TV Engine

1st André Flores
up201907001@fe.up.pt

2nd Guilherme Garrido
up201905407@fe.up.pt

3rd Luís Lucas
up201904624@fe.up.pt

***Abstract*—This document serves as a report for the project of the curricular unit of Information Processing and Retrieval (PRI). The result of the project is an information search system. The document starts by detailing the chosen dataset and the respective data preparation activities, including a processing pipeline, data operations, the final dataset, a conceptual data model, and data characterization.**

The theme of this particular work is Movies and TV Shows from Netflix.

I. Introduction

The objective of this project is to develop an information search system, which includes work on data collection and preparation, information querying and retrieval, and retrieval evaluation. The group chose a dataset related to movies and shows as it has both textual (e.g. movie's description) and numeric (e.g. movie's runtime) data. TV Engine is a movie/show search system that includes recommendations to other similar items, based on several parameters.

II. Dataset

The dataset used in this project refers to movies and shows available on Netflix streaming in July 2022, extracted by Victor Soeiro. It was obtained from Kaggle (License CC0: Public domain) and originally included two files: titles.csv and credits.csv, with 5850 and 77801 records, corresponding to 1,93MB and 3,63MB, respectively.

A. Data processing pipeline

Analyzing the original data, using OpenRefine exhaustively, allowed the identification of inconsistent records, mostly in the titles.csv file. Some records were deleted: duplicates and records with missing data (described in the next section). Then, it was decided to split the data from the titles.csv, extracting the information related to ratings/scores and genres. Therefore, two new files were created: scores.csv and genres.csv.

B. Data operations

Inconsistent data were mostly found in the cleaned titles.csv file. Duplicate titles (1 record removed) and missing descriptions(18 records removed), records with no information about age certification (2619 records set to 'unrated' to not disturb statistics), and the movie´s runtime declared as 0 (14 records replaced by 'null' to no disturb statistics) were the data issues found. These deletions led to the need to remove the connected rows in credits.csv, so actors and directors of the movies/show deleted were removed as well as they have no value now.

Also, in the file created regarding the scores, some records were found without values related to the score, voting, and popularity on the IMDb and TMDB platforms. The information about TMDB could be fetched from its website as the movies/shows are identified by title.id which is available in every record. On the other hand, the IMDb system works with a different identifier (imdbId in the scores.csv) which

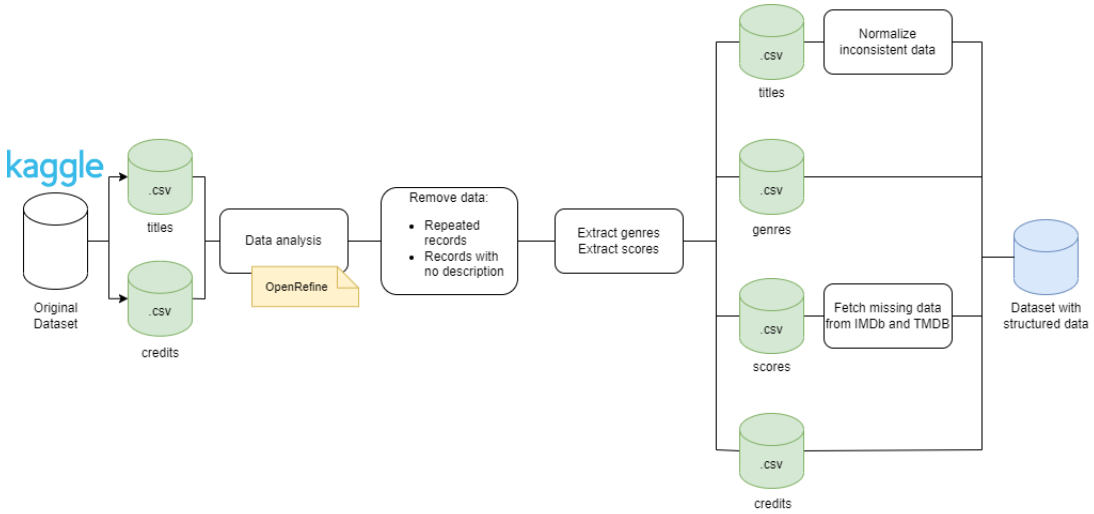

Fig. 1. Data processing pipeline.

is missing in 403 records. These records cannot have their IMDb scores and votes fetched so the values were left empty.

No more inconsistencies were found.

