Computing Science & Mathematics
University of Stirling

CSCU9DC Assignment 2024

Due: Wednesday 17th April at 16:00

Movie Rating Analysis

Your task in completing this assignment is to analyse a range of movie reviews and extract the highest ranked films for a given genre of movie in a specific range of years and across all years.

You have been provided with a list of over 100,000 ratings from a group of reviewers in a file called 'ratings.txt'. Each line in the file contains an individual user's rating for a particular movie as follows:

```
387   Kickboxer  Action     1989  2.5
599   Lionheart  Action     1990  0.5
599   Tornado!   Action     1996  2
599   Steel      Action     1997  1
217   Steel      Action     1997  3
140   Yojimbo    Action|Adventure  1961  1
23    Yojimbo    Action|Adventure  1961  3
...
```

The columns are organised as follows: Reviewer ID, Movie Title, Genres, Year, Rating. The genres entry will contain 1 or more genres separated by a vertical bar. 2 smaller versions of the above data are also supplied in the files r5.txt and r100.txt and contain 5 rows of data and 100 rows of data respectively.

You have also been supplied with a file called 'years.txt' which contains a list of the years that movies should have been released in for them to be included in your analysis. If a movie release date is not in this list, it should be excluded from the results of the analysis. If however this file is empty, all ratings and years should be included in the results.

You are required to implement a Hadoop based Map/Reduce solution that will, *in a single run*, extract the highest ranked films for a given genre of movie filtered on the listed years in the 'years.txt' file. Your code must use data that is stored on HDFS and execute correctly by processing this data using map and reduce processes. You may implement your solution in a standard Python map/reduce streaming form (as used in the supplied runhadoop.sh script), in Spark using RDDs (use of standard data frames is not a valid solution) or in Java by creating custom mapper, combiner and reducer classes. Please choose only one of these solutions and note clearly at the start of your report which approach you have chosen and why.

The number of results and genres you will get back will vary depending upon the size of the data set that you use (for example, in the data set with only 5 ratings, there are only two genres present (Action and Adventure) and only 4 movies are rated.

Your final output should be in the following format where each genre shows the movie that has the highest average ranking for a given genre along with the average ranking for that movie based on the range of years to be analysed (p*lease note that the following example output is* ***not*** *the correct answer – it is just used to demonstrate the required structure of the results and to give you an idea of what is required).*

```
Action      Yojimbo    4.2
Adventure   Yojimbo    4.2
Animation   Zootopia   3.9
Children    Zootopia   3.9
Comedy      Zootopia   3.9
Crime       Machete    3.2
Drama       Ben-Hur    3.9
Fantasy     Highlander 3.7
Film-Noir   Third Man  4.1
Horror      Aliens     4.0
IMAX        Avatar     3.6
Mystery     Hanna      3.3
Romance     Twister    3.3
Sci-Fi      Aliens     4.0
Thriller    Goldfinger 3.8
War         Troy       3.4
Western     Rango      3.6
```

You should produce 2 sets of results, one where the years.txt file contains the listed set of years and one where the years.txt file is empty and indicates that all years should be included. You should find that your results are different when comparing all years versus the selected set of years.

In order to ensure that a movie rating with only a single review and a maximum score of 5 does not dominate the results, you are required to only rank movies with at least 15 ratings. You should set a variable to specify this minimum and be able to change it from a value of 1 for testing purposes (when working with a small set of data where most movies will have less that 15 ratings) up to 15 when running your solution against the full set of 100,000 ratings.

Once you complete the assignment, you should submit the following items depending upon which solution you decide to implement (template versions of each of the solution types are provided):

1. Solution Type:
   - Python Streaming solution: mapper.py, combiner.py, reducer.py
   - Spark Solution: movieSpark.py
   - Java: MovieRatings.java
2. The outputs your program produced on Hadoop as text files (not a screen shot or image)
3. A report detailing your design and analysis of possible solutions

Step 1, Design – 50 Marks

If you have chosen the Python streaming or Java based solution, consider the Map/Reduce design you have chosen to implement and compare developing two solutions: one without a Combiner and one with a Combiner. Discuss what keys and values the mapper will emit compared with the Combiner or Reducer and how this affects the efficiency of your solution. Some points to consider are how much data will be moved across the network and the maximum number of different reducers that could be used in your design.

