**Report Document**

**Cloud Project No: 4**

**Team Members: Radhika Suresh, Thiruvengadam Kalai Kannan**

**Questions chosen: 2, 3**

**Question 2: Thiruvengadam Kalai Kannan**

**Question 3: Radhika Suresh**

**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.

**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 Spark etc., studied and understood.
* Understanding what is required from the twitter data to answer the question (Example, ‘Screen\_name’ of the tweeter, day and date of tweet in ‘created\_at’) and how to extract it from the JSON data.
* Defining what should be computed and output.
* Writing the python code in interactive mode (pyspark) and as a python file.
* Executing the code locally on few data files to check correctness of code.
* Running the code on the cluster to get the output.

**ALGORITHM:**

* The SparkContext is created and each twitter textfile is loaded.
* The getDay() function is used to get a key-value pair of Day and Date for each PrezOno tweet after extracting the required data from the json string.
* The day and date of all PreOno’s tweets is in a RDD which is transformed into a python list and sorted and stored in list ‘days’.
* 1. Initialize the required variables.

2. For each (day,date) pair from sorted list ‘days’, do the following:

a. Assign the key (day) and value (date).

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

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

Update dictionary ‘output’ with average ‘avg’ as key and ‘day’ 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.

* The ‘day’ with maximum average is calculated from the dictionary and displayed with the average.
* The SparkContext is stopped.

Logic for average calculation:

The program increments ‘summ’ for every occurrence of same ‘key’ (day).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 ‘day’ and dictionary ‘output’ is updated with average ‘avg’ as key and ‘day’ as value for calculating the day 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 Monday and 3 times another Monday, summ = 5, count =2 , avg = 2.5.

**RESULT** :

The observation made from analysing the twitter data for PrezOno’s tweets is that PrezOno tweets the most on Friday with the maximum average tweets of 1.22 in that day of the week.

Here is the plot of expected number of tweets from PrezOno for each day of the week from his everyday tweets. The expected number of tweets is the average tweets every week in that day. For example, if he alternates between 2 and 3 tweets on Monday, his average for Monday would be 2.5

**Question 3**: 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 an 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 in spark.
* 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.

**Algorithm formulated to solve this problem**:

* The json parsing is the first step to be done.
* The getText is used to return the text of the tweet when a full tweet object is provided.
* The returnTrueIfPrez function extracts the screen name and returns true when screen name is PrezOno.
* The returnTrueIfNotPrez function extracts the screen name and returns true when screen name is not PrezOno.
* The input i.e the data set is given as a command line argument.
* **Filter** extracts each element in the sequence for which the function returns **true**.
* The filter is applied to each element in the sequence for which the functions.
* The prez\_tweets variable stores all the items which are obtained after applying filter to the function returnTrueIfPrez.
* The non\_prez\_tweets variable stores all the items which are obtained after applying filter to the function returnTrueIfNotPrez.
* The prez\_texts stores the text of tweets posted by PrezOno using flattmap function which provides the one to many correspondence.
* The prez\_lengths is a RDD that stores the length of the tweets posted by PrezOno.
* The non\_ prez\_texts stores the text of tweets posted by others using flatmap function which provides the one to many correspondence.
* The non\_prez\_lengths is a RDD that stores the length of the tweets posted by others.
* The average tweet length of PrezOno is calculated by applying stats function to the RDD prez\_lengths.
* The average tweet length of nonono tweets is calculated by applying stats function to the RDD non\_prez\_lengths.

**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).