TVDB – Television Database

Assignment Report

# Abstract

The theme of this assignment is popular TV shows. For the purpose of this assignment we have limited the number of TV shows in our model to 50. The data collected for this assignment comes from a myriad of sources such as web scraping of imdb.com using Python, crawling Twitter with the use of an API implemented with Python, and a dataset consisting of the ratings of TV shows on Netflix. The goal of this assignment is to create a conceptual model that can integrate all these three sources.

# Thematic Relationship

The TV Show entity is related to the Netflix entity through the title i.e. the name of the TV show. It is a one-one relationship.

The Episode entity is related to the TV Show entity via a many-one relationship. One TV show contains one or more episodes. One episode can belong only to one TV Show.

The tweet entity is related to the TV Show via a many-to-many relationship as there can be many tweets about a TV Show and one tweet can talk about multiple TV shows.

# ER Diagram

Netflix

title

rating

ratingLevel

ratingDescription

release year

user rating score

user rating size

Tweets

Favorite Count

Tweet

Date

Episode

Title

Duration

Rating

Director

Season

TV SHOW

Title

Rating

Release Year

Genre

Runtime

Votes

Description

No. of Seasons

No. of Episodes

# Data Sources

We have chosen our primary source of data to be imdb.com from which we have scraped TV Show and Episode data. We have chosen Twitter to get a public perception of the TV Show by using metrics such as Favorite Count, Follower count, etc.. To get a perception of ratings outside of IMDB we are also using a CSV file from Kaggle on Netflix rating of TV Shows. The actor information is obtained from a TSV file provided by IMDB.

## Web Scraping

The kind of data we incorporate into this model using web scraping is information on TV shows such as rating, genre, votes, etc..

The web scraping module of this assignment is implemented using BeautifulSoup and Requests packages to scrape the data. We hit a specific web page <https://www.imdb.com/search/title?title_type=tv_series> and made sure that the HTTP response that we receive is 200 to signify a valid response.

We then created a container to hold the attributes of the TV show that we are interested in scraping

This process is repeated to get the season information. Since we do not know beforehand the number of seasons in each TV show, we look at the maximum value under the class “seasons-and-year-nav” and iterate accordingly by changing the URL to get the list of episodes under each season.

The data in the container is then transferred to a pandas data frame to be cleaned and manipulated. The data was cleaned by converting the runtime to an integer column e.g. if the runtime of an episode is represented as “50 min”, we removed the “ min” using regular expressions and converted the remaining numbers represented in character form to integer.

## API

Twitter API is used to make API calls on the hashtags that are generated by scraping IMDB (as explained above) for the top 50 TV shows and creating hashtags based on the name of the show and searching for those hashtags on twitter.

The list of hashtags is stored in a data frame in Python and we used a for loop to go through the list of hashtags and search Twitter for these hashtags.

## CSV

Our third source of data comes from csv file on Kaggle on Netflix ratings of TV shows.

After importing the CSV file into a pandas dataframe we checked for nulls and duplicates. The nulls in the numeric columns were replaces with the mean value for that column. The duplicate rows are deleted.

# Audit completeness

The data that we have collected for this assignment may not be the same scale as real world data but it is similar in quality to the real world data. Especially in the case of tweets where we use same set of tweets that anybody with access to the free Twitter API would. We scrape the same imdb.com website that anybody else would. Perhaps a real-world application would not just crawl one website but several.

# Audit consistency/uniformity

We have strived to keep our data consistent by using pandas functions to remove nulls and replace them wherever possible with appropriate values, removed duplicate records and created derived columns in data frames so that it is easier to analyze the dataset.

To keep the scope of this assignment limited, we have chosen to extract only a few attributes of a tweet using the Twitter API namely: Tweet, Created Date, and Favorite Count.

# Conclusion and Inference

Therefore my using multiple sources of data such as imdb.com, Twitter API and a Netflix TV show rating dataset, we have created a conceptual schema of our data on TV Shows.

# License

The text in the document by Shriram Karthikeyan and Darshan Durve is licensed under CC BY 3.0 https://creativecommons.org/licenses/by/3.0/us/

The code in the document by Shriram Karthikeyan and Darshan Durve is licensed under the MIT License https://opensource.org/licenses/MIT

# Citations

https://www.kaggle.com/abhikaggle8/netflix-rating-distributions

https://stackoverflow.com/questions/18689823/pandas-dataframe-replace-nan-values-with-average-of-columns

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html

https://github.com/nikbearbrown/INFO\_6210