PROGRAMMING ASSIGNMENT 4

ANALYZING THE MOVIELENS DATASET USING SPARK
VERSION 1.1

DUE DATE: Wednesday November 19th, 2025 @ 8:00 pm

OBJECTIVE

The objective of this assignment is to gain experience in developing Java Spark programs. As part of this assignment, you will be working with the MovieLens dataset that describes ratings and free-text tagging activities from MovieLens, a movie recommendation service. This dataset was created by GroupLens and primarily hosted at Kaggle. You will be using Apache Spark (version 3.5.0) to implement this assignment.

This assignment must be done individually and will account for **10 points** towards your cumulative course grade. There are a few components to this assignment, and the points-breakdown is listed in the remainder of the text. This assignment may be modified to clarify any questions (and the version number incremented), but the crux of the assignment and the distribution of points will not change.

This assignment is to be done individually.

Generative AI Use and Consequences

Use of AI tools such as ChatGPT, Claude, Github Co-Pilot, or anything of their kind to write or "improve" your code or written work at *any* stage is prohibited; this includes the ideation phase. It is your responsibility to ensure that you don't have the GitHub Co-Pilot extension installed in your IDE; assignment solutions generated by Co-Pilot aren't written by you. Turning in code or an essay written by generative AI tools will be treated as turning in work created by someone else, namely an act of plagiarism and/or cheating. At a minimum, this will result in a 100% deduction (i.e., you will receive a −10/10). To ensure fairness and maintain integrity, grading will also include code reviews, interviews, and on-the-spot code modifications.

Ultimately, you will get out of the class what you put in. Simply copying and pasting code from generative AI tools is not only unethical, it robs you of the chance to learn. Here are four reasons why these generative AI tools undercuts your own education:

1. They take away the struggle that leads to understanding. They rob you of the ability to think and learn the concepts for yourself. Solving problems yourself is how concepts stick. If the AI does the work, what's left for you to learn?
2. You will struggle with the in-classroom quizzes and exams where you will not have access to these tools.
3. Yes, AI tools will become an important part of a software engineer's workflow. But to use them effectively later, you first need solid expertise in the subject matter; and, that only comes from practicing *without* them.
4. These tools are prone to generating imperfect or even incorrect solutions, so trusting them blindly can lead to bad consequences.

1 Cluster setup

For this assignment, the Hadoop cluster with Spark running on every node has been already set up for the auto-grader. Datasets have already been staged too. Your programs will process the staged Datasets.

2 Analysis of the MovieLens Dataset

You should develop a Java Spark program that leverages the Dataset construct to process the main *and* supplementary datasets to answer the questions.

We will run your program with the following command:
spark-submit ––deploy-mode cluster solution.jar <genre> <year1> <year2> … <yearN> <input-folder> <output-folder>

Use only data for the given years to answer **all** questions. Use data for the given genre to answer **only** questions 2 and 10.

Q1.	How many movies were released every year? Ignore movies with no year. Output must be comma-separated pairs of year and number of movies. Pairs must be sorted by year. Example output: 1980, 14 1981, 16 1982, 20 …	**1pt**
Q2.	How many movies have the genre "<genre>" ? Output a single number. Output format: 75	**1pt**
Q3.	Rank all genres in the order of their average rating. A movie may span multiple genres; such a movie should be counted in all the genres. Output format: Horror, 4.559 Comedy, 4.333 Action, 3.962 …	**1pt**
Q4.	What is the average number of genres per movie each year? Same output format as Q1. 1980, 3.515 1981, 2.763 1983, 3.137 …	**1pt**

Q5.	What are the top-3 combinations of genres that have the highest average ratings? Output format: Horror|Comendy, 3.676 Action|Adventure, 3.524 Romance|Comedy|Drama, 3.134	**1pt**
Q6.	Rank the users by number of ratings given. Each row should be a comma separated pair of user id and number of ratings. Show only top 20 users. Output format: 274373, 17 439367, 15 432232, 14 …	**1pt**
Q7.	For each year, what month has the most reviews? Output format: 1980, February 1981, January …	**1pt**
Q8.	What month of the year has the highest median review score? Output a single month: December	**1pt**
Q9.	For each genre, how many unique users have written bad reviews (score < 2.5)? Sort genres alphabetically. Output format: Action, 455 Adventure, 462 Comedy, 163 Drama, 314 Horror, 555 Sci-fi, 540 Western, 356 …	**1pt**
Q10.	What are the 3 most common words used inside tags for <genre> movies with average scores above 4.0? Output format for <genre> == "Horror": scary, terrifying, great	**1pt**

Store the answers to the questions in hdfs inside <output_folder>/Q<number>. For example if output folder is /output, then the answer to question 1 should be stored inside /output/Q1.

3 Dataset Description

For this assignment you will be using the MovieLens 20M Dataset (do not download from here).

The datasets describe ratings and free-text tagging activities from MovieLens, a movie recommendation service. It contains 20,000,263 ratings and 465,564 tag applications across 27,278 movies. These datasets were created by 138,493 users between January 09, 1995 and March 31, 2015. The dataset was generated on October 17, 2016.

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

The datasets are contained in six files. The files are stored in hdfs inside <input_folder>/

- tags.csv that contains tags applied to movies by users:
  - userId
  - movieId
  - tag
  - timestamp

- ratings.csv that contains ratings of movies by users:
  - userId
  - movieId
  - rating
  - timestamp

- movies.csv that contains movie information:
  - movieId
  - title
  - genres

- links.csv that contains identifiers that can be used to link to other sources:
  - movieId
  - imdbId
  - tmbdId

- genome-scores.csv that contains movie-tag relevance data:
  - movieId
  - tagId
  - relevance

- genome-tags.csv that contains tag descriptions:
  - tagId
  - tag

References:

1. https://www.kaggle.com/grouplens/movielens-20m-dataset (do not download from here)
2. F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages.

4 Provided Resources

You can download the dataset here (http://www.cs.colostate.edu/~csx55/ml-20m.zip) Note that this is what has been staged for you in the cluster. If you download the dataset from other sources it might be slightly different in format.

Gradle file:

```
plugins {
    id 'java'
}

repositories {
    mavenCentral()
}

dependencies {
    compileOnly 'org.apache.spark:spark-core_2.13:3.5.0'
    compileOnly 'org.apache.spark:spark-sql_2.13:3.5.0'
}

application {
    mainClass = 'Main'
}

jar {
    manifest {
        attributes 'Main-Class': 'Main'
    }
}
```

5 Deductions

There will be a **10-point deduction** if any of the restrictions below are violated.

1. You should not implement this assignment as a stand-alone program.
2. You should not implement this assignment using anything other than Spark. Implementing your own framework or using a 3rd party framework (that is not Spark) to implement this assignment is not allowed.

Version Change History

This section will reflect the change history for the assignment. It will list the version number, the date it was released, and the changes that were made to the preceding version. Changes to the first public release are made to clarify the assignment; the spirit or the crux of the assignment will not change.

Version	Date	Comments
1.0	10/22/2025	First public release of the assignment.
1.1	11/05/2025	Updated question 6 and dataset instructions. Added Gradle file.