C. Final dataset

Title.csv keeps the information about the movie/show and has the following columns:

- id - movie/show identifier on JustWatch platform
- title - the title of the movie/show
- type - MOVIE or SHOW
- description - brief description of the movie/show
- release year - the year of release
- age certification - adequate age for the audience of the movie/show
- runtime - length of the movie or episode (show)
- production countries - list of countries that produced the movie/show
- seasons - number of seasons available for the show

Credits.csv keeps the information about the actors and directors of the movie/show:

- person id - person identifier on JustWatch platform
- id - movie/show identifier
- name - the name of the actor/director
- character - the name of the character the actor played
- role - ACTOR or DIRECTOR

Genres.csv columns:

- id - movie/show identifier
- List of all genres present in the original titles.csv. Each cell has 1 as value if the corresponding movie/show is considered from the corresponding genre and 0 otherwise.

Scores.csv columns:

- id - movie/show identifier on JustWatch platform
- imdb id - movie/show identifier on IMDb platform
- imbd score - movie/show score on IMDb platform
- imdb votes - movie/show votes on IMDb platform
- tmdb popularity - movie/show popularity on TMDB platform
- tmdb score - movie/show score on TMDB platform

D. Conceptual data model

After analyzing the original data, it was decided to split the titles domain, which included genres and scores. So it ended up with four main domains (Figure 2): titles, credits, genres, and scores, each corresponding to a file. The attributes in the model are the columns of the respective data structure and already were described in the last section ”Final dataset”. A title can have multiple credits and an individual can be credited in multiple titles. A title has only one list of genres and each list is associated to just one title. A title has a set of scores and a set of scores is associated to only one title.

E. Makefile

The makefile produced follows the process of data operations shown in the pipeline. It starts by deleting the entries with duplicate titles or without descriptions. Then, it creates new

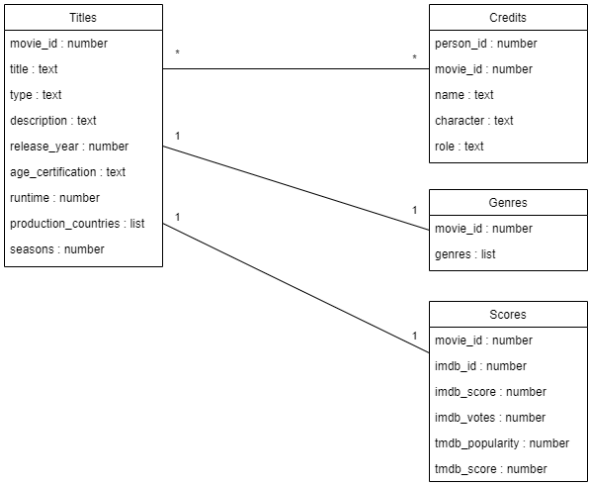

Fig. 2. Conceptual data model.

files for genres and scores and deletes the respective columns from titles. Also alters records with no information about age certification, setting it to ’unrated’, and records with 0 runtime, replacing it with ’null’. Furthermore, the makefile updates the records with missing scores by fetching them from the IMDb and TMDB websites. Python scripts were used to create the genre file and to fetch scores.

F. Data characterization

In this section some histograms are presented to help understand the content of the project’s dataset.

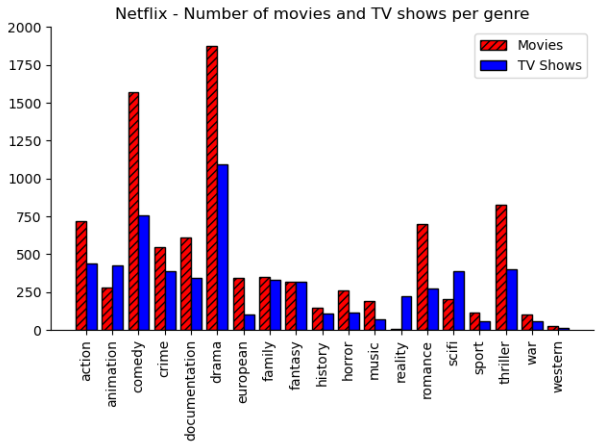

Fig. 3. Movies/shows distribution per genre.

The titles.csv file has almost 6000 registers. Figure 3 shows the distribution by genres and by type (movie or show). It is visible that drama, comedy, thriller, action, and romance are the most common genres of movies/shows, and there are significantly more movies than shows.

The next histograms (Figures 4 and 5) show the evolution of the production of movies and shows. Only 3,5% of titles were produced before the year 2000 so it was decided not to include that data here.

The movies followed an exponential growth until 2017, peaking in 2018 and decreasing in the following years (Figure 4). The TV shows followed a similar evolution until 2018 but on a smaller scale, peaking only in 2021 (Figure 5).

