Lab Course: Distributed Data Analytics
Exercise Sheet 8

Mofassir ul Islam Arif
Information Systems and Machine Learning Lab
University of Hildesheim

Submission deadline: Monday July 01, 23:59PM (on LearnWeb, course code: 3116)

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a zip or a tar file containing two things a) python scripts and b) a pdf document.
2. In the pdf document you will explain your approach (i.e. how you solved a given problem), and present your results in the form of graphs and tables.
3. The submission should be made before the deadline, only through learnweb.

Distributed Computing with Apache Spark

Apache Spark is the third framework we are going to use in this lab course. In earlier labs we have worked with a message passing model (MPI) and computational graph model (TensorFlow). Apache Spark provide an abstraction called a resilient distributed dataset (RDD) (latest flavors provide Dataframes). It provides two sets of function to manipulate RDDs 1) Transformations and 2) Actions Apache Spark tutorial guide is available at `https://spark.apache.org/docs/latest/rdd-programming-guide.html`.

Exercise 1: Apache Spark Basics (10 points)

Part a) Basic Operations on Resilient Distributed Dataset (RDD) (4 points)

Let's have two lists of words as follows:

- `a = ["spark", "rdd", "python", "context", "create", "class"]`
- `b = ["operation", "apache", "scala", "lambda","parallel","partition"]`

Create two RDD objects of `a`, `b` and do the following tasks. Words should be remained in the results of join operations.

1. Perform `rightOuterJoin` and `fullOuterJoin` operations between `a` and `b`. Briefly explain your solution. (1 point)
2. Using `map` and `reduce` functions to count how many times the character `"s"` appears in all `a` and `b`. (1 point)
3. Using `aggregate` function to count how many times the character `"s"` appears in all `a` and `b`. (1 point)

Part b) Basic Operations on DataFrames (6 points)

Use dataset `students.json` (download from learnweb) for this exercise. First creating DataFrames from the dataset and do several tasks as follows:

1. Replace the `null` value(s) in column `points` by the mean of all points. (0.5 point)
2. Replace the `null` value(s) in column `dob` and column `last_name` by `"unknown"` and `"--"` respectively. (0.5 point)

3. In the `dob` column, there exist several formats of dates, e.g. `October 14, 1983` and `26 December 1989`. Let's convert all the dates into `DD-MM-YYYY` format where `DD`, `MM` and `YYYY` are two digits for day, two digits for months and four digits for year respectively. (2 points)
4. Insert a new column `age` and calculate the current age of all students. (1 point)
5. Let's consider granting some points for good performed students in the class. For each student, if his point is larger than 1 standard deviation of all points, then we update his current point to 20, which is the maximum. See Annex 1 for a tutorial on how to calculate standard deviation. (2 points)
6. Create a histogram on the new points created in the task 5. (1 point)

You should report a snapshot of the final dataset in your submitted .pdf file.

Exercise 2: Manipulating Recommender Dataset with Apache Spark (10 points)

For this exercise you will use movielens10m dataset available at `https://grouplens.org/datasets/movielens/10m/`. The movielens dataset is a rating prediction dataset with ratings given on a scale of 1 to 5. Specifically, you will be working with Tags Data File Structure `tags.dat`, which contains data in the form "UserID::MovieID::Tag::Timestamp". You have to solve following questions using Apache Spark transformations and actions.

1. A tagging session for a user can be defined as the duration in which he/she generated tagging activities. Typically, an inactive duration of 30 mins is considered as a termination of the tagging session. Your task is to separate out tagging sessions for each user.
2. Once you have all the tagging sessions for each user, calculate the frequency of tagging for each user session.
3. Find a mean and standard deviation of the tagging frequency of each user.
4. Find a mean and standard deviation of the tagging frequency for across users.
5. Provide the list of users with a mean tagging frequency within the two standard deviation from the mean frequency of all users.

Bonus (Optional Question): Analysis of Movie dataset using Apache Spark MapReduce (5 points)

Instructions for Bonus Question

You will only get points if you have a complete solution. There are no points for a partial solution.

Problem statement:

For this exercise you will use movielens10m dataset available at `http://files.grouplens.org/datasets/movielens/ml-10m-README.html`. The movielens10m dataset consists of 10 million rating entries by users. There are two main files required to solve this exercise 1) rating.dat and 2) movie.dat. If the dataset is large for your system, you can look at movielens1m. The `rating.dat` file contains userId, movieId and ratings (on scale of 1 to 5). The `movie.dat` file contains information about movies i.e. movieId, Title and Genres. You have to perform following tasks using map and reduce function of Apache Spark.

1. Find the movie title which has the maximum average ratings?
2. Find the user who has assign the lowest average ratings among all the users the number of ratings greater than 40?
3. Find the movie genre with the highest average ratings?

[Hint: You might need to merge/combine `movie.dat` and `rating.dat`.]

Related reading material

1. Parallel Speedup analysis https://portal.tacc.utexas.edu/c/document_library/get_file?uuid=e05d457a-0fbf-424b-87ce-c96fc0077099&groupId=13601

2. Spark launching configuration https://spark.apache.org/docs/latest/submitting-applications.html