67-364 / 95-885 Data Science, Spring 2019

Assignment 8: Working with Spark and Pig

Due: Friday, May 3, 11:59 pm

In this assignment you will explore two problems using Spark and Pig. While typically this would be an assignment due in one week, given that this is the last assignment for this course, there is a relaxed time frame of 2 weeks for completion. Hence flex days cannot be applied to this assignment.

1. **Expressing PageRank using Spark.** The notebook `spark-pagerank-exercise.ipynb` provides an outline of expressing the PageRank algorithm using RDDs. Review the notebook to identify the portions that need to be completed. You will find it helpful to review the various instances of the PageRank algorithm we have discussed in class e.g., the Excel spreadsheet simulation, the slides discussing step by step the algorithm. The more time one spends understanding the nuances of the algorithm, the less time it will take to complete the notebook. The starter code shows output from `graph-1.txt`, which we saw in an earlier assignment. Do test it on other graphs also (e.g., `graph-2.txt` and `wikipedia-example.txt`).

2. **Most popular gender-neutral names.** In class we have looked at the baby names data set. The data set is 117MB. Hence rather than create 60+ copies of the same data set, please link to my copy with the following command from bridges:

```
% ln -s ~raja/pub/babynames2017.csv .
```

This will create a *symbolic link* called `babynames2017.csv` in your current directory. Check the contents of the file with:

```
% more babynames2017.csv
```

Exit from `more` by pressing `q`.
Data is of the form `state, gender, year, name, number`. The first three lines from the file are below:

```
AK,F,1910,Mary,14
AK,F,1910,Annie,12
AK,F,1910,Anna,10
```

In this question you will write a Pig program to answer the question: *which are the most popular gender neutral names*. When working with large data sets, you will typically develop and debug your program against a smaller representative set. First, by hand, create a file, `test.csv` with the same structure of `babyname2017.csv` but with only 6 rows in it. Make up the data any way you want, but ensure that your test data demonstrates the type of filtering described below:

1. Determine all male_names and their respective counts
2. Determine all female_names and their respective counts
3. Join the two relations from steps 1 and 2
4. Filter the joined relation when the ratio of the count of male names to female names is within the closed interval [0.25, 4]

5. Sort the final answer in descending order of the sum of total number of male and female names

Note that each of the above steps may translate to more than one Pig operation. Test your code along the lines of:

```
% pig -x local -p input=test.csv neutral_names.pig 2> TRACE
```

Finally test your code with

```
% pig -x local -p input=babynames2017.csv neutral_names.pig 2> TRACE
```

It is highly recommended that you develop the program incrementally --- write a couple of pig statements, test your code against `test.csv` and gradually build the program.

As we discussed in class, `2>` redirects `stderr` output. Just `>` redirects `stdout` output. You can redirect the output of the pig program to a file with something like:

```
% pig -x local -p input=babynames2017.csv neutral_names.pig 2> TRACE > ANS
```

No output will appear on your screen; all output will be redirected to the file `ANS`.

To guide your development efforts, a line of my output is:

```
((Jamie,M),82135,(Jamie,F),265301,347436,0.30959174673295614)
```

Few points to note:

1. Parameterize your code to take the input file from the command line (as shown above)
2. Note that grouping has been done by both `name` and `gender`. The above line of output is to be read as: there are a total of `82135` baby boys with the name `Jamie` and `265301` baby girls with the name `Jamie`, `82135+265301 = 347436, and 82135/265301 = 0.30959174673295614`
3. Sort your final output in descending order by the total number of babies with that name. In the line given above you would sort by the column with the number `347436`.
4. Pig does integer division. Hence `1/2` would be `0`. Cast the number to a `float` before performing the division to get a real answer
5. Give aliases to all field. Do not use positional parameters e.g., `$1`
6. Your output will consist of 647 lines.
7. Usual style guidelines apply to Pig code too. Most significantly reasonably comment your code and ensure that each line of code is not longer than 70 characters (so that horizontal scrolling is not needed).

You may need to refer to some additional constructs of Pig beyond what we've discussed in class. Please refer to the online documentation available off of the course schedule.

What to submit:

A single zip file, `a8.zip`, with:

1. `spark-pagerank-exercise.ipynb`
   `graph-1.txt`
   (no need to submit `graph-2.txt` or `wikipedia-example.txt`)

2. `neutral_names.pig`
   `test.csv`

   Do not submit `babynames2017.csv`