An Exploratory Analysis of Televsion Show Data from IMDB and The Movie Database.

Carroll – x17501726
BSc. Computing
National College of Ireland
Dublin, Ireland.

***Abstract*—Since the new millennium, television shows have experienced a huge rise in both popularity and success. As a result of this surge in volume and popularity of television shows, a vast amount of data on television shows has been recorded and published online on databases such as Internet Movie Database (IMDb) and The Movie Database (TMDb. This report details that there is a strong correlation between each databases average rating for a television show, that there is a correlation between the popularity levels of television shows and ratings but not a strong one and that, on average, most genres have a similar average rating, with little variance between them.**

Keywords—IMDb, Internet Movie Database, TMDb, The Movie Database, television, television show, KDD, Data visualization, R.

I. Introduction

Since the new millennium, television shows have experienced a huge rise in both popularity and success. Pichard [1] debated that starting at the dawn of the millennium, American television began enjoying a Second Golden age as a result of an improvement in aesthetics and storytelling, a significant decrease in the difference of production quality of cable and network television series, and the massive success of series during this period. Pichard also states that this renaissance was sparked by cable channels, such as HBO during the 1990's. The new Golden Age of television centred on tragic dramas such as The Sopranos [2], The Wire [2], Avatar: The Last Airbender [3], Buffy the Vampire Slayer [2], Breaking Bad [2], and Game of Thrones [4].

As a result of this surge in volume and popularity of television shows, a vast amount of data on television shows has been recorded and published online. Internet Movie Database (IMDb) is one of the most widely used and respected resource for movie and television content [1]. It is made up of both user and critic reviews alike, ratings and other details about movies and television shows. It contains a large volume of data on television shows from their general information to information on each season and each individual episode.

The Movie Database (TMDb), however, is entirely made up of information including ratings, reviews, popularity, descriptions, and summaries added by users of the website [2]. TMDb represents data that is generated and maintained by its users. As a result, TMDb's data represents the sentiments of the public for a given television show.

This report seeks to compare IMDb and The Movie Database television show datasets to find any correlation between each datasets ratings of television shows, the correlation between a high level of popularity in television shows and ratings, and if certain genres, on average, get higher ratings than others.

II. Literature Review

A. Halloween Episodes Get Higher IMDb Ratings

Pavlik [7] found, using IMDb data, that for each season of a television show Halloween episodes received higher ratings on average than the average rating of regular episodes. This investigated was carried out on twenty-four thousand episodes from one thousand seasons of one hundred and eighty-four television shows.

Pavlik gathered the data by getting a list of Halloween episodes from Wikipedia [8]. Then, using an R script, Pavlik, searched for the series name on IMDb [5] and collect the associated IMDb ID number. The ID number was then used to iterate over the pages of the shows episodes to get the episodes ID numbers. The episodes were then iterated over to gather details rating, genre, and release date.

Pavlik then conducted a paired t-test comparing the Halloween episode rating and average rating of regular episodes from that season. A paired t-test was selected for this to check if the paired observations came from the same distribution or not. The result of this investigation revealed that Halloween episodes receive an increase of 0.9 in average rating. Pavlik notes that although this is a small increase, it is a significant one.

Pavlik's investigation has provided results that are interesting and hold potential value to businesses. However, the findings rely on assumptions. Pavlik assumes in this investigation that the dependent variables such as season ratings, regular episode ratings and the Halloween episode rating from IMDb are normally distributed. Pavlik later states that the data did not pass a normality test. This assumption is a detriment to the data and its findings. Similarly, Pavlik notes that this trend is influenced by the variation in each season's ratings. As a result, the findings can be skewed to be particularly favourable to the findings of the investigation or to its disadvantage.

However, Pavlik introduced t-tests to account for variability between the Halloween episode rating and that of the regular episodes' mean rating. This introduces an increased element of reliability for the findings of the investigation as it helps account for differences in ratings of episodes between seasons. As result, the effects of the potentially skewed ratings discussed previously are lessened.

B. Recommendation System Using NLP Tools

Kapoor et al. [9] used data from The Movie Database (TMDb) [6] to create a movie review android application. The main aim of the app is to perform a sentiment analysis on each review left by a user and based on if the sentiment is positive or negative, the movies review will be updated accordingly. The application works by sending users reviews to the apps server and using a support vector machine (SVM) model, process the review and classify it as positive or negative.

Kapoor et al. used the TMDb API [13] to gather movie data on a daily basis. The API possesses many end points; however, Kapoor et al. only utilised the movie endpoint for the purposes of this research. The data was refreshed this often in order to reflect changes for movie sections in their app such as most popular, or top-rated. During the android application development section of this research, Kapoor et al. gathered movie data from TMDb with attributes such as name, genre, rating, reviews, actors, and earnings.

However, for the purposes of the research the most important data that was gathered was the name, review, and genre data as these are the key elements that are provided for the sentiment analysis. Kapoor et al. are conducting very interesting research with the aid of TMDb data, particularly by using the review data from the general public for use in a sentiment analysis. This mitigates the use of non-organic reviews, such as those created by the research team, and helps to create a stronger algorithm for the sentiment analysis as it can adapt and learn from naturally added reviews from TMDb. It is interesting to note that Kapoor et al. did not utilise the TMDb API POST request functionality for their users ratings or reviews, or for the ratings that resulted from the sentiment analysed reviews to ascertain what impact this would have on the TMDb movie sections such as most popular or top rated.

