INF553 Foundations and Applications of Data Mining

Summer 2019

Assignment 1

Deadline: May. 30th 11:59 PM PST

1. Overview of the Assignment

In assignment 1, you will complete two tasks. The goal of these tasks is to let you be familiar with Spark operation types (e.g., transformations and actions) and perform data exploration tasks on the Yelp dataset (https://www.yelp.com/dataset).

2. Requirements

2.1 Programming Requirements

a. You must use Python to implement all tasks. You can only use standard python libraries (i.e., external libraries like numpy or pandas are not allowed). There will be **10% bonus** for each task if you also submit a Scala implementation and both your Python and Scala implementations are correct.

b. You are required to only use Spark RDD in order to understand Spark operations more deeply. You will not get any point if you use Spark DataFrame or DataSet.

2.2 Programming Environment

Python 3.6, Scala 2.11 and Spark 2.3.3

We will use these library versions to compile and test your code. There will be a 20% penalty if we cannot run your code due to the library version inconsistency.

2.3 Write your own code

Do not share code with other students!!

For this assignment to be an effective learning experience, you must write your own code! We emphasize this point because you will be able to find Python implementations of some of the required functions on the web. Please do not look for or at any such code!

TAs will combine all the code we can find from the web (e.g., Github) as well as other students' code from this and other (previous) sections for plagiarism detection. We will report all detected plagiarism.

2.4 What you need to turn in

Your submission must be a zip file with name: **firstname_lastname_hw1.zip** (all lowercase). You need to pack the following files in the zip file (see Figure 1):

a. two Python scripts, named: (all lowercase)

firstname_lastname_task1.py, firstname_lastname_task2.py

b1. [OPTIONAL] two Scala scripts, named: (all lowercase)

firstname_lastname_task1.scala, firstname_lastname_task2.scala

b2. [OPTIONAL] one jar package, named: **firstname_lastname_hw1.jar** (all lowercase)

c. You don't need to include your results. We will grade on your code with our testing data (data will be in the same format).
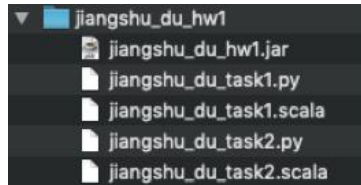

Figure 1: Submission Structure

3. Yelp Data

In this assignment, you will explore the Yelp dataset. You need to download the original JSON files HERE (https://www.yelp.com/dataset). You are going to use the entire review.json and business.json for assignment 1.

You can find the metadata of the datasets here (https://www.yelp.com/dataset/documentation/main).

4. Tasks

4.1 Task1: Data Exploration (7 points)

You will explore the dataset, review.json, containing review information for this task, and you need to write a program to automatically answer the following questions:

A. The number of reviews that people think are useful (The value of tag 'useful' > 0) (1 point)

B. The number of reviews that have 5.0 stars rating (1 point)

C. How many characters are there in the 'text' of the longest review (1 point)

D. The number of distinct users who wrote reviews (1 point)

E. The top 20 users who wrote the largest numbers of reviews and the number of reviews they wrote (1 point)

F. The number of distinct businesses that have been reviewed (1 point)

G. The top 20 businesses that had the largest numbers of reviews and the number of reviews they had (1 point)

Input format: (we will use the following command to execute your code)

```
Python: $ ./bin/spark-submit firstname_lastname_task1.py <input_file_name> <output_file_name>

Scala:  $ ./bin/spark-submit --class firstname_lastname_task1 firstname_lastname_hw1.jar <input_file_name> <output_file_name>
```

Param: input_file_name: the name of the input file (review), including file path

Param: output_file_name: the name of the output JSON file, including file path

Output format:

IMPORTANT: Please strictly follow the output format since your code will be graded automatically.

a. The output for Questions A/B/C/D/F will be a number. The output for Questions E/G will be a list, which is sorted by the number of reviews in the descending order. If two user_ids/business_ids have the same number of reviews, please sort the user_ids /business_ids in the alphabetical order.

b. You need to write the results in the JSON format file. You must use **exactly the same tags** for answering each question.

```json
{
  "n_review_useful": 11111,
  "n_review_5_star": 11111,
  "n_characters": 11111,
  "n_user": 11111,
  "top20_user": [["ABCDEFGHIJKLMNOPQ", 1111], ..., ["BCDEFGHIJKLMNOPQR", 1111]],
  "n_business": 11111,
  "top20_business": [["QPONMLKJIHGFEDCBA", 1111111], ..., ["RQPONMLKJIHGFEDCB", 11111]]
}
```

Figure 2: JSON output structure for task1

4.2 Task2: Exploration on Multiple Datasets (5.5 points)

In task2, you are asked to explore two datasets together containing review information (review.json) and business information (business.json) and write a program to answer the following questions:

A. What are the average stars for each state? (**DO NOT** use the stars information in the business file) (2.5 point)

B. You are required to use two ways to **print** top 5 states with highest stars. You need to compare the time difference between two methods and explain the result **within 1 or 2 sentences**. (3 point)

    Method1: Collect all the data, and then print the first 5 states

    Method2: Take the first 5 states, and then print all

Input format: (we will use the following command to execute your code)

```
Python: $ spark-submit firstname_lastname_task2.py <input_file_name1> <input_file_name2> <output_file_name1> <output_file_name2>
Scala: $ spark-submit --class firstname_lastname_task2 firstname_lastname_hw1.jar <input_file_name1> <input_file_name2> <output_file_name1> <output_file_name2>
```

Param: input_file_name1: the name of the input file (review), including file path

Param: input_file_name2: the name of the input file (business), including file path

Param: output_file_name1: the name of the output file for Question A, including file path

Param: output_file_name2: the name of the output JSON file for Question B, including file path

Output format:

a. You need to write the results for Question A as a file. The header (first line) of the file is "states, stars". The outputs should be sorted by the average stars in descending order. If two states have the same stars, please sort the states in the alphabetical order. (see Figure 3 left)

b. You also need to write the answer for Question B in a JSON file. You must use **exactly the same tags** for the task.


```
state,stars
abcd,5.0
bcde,5.0
cdef,5.0
defg,5.0
abce,4.9
```

```
{
    "m1": 0,
    "m2": 0,
    "explanation":""
}
```

Figure 3: Question A output file structure (left) and JSON output structure (right) for task2

5. Grading Criteria

(% penalty = % penalty of possible points you get)

1. You can use your free 5-day extension separately or together.

2. There will be 10% bonus if you use both Scala and Python.

3. If we cannot run your programs with the command we specified, there will be 80% penalty.

4. If your program cannot run with the required Scala/Python/Spark versions, there will be 20% penalty.

5. If our grading program cannot find a specified tag, there will be no point for this question.

6. If the outputs of your program are unsorted or partially sorted, there will be 50% penalty.

7. If the header of the output file is missing, there will be 10% penalty.

8. We can regrade on your assignments within seven days once the scores are released. No argue after one week. There will be 20% penalty if our grading is correct.

9. There will be 20% penalty for late submission within a week and no point after a week.

10. There will be no point if the total execution time exceeds 15 minutes.

11. Only when your results from Python are correct, the bonus of using Scala will be calculated. There is no partially point for Scala. See the example below:

Example situations

Task	Score for Python	Score for Scala (10% of previous column if correct)	Total
Task1	Correct: 7 points	Correct: 7 * 10%	7.7
Task1	Wrong: 0 point	Correct: 0 * 10%	0.0

Task1	Partially correct: 2 points	Correct: 2 * 10%	2.2
Task1	Partially correct: 2 points	Wrong: 0	2