CSE 482: Big Data Analysis (Spring 2020) Homework 2
Due date: Monday, February 19, 2020

Please make sure you submit a PDF version of your homework via D2L.

1. Write the corresponding HDFS commands to perform the tasks described for each question below. Type `hadoop fs -help` for the list of HDFS commands available. You can also refer to the documentation available at `https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html`. To double-check your answers, you should test the commands to make sure they work correctly.

   (a) Suppose you are connected to a master node on AWS running hadoop with Linux operating system. Assume you have created a data directory named `logs` (on the Linux filesystem of the master node), which currently contains 1000 Web log files to be processed. Write the hadoop DFS commands needed to upload all the Web log files from the `logs` directory to the directory named `/user/hadoop/data` on HDFS. Assume the `/user/hadoop/data` directory has not existed yet on HDFS. Therefore, you need to create the directory first before transferring the files.

   (b) Write the HDFS command to move the Web log files from the `/user/hadoop/data` directory on HDFS to a shared directory named `/user/share/` on HDFS. After the move, all the files should now be located in the `/user/share/data/` directory. Write the HDFS command to list all the files and subdirectories located in `/user/share` directory. To make sure the files have been moved, write the corresponding HDFS command to list all the files and subdirectories located in the directory named `/user/hadoop` to verify that the `data` subdirectory no longer exists.

   (c) Suppose one of the files located in the `/user/share/data/` directory named `2020-01-01.txt` is corrupted. You need to replace the corrupted file with a new file named `2020-01-01-new.txt`, which is currently located in the `logs/new` directory on the local (Linux) filesystem of the AWS master node. Write the HDFS commands to (1) delete the corrupted file from `/user/share/data/` directory on HDFS, (2) Upload the new file from `logs/new` directory to the `/user/share/data/` directory on HDFS, and (3) rename the new file on HDFS from `2020-01-01-new.txt` to `2020-01-01.txt`.

   (d) Write the HDFS command to display the content of the file `2020-01-01.txt`, which is currently stored in the `/user/share/data/` directory on HDFS. As the file is huge, write another HDFS command to display the last kilobyte of the file to standard output.

2. Consider a Hadoop program written to solve each computational problem and dataset described below. State how would you setup the (key,value) pairs as inputs and outputs of its mapper and reducer classes. Assume your Hadoop program uses TextInputFormat as its input format (where each record corresponds to a line of the input file). Since the inputs for the mappers are the same (byte offset, content of the line) for all the problems below, you only have to specify the mappers' outputs as well as reducers' inputs and outputs. You must also explain the operations performed by the map and reduce functions of the Hadoop program. If the problem requires more than one mapreduce jobs, you should explain what each job is trying to do along with its input and output key-value pairs. You should solve the computation problem with minimum number of mapreduce jobs.

   Example:

   Data set: Collections of text documents.

   Problem: Count the frequency of nouns that appear at least 100 times in the documents.

   Answer:

   (i) Mapper function: Tokenize each line into a set of terms (words), and filter out terms that are not nouns.

   (ii) Mapper output: key is a noun, value is 1.

   (iii) Reducer input: key is a word, value is list of 1's.

   (iv) Reduce function: sums up the 1's for each key (noun).

   (v) Reducer output: key is a noun, value is frequency of the word (filter the nouns whose frequencies are below 100).

   (a) **Data set:** Car for sale data. Each line in the data file has 5 columns (seller_id, car_make, car_model, car_year, price). For example:

   ```
   1234,honda,accord,2010,10500
   2331,ford,taurus,2005,2400
   ```

   Problem: Find the median price (over all years) for each make and model of vehicle. For example, the median price for ford taurus could be 8000.

   (b) **Data set:** Netflix movie rental data. Each record in the data file contains the following 4 columns: userID, rental_date, movie_title, movie_genre. For example, the record

   ```
   user111 12-20-2019 star_wars scifi
   user111 12-21-2019 aladdin animation
   user111 12-25-2019 lion_king animation
   ```

   Problem: Find the favorite movie genre of each user. In the above example, the favorite genre for user111 is animation.

(c) **Data set:** Youtube subscriber data. Each line in the data file is a 2-tuple (user, subscriber). For example, the following lines in the data file:

```
john mary
john bob
mary john
```

show that mary and bob are subscribers of John's Youtube videos.

Problem: Find all pairs of users who subscribe to each others' videos. In the example above, john and mary are such pair of subscribers, but john and bob are not (since john does not subscribe to bob's videos)

(d) **Data set:** Loan applicant data. Each line in the data file contains the following attributes: marital status, age group, employment status, home ownership, credit rating, and class (approve/reject).

```
single, 18-25, employed, none, poor, reject.
single, 25-45, employed, yes, good, approve.
```

Problem: Compute the entropy of each attribute (marital status, age group, etc) with respect to the class variable.

(e) **Data set:** Document data. Each record in the dataset corresponds to a document with its ID and set of words that appear in the document. For example, the following records contain the set of words that appear in documents 12345, 12346, and 12347, respectively.

```
    12345 team won goal result
    12346 political party won election result
    12347 lunch party restaurant
```

Problem: Compute the cosine similarity between every pair of documents in the dataset. Given a pair of documents, say, u and v, their cosine similarity is computed as follows:

$$\text{cosine}(u, v) = \frac{n_{uv}}{\sqrt{n_u \times n_v}},$$

where n_{uv} is the number of words that appear in both u and v, n_u is the number of words that appear in document u and n_v is the number of words that appear in document v. For the above example, cosine(12345,12346) $= 2/\sqrt{20}$ whereas cosine(12346,12347) $= 1/\sqrt{15}$. Hint: You will need two mapreduce (Hadoop) jobs for this problem.

3. Download the data file `Titanic.csv` from the class Web site. Each line in the data file has the following comma-separated attribute values:

   ```
   PassengerGroup,Age,Gender,Outcome
   ```

   For this question, you need to write a Hadoop program that computes the mutual information between every pair of attributes. The reducer output will contain the following key-value pair:

   - key is name of attribute pair, e.g., (Age, Outcome).
   - Value is the their mutual information.

   Deliverable: Your hadoop source code (*.java), the archived (jar) files, and the reducer output file, which must have 2 tab-separated columns: attribute pair and its mutual information value.