Analysis of Key Factors Influencing TV Shows' Success

Aakarsha LNU
alnu2@binghamton.edu
SUNY Binghamton
Binghamton, New York,
USA

Riya Yeshwant Thakur
rthakur1@binghamton.edu
SUNY Binghamton
Binghamton, New York,
USA

Aayushi Ahlawat
aahlawa1@binghamton.edu
SUNY Binghamton
Binghamton, New York,
USA

Ishan Bagchi
ibagchi1@binghamton.edu
SUNY Binghamton
Binghamton, New York,
USA

Tarun Tiwari
ttiwari1@binghamton.edu
SUNY Binghamton
Binghamton, New York,
USA

ABSTRACT

These days, television and movies are the two main forms of entertainment. Everybody enjoys watching and reading fascinating stories. When we watch television, we observe that many people discuss their experiences. On social media, anyone can publicly voice their opinions on any subject. First off, the big social networking sites like Twitter, Instagram, TMDB, and Reddit create a ton of information about user behaviors, news, rising trends, etc. There is a huge amount of user-generated content available, and this al- ters the environment significantly. The genre, the cast, the budget, the location of production, the number of seasons, the number of episodes, the ratings, and the area all play a role in determining whether a television program is successful or not. Thus, by examining these elements for a range of programs, we aim to predict which TV show has a higher chance of becoming successful.

KEYWORDS

Tv shows, Genre, Cast, Budget, Reviews, Region, Twitter, Reddit, TMDB

1 INTRODUCTION

We all watch a lot of television. Who does not enjoy having some spare time on the weekend? With the increasing popularity of OTT platforms and the media business, individuals have more options for different TV series and movies, which has increased competition in the media sector. People in the entertainment industry are always seeking for new strategies to make their series or movie a success. Thus, data analysis is becoming increasingly important in managerial decision-making in media enterprises throughout the world. The dependence of subscription video on demand providers, such as Netflix, on data analytics to support choices regarding new content investment is widely documented.

Tmdb is a website that provides millions of movies, TV shows for people to explore. It provides us with the ability to search for the various genre movies, TV shows, the most popular movies or TV shows currently streaming on TV, theaters or available for rent. The platform also provides users to post reviews about the movies/TV shows and provide ratings. The website provides recommendations based on the movies searched or watched by the user. We will make use of the TMDB API which is available on the TMDB API Developers page. The aim of this project is to determine the success of the TV show based on major factors such as budget,cast etc.

Twitter is an open social network that people use to converse with each other in short messages, known as tweets. People publish their comments regarding actors, production companies, and TV shows, which is useful for our study of the most popular and trending shows and their dependence on various factors.

Reddit is a social news aggregation, content review, and debate platform. The website's material, which includes links, text entries, photographs, and videos, is contributed by registered users and is then rated by other users. Posts are arranged into user-made boards called "communities" or "subreddits" according to their subjects. When there are enough upvotes, posts that have received the most upvotes will eventually appear on the first page of thewebsite.

2 DATA COLLECTION

This project will be collecting data from 3 data sources, namely, TMDB, Twitter, and Reddit. Below 3 sources specify the details of data collections from 3 sources:

2.1 TMDB

Our data will be gathered from The Movie Database (TMDB), a community-built TV and movie database founded in 2008. Since 2008, the number of submissions to this database has expanded year after year. With over 400,000 developers and businesses using this platform, TMDB has established itself as a top source ofmetadata.

- GET /tv/tv_id: Would fetch show's budget,cast,region,etc which would help determine their dependency on the show's rank.

- GET /tv/tv_id/changes: Tv shows details by id also provides next and previous episode details.

  GET /tv/tv_id/content_ratings: Individual rating depending on people's liking.

- GET /tv/tv_id/reviews: Individual reviews that would affect the rating of a show.

- GET /discover/tv: Discover Tv shows by different types of data like average rating, number of votes, genres, etc.

- GET /tv/top: API to get most popular TV show list.

- GET /tv/tv_id/watch/providers: Get list of watch providers for a tv show.

2.2 TWITTER

Using Python code, the stream of tweets is collected. Real-time tweets will be gathered using the Twitter Streaming API, resource URL- https://api.twitter.com/2/tweets/sample/ stream. A number of parameters that can be helpful for our project's subsequent research will be added to the end of the URL. We will be using the twitter fields available which are: Id(Unique identifier of the Tweet), Tweet(The content of the Tweet), CreatedAt(Creation time of the Tweet),Public Metrics (Engagement metrics for the Tweet at the time of the request such as count of retweets, replies and likes), Entities(Contains details about text that has a special meaning in a Tweet such as mentions, hashtags and other details), Annotations(Contains context annotations such as domain(domain name for the Tweet), Language(Language of the Tweet, if detected by Twitter).