III. Data

The IMDb and TMDb datasets are suitable for the topic of this report as they both contain extensive information including title, average ratings, votes, popularity, and runtimes about television shows across multiple genres and decades of television. These elements supply the foundation for the objectives set out for this report and are essential to the findings of this report.

Similarly, the datasets both complement each other as they each possess similar attributes such as title, average rating, and genre for each television show. However, each possess unique attributes such as popularity, title type, and start and end year, that when combined, provide an insightful and strong dataset. For example, findings related to the correlation between a television shows ratings and popularity could not be found without these complimentary datasets.

However, other datasets were considered for use within this report. Data from the Trakt.tv [10] Application Programming Interface (API) [11] was considered for use. Trakt.tv is a website that allows users to track the movies and television shows they watch by integrating with their media center or computer. They have published their API online, which allows for the retrieval of various data about television shows such as seasons, episodes, genres, comments about the show and the cast. However, from investigating the IMDb, TMDb and Trakt.tv datasets, it was clear that the IMDb were more complimentary and were more aligned with the objectives of this report. A summary of all the datasets used in this report is shown in "Table I".

TABLE I

Dataset:	IMDb Basics (imdb-basics-title-type-genre.csv)	IMDb Ratings (imdb-tv-show-avg-rating-and-votes.csv)	IMDb Basics and Ratings Merged (imdb-dataset-merged-uncleaned.csv)	TMDb Data (tmdb.csv)
Size:	85.5 MB	17.8 MB	76.7 MB	650 KB
Data Structure:	Structured	Structured	Structured	Structured
Number of Files:	1	1	1	1
File Format:	CSV	CSV	CSV	CSV
Instances:	1,048,575	1,048,575	1,048,575	18,985
Number of Attributes:	9	3	11	5
No. Numeric Attributes:	3	2	5	3
No. Categorical Attributes:	3	0	3	1
No. Text Attributes:	3	1	3	1
Data Types:	Characters (chr) and Integers (int)	Characters (chr), numbers (num), and Integers (int)	Characters (chr), numbers (num), and Integers (int)	Characters (chr), numbers (num), and Integers (int)

The IMDb dataset was acquired from their website [5] by downloading the IMDb Basics [12] (name.basics.tsv.gz), which is comprised of information such as title type, title, star and end years, runtime, and genre, on all IMDb titles. IMDb Basics [12] (imdb-basics-title-type-genre.csv), as seen with sample data in "Table II", is comprised of information for every IMDb title. In order to have one IMDb dataset to work on, both of these datasets were merged by title into one file.

TABLE II

IMDb Basics (imdb-basics-title-type-genre.csv) – Sample Data								
tconst	***titleType***	***primaryTitle***	***originalTitle***	***isAdult***	***startYear***	***endYear***	***runtime Minutes***	***genres***
tt0141842	tvSeries	The Sopranos	The Sopranos	0	1999	2007	55	Crime, Drama

IMDb Ratings [12] (title.ratings.tsv.gz), as shown with sample data in "Table III", is comprised of information on average rating and number of votes for every IMDb title.

TABLE III

IMDb Ratings (imdb-tv-show-avg-rating-and-votes.csv) – Sample Data		
tconst	***averageRating***	***numVotes***
tt0141842	9.2	291726

In order to have one IMDb dataset to work on, both of these data sets were merged (imdb-dataset-merged-uncleaned.csv) by ID (tconst) into one file, as shown with sample data in "Table IV".

TABLE IV

IMDb Basics and Ratings Merged (imdb-basics-title-type-genre.csv) – Sample Data									
titleType	***primaryTitle***	***originalTitle***	***isAdult***	***startYear***	***endYear***	***runtimeMinutes***	***genres***	***averageRating***	***numVotes***
tvSeries	The Sopranos	The Sopranos	0	1999	2007	55	Crime, Drama	9.2	291726

The TMDb [10] dataset was retrieved from their API [11]. This was achieved by going to the tv/popular [13] endpoint and retrieving the data from there. This required using the correct http verb, GET, going to the correct endpoint (tv/popular), acquiring, and utilising an API key in the request and supplying the parameters necessary, page and language. However, the data from tv/popular is paginated so it required writing a for loop to go over every page of the tv/popular endpoint, gathering the data in order to combine it and save it to one file (tmdb.csv), as shown along with sample data in "Table V".

TABLE V

TMDb Data (tmdb.csv) – Sample Data				
name	***popularity***	***tmdbNoOfVotes***	***tmdbAverageRating***	genre_***ids***
The Sopranos	65	1199	8.4	Drama

IV. METHODOLOGY

For this report, the Knowledge Discovery in Databases (KDD) was chosen. This was the most appropriate option considering datasets and objectives of the report.

A. Selection

The selection of datasets for this report involved choosing the IMDb and TMDb datasets as they both provide consistent, reliable sources of data on television shows. Similarly, these datasets are complimentary and can achieve the objectives of the report. The IMDb Basics (imdb-basics-title-type-genre.csv) and IMDb Ratings (imdb-tv-show-avg-rating-and-votes.csv) datasets both contained data that would be necessary to achieve the goals of the report. In order to work with these datasets and for ease of use, they were converted from TSV filetypes to CSV prior to the selection process.

