SI 601 Winter 2016 Homework 3 (100 points)

Due at 5:30pm on Wednesday, Jan. 27, 2016

In this homework you will do web scraping, parse the retrieved HTML pages and make some REST Web API calls. Sounds interesting? Read on.

IMDB is a well-known movie ratings website. Another popular movie info web site is http://www.themoviedb.org/. The question is: are the ratings from IMDB and themoviedb.org correlated? We'll find out in this homework.

You'll be using these modules for this homework: `urllib2, BeautifulSoup, json, and time.` Instructions for installing BeautifulSoup are in Lecture 3 slides.

Step 1. (15 points)

Fetch the top 200 Sci-Fi movies based on user ratings from IMDB starting from this URL:

http://www.imdb.com/search/title?at=0&genres=sci_fi&sort=user_rating&start=1&title_type=feature

The above URL contains the top 50 Sci-Fi movies. More movies are available by following the "Next" link. Note how the "start=1" part of the URL changes when you follow the "Next" links. Save each fetched page in a HTML file. The saved HTML files should look similar to the provided step1_top_scifi_movies_*_desired_output.html

The top Sci-Fi movies list changes over time. Therefore, it is possible that the pages you saved are somewhat different than the provided examples.

Also note that a few movies have titles or actors with, e.g. accented characters. **Make sure you use the `utf8` encoding to write out the HTML to use Unicode and preserve any non-English characters.**

Step 2. (40 points)

Parse the HTML pages you saved in Step 1 with BeautifulSoup, extract movie information for top 200 Sci-Fi movies and save the result in a tab-delimited file named step2_top_200_scifi_movies.tsv. The TSV file should have 5 columns and 200 data rows.

The 5 columns are: Rank, IMDB ID, Title, Year, Rating

The IMDB_ID is the part that sits between last two slashes in the movie URL in the table. For example, in

<a href="/title/tt2395427/" title="Avengers: Age of Ultron (2015)">

the IMDB ID is tt2395427. Note that some movies do not have user ratings yet, and in this case, the rating should be "NA" in the output file.

Your Step 2 output file should look like `step2_top_200_scifi_movies_desired_output`.tsv

Step 3. (20 points)

Use the Web service http://www.themoviedb.org/documentation/api to get themoviedb.org rating for each of the top 200 movies using the IMDB ID you collected in Step 2. To use this API, you will need to register for a free account and get your API Key, which is a long string. For example, if your API key is sfhdskfhdskfhsakfd, this URL fetches a JSON response for the movie "Star Wars: Episode VI - Return of the Jedi", which has IMDB ID "tt0086190":

```
http://api.themoviedb.org/3/find/ tt0086190?api_key=sfhdskfhdskfhsakfd&external_source=imdb_id
```

You should get this JSON response:


```
{"movie_results":[{"adult":false,"backdrop_path":"/bvJOpyHYWACDusvQvXxKEHFNjce.
jpg","genre_ids":[28,12,878],"id":1892,"original_language":"en","original_title
":"Star Wars: Episode VI - Return of the Jedi","overview":"As Rebel leaders map
their strategy for an all-out attack on the Emperor's newer, bigger Death Star.
Han Solo remains frozen in the cavernous desert fortress of Jabba the Hutt, the
most loathsome outlaw in the universe, who is also keeping Princess Leia as a
slave girl. Now a master of the Force, Luke Skywalker rescues his friends, but
he cannot become a true Jedi Knight until he wages his own crucial battle
against Darth Vader, who has sworn to win Luke over to the dark side of the
Force.","release_date":"1983-05-
25","poster_path":"/jx5p0aHlbPXqe3AH9G15NvmWaqQ.jpg","popularity":4.448141,"tit
le":"Star Wars: Episode VI - Return of the
Jedi","video":false,"vote_average":7.7,"vote_count":2428}],"person_results":[],
"tv_results":[],"tv_episode_results":[],"tv_season_results":[]}
```


The "vote_average":7.7 value is the themoviedb.org rating we would like to compare to IMDB ratings.

IMPORTANT! You MUST pause at least 5 seconds between EVERY HTTP request to themoviedb.org. If you don't do this, and send requests continuously in a loop with no delay, the server will reject your request after a few requests.

Use the `sleep(x)` function in the `time` module after each HTTP request to pause *x* seconds.

Save your results in a text file named step3.txt that contains the IMDB ID and the JSON string (separated by a tab) for each movie on each line. The file should look like step3_desired_output.txt

Step 4. (20 points)

After you verify that your step 3 output is correct, **you can comment out your HTTP request code for step 3 to avoid running that time-consuming step from now on**. Now open the file you saved in step 3, load the JSON string on each line into a variable, extract just the 'vote_average' numbers, and then join it with the IMDB data based on the IMDB IDs.

If a movie does not have an IMDB rating or themoviedb rating (rating of 0 means no rating), discard it. Save your results in a CSV file named step4.csv, which should have the following columns:

IMDB ID, Title, Year, IMDB Rating, themoviedb Rating

The file should look like `step4_desired_output.csv`

Step 5. (5 points)

Load your `step4.csv` into Excel or Google Doc and make a scatter plot that looks like step5_example_plot.pdf.

What to submit:

A zip file named si601_w16_hw3_***youruniquename***.zip that contains:

1. Your python source code file

2. All of your output files for step 1 through step 5