CS 839: Data Science

Project Stage 2 - Report

Team Members

- Arun Jose (jose4@wisc.edu)
- Kundan Kumar (kkumar36@wisc.edu)
- KN Sushma (kudlurnirvan@wisc.edu)

Introduction

The project involves selecting two web data sources from which structured data can be extracted by using the Rule-based Wrapper Construction method. We have chosen to extract movies from our two web sources, based on a reasonable amount of data consistency between matching entities.

Web Sources Type

The entity we selected for this project is movies from where we will be extracting information. The two sources we have identified are the two most popular movie review websites:

1. IMDB

IMDB is an online database of information related to films, television programs, home videos and video games, and internet streams, including cast, production crew and personnel biographies, plot summaries, trivia, and fan reviews and ratings.

2. Rotten Tomatoes

Rotten Tomatoes is an American review-aggregation website for film and television.

These two sources have overlapping real-world attributes which will be used in further project stages for entity matching.

Data Extraction Techniques

We first eyeballed the attributes present in both the web pages and we finalized the attributes of 'Movie Name', 'Year of release', 'Genre', 'Rating', 'Directed By' and 'Runtime of movie' to be extracted. In order to ensure that we have sufficient matching attributes in the extracted data from both the websites, we applied some filters like top rated movies and sorting by Tomatometer.

<u>Extracting movies' information from RottenTomatoes:</u>

1. Extracted around 5000 movie links after applying the above mentioned filters to the webpage. This step had a challenge as the response from the filters were not paginated, So we exploited the AJAX call that the webpage makes for reloading every 30 set of movies. We used that AJAX call URL (which has Page Number parameter mentioned in the request) to get movies list. We made exactly 170 AJAX call to get hyperlinks for 5000 movies. We dumped these hyperlinks into a text file.
   URL used to extract hyperlinks: RottenTomatoes
2. Once we obtained the hyperlinks for the selected movies, we looped through each one of them and parsed the HTML response using BeautifulSoup python package. Some of the extractions involved finding the attribute by TAG name, while some others were selected by parsing the javascript content of the webpage. We were able to extract the below attributes from RottenTomatoes website and we dumped this information into a CSV file.

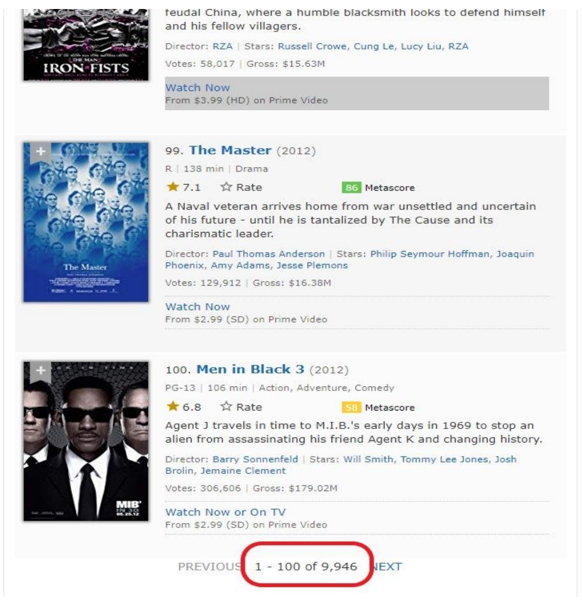
Attributes	**Description**
Name	Name of the movie
Year	The year when the movie was released (Ex: 2005)
Rating	Rating of the movie (like, R: Restricted, G: General Audiences, PG: Parental Guidance Suggested)
Genre	The category of the movie (like: Drama, Comedy, Action, Horror, Romance etc)
Directed By	Name(s) of director(s) who directed the movie
Runtime	The duration of the movie (120 minutes)

We extracted 5394 tuples from the RottenTomatoes website.

