CS 839 - Data Science
Project Stage 2

Karthik Chandrashekar
karthik.chandrashekar@wisc.edu

Kartik Sreenivasan
ksreenivasa2@wisc.edu

Niveditha Hariharan
nhariharan@wisc.edu

Due Date: April 12, 2019 at 11:59 PM

1 Data Sources

We extracted movie data from two websites, namely `http://www.imdb.com` and `http://www.rottentomatoes.com`. From both these sites, we scraped lists of movies and extracted several attributes including movie titles, the release data, the director and so on. Note that both these websites maintain massive databases of movies. In order to increase the probability of having matches we extracted movies in more or less increasing order of their release date.

2 Method of Extraction

We essentially followed the mechanism of template based extraction. After manually viewing the source (inspect element from browser) of both webpages, we looked for patterns that described delimiters specific to each webpage in order to separate out the different attributes. Note that these are typically *div* and *class* names. Once we had this, we used the python tool *Beautiful Soup* to extract the *Html* source from the page and construct a list of entities. We then write this into a csv. Note that the entire process can be automated since the URL of each subsequent page follows a pattern. For example, in IMDB, the URL of page $page_number$ looks like

`https://www.imdb.com/list/ls057823854/?st_dt=&mode=detail&page=page_number&ref_=ttls_vm_dtl&sort=list_order,asc`

Therefore, we just ask the python script to subsequently increase the value of *page_number* until we get as many tuples as we want. More specifics can be gleaned by viewing the code as it is not very complicated. It mostly consists of regex searches which are aided by the *find* method in *Beautiful Soup*.

3 Description of Entity - Movie

There isn't much to explain here since nearly everybody is aware of what a movie is. These are typically movies released in the US so they are mostly Hollywood movies. The search filter input into IMDB and Rotten Tomatoes are slightly different in that for IMDB we simply asked for list of movies in increasing order of release date. Whereas for implementational ease, we requested top movies of each year in Rotten Tomatoes. The data turned out to be more structured this way. We extracted the same attributes in both cases, namely:

1. Movie Title
2. Year the movie was released
3. Director of said movie
4. Total runtime
5. Genre
6. Rating
   (a) IMDB: Rating on a scale of 0-10
   (b) Rotten Tomatoes: Tomatometer is a percentage on a scale of 0-100
7. Description
8. Certificate (PG13, R, etc)

Number of movies in Table A (IMDB): 7355

Number of movies in Table B (Rotten Tomatoes): 3066

Therefore, both tables should look more or less identical. Of course, there will be differences based on how the data was input and the *ratings*. Also, the entities will not be identical since there will be some movies in one that we did not extract from the other.

P.S: If you wish to get a feel for the data, you can simply look at the notebook we've created for that purpose from: `https://github.com/karthik-c/CS839_Project_WebPage/blob/master/stage2/CODE/explore_data.ipynb`

4 Open Source Tools

4.1 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python is widely used in the industry and for good reason. We used it here as our primary scripting language that enabled us to do all this.

4.2 Beautiful Soup

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

We used *Beautiful Soup* in conjunction with Python, using the HTML parser to do both extract the source of the webpage and then navigate the HTML code to extract our entities.

4.3 Pandas

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

We used *pandas* only to read the final csvs as tables to verify our work in the jupyter notebook mentioned above.

4.4 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

We used *Git* and specifically *Github* to both collaborate as well as host our code and project.

4.5 LaTeX

LaTeXis a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeXis the de facto standard for the communication and publication of scientific documents. LaTeXis available as free software.

We used LaTeXto create this report.