CS 6350 - ASSIGNMENT 3

Please read the instructions below before starting the assignment.

- You are to do the questions in Spark using either Scala or PySpark. The idea is to show you how Spark makes computing easy and efficient.

- You should use a cover sheet, which can be downloaded at: http://www.utdallas.edu/~axn112530/cs6350/CS6350_CoverPage.docx

- You are allowed to work in pairs i.e. a group of two students is allowed. Please write the names of the group members on the cover page. Only one submission per team is required.

- UTD cluster can become slow or even unresponsive if too many people using Spark at the same time. It is your responsibility to start early and submit the assignment by the deadline. No extensions will be allowed.

- You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After that, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.

- Please ask all questions on Piazza, and not through email to the instructor or TA.

ASSIGNMENT 3

In this assignment, you will use the **Yelp Dataset** again to answer the questions below.
In this assignment, you will have to do the questions using Spark, either using Scala or PySpark
Spark can be used either in local mode or yarn mode, but it's advisable to run it in YARN mode:

To use local mode
    spark-shell --master local
To use yarn mode
    spark-shell --master yarn

NOTE: Spark is memory intensive. UTD cluster will get slow or even become unresponsive if many students log on at the same time. You are strongly advised to start your assignment early and not wait till the last moment. You can also use your personal virtual cluster, such as Docker or Hortonworks.
There will be no extensions granted for the assignments. Claiming that the cluster is slow or unresponsive or the program is not running is not an acceptable excuse.

The dataset files are located in hdfs in the following path,

/yelp/business/business.csv
/yelp/review/review.csv
/yelp/user/user.csv

If somehow the files disappear from the above HDFS location, you can also download them from:

http://www.utdallas.edu/~axn112530/cs6350/yelp/

Dataset Description.
The dataset comprises of **three** csv files, namely user.csv, business.csv and review.csv. Note that some of the content, such as id fields are encoded. Note that the files are separated by "^" character.

1. Business.csv file contain basic information about local businesses.
Business.csv file contains the following columns
"business_id","full_address","categories"

'business_id': (a unique identifier for the business)
'full_address': (localized address),
'categories': [(localized category names)]

2. Review.csv file contains the star rating given by a user to a business. Use user_id to associate this review with others by the same user. Use business_id to associate this review with others of the same business.

review.csv file contains the following columns
"review_id","user_id","business_id","stars"
'review_id': (a unique identifier for the review)
'user_id': (the identifier of the reviewed business),
'business_id': (the identifier of the authoring user),
'stars': (star rating, integer 1-5),the rating given by the user to a business

3. user.csv file contains aggregate information about a single user across all of Yelp
user.csv file contains the following columns "user_id","name","url"
user_id': (unique user identifier),
'name': (first name, last initial, like 'Matt J.'), this column has been made anonymous to preserve privacy
'url': url of the user on yelp

Q1. List the business_id , full address and categories of the Top 10 highest rated businesses using the average ratings.

This will require you to use **review.csv** and **business.csv** and join them on the common key (business_id)

Please use reduce side join to answer this problem.

Sample output:

business id	full address	categories	avg rating
xdf12344444444,	CA 91711	List['Local Services', 'Carpet Cleaning']	5.0

Q2. Read a user name from the command line and find the average of their review rating.

For example, if the command line argument is "Matt J", you need to output the average review ratings of that user.

Q3. **List the 'user id' and 'stars' of users that reviewed businesses located in Stanford.**

You would need to filter the business.csv files by addresses that contain the word "Stanford". There is no need for any aggregation operation.

Q4. List the user_id , and name of the top 10 users who have written the most reviews.

Q5. List the business_id, and count of each business's ratings for the businesses that are located in the state of TX