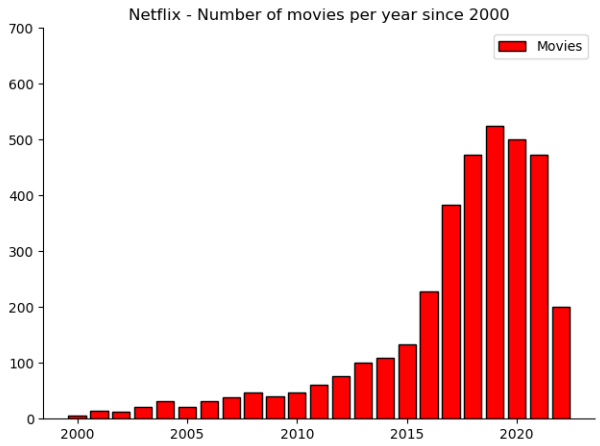

Fig. 4. Movies distribution per year since 2000.

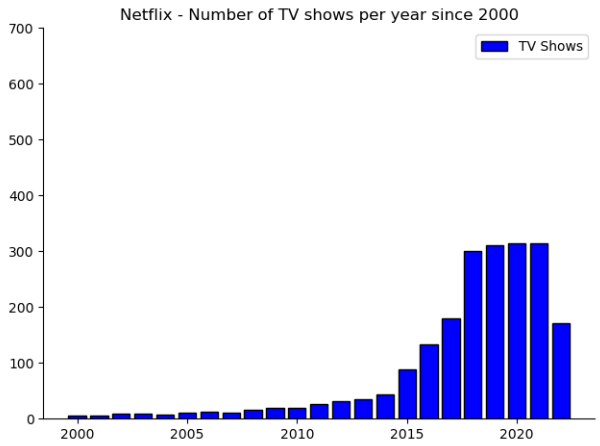

Fig. 5. Shows distribution per year since 2000.

Figures 6 and 7 show, in histograms, the average score, on the IMDb platform, of movies and shows, respectively, grouped by the year of production. It can be concluded that the audience tends to prefer older productions, as the scores given to the more recent movies are lower than the older ones, and they also prefer TV shows to movies.

Figure 8 shows the average scores of movies/shows grouped by genre. TV shows are usually better rated than movies from the same genre. Animation, european, horror, romance, and thriller have the biggest differences between movies and shows.

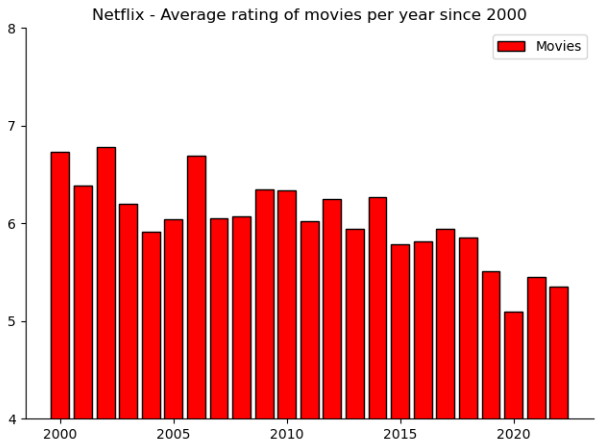

Fig. 6. Average movie score per year since 2000.

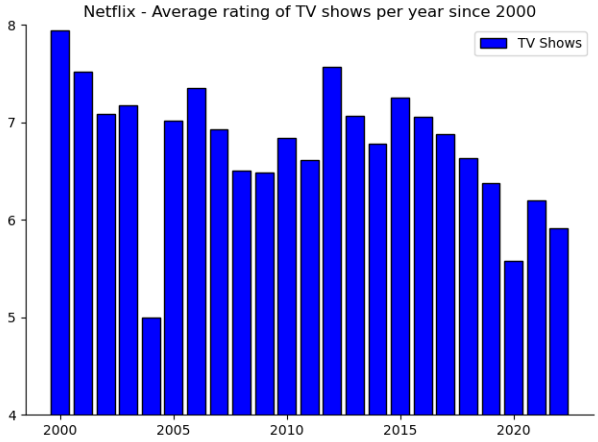

Fig. 7. Average TV show score per year since 2000.

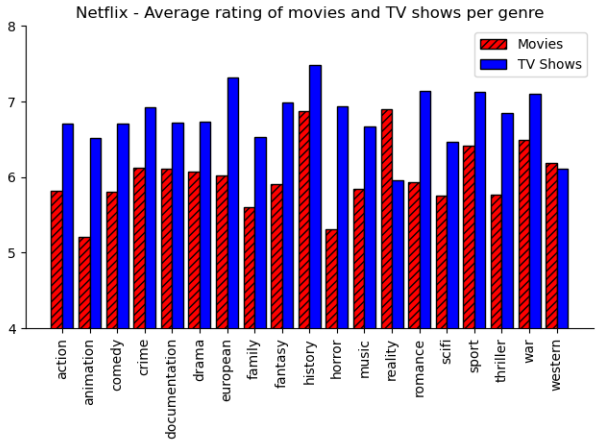

Fig. 8. Average movie/show score per genre.