Similarly, for the TMDb data (tmdb.csv), the TMDb API "tv/popular" endpoint was selected to gather data on multiple television shows. A large amount of data was extracted as "JSON" [14], and then parsed to in order to make it easier to work with. The attributes "name", "popularity", "voteCount", "voteAverage", and "genreIds" were selected as necessary for the report.

B. Pre-processing

In order to clean the IMDb dataset, it was first necessary to combine the IMDb Basics and Ratings file in order to see what cleaning needed to be done on the data after it had been merged. They were merged by ID (tconst). There were columns present in the IMDb Basics and Ratings datasets that were not needed for the purposes of the analysis. Columns including "originalTitle", "isAdult", and "titleType" were removed from the dataset as result.

It was then necessary to deal with missing values. For columns such as "avgRating" and "numVotes", where the data type was number (num), there were a lot of missing values. In order to gain a consistent dataset imputation was carried out. The mean of "avgRating" was calculated and injected into the entries on the "avgRating" column where NA values were present. The same process was carried out for the "numVotes" NA values. The dataset attributes "primaryTitle", "genres", "averageRating" and numVotes" were changed to "name", imdbGenres", "imdbAverageRating" and "imdbNoOfVotes" respectively to prepare for transformation. The dataset then began to take shape and any rows in the dataset that were not of "titleType" that did not contain "tvSeries" were removed as they were not relevant to the analysis objectives.

In order to clean the TMDb dataset, a number of processes were carried out. Firstly, the attributes "name", "popularity", "voteCount", "voteAverage", and "genreIds" were selected to be a part of the saved dataset as they are complimentary to the IMDb dataset and the report objectives. Then any columns that contained entries with NULL/NA values were removed from the dataset. This was the most appropriate option to do as it was causing issues with saving the datafile due to the code not knowing what to do with the NA values. When the data was saved successfully, it contained duplicate entries with the same data. An example of this was multiple rows containing the same data on the television show "The Mandalorian". These duplicate rows were removed as they were redundant data.

Similarly, when the data was gathered from the API, the genres for each were assigned as ID numbers for each genre, rather than the genre name, hence the column name "genreIds". To deal with this, the data was checked to see how many unique genre ID's were present. The genre name associated with each "genreIds" was then manually gathered" and replaced each genre ID with the associated genre name throughout the dataset. Data of each unique entry present in the "genreIds" column was then viewed again to make sure this had worked correctly. These steps were done as the names of each provide more consistent and valuable data to the report. The names of other attributes such as "vote_average", "vote_count", "genreIds", and "ï..name" were also changed to "tmdbAverageRating", "tmdbNoOfVotes", "tmdbGenres" and "name" respectively for standardisation and to prepare for transformation.

TMDb calculates and defines the figures for its tv/popular "popularity" attribute by the number of votes, and views for the current day, the number of users who marked the show as a favourite and/or added it to their watchlist that day, the total number of votes and by the previous days score. As result of this and from the standpoint that this would have no detrimental effect on the figures contained within the "popularity" attribute, they were then rounded to the nearest whole number.

C. Transformation

For the transformation of the datasets, the IMDb Basics and Ratings datasets were merged together by ID (tconst). Once these datasets were merged, the attribute "primary title was changed to "name" and likewise with the TMDb dataset, "ï..name" was changed to "name". The merged IMDb dataset and the TMDb dataset were then merged together by name attribute.

This, however, led to a lot of rows with missing values as some television shows that were present in each respective dataset were not present in the other and vice versa. In order to deal with this, the rows that contained missing value were found and counted to assess the situation. The attributes "tmdbNoOfVotes", "tmdbAverageRating", and "popularity" were found to have a significant amount of missing values. The mean value was found for "tmdbNoOfVotes", "tmdbAverageRating", and "popularity" and injected into each column, respectively.

The "imdbGenres" and "genre_ids" attributes were still separated so they were then merged into one "genres" attribute column separating the values by a comma. The "imdbGenres" and "genre_ids" attribute columns were then removed from the dataset. However, as a result there were duplicate entries in each rows "genres" column as a result of both the IMDb and TMDb datasets both containing the same genre categorical information on a television show. An example of this is a row containing "Comedy, Action, Comedy" in the genre attribute was transformed to "Comedy, Action". These duplicate values were removed. The final transformed dataset was then saved.

D. Interpretation/Evaluation

The first objective of this report seeks to find out if there is a correlation between IMDb and TMDb ratings for television shows. In order to interpret the data and complete this objective, a data frame was created that took a subset of the merged IMDb and TMDb attributes, namely "imdbAverageRating" and "tmdbAverageRating". A bar chart of "imdbAverageRating" against each incremental rating was then plotted. Similarly, this was applied to the "tmdbAverageRating" data too. This was done in order to get an idea of what levels of ratings each attribute possessed at an incremental level. Finally, a scatter graph was then plotted in order to see the relationship between the attributes in a clearer way.

