Big Data Assignment

ISTD, SUTD

March 29, 2024

Group Assignment

This is a group assignment. You are supposed to work as a team based on your project group. Each group is only required to submit one copy of your solution through eDimension.

Deadline

14 April 2024 23:58.

Synopsis

In this assignment, you are supposed to develop a big data application to process user generated data capturing customer opinions towards brands and businesses.

Your tasks include loading the raw data into the Hadoop distributed file system (HDFS), performing data transformation and cleaning using Spark framework. Finally, you should be able to conduct some descriptive analytics on the cleaned data.

Common Requirements

For all the questions,

- 1. You may use RDD API and/or Dataframe API. You are not allowed to use Spark SQL API. e.g. spark.sql("SELECT") will not be rewarded with any mark.
- 2. The answer should not be dependent on the results from other questions, i.e. for each answer script, it should start from reading the given raw data file. However, you may develop your own library to be shared between answers for different questions.
- 3. There should be only *one* call to read() and *one* call to write() in each answer script.

4. Place the input data in the designated HDFS path and output data should be written in the designated HDFS path.

Submission

You are supposed to submit the following

```
1. q1.py
2. q2.py
3. a2.py
4. q4.py
5. q5.py
6. hw2.sh
```

Each py file should contain the solution to the correspondent question, i.e. q1.py for question 1, q2.py for question 2 so on and so forth. The script hw2.sh sets up the needed folders and data in HDFS and submits the python scripts to the spark cluster.

Templates of these six files are given to you. Besides editing the python template files, please read through the hw2.sh file and update the following sections

```
# change the following according to your student numbers
echo "1001234,1003456"

# change the following according to your environment
hdfs namenode="localhost"
```

You may assume that during the grading process, the data folder ./data, the .py files and hw2.sh file are placed under the same linux folder.

Part 1

Data

In part 1 we are looking at the restaurant review data extracted from

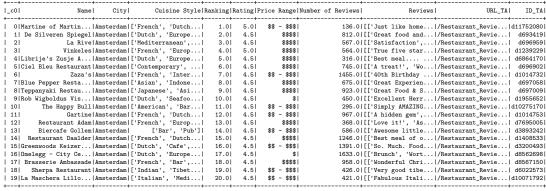
https://www.kaggle.com/damienbeneschi/krakow-ta-restaurans-data-raw

You **don't** need to and are **not recommended** to download the data from kaggle. Please make use of the data provided to you along with this assignment.

Copy the CSV file TA_restaurants_curated_cleaned.csv into HDFS path /assignment2/part1/input/. Load it into a Dataframe or RDD, we may observe that the data set has the following schema

```
hdfs_nn = "localhost"
df = spark.read.option("header",True)\
```

and a preview of the data records looks like the following:

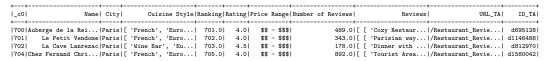


only showing top 20 rows

Question 1 (1 mark)

Develop a Spark application that cleans up the CSV file by removing rows with no reviews or rating < 1.0. Write the output as CSV into HDFS path /assignment2/output/question1/.

Sample output



Question 2 (1 mark)

Develop a Spark application that finds the best and the worst restaurants for each city for each price range (in terms of rating). Write the output as CSV

into HDFS path /assignment2/output/question2/. For simplicity, you can ignore rows with Price Range field as null.

You may use RDD API and/or Dataframe API. You are not allowed to use Spark SQL API.

Sample output:

+	+		+	+		+			+
_c0	Name	Cityl	Cuisine Style Rar	nking Rating F	rice Range Number	of Reviews	Reviews	URL_TA	ID_TA
3198	Pietersma Snacks	Amsterdam ['Dutch', 'Europ 32	209.0 5.0	\$I	null	[[],[]] /Restauran	t_Revie d10	0587448
2932	Grillroom Sabba	Amsterdam ['Middle Eastern'] 29	942.0 2.5	\$1	12.0 [['This is a gr /Restauran	nt_Revie de	6464568
1503	1 Chefalyon	Lyon ['Pub', 'Gastrop 14	485.0 5.0	\$\$\$\$	null	[[],[]] /Restauran	t_Revie d12	2408653
2605	Papagayol	Lyon	['Diner'] 26	606.0 2.0	\$\$\$\$	33.0	[[],[]] /Restauran	nt_Revie di	1329792
2951	le bountje	Brussels ['Belgian', 'Eur 29	952.0 5.0	\$\$ - \$\$\$	null	[[], []] /Restauran	t_Revie di	1563747
3009	Belga & Col	Brussels	['European']	null -1.0	\$\$ - \$\$\$	null	[[], []] /Restauran	t_Revie d13	3531979
2462	Sushi Express	Stockholm ['Japanese', 'Su	null 5.0	\$\$ - \$\$\$	2.01	[[], []] /Restauran	nt_Revie d13	3344590