If you have chosen a Spark based solution, you will need to have multiple map and reduce stages in order to arrive at the correct final answer. Discuss in detail what each of the stages need to do, what alternatives you investigated for each stage and what the trade-offs were.

For all solutions, discuss your key/value choices at each stage of the process and why you decided to use a particular key/value structure, what alternatives you considered and why you decided that they were less efficient or impractical. This should be related to aspects of maximising parallelism and making the best use of the Hadoop distributed processing cluster.

You should also address whether or not your solution is fully optimal for the task and if you think it could be improved. If you think it could be improved but could not work out how to code it, discuss what type of changes would need to be made to make your solution more effective and why these changes would improve performance.

Step 2, Implementation – 50 Marks

Using the sample code provided on the assignment page as a starting point, modify this code to produce an efficient solution that works according to the design choices discussed above.

Each version of the sample code is just a revised version of the original Word Count example from the lectures and practicals and just counts each review type. It will need significant changes to meet the desired requirements but does break up each line of data into its component parts.

For the Python streaming approach, you can use the simhadoop.sh script to test your code within a terminal on either your own PC or the Jupyter Lab setup. This version of simhadoop.sh will also produce intermediary files for the mapper output (mapout5.txt), combiner output (comout5.txt) and the reducer output (results5.txt). Please also be aware that when you use the simhadoop script, all the data from the mapper or combiner will be sent to a single reducer. On Hadoop this may not be the case and multiple independent reducers could run. If you have built a solution that relies on only one reducer working, it may work fine with the simhadoop script but not work on Hadoop and will likely be incorrect. It is also likely to be highly inefficient so a redesign would be worth considering.

For the Spark and Java approaches, you will need to test your code directly on Hadoop and will get relatively little feedback regarding any errors that occur. In the Spark case, you may at least get error reports sent to your results.txt file which you can use to help debug your code. In this case, make sure you close down any tabs in Jupyter Lab that show the contents of results.txt from prior runs before running your code again otherwise you will not see the updated results of your test run.

Your final submitted code and output should be run with the full *ratings.txt* file on the Hadoop server and the output of this with a full and an empty version of the years.txt file should be submitted along with your code. *Please note that actual text files should be submitted - not images of them in your report.*

Submission Details

Please write up your work in a report as detailed above and submit it as a PDF, Word or Pages document along with your code on the assignment page on Canvas. Ensure that you also include your actual Hadoop output. Please write your 7 digit student ID number on the front of your report and also in the code files as a comment near the top. ***Do not*** *provide your name* in any of these files. *This assignment is worth 100% of the marks for the CSCU9DC module and the deadline for submission is Wednesday the 17th of April at 16:00 BST.*

Late Submission

If you cannot meet the assignment hand in deadline and have good cause, please see the module coordinator to explain your situation and discuss an extension as soon as possible after the problem becomes known. Coursework will be accepted up to seven days after the hand-in deadline (or expiry of any agreed extension) but the mark will be lowered by three marks per day or part thereof. After seven days the work will be deemed a non-submission and you will receive an X for this assessment.

Plagiarism

Work which is submitted for assessment **must be solely your own work**. Plagiarism means presenting the work of others as though it were your own. The University takes a very serious view of plagiarism, and the penalties can be severe (ranging from a reduced mark in the assessment, through a fail mark for the module, to expulsion from the University for more serious or repeated offences). We check submissions carefully for evidence of plagiarism, and pursue those cases we find.

Do not be tempted to give your code to other students to help show them how to solve the problem since you will be implicated in a plagiarism case and will receive the same penalty as any student who is found to have submitted work that is similar or the same as your own.