CS6065 Cloud Computing
P3
Hadoop for Real Problems
Report

Sayali Pendharkar M08844056
Laxmi Janakiraman M08844347
Suprabha Shreepad Hegde M08879427

Question 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.

Technologies and Languages:
Hadoop Cluster and MapReduce jobs
Programming language-Python

Analysis Method:
The question requires one to determine which day of the seven days of the week, President Santa Ono tweets the most on average. It also involves plotting a graph of averages of all days of the week, indicating the expected number of tweets per day. This requires performing a MapReduce operation over the entire Twitter dataset present.

Mapper:
The twitter dataset is present in the JSON format. This is parsed using the json package on Python. The fields in the dataset required for solving this question are:
created_at,screen_name and user.
The tweets by President Ono are determined by placing a condition on the Name such that only tweets with screen_name=PrezOno are selected. The field created_at is split to determine the day which is the key in the key-value pair sent as output from Mapper.

Pseudo code for Mapper:

1. Read line from standard input
2. Parse using json.loads()
    a. tweets.append(json.loads(line))
    b. Tweets is a list of tweets from the dataset
3. For each tweet in tweets

. If screen_name==’PrezOno’
a. day=day from created_at of tweet
b. key-value=<day,1>

Sort:
Sort is in-built in Hadoop and the values are sorted based on the key. For example all tweets from Monday are sorted together.

Reducer:
The sorted output from Mapper is passed into a predetermined number of reducers. The reducers use the <key-value> pairs to calculate average for each day and add it to a dictionary (avg) with the day as key. Hence the dictionary has 7 elements with day as key and the value as average no of tweets by President Ono on that particular day.

Assumption: Each year contains 365 days and 52 weeks. Hence each day of the week occurs 52 times and hence the average for each day is calculated by dividing by no. of tweets by 52

Pseudo code for Reducer:

1. For each line in standard input

a. If day is already the key
b. Increment the no of tweets and calculate average
c. Else make the new day as key and continue
d. Add average to dictionary

2. Calculate the average for last day and add to dictionary

Results:

Input to Mapper:

The input to mapper is the twitter data in HDFS

Output of Mapper:

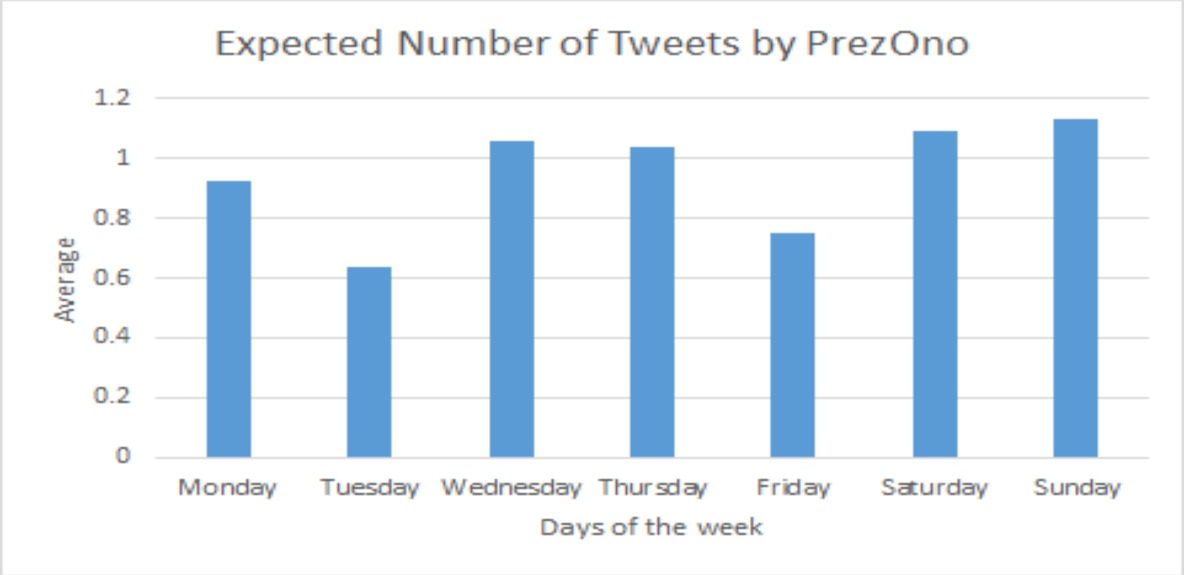
The output of Mapper is a <key-value> pair- <day,1> for each tweet from President Santa Ono. This can be used to calculate the total number of tweets on each day by President Ono.

Input to Reducer:

Sorted list of key-value pairs from Mapper

Output of Reducer:

The output of the reducer is a dictionary consisting of the averages tweets by PrezOno for all the days and the day that he is hs tweeted the most on average.
A graph plotting the expected number of tweets by PrezOno on each day.



Graph_1:Expected Number of Tweets by PrezOno

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

Technologies and Languages:
Hadoop Cluster to run MapReduce jobs
Python as the coding language

Analysis Method:
The task is to compare the average tweet length of President Santa Ono's tweets to the average tweet length of all other users. The result expected involves the average tweet length of President Ono's tweets and the comparison (ratio) of his average tweet length to the average tweet length of all other users.
We are calculating the average tweet length for each user and associating the average with the user's screen name, which is unique. Later, we are finding the collective average of all the users except President Ono. To compare President Ono's tweet length to the average of all others, we are dividing his average with the collective average of all others.

Pseudo code for mapper:
1. Read every tweet in JSON format.
2. Parse the tweet and find the user's screen name.
3. Parse the tweet and find the length of the user's tweet.
4. Print the key-value pair <user's screen name, tweet length> for every tweet in the input.

