

# Scalable Data Mining (Autumn 2021)

## Assignment 1 (Full Marks: 100)

### Steps for Spark installation:

1. Follow the guidelines given in this link to install Spark in your system:

<https://medium.com/@josemarcialportilla/installing-scala-and-spark-on-ubuntu-5665ee4b62b1>

**Instructions:** Please submit your answers (code+output+your way to approach the problem) to the following questions as a write-up in a PDF file via Moodle.

### Question 1 (Marks = 30)

In this assignment, you have to use Spark to have a look at the [Movie Lens dataset](#) containing user generated ratings for movies. The dataset comes in 3 files:

- ratings.dat contains the ratings in the following format: UserID::MovieID::Rating::Timestamp
- users.dat contains demographic information about the users:  
UserID::Gender::Age::Occupation::Zip-code.
- movies.dat contains meta information about the movies: MovieID::Title::Genres

Please read the readme file in the zip folder for further information.

=====

#### (10 points):

- a) Download the ratings file, parse it and load it in an RDD named ratings.
- b) How many lines does the ratings RDD contain?

#### (20 points):

- c) Count how many unique movies have been rated.
- d) Which user gave the most ratings? Return the userID and number of ratings.
- e) Which user gave the most '5' ratings? Return the userID and number of ratings.

## Question 2 (Marks = 40)

Using the same data file from Question 1, perform the following operations:

**(20 points):**

- a) Read the movies and users files into RDDs. How many records are there in each RDD?
- b) How many of the movies are a comedy?
- c) Which comedy has the most ratings? Return the title and the number of rankings. Answer this question by joining two datasets.

**(20 points):**

- e) Compute the number of unique users that rated the movies with movie\_IDs 2858, 356 and 2329 **without using an inverted index**. Measure the time (in seconds) it takes to make this computation.
- f) Create an inverted index on ratings, field movie\_ID. Print the first item.
- g) Compute the number of unique users that rated the movies with movie\_IDs 2858, 356 and 2329 **using the above calculated index**. Measure the time (in seconds) it takes to compute the same result using the index.

## Question 3 (Marks = 30)

Download the file from this link on google drive: [data2\\_1](#) . Write a function to load this data in an RDD and name it as 'Assignment\_1'. Make sure you use a case class to map the file fields.

Each line in this file contains the following fields: debug\_level: **String**, timestamp: **Date**, download\_id: **Integer**, retrieval\_stage: **String**, rest: **String**

Example: **DEBUG, 2017-03-24T12:06:23+00:00, ghtorrent-49 -- ghtorrent.rb: Repo Shikanime/print exists**

Here, debug\_level = DEBUG ; timestamp = 2017-03-24T12:06:23+00:00 ; download\_id = ghtorrent-49 ; retrieval\_stage = ghtorrent.rb ; rest = Repo Shikanime / print exists

- a. Create a function that given an RDD and a field (e.g. download\_id), it computes an inverted index on the RDD for efficiently searching the records of the RDD using values of the field as keys.
- b. Compute the number of different repositories accessed by the client 'ghtorrent-22' (without using the inverted index).
- c. Compute the number of different repositories accessed by the client 'ghtorrent-22' using the inverted index calculated above.

## Submission Instructions:

**You will submit 1 file** using the filename *RollNo\_AssignmentNo.pdf* with the following details:

- (1) description/logic of how you are going to use Spark to solve each problem using Scala,
- (2) the code snippets for each problem and
- (3) their respective outputs.