Similarly, the second objective of this report is to find the correlation between a high level of popularity in television shows and ratings. In order to fulfil this objective, a data frame that took a subset of the final IMDb and TMDb dataset was created. This data frame contained the "imdbAverageRating", "tmdbAverageRating" and "popularity" attributes. A plot of "imdbAverageRating" against "popularity" was then created. Similarly, this was done for "tmdbAverageRating" against popularity". These plots were created in order to get an idea of each datasets ratings versus popularity.

The melt function was then applied to the data and a scatter plot was created with popularity against the IMDb and TMDb attributes individually. This was done in order to interpret the data and to differentiate between the IMDb and TMDb average ratings versus the popularity of television shows.

The final objective of the report is to find out if certain genres, on average, get higher ratings than others. This objective was achieved by firstly created a corpus of the merged IMDb and TMDb datasets genre attribute. To aid this, all the words in the corpus were converted to lowercase to avoid any issues where the same word was differentiated from itself due to a capital letter. Then any punctuation and any excess whitespace were removed.

A term document matrix was then created from the corpus to store the frequency of each genre type. The mean average rating for the top four most common and top 4 least common genres was then found. The top four most and least common genres was used as there was twenty-six genres present in the dataset and for visualisation purposes displaying that number of genres was not feasible. The top four most and least common genres paint the picture of the correlation between genre ratings sufficiently in order to meet the objective.

Each row that contained each of the most and least common genres was found. For each of these, the mean of the average IMDb and TMDb rating was found and then stored in a data frame. These average ratings for the most common genre types, comedy, drama, family, and animation, and the least common genre types short, biography, war, and horror were plotted. This shows the correlation between genres and average ratings.

V. Implementation

The R application code workflow starts with setting working directory for the script to run in. This must be done in RStudio and works by setting the working directory to the folder that the script and other files are in. This allows the script to then get and work with the initial datasets it needs, IMDb Basics and Ratings as they are all in the same folder. The script then installs the packages it needs to run the script successfully: "dplyr", "httr", "jsonlite", "rlist", "data.table", "rtidy", "ggplot2", "ggthemes", "reshape", "tm", and "wordcloud" if they are not already installed. The script then loads in the IMDb datasets that are in CSV format in as data frames. The script then runs the pre-processing and transformation processes on these files and saves it as a CSV file.

The script then uses the "httr" R packages to get the TMDb dataset from their API. This is done by using the HTTP GET verb along with the URL for the tv/popular endpoint, an API key, and the necessary parameters to retrieve the data. The data at this endpoint is paginated so it was necessary to write a for loop in the R script to go through each page and save the data. Rows containing null values are removed from the data frame before it is saved as a CSV file. The script then runs the pre-processing and transformation processes on this file and saves it.

The last saved CSV of the IMDb and TMDb datasets are then loaded in before transformation takes place and they are merged. After they are merged, pre-processing and transformation processes are carried out on the data to deal

with missing values and small issues that have arisen as a result of the merge. The final dataset is saved as a CSV file.

The final dataset is then used to create visualisations of the data, this is done to target the objectives of the report. The script makes use of the dataset and the "ggplot2" library to create graphs that accomplish the objectives of the report. The resulting graphs are then saved.

The workflow of this application is automatic. The script must be opened in RStudio in order for the code for setting the working directory to work correctly. Once this is ensured, the rest of the code will run correctly. The projects workflow can be seen in "Fig. 1".

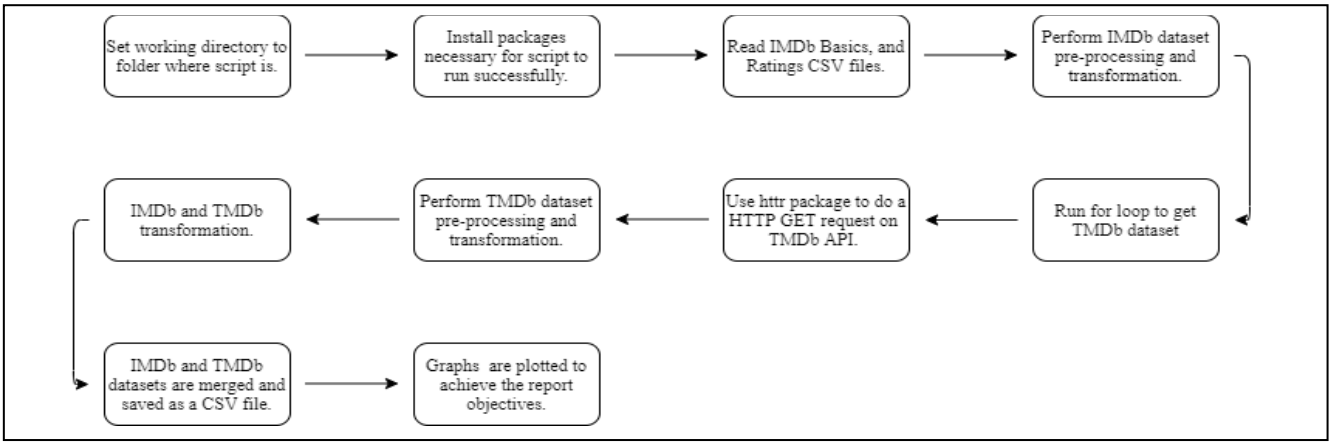

Fig. 1. Project workflow diagram.