The key value pairs are then sorted and passed to the reducer.

Pseudo code for reducer:
1. Read sorted input of key-value pairs.
2. Create a dictionary containing each user's screen name as the key and the list of the length of all his tweets as the value.
3. Find the average tweet length for each user.
4. Find the average tweet length for all other users collectively except for President Ono.
5. Find the ratio of average tweet length of President Ono's tweets to the collective average tweet length of all other users.

Results:

Input to the mapper:
Twitter data files containing tweets in JSON format.

Output from the mapper:
A key value pair <user's screen name, tweet length> for every tweet

Input to the reducer:
Sorted output from mapper. This sort is performed on the user's screen name.

Output from the reducer:

Average tweet length of President Ono's tweets. Ratio of President Ono's tweets to the collective average tweet length of all other users.

Question 10:
Detect the proportion of bad words in a tweet. Plot bad word proportion by hour for all 24 hours.

Technologies and Languages:
Hadoop Cluster and MapReduce jobs
Programming language-Python

Analysis Method:
The question demands for a proportion of bad words in a tweet and to plot a graph with bad word proportion by hour for all 24 hours. Initially a bad word list is built. In our case we have taken the list from http://www.hyperhero.com/en/insults.htm .

Assumption:
All the words present in the list (http://www.hyperhero.com/en/insults.htm) are considered to be bad. From 'created_at' value, if the hour is 07:00:00 to 07:59:59 we have considered it as 7th hour.

Pseudo code for Mapper:

1. Read every tweet from standard input
2. Decode JSON and get the values of tweet message and tweet hour for every tweet.
3. Check if the word present in a tweet message matches any word in bad word list.
4. Calculate the total words and the total bad words present in a tweet.
5. Pass <key,value> pair <hour, totalwords_badwords> to the reducer.

Pseudo code for Reducer:

1. For each line in standard input
   i. If hour is already the key
   ii. Get the total words and the bad words count
   iii. Else make the new hour as key and continue
   iv. Add totalword_badword to dictionary.
2. Get the total word count and total bad word count for each hour from the dictionary.
3. Calculate the ratio of total word count to total bad word count.
4. Print for each hour the ratio of total word count to bad word count.

Results:

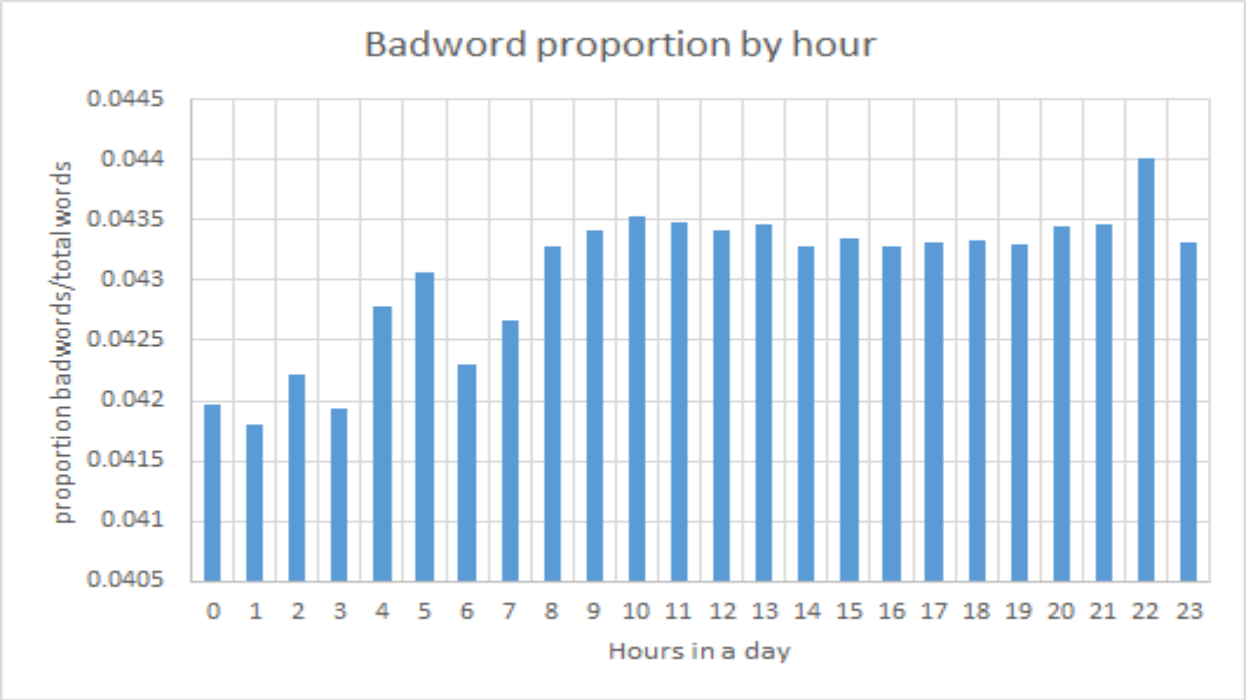
Input to Mapper:
The input to mapper is the twitter data in HDFS

Output of Mapper:
The output of Mapper is a <key-value> pair- <tweet_hour , totalwords_totalbadwords> for all tweets in different hours of a day.

Input to Reducer:
Sorted list of key-value pairs from Mapper

Output of Reducer:
The output of the reducer is a list of proportion of bad words present in all tweets for each hour (0-23) in a day.

A graph plotting the expected bad word proportion to hour is shown in graph 2.



Graph_2: Bad word proportion by hour