Question 3 (1 mark)

Develop a Spark application that extracts the three cities with the highest and lowest average rating per restaurant. Combine them, sorted, such that the output looks like this:

For instance:

City AverageRating RatingGroup +				
Athens 4.241316931982634 Top London 4.178003263308178 Top Krakow 4.164012738853503 Top Geneva 3.97270245677889 Bottom Helsinki 3.9153318077803205 Bottom Brussels 3.900580875781948 Bottom		City	AverageRating	RatingGroup
Krakow 4.164012738853503 Top Geneva 3.97270245677889 Bottom Helsinki 3.9153318077803205 Bottom Brussels 3.900580875781948 Bottom			4.241316931982634	
Helsinki 3.9153318077803205 Bottom Brussels 3.900580875781948 Bottom	1			•
Brussels 3.900580875781948 Bottom	1			
	+-	+		+ Bottom

Write the output as CSV files into HDFS path /assignment2/output/question3/.

Question 4 (1 mark)

Develop a Spark application that counts the number of restaurants by city and cuisine style.

The output should something like the following:

_			
	City		'
١	Amsterdam	Vietnamese	24
Bratislava		Hungarian	3
-	Brussels	International	74
-	London	Kosher	26
1	Lvon	Mediterranean	1 801

| Lyon| German| 2|

Write the output as CSV files into HDFS path /assignment2/output/question4/.

Part 2

In this second part of the assignment, we investigate the movie credit data extracted from

https://www.kaggle.com/tmdb/tmdb-movie-metadata?select=tmdb_5000_credits.csv

You don't need to and are not recommended to download the data from kaggle. Please make use of the data provided to you along with this part.

For part 2, instead of CSV, we consider the input file in **Parquet** format. Parquet format is a compressed column based format which is optimized for parallel processing. For more details of parquet file format, refer to the following documentation:

https://spark.apache.org/docs/latest/sql-data-sources-parquet.html

Copy the tmdb_5000_credits.parquet file into HDFS path /part2/input/. Load it into a Dataframe or RDD, we may observe that the data set has the following schema

If we take a look at the first rows of the data, we see the following:

```
+----+
|movie id|
                    title|
                                      cast|
Avatar|[{"cast id": 242,...|[{"credit id": "5...|
    285|Pirates of the Ca...|[{"cast_id": 4, "...|[{"credit_id": "5...|
                  Spectre|[{"cast_id": 1, "...|[{"credit_id": "5...|
  206647
   49026|The Dark Knight R...|[{"cast_id": 2, "...|[{"credit_id": "5...|
               John Carter|[{"cast_id": 5, "...|[{"credit_id": "5...|
   495291
              Spider-Man 3|[{"cast_id": 30, ...|[{"credit_id": "5...|
    559|
                  Tangled|[{"cast_id": 34, ...|[{"credit_id": "5...|
   38757|
   99861|Avengers: Age of ...|[{"cast_id": 76, ...|[{"credit_id": "5...|
    767|Harry Potter and ...|[{"cast_id": 3, "...|[{"credit_id": "5...|
```

Question 5 (2 marks)

Develop a Spark application that finds the pairs of actors/actresses that are co-cast for **at least** 2 movies. The output should be in a (set of) **Parquet** files in the following schema:

movie_id, title, actor1, actor2

Note that the result should not contain any repetition, e.g.

49026, The Dark Knight Rises, Michael Caine, Christian Bale is considered as a duplicate entry of

49026, The Dark Knight Rises, Christian Bale, Michael Caine

The output should be something like the following:

+-		+		+
m	novie_id	title	actor1	•
1	•	One Man's Hero	James Gammon	Tom Berenger
1	9942	Major League	James Gammon	Tom Berenger
1	285 P	irates of the Ca	David Bailie	Ho-Kwan Tse
1	58 P	irates of the Ca	David Bailie	Ho-Kwan Tse
1	921	Cinderella Man	Michael Stevens Conrad	Bergschneider
1	14577	Dirty Work	Michael Stevens Conrad	Bergschneider
1	16290	Jackass 3D Dimi	try Elyashkevich	Manny Puig
1	12094	Jackass Number Two Dimi	try Elyashkevich	Manny Puig
1	9012	Jackass: The Movie Dimi	itry Elyashkevich	Manny Puig

Hint

You should be able to extract the needed info from the movie_id, title and cast columns.

Good luck!