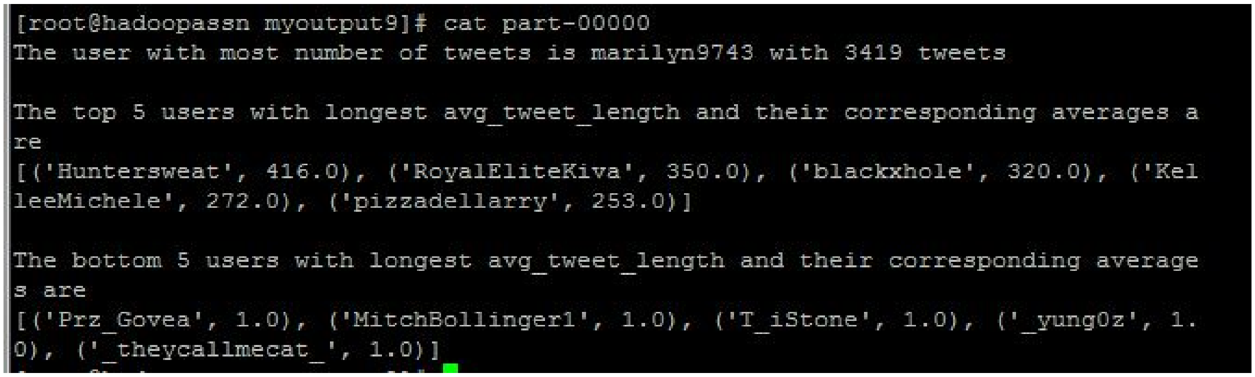
Mapper gives the output to the reducer and the key is the usernames ie the screen names. It compares with the old_word ie none. Here I am trying to find the average length of each person's tweet.

To find the avg of the tweet length:

Avg length = length of all tweet of a screenname/number of tweets made by the username.

In order to sort according to the avg length, I took an empty dictionary and added the unique screen names and its avg. I used the operator library for to sort the elements of the dictionary according to the values rather than keys.After sorting the values I printed the first 5 names and the corresponding top avg lengths and bottom 5 names corresponding lowest avg lengths.

The final output of the reducer is:



```
[root@hadoopassn myoutput9]# cat part-00000
The user with most number of tweets is marilyn9743 with 3419 tweets

The top 5 users with longest avg_tweet_length and their corresponding averages a
re
[('Huntersweat', 416.0), ('RoyalEliteKiva', 350.0), ('blackxhole', 320.0), ('Kel
leeMichele', 272.0), ('pizzadellarry', 253.0)]

The bottom 5 users with longest avg_tweet_length and their corresponding average
s are
[('Prz_Govea', 1.0), ('MitchBollinger1', 1.0), ('T_iStone', 1.0), ('_yung0z', 1.
0), ('_theycallmecat_', 1.0)]
```

The command run for this twitter analysis is

hadoop jar /root/hadoop-2.7.1/share/hadoop/tools/lib/hadoop-streaming-2.7.1.jar -input /data/twitter -output myoutputq5 -file *.py -mapper q5_map.py -reducer q5_reduce.py

Number of mappers: 3213
Number of reducers: 1