**Report Document**

**Cloud Project No: 3**

**QUESTION 1:**

What hour of the day does @PrezOno tweet the most on average, using every day we have twitter data?  Include a plot of the expected number of tweets for each hour of the day, for those he did tweet.

**APPROACH**:

* The question was selected, the twitter data downloaded from the HDFS and the data was analyzed.
* The JSON format of the twitter data was analyzed to understand the format.
* The requirements for solving the problem, i.e. working with JSON data using python, working on Hadoop cluster, working with virtual machine on CEAS cloud etc., studied and understood.
* Understanding what is required from the twitter data to answer the question (Example, ‘Screen\_name’ of the tweeter, time of tweet in ‘created\_at’) and how to extract it from the JSON data.
* Defining what mapper and reducer should compute and output.
* Writing the python code for mapper and reducer.
* Executing the code locally on few data files to check correctness of code.
* Running the code on the cluster to get the output.

**ALGORITHM:**

Mapper outputs the hour and date only if the tweet is created by ‘PrezOno’.

Reducer increments ‘summ’ for every occurrence of same ‘key’ (hour).It increments ‘count’ for every change in ‘val’(date) and so ‘count’ is initialized to 1 to account for the first date. Average is calculated at every transition in key, displayed with hour and dictionary ‘output’ is updated with average ‘avg’ as key and ‘hour’ as value for calculating the hour with maximum average. This step is repeated at end of for loop for last set of key. For example, if PrezOno tweets 2 times one day and 3 times another day at hour 12, summ = 5, count =2 , avg = 2.5.

**Mapper (mapono.py):**

1. For each line (tweet in JSON format) from standard input, do the following:

a. Convert the JSON string into python dictionary using json.loads() function.

b. Extract the string value for key ‘created\_at’ from the dictionary and slice the string to get hour and date.

c. Extract the string value for ‘screen\_name’ in ‘user’ from the dictionary and convert to lowercase to get tweeter.

d. If tweeter is ‘prezono’, then print the hour and date to standard output.

**Reducer (redono.py):**

1. Initialize the required variables.

2. For each line (output of mapper) from standard input, do the following:

a. Split the line to get the key (hour) and value (date).

b. If the key value changes from previous input’s key,

Compute average and print the hour (key), avg, summ and count to standard output.

Update dictionary ‘output’ with average ‘avg’ as key and ‘hour’ as value

Reset the variables to initial value.

c. Increment summ for each occurrence of key.

d. Increment count for each transition of date.

3. Repeat sub steps of 2b for last set of key.

**RESULT** :

The observation made from analysing the twitter data for PrezOno’s tweets is that PrezOno tweets the most between 3pm and 4pm (i.e. 15.00.00 to 15.59.59) with the maximum average tweets of 1.3 in that hour.

Here is the plot of expected number of tweets from PrezOno for each hour of the day from his everyday tweets. The hour is represented in 24 hour format. For example, the hour between 3pm and 4 pm is represented by hour 15. The expected number of tweets is the average tweets every day in that hour. For example, if he alternates between 2 and 3 tweets per day, his average would be 2.5

**Question 2**: How does @PrezOno’s tweet length compare to the average of all others?  What is his average length?  All others?

**Approach to the problem:**

* The twitter data set was first downloaded and the format of the data was first was found out. It was in the json format arranged in a hierarchy. My question was to find out average tweet length of PrezOno and the average tweet length of other people who have tweeted and finally compare them.
* The next step was to find out the attributes from the twitter data in json format. The attributes that were required to answer the questions: Screen\_name and the text attribute which had the text content of the tweet.
* The next step was to formulate a map-reduce algorithm to solve this problem.
* The next step was to choose a language to practically implement it.
* Python is the language that is chosen to implement this.
* Then average tweet length of Prezono and the average tweet length of others were then calculated.

**Analysis:**

* The analysis of the problem consists of the following steps:
* Analysing the twitter data set that is used to answer the question.
* Listing down the attributes that are required to solve this problem.
* Analysing and understanding what those attributes represent.
* Finding the average tweet length of PrezOno and the average tweet length of others.
* Finding out the ratio between the average tweet length of PrezOno and the average tweet length of others.
* The final step is to compare them and then give the result of the analysis.

**Map–Reduce Algorithm formulated to solve this problem**:

**Mapper:**

* The json data is first parsed and then stored.
* For every line in the data set the text attribute that has the tweet message is extracted.
* For every line in the data set the screen name attribute is then extracted.
* The length of the tweet messages are then calculated and stored.
* If the screen name is Prezono then the (key,value) pair is (‘ono’, string\_length\_of\_the\_message)
* Else for the other people who have tweeted the (key,value) pair is (‘notono’, string\_length\_of\_the\_message).
* These values are passed as the input to the reducer.

**Reducer**:

* For every line in the standard input stream the data is split by a tab space.
* If the key is ‘ono’ the cumulative sum and the count are calculated and stored separately.
* The cumulative sum and the total count for non-ono tweets are then calculated separately.
* The average length ono’s tweet is calculated.
* The average length of nonono tweets is calculated.
* Finally the ratio between average length ono’s tweet and the average length of nonono tweets is calculated.

**Results:**

The average tweet length of prezono: 35578 / 341 = 104.3

The average tweet length of nonono tweets: 496716166 / 6079961 = 81.6

Ratio : 1.278 (Comparison).