R is the programming language that was used for this projects code. It was used throughout the project to carry out the processes in the KDD methodology. Various R packages were used in the script too such as "dplyr", "httr", "jsonlite", "rlist", "data.table", "rtidy", "ggplot2", "ggthemes", "reshape", "tm", and "wordcloud". These packages were essential to both getting, formatting the datasets for the project correctly and creating the visualisations. The API used in the project was TMDb's API. The endpoint accessed was tv/popular. The "httr" package was used to access this API's endpoint and successfully retrieve the data from it using the HTTP GET verb, a URL, an API key, and parameters. The "jsonlite" package was used to parse and format the data the httr package retrieved. The other packages aided carrying out pre-processing, transformation, and visualisation processes on the datasets and to create visualisations. Reading and writing to CSV files was also used a lot throughout the script. This was done so as to have backups and data "checkpoints" throughout the project.

The main technical challenge faced throughout this project was learning how to use R. This project was my first foray into using R and it takes time to learn any language. I had never used a language to work with data in this way, so I was slower than usual to learn. R also contains a lot of functionality; this was overwhelming and tough to know where to begin. However, once I started, I slowly overcame this issue and gained confidence on how to approach coding with R.

The pre-processing and transformation that I carried out with R also took some time to get to grips with. It took time to learn how to write the code for what exactly I wanted to do in these KDD sections but once I figured out each step, it became easier to advance in.

Another issue was learning how to interact and get data from API's. A lot of API documentation is vaguer than I would like so it was a struggle at times to learn how to interact and utilise API endpoints and functionality. This came mostly in how to use R packages to get API data, what parameters I needed to specify and where in the request to place the API key.

VI. Results and Findings

A. *To investigate the correlation between IMDb ratings and TMDb ratings of television shows*

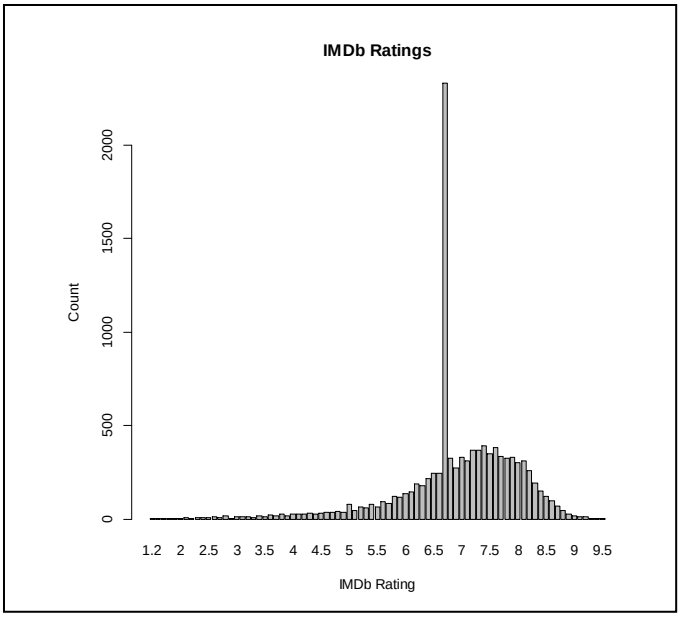

Fig. 2. IMDb Ratings.

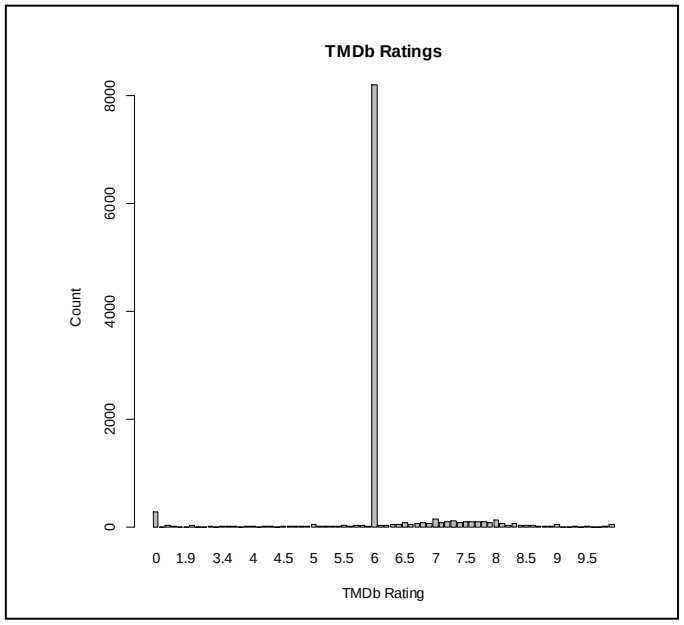

Fig. 3. TMDb Ratings.

From reading Fig 2, it can be observed that there are more higher ratings present in the IMDb Ratings data. This means that there are more television shows that received a rating greater than 5.0 than ones that received a rating less than 5.0. As can be clearly seen from the graph, a large number of shows received a rating of 6.9. This is a result of injecting the mean of the "imdbAverageRating" into rows with NA values as part of the pre-processing section of KDD. This particularly high increment of ratings thus is an outlier. The majority of IMDb Ratings are in the 6.5 to 8 .5 range.

