MIS 284N - Big Data and Distributed Programming
Project 1 - Movie Recommender using Map-Reduce

Instructor: *Ramesh Yerraballi*
TA: *Aastha Tripathi*
Semester: *Fall 2018*
Due Date: 11:59pm, Monday 9/17

This project is based on Map-Reduce Framework. In these you will get to work with Spark and will get to know how does spark work, what functionalities does spark provide, what does map-reduce framework do and why is it useful.

In this project you will be implementing a basic movie recommender system. You will be given a dataset where there are multiple csv files. These csv files have data corresponding to movie ratings.

For example, `movies.csv` has

```
[movieId (some unique number), movie title, genre]
```

and there are other files which provide the data for what user gave what ratings to which movies and so. You need to download the corrsponding data from - http://files.grouplens.org/datasets/movielens/ml-latest-small.zip

You will be given a user's **`userId`** and a **`movieId`** the user liked. Based on the given `movieId` and `userId` you need to recommend top 5 movies for the given user (**userId**).

For example, you can find out the people/other users who have also liked the given movie (**movieId**). Once these users are known you can get the other movies they have rated and then aggregate the rating of the other movies given by these users and give the top 5 rated movies.

To be more creative and for better recommendations, you might want to use other data given in the csv files like genre, tags and so. (How? - You have to figure this out!)

Notes:

1. User liked a movie can be quantified as user giving a rating of more than **`avgRating`** to the given movie.
2. To get you started on how to use the MapReduce Framework and its different functions, find a helper notebook here.
3. Assume in your code you have a constant input named:

```
givenMovieId = '31';
givenUserId = '1';
avgRating = 3.0;
```

The given numbers like 31, 1 are just examples and will vary in testing. **`givenMovieId`** corresponding to the Id the user liked, **`givenUserId`** is the userId and **`avgRating`** is the rating above which we will consider the user liked a movie.

Return **LIST** of top 5 movies you are recommending (**Hint**: see **top** or **takeOrdered**)
For example - [('movie title1', 'aggregate rating'), ('movie title2', 'aggregate rating')....]

What to turn in:
A zip folder which will have:

1. Jupyter Notebook
2. A brief report on what features you used for recommendation. And a brief explanation of flow of your code. For example, what RDD does what or, why it was created. You can merge the report into the Notebook as we did with the exercise we did in class.
3. datasets folder which will have all the csv files you are using in your notebook.
4. Notebook should use relative path to the csv files in datasets folder.
5. Name of the zip folder - <your_name>_<your_partner_name>.zip