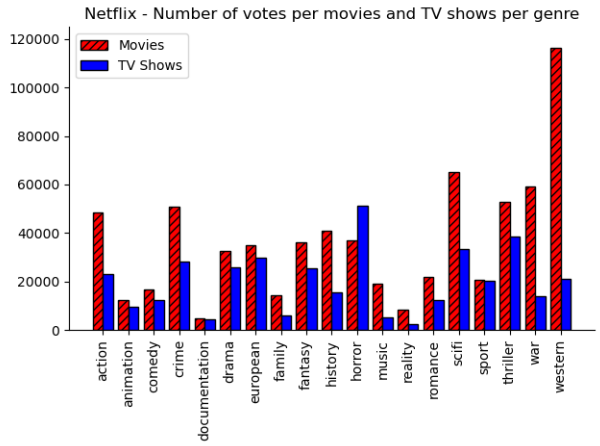

Fig. 9. Average votes on movie/show per genre.

Figure 9 illustrates the average number of votes per movie/show grouped by genre, on the IMDb platform. Movies have usually more interaction from the audience. The most voted genres are action, crime, horror, scifi, and thriller. Western movies have such a high number of votes due to *Django Unchained*'s 1,5 million votes being one of the only 28 movies from this genre.

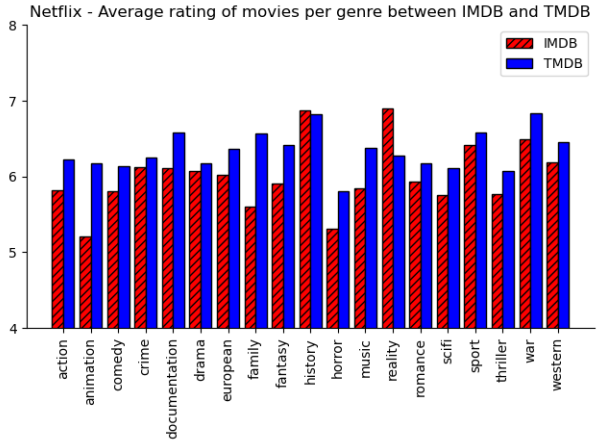

Fig. 10. Average movie score per genre IMDb vs TMDB.

Figure 10 compares the average score of movies, grouped by genre, on the platforms IMDb and TMDB. No genre has a difference higher than 1 point on a 1-10 scale. TMDB has generally better scores for the same kind of movie.

Figure 11 does the same comparison between platforms for TV shows. As in the previous histogram, TMDB has better scores than IMDb for the same genre, but with no significant differences.

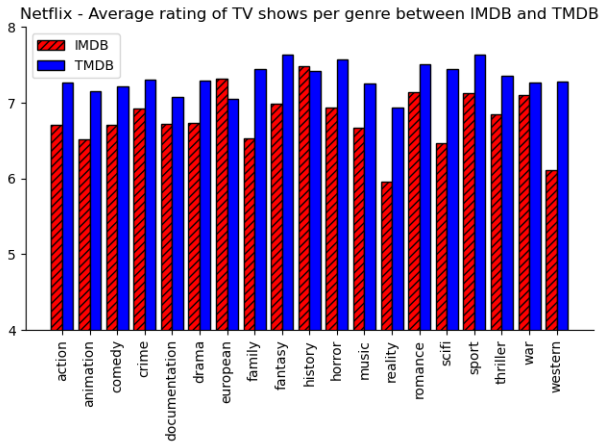

Fig. 11. Average show score per genre IMDb vs TMDB.

G. Search scenarios

The next step in this project is to implement and use an information retrieval tool on the dataset. Below are some examples of interesting retrieval scenarios:

- What shows are similar to 'Game of Thrones'?
- What actors are similar to George Clooney?
- What are the best horror movies based on people´s classifications?
- Has Emilia Clarke participated in any show since 2020?

III. Conclusions

This document described the process of data preparation on a dataset of movies and TV shows. The dataset and consequent data operations were described, as well as the defined processing pipeline and makefile. The conceptual data model was presented and the data was analyzed through some histograms.

The expected work for this stage was successfully concluded. In the next stages, the implementation of an information retrieval tool on the dataset and its exploration with free-text queries is expected.

References

[1] –. (2022, Oct) IMDb [Online]. Available: https://www.imdb.com/
[2] –. (2022, Oct) TMDB [Online]. Available: https://www.themoviedb.org/