From reading Fig 3, it can be observed that there are more higher ratings present in the data. However, there are a significant amount present in the 0 to 2 range. These are most likely from television shows with no rating present. There are more television shows that received a rating higher than 6.5 than ones that received a rating lower than 6.5. As can be clearly seen from the graph, a large number of shows received a rating of 6.0. This is a result of injecting the mean of the "tmdbAverageRating" into rows with NA values as part of the pre-processing section of KDD. This particularly high increment of ratings thus is an outlier. The majority of IMDb Ratings are in the 6.5 to 9 range. The conclusion that can be drawn from reading these graphs is that the majority of shows present in the dataset received a strong rating of greater than 6.0.

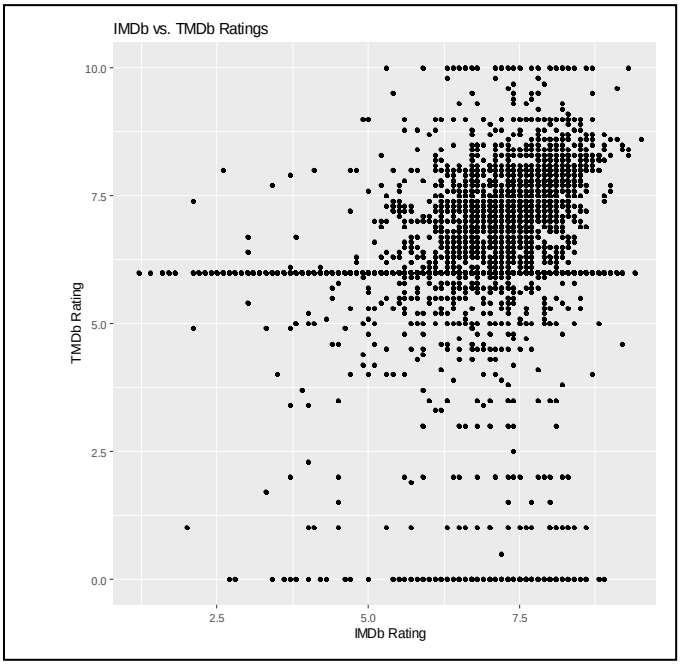

Fig. 4. IMDb vs TMDb Ratings.

Fig. 4 shows the correlation between IMDb and TMDb ratings. As a result of these findings, it can be said that apart from the outlier mean values, there is a strong correlation between IMDb and TMDb ratings for television shows in the range 6.0 to 8.5. I did not find this result to be all that surprising, it makes sense that rating across the different platforms are similar. However, I did expect there to be more of a difference as IMDb ratings include critic reviews, which can be harsh, and TMDb ratings are entirely made up of user ratings.

B. To investigate the correlation between the popularity of television shows and ratings

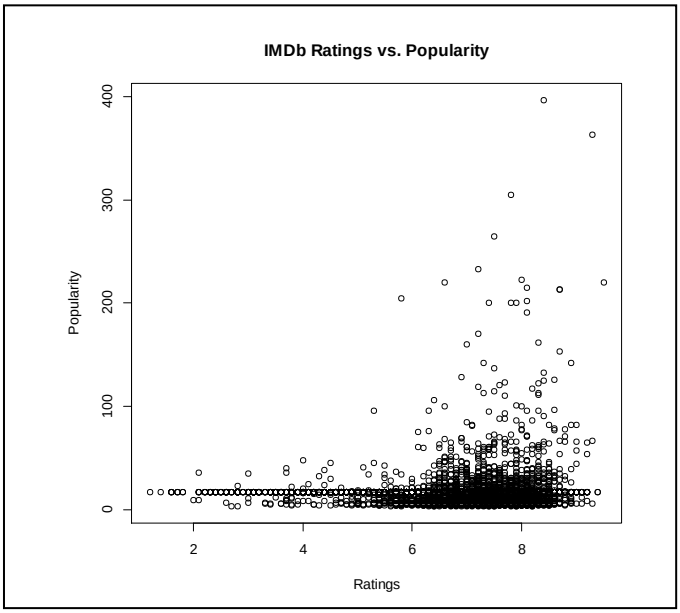

Fig. 5. IMDb Rating vs. Popularity.

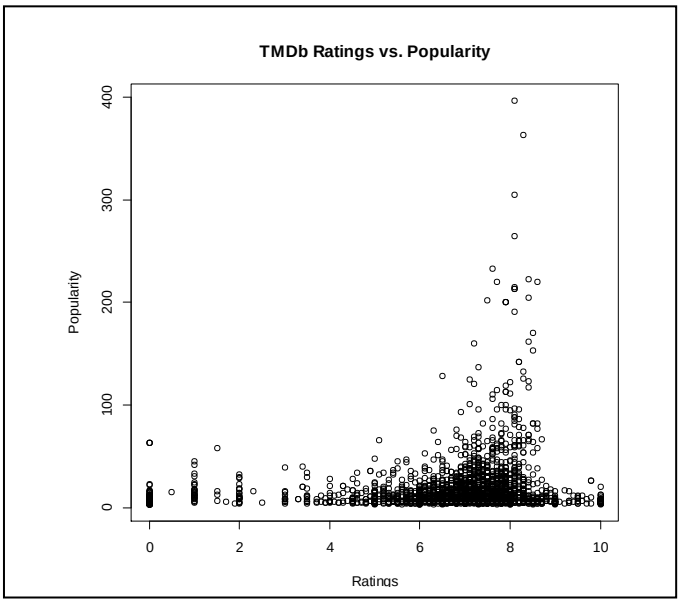

Fig. 6. TMDb Rating vs. Popularity.

