CSE 482: Big Data Analysis (Spring 2019) Homework 4
Due date: Monday, April 1, 2019

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

1. Write the corresponding HDFS commands to perform the following tasks. Each of these tasks must be accomplished with a single HDFS command. Hint: type `hadoop fs -help` for the list of 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're encouraged to test the HDFS commands to make sure they work correctly.

   (a) Make a directory on HDFS named `/user/hadoop`.

   (b) Upload a file named `patients.csv` located in the current working directory of the local filesystem to `/user/hadoop` on HDFS.

   (c) Rename the uploaded file on HDFS from `patients.csv` to `data.csv`.

   (d) List all the files and subdirectories located in the directory named `/user/cse482` on HDFS.

   (e) Move the uploaded file from its current directory on HDFS to another directory `/user/cse482` on HDFS.

   (f) Make a copy of the file `data.csv` from the `/user/cse482` directory to `/user/patients/` directory on HDFS. The copied file should be named as `patients.csv`.

   (g) Display the content of the HDFS file `/user/patients/patients.csv`.

   (h) Delete the file `/user/cse482/data.csv` on HDFS.

   (i) Merge all the files located in the HDFS directory `/user/cse482/results` into a single file named `output.txt` to be stored in the current working directory of the local filesystem.

   (j) Download the file named `/user/patients/medication.txt` on HDFS to the current working directory of the local filesystem.

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:** Amazon book ratings data. Each line in the data file has 4 columns (reviewer_id, book_id, book_genre, rating), where ratings are integer-valued ranging from 1 to 4.

   Problem: Identify the highest rated book, i.e., the book with highest average rating, for each book genre. Note that each book can have more than one ratings (e.g., by different reviewers).

   (b) **Data set:** Movie preference data. Each record in the data file contains the movie title and list of users who liked the movie. For example, the record

```
jaws  user111 user134 user313 user5812
star_wars user111 user313 user388 user4422
```

   Problem: For each pair of users, count the number of movies they both liked. The output may exclude pairs of users who do not have any movies they both liked.

   (c) **Data set:** Maximum and minimum daily temperature readings for weather stations from around the world. Each line in the data files has 4 columns (station id, date, max temperature, min temperature).

Problem: Find the station id and date of anomalous temperature readings in the dataset. A temperature reading is anomalous if the minimum daily temperature exceeds the maximum temperature for the given day.

(d) **Data set:** Instagram friendship graph. Each record corresponds to an Instagram user, followed by a list of his/her friends. For example, the graph data may contain the following records:

```
john123 mary456 tom312 lee222
mary456 john123
tom312 john123 lee222
lee222 john123 tom312
```

The first line above states that mary456, tom312, and lee222 are friends of john123.

Problem: Find pairs of Instagram users who are not friends with each other but who share one or more common friends. This is known as the "friend-of-a-friend" (FOF) problem. For example, mary456 and tom312 are both friends of john123, but they are not friends with each other. The Hadoop program should only output the pair (u, v) if $u < v$. In the previous example, the program should only output the pair (mary456, tom312) but not (tom312, mary456).

(e) **Data set:** Cancer data. Each line in the data file corresponds to a patient with the following nominal-valued attributes: patientID, gender, marital_status, Smoker, Weight_class, and Class, where the Class attribute has value yes or no to indicate whether the patient has cancer.

```
12345, female, married, smoker, normal, yes.
136666, male, single, nonsmoker, normal, no.
14423, male, married, smoker, overweight, yes.
```

Problem: Compute the gini index for each of the following attributes: gender, marital_status, smoker, and weight_class, based on the distribution of their class values.

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

```
preg,plas,pres,skin,insu,mass,pedi,age,class
```

For this question, you need to write a Hadoop program that computes the covariance value between every pair of attributes (except for the class). For example, the covariance value between any given pair of attributes (x, y) can be calculated as follows:

$$cov(x, y) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}),$$

where N is the number of rows in the dataset, $\bar{x}$ is the average value of attribute x and $\bar{y}$ is the average value of attribute y.

Deliverable: Your hadoop source code (*.java), the archived (jar) files, and the reducer output file, which must have 2 tab-separated columns, (attribute1,attribute2) and its covariance value. You can use the column ID to represent each attribute instead of the actual attribute name.

4. Download the sample dataset `lastFM.csv` from the class website. The dataset contains information about the number of times each user played a song performed by an artist on the LastFM streaming radio website. For example, the first 2 lines of the data file is as follows:

```
1,all:my:faults,288
1,all_ends,229
```

The first column corresponds to user ID, the second column is the artist name, and the third column is the number of times the user played the song performed by the artist.

For this question, you need to write a Hadoop program that returns the name of the favorite artist for each user (i.e., the artist with the most number of plays). The output of the reducer must have 2 columns: userID and the name of their favorite artist.

Deliverable: Your hadoop source code (*.java), the archived (jar) files, and the reducer output file.