SI 618 Fall 2022 Homework 6 (100 points)

Data to be used in this homework:

On the Greatlakes cluster, the data file can be found in the following location:

/scratch/siads618f22_class_root/siads618f22_class/shared_data/hw6_data/book_ratings_data_jsonl.json

This file is a combined and cleaned version of the Goodreads book reviews data files.

Note that you do not need to download the dataset yourself as it is already put into Greatlakes cluster.

Average Review Scores by Category for Authors

The goal is to compute the number of reviews, average review score (rounded to two decimals), number of reviews from active users(User_id won't be null) and number of ratings from inactive users for each category for each author. If a review is for a book under multiple categories and multiple authors, its review values should be attributed to all of the categories for all those authors. For this analysis, we are interested in using only reviews that have a helpfulness value of at least 25% (eg. A 10/10 helpfulness review is 100% helpful and a 4/5 helpfulness review is 80* helpful) and contain a review summary. Reviews for books missing an author should be considered as having an author named 'Unknown'. In the same way, reviews for books missing categories specifications should fall under the 'Unknown' category.

Your final result should be a TSV file that matches the provided si618_hw6_preferred_output.tsv file.

In this desired output file, each row contains 6 columns, which are separated by a tab. For example, consider this following row:

```
A A Milne       Juvenile Fiction        4.71    103     82      21
```

This means the author "A A Milne" in the category of "Juvenile Fiction" has an average review score of 4.71 from 103 reviews. Out of this 103, 82 reviews came from users who were active and 21 reviews came from users who were inactive.

The rows in the output file should be sorted in ascending order of author names and the categories under each author are sorted by the decreasing order of the number of reviews for that category under that author. Further ties are resolved by the ascending order of the category names.

Data entry errors and inconsistencies in letter casing are present in the dataset. For eg.,

- 'Engineering' and 'ENGINEERING' are considered different
- First and second names of authors/publishers are themselves entered as categories

Multiple authors/categories being separated by commas can be ambiguous. Please DO NOT make any changes in these cases. In a sample cases of authors like ‘Jack, John and Company, inc’ all three values separated by commas - ‘Jack’, ‘John and Company’ and ‘inc’ - are treated as individual authors in the preferred output. As a result, you can find some strange author/category names like ‘c’, ‘rd’, ‘univ’ and ‘1939-1945’. Please leave these intact in your output without making any changes in order to meet the preferred-output. However, while parsing these comma-separated strings to get the individual values, DO NOT consider the white space or empty strings between commas as individual values.

You MUST use Spark to do this homework. A non-Spark solution will not get any credit. You must also generate the output by submitting your python file as a batch job (we use bash shell) on Greatlakes.

HINT: You can modify from the provided example code starter_code.py. Save it as “youruniqname_si618_hw6_solution.py”

Batch Job submission instructions:

Before we begin, you will need a login account for accessing Greatlakes under your uniqname. If you do not have one already, please request one at the earliest possible here.

Login and File uploads to Greatlakes

- Please login to the great-lakes terminal by using the following command **‘ssh uniqname@greatlakes.arc-ts.umich.edu’.** Make sure you are connected through MWireless or the UMVPN if you are accessing from a different network.
- You can use Cyberduck/WinSCP to upload your batch-job and python_solution files to your directory in the Greatlakes cluster. You can also upload files by accessing your home directory at ‘https://greatlakes.arc-ts.umich.edu/pun/sys/dashboard/files/fs/home/uniqname’.

Submitting a batch job

- The only modification in the batch-job.sh file that is recommended is adding the right uniquename in the last line where the python source code file is referenced
- You can use the **sbatch batch-job.sh** command to submit the batch job. This would submit a batch job and print an id for the job. You can use the **tail -100 slurm-<batch_job_id>.out** to print the last 100 lines of the output stream that is printed by executing the batch job. This can be used to debug your code.
- If you receive the following error: ‘When running with master 'yarn' either HADOOP_CONF_DIR or YARN_CONF_DIR must be set in the environment’, please run the following command in the great-lakes terminal : export HADOOP_CONF_DIR=$HADOOP_HOME/etc/Hadoop
- The solution file for this homework should ideally run within a couple minutes. Please do not leave your batch job running for long durations. If your code is

suboptimal and taking very long durations, please use **‘scancel <batch_job_id>’** command to cancel your job and avoid wastage of computing resources. You can check your job queue using the **‘sq uniqname’** command.

- Once the job is complete and the output files are generated, you can concatenate them using the **cat uniqname_si618_hw6_output/part*> uniqname_si618_hw6_output.tsv** command

What to submit:

Submit a zip file named uniqname_si618_hw6.zip containing:

- Your Python source code: uniqname_si618_hw6_solution.py
- The tsv output file: uniqname_si618_hw6_output.tsv