From reading Fig. 5, it can be seen that there is a correlation between the popularity of a television show and IMDb ratings. However, there is not a strong correlation between the two. The majority of values present in Fig. 5 have a rating of 5.5 to 8.5 or so, meaning the shows have a strong rating. However, the corresponding popularity level is generally not that high. Most television shows in this range have a popularity level of less than 100. The strongest correlation, however, between popularity and ratings is present in the 7.5 to 8.5 range of the ratings.

The same can largely be said for the TMDb ratings and popularity of a television show, as seen in Fig. 6. However, there is a stronger correlation present in the TMDb data for ratings and popularity than the IMDb data. It can be seen that the TMDb ratings and popularity have strong correlation around the 8.0 rating range. As a result of this, it can be said that TMDb ratings data has a stronger correlation to popularity than the IMDb ratings data and popularity.

Fig. 7. IMDb and TMDb Ratings vs. Popularity.

From reading Fig. 7, it can be seen that the TMDb ratings data has a stronger and more consistent correlation to the popularity level of a television show than the IMDb ratings. This could be because the popularity data came from the TMDb originally. To conclude, there is not a strong correlation between the ratings and popularity level of a television show, however, there is a correlation. I would have expected this correlation to be a lot stronger across the board as it is the logical answer. I think this reenforces the belief that a television show may be highly reviewed and rated, but that does not mean it will be popular or enjoyed by the general public.

C. To find the correlation between genres and ratings

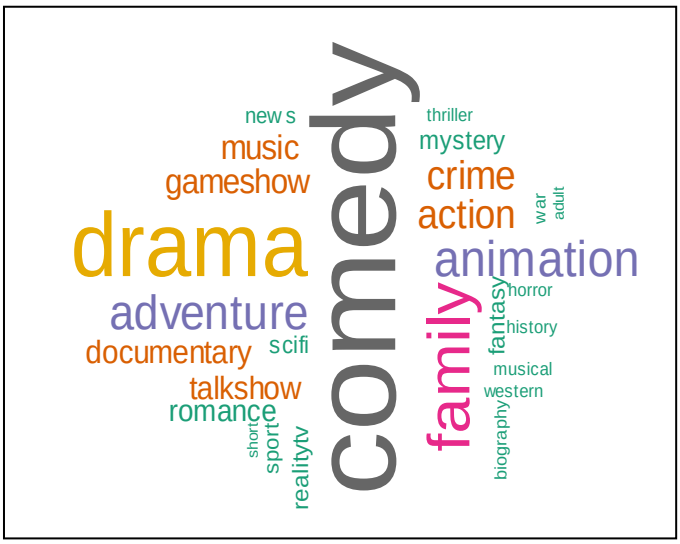

Fig. 8. TMDb Rating vs. Popularity.

From reading Fig. 8, the most common genres in the merged IMDb and TMDb dataset can be seen. I was surprised to see that comedy, as the largest word, was the most frequent genre. I expected there to be more titles around gameshows and westerns, as they are such long standing frequent types of television shows that I have seen. I expected there to be a higher frequency of genres like sport and sci-fi as there seems to be a large amount of television shows airing at the moment that would fit into these genres.

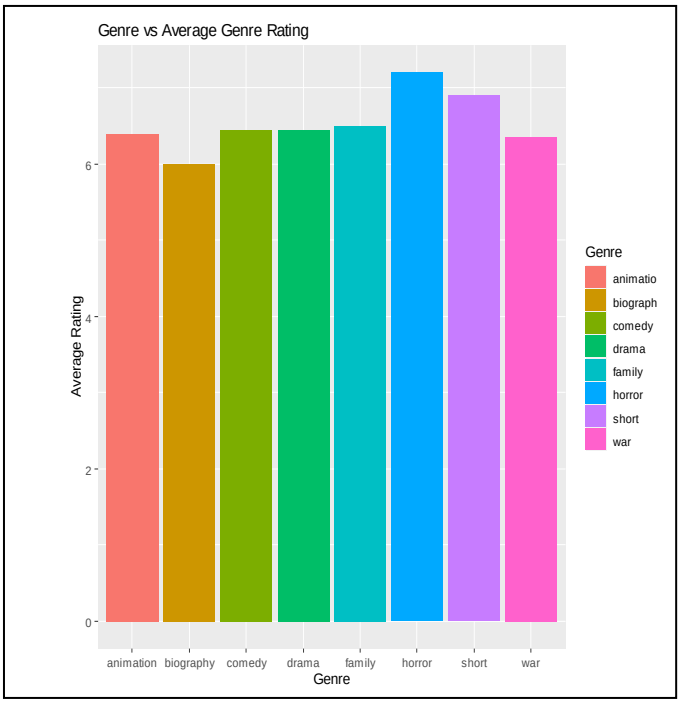

Fig. 8. TMDb Rating vs. Popularity.

