DATA SCIENCE REPORT

STAGE 2

EXTRACTING MOVIES FROM IMDB AND FILMCRAVE WEB SOURCES

Team Members

1. Saranya Baskaran

2. Shivanee Nagarajan

3. Varun Ramesh

For project stage 2, we have extracted movies from two different web sources using rule-based wrapper construction methods.

1. Web Data Sources

We have chosen IMDB and FilmCrave websites for movie reviews. Both of these websites are very popular movie review sites with hundreds and thousands of movies. Also, we made sure to collect movies with similar filters applied in both the web sources so that there is a considerable amount of overlap which is essential for the upcoming stages of the project.

IMDB:

IMDB is one of the most popular and reliable movie review sites. It has a good collection of data to extract.

Sample URL:
http://www.imdb.com/search/title?year=2017,2017&title_type=feature&sort=moviemeter,asc&page=1

We started with movies released in 2017 and went till 2005 in backwards by picking the top 250 movies in each year (sorted by popularity). Each page has a list of movies (~50 per page) and hence we kept going till first 5 pages in each year to pick ~250 movies per year.

Filmcrave:

Filmcrave is an online movie social network that allows users to write movie reviews, share movie lists, watch trailers and interact with other members

Sample URL:
http://www.filmcrave.com/list_top_movie.php?yr=2017

Similarly, the start URL was modified to change the year and approximately 250 movie entities where extracted for each year. We made sure to pick similar data from both the sources.

2. Extraction of Structured Data

From both the web sources structured data is extracted using Python Scrapy tool and is stored as tables in a csv file

IMDB Extraction:

IMDB spider does the actual crawling and extraction of data form IMDB URLs. All the URLs that were to be crawled were constructed and appended to the list of start URL's.

The next step was to analyze the HTML DOM Tree structure followed by IMDB pages and to verify whether all the pages had similar html structure. For each of the attribute to be extracted, a Xpath selector or a CSS selector was identified and rules were return accordingly in the spider. Attributes such as director, actors were multivalued and not within a single tag in the web page. Hence some special processing was done to identify the list of directors and actors from the web page

Filmcrave Extraction:

Similar to IMDB, for filmcrave also the HTML DOM structure was analyzed to figure out the exact Xpath selectors and CSS selectors to be used for each of the attribute. Then rules were written accordingly in the spider. A couple of attributes were not within a well-formed html structure. Hence special handling was done for them in the spider code.

We defined a pipeline in the scrapy project settings file which would help in inserting the data into a csv file. The list of movies scraped out of a page will then be passed onto the data insertion method in the pipeline. Here the logic of merging the schema into a common one and then inserting data as a table into a csv file is available.

3. Entity & Table Details

The entity that we extracted are movies. We collected data such as actors, directors, movie rating etc from both the movie review sites as described above. We have taken all information provided by each site and then merged the results to form a common schema which basically is the intersection of all attributes given by both the data sources.

Table 1: IMDB Web source

- **Number of Tuples**: 3750
- **Schema:**
    - Title – String (Movie Name)
    - Year – Integer (Just the year of release)
    - Mpaa Rating – String (this is a combination of string and numbers)
    - Overall Rating – Float (Total rating of the movie. It is measured out of 5)
    - Genre – String (It is a list of genres that the movie falls in separated by /)
    - Directors – String (List of directors who directed the movie. It can be 1 or more, A comma separated list)
    - Stars – String It includes all actors who acted in the movie. A comma separated list)
    - Plot – String (A short text description of the story)
    - Duration – Float (Length of the movie in minutes)

Table 2: FilmCrave Web source

- **Number of Tuples**: 3220
- **Schema:**
    - Title – String (Movie Name)
    - Year – Integer (Just the year of release)
    - Mpaa Rating – String (this is a combination of string and numbers)
    - Overall Rating – Float (Total rating of the movie. It is measure out of 5)
    - Genre – String (It is a list of genres that the movie falls in separated by/)
    - Directors – String (List of directors who directed the movie. It can be 1 or more. A comma separated list)
    - Stars – String It includes all actors who acted in the movie. A comma separated list)
    - Plot – String (A short text description of the story)

We cleaned and then merged the schema of both the sources to form the following common schema.

Column	Description	Type
Title	Title of the movie	String
Overall Rating	Overall rating of the movie	Float
Year	Year of release	Integer
Genre	Genres of the film	String
MPAA_Rating	MPAA rating obtained by the movie	String
Directors	Directors of the movie	String
Actors	Stars of the movie	String
Plot	Description of the movie plot	String

4. Open Source Tools

The open source tool we used is Scrapy.

It is one of the most widely used Python based web crawling framework to extract structured data from web pages. It helps in extracting data using CSS classes or Xpath selectors. Also, in Scrapy, pipelines can be used to define a process to extract and store the data in a structure/format as needed.

Scrapy's main components are spiders which does the actual crawling. For each web source, based on that web page layout a set of instructions can be written in the spider and it will asynchronously crawl all pages based on the start URLs given and gives back the data which can later be stored in well-structured format. It is easy to learn and an efficient tool for web crawling. Hence it makes it easier to build large scale projects and maintain them using scrapy.