<u>Extracting movies information from IMDB:</u>

1. The first step was to extract the list of all the required movie links to a file, which could be used from then on as our source to crawl data. The idea was to parse the HTML of this search space and filter out only the hyperlinks corresponding to the movies. For this, we used BeautifulSoup and parse the HTML of a search page for the 100 movie names

present in it. The URL was modified via a loop to navigate between the different search pages. There were a 100 movies per page, so we queried the same over 52 pages to get around 5000 movie links with a buffer of 200 more. This was saved in our text file labelled “imdb_movie_hyperlinks.txt”
URL used to extract hyperlinks: IMDB

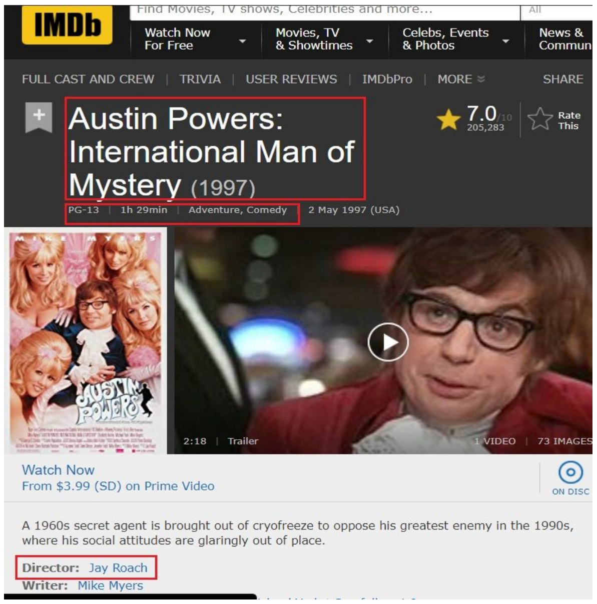


feudal China, where a humble blacksmith looks to defend himself and his fellow villagers.
Director: RZA | Stars: Russell Crowe, Cung Le, Lucy Liu, RZA
Votes: 58,017 | Gross: $15.63M
Watch Now
From $3.99 (HD) on Prime Video

99. The Master (2012)
R | 138 min | Drama
★ 7.1 ☆ Rate 86 Metascore
A Naval veteran arrives home from war unsettled and uncertain of his future - until he is tantalized by The Cause and its charismatic leader.
Director: Paul Thomas Anderson | Stars: Philip Seymour Hoffman, Joaquin Phoenix, Amy Adams, Jesse Plemons
Votes: 129,912 | Gross: $16.38M
Watch Now
From $2.99 (SD) on Prime Video

100. Men in Black 3 (2012)
PG-13 | 106 min | Action, Adventure, Comedy
★ 6.8 ☆ Rate 58 Metascore
Agent J travels in time to M.I.B.'s early days in 1969 to stop an alien from assassinating his friend Agent K and changing history.
Director: Barry Sonnenfeld | Stars: Will Smith, Tommy Lee Jones, Josh Brolin, Jemaine Clement
Votes: 306,606 | Gross: $179.02M
Watch Now or On TV
From $2.99 (SD) on Prime Video

PREVIOUS 1 - 100 of 9,946 NEXT



Austin Powers: International Man of Mystery (1997)
★ 7.0/10 205,283
PG-13 | 1h 29min | Adventure, Comedy | 2 May 1997 (USA)
Watch Now
From $3.99 (SD) on Prime Video
A 1960s secret agent is brought out of cryofreeze to oppose his greatest enemy in the 1990s, where his social attitudes are glaringly out of place.
Director: Jay Roach
Writer: Mike Myers

2. Now that we have the list of hyperlinks to the movies, the next step was to run a loop through these links and extract the required features from them. Again, we have implemented HTML parsing using BeautifulSoup by looking at the relevant tags and classes. These features were formulated into a data tuple and appended to a csv file labelled “imdb_data.csv”.

Attributes	**Description**
Name	Name of the movie
Year	The year when the movie was released (Ex: 2005)
Rating	Rating of the movie (like, R: Restricted, G: General Audiences, PG: Parental Guidance Suggested)
Genre	The category of the movie (like: Drama, Comedy, Action, Horror, Romance etc)
Directed By	Name(s) of director(s) who directed the movie
Runtime	The duration of the movie (2hr 10 min)

We extracted 4885 tuples from the IMDB website.

Open-Source Tools

We chose a Python open source library called ‘BeautifulSoup‘ for the extraction of entities from HTML web pages. We used BeautifulSoup to parse the HTML content of the page as well as to read the Javascript content. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.

References

https://www.imdb.com/
https://www.rottentomatoes.com/
https://pypi.org/project/beautifulsoup4/
https://en.wikipedia.org/wiki/Beautiful_Soup_(HTML_parser)