From reading Fig. 8, which represents the top four most common genres (comedy, drama, family and animation), and top four least common genres (short, biography, horror and war), it can be seen that there is not high levels of variance between them.

This surprised me as I expected there to be more variance between the genres, especially in the most common versus the least common genres. I expected that a genre like comedy, which is generally well-favored in terms of television, would have a higher average rating. However, I expected a genre like biography to have the weakest average rating and it does.

This may be due to people watching biographies about people they admire and not liking how they were portrayed or perhaps were annoyed due to finding out unsavory facts about the person they admire. I was surprised to see that horror had the highest average rating as horror movies can be incredibly cliché at times.

However, a lot of titles in the horror genre do have a cult following so perhaps this accounts for some of the high rating. To conclude, there is not a high level of variance between genres and their average ratings, and that the horror genre was received the highest average rating and biography received the weakest.

VII. Conclusions and Future Work

I learned a huge amount about programming with R, best practices for dealing with data, how to extract data from an API, how to carry out KDD methodology on datasets. I learned a lot by trial and error by attempting these things. There are many lessons that I have learned from carrying out this report that I will carry into any future projects.

In terms of what I would do differently if I were to do the CA again, I would definitely make a greater effort to get ahead of the selection and pre-processing sections of KDD for the report. Coming up with an strong idea that's viable for the project and then finding datasets that would be suitable for the project is a very long and time consuming process and no other parts of the CA can be started until these sections have been completed.

I would also look deeper into the dataset and gather more insights from it. There is a plethora of insights to be gleaned from a dataset such as this and I feel like I have only scratched the surface. Insights that are valuable from a business perspective would especially be interesting to investigate.

I would also avoid using the mean to deal with missing data as this may have skewed the results and introduced bias in the data. This was done with the intention of avoiding issues with the data due to NA values and having a more concrete dataset to draw insights from. However, this may have unintentionally had an impact on the results of the correlation between columns as well as the other insights.

Similarly, I would also introduce the Trakt.tv API to the dataset to investigate the objectives of the report. This could introduce more interesting data for the insights as well as corroborate the one I have found already.

References

[1] A. Pichard, Le nouvel âge d'or des séries américaines, 1st ed. France: Editions Le Manuscrit, 2011, pp. 1-224.

[2] A. Sepinwall, The revolution was televised, 1st ed. New York City: Gallery Books, 2013, pp. 32-336.

[3] M. Sholars, "HuffPost is now a part of Verizon Media", Huffingtonpost.ca, 2020. [Online]. Available: https://www.huffingtonpost.ca/mike-sholars/animation_b_4833456.html. [Accessed: 17- Nov- 2020].

[4] J. Plunkett and J. Deans, "Kevin Spacey: television has entered a new golden age", the Guardian, 2020. [Online]. Available: https://www.theguardian.com/media/2013/aug/22/kevin-spacey-tv-golden-age. [Accessed: 17- Nov- 2020].

[5] Unknown, "IMDb | Help." https://help.IMDb.com/article/IMDb/general-information/what-is-IMDb/G836CY29Z4SGNMK5?ref_=helpsect_cons_1_1# (accessed Nov. 15, 2020).

[6] Unknown, "General FAQ — The Movie Database (TMDb)." https://www.themoviedb.org/faq/general (accessed Nov. 15, 2020).

[7] K. Pavlik, "Halloween Episodes Get Higher IMDb Ratings", Kaylin Pavlik, 2017. [Online]. Available: https://www.kaylinpavlik.com/halloween-tv-episodes-are-better-than-regular-episodes/. [Accessed: 18- Nov- 2020].

[8] Unknown, "Wikipedia", Wikipedia.org, 2020. [Online]. Available: https://www.wikipedia.org/. [Accessed: 18- Nov- 2020].

[9] N. Kapoor, S. Vishal and K. K. S., "Movie Recommendation System Using NLP Tools," 2020 5th International Conference on Communication and Electronics Systems (ICCES), COIMBATORE, India, 2020, pp. 883-888, doi: 10.1109/ICCES48766.2020.9137993.

[10] Unknown, "Automatically track TV & movies you're watching", Trakt.tv, 2020. [Online]. Available: https://trakt.tv/dashboard. [Accessed: 18- Nov- 2020].

[11] "Unknown, "Trakt API · Apiary", Trakt.docs.apiary.io, 2020. [Online]. Available: https://trakt.docs.apiary.io/#reference/shows/ratings/get-show-ratings. [Accessed: 18- Nov- 2020].

[12] Unknown, "IMDb Datasets", Datasets.imdbws.com, 2020. [Online]. Available: https://datasets.imdbws.com/. [Accessed: 18- Nov- 2020].

[13] Unknown, "API Docs", Developers.themoviedb.org, 2020. [Online]. Available: https://developers.themoviedb.org/3/tv/get-popular-tv-shows. [Accessed: 18- Nov- 2020].

[14] D. Crockford, "JSON", Json.org, 2020. [Online]. Available: https://www.json.org/json-en.html. [Accessed: 20- Nov- 2020].