The implementation of the collection program used Python's request library for sending the stream request. We have used the Bearer token as a credential. This credential is generated once we have created a developer account on Twitter. The response request is sent using the GET method and we collect the JSON response. We will be collecting every field required for each line in response. Context annotations are delivered as a context annotations field in the payload. These annotations are inferred based on semantic analysis (keywords, hashtags, handles, etc) of the Tweet text and result in domain and/or entity labels. Context annotations can yield one or many domains. We will use the Domain "TV Show" provided by the Twitter API.

2.3 REDDIT

The task of collecting subreddits will be accomplished by constructing a Python script, but before that, we will be obtaining a temporary OAuth token from Reddit. Then, we must create an API to read the data and stream new postings. Various sub-reddits will be supplied as parameters in order to read or stream the most recent post, generating real-time data. Additionally, a complete dataframe that includes the post's title, date and time, content, and user reactions can be created. A few other parameters, like the following, could be added to further optimize the data:

Limit: in which maximum items that needs to be returned will be given before/after: specify that we only want to search for the posts that are before or after another.

3 NAPKIN MATH

Currently we have data for 4728 TV shows in TMDB, and for approximately 1000 different TV shows' data changes every day, which includes changes in their reviews, TV show details, number of episodes, etc.

4 METHODOLOGY

Following are the steps for data extraction methodology that we have implemented in the following three datasets:

4.1 TMDB

- Firstly, we have used a GET discover api to get the results of all the TV shows sorted in descending order with respect to their "first air date". We have extracted the TV show details based on each page returned, and stored it in a collection named "tmdb".
- With the help of the tmdb database created above, we get a list of released TV shows and for each of them we extract the list of reviews generated, and add the tv id associated with it to our "reviews" collection.
- We get the list of all TV show ids using the tmdb database collection. Using which we update our tmdb database, and add TV id for each TV shows and update its data.
- We create a third collection named "TV changes" which stores all the changes occured to the TV shows within the last 24 hours using the change api. If the results list is greater than 0 we check to see if the changes associated with a TV id exists in our tmdb database. Further, we focus on change keys namely, languages, origin country, overview, production companies, status, type, created at, episode runtime, genres, names, production countries, tagline to get these changes. We save the respective changes to change collection with the TV id.
- If the changes associated to a TV id does not exist in the tmdb database, we add that TV show id data to our tmdb collection as a new TV id details. We map the TV id and update our primary tmdb collection.

4.2 Twitter

- Firstly, we created an account in the Twitter Developer Portal and hosted an application to get all the following keys required for authorization: API Key, Access Token, Secret Key.
- Secondly, we created a MongoDB account for saving the database. We created a cluster with a database and a collection for storing the twitter sample stream data. A connection string is generated using username and password for connecting it to our web app.
- We are sending an http request to sample stream api to extract the 1 percent volume sample data. Further, the URL endpoint is modified to include the following fields: context annotation, author id, public matrix, language, promoted matrix and an expansion for including users.fields: username, name, location of the included users in the tweet posted.
- Once the API is called, we store the response and check if the tweet extracted contains the field context annotation and if the language is English. If the following conditions are true, then only the particular tweet are fetched. Further, we are filtering out those tweets which consists of the domain as "TV Shows" in the annotation and storing them to our collection "twitter_stream".
- Apart from the collected tweets, Tweet fetched time is also getting stored simultaneously in the field "tweet fetched time" i.e, the current time when the tweet was fetched. This key-value pair is used to keep a track of the count of tweets

extracted in an hour. This count is extracted using a query sent to our "twitter_stream" collection.

- We have created another collection for storing the count of tweets fetched hourly basis which will consist of the keys: count, source (which indicates whether it is a twitter, reddit, or tmdb data), start_time, and end_time.

4.3 Reddit

- A temporary OAuth is requested from reddit with the help of username and password which are stored in data variable.
- Header information is stored which provides a brief description of our application.
- Further, a request is sent to grant access for an OAuth token and the resultant token is converted to JSON followed by pulling the access token value.
- Authorization is added to header's dictionary.
- Next process is to make request to retrieve data from the list of subreddits which include television, tv details, and also some individual tv shows. Using the list, we retrieve subreddit details and store the response.
- To get the desired data, we applied a filter so that each post can be accessed by looping through each item in res.json() ['data']['children'].
- After retrieving a large amount of dynamic data from the selected subreddit like "/r/television", it is stored in the respective database collection which can be further analysed.

5 CONCLUSION

The objective of this project is to collect data from 3 data sources-TMDB, Twitter, and Reddit using the rest API respectively. Data collection is the first and foremost step in building any social media data science pipeline as further analysis is done on the extracted data. We have applied and used various fields to fetch the desired data of our interest and saved them in 3 different clusters for each of them specifically in mongodb database. Our future scope is to analyze the collected data based on further extraction of primary fields from the raw data, and determine the essential factors involved in contributing towards the success rate of any TV show002E