PRI2138 Project Report

IMDb Data

Gonçalo Teixeira, Pedro Pinto, Pedro Azevedo
Faculdade de Engenharia, Porto

ABSTRACT

Information has never been more important, and has never been more available has well. This means that the processing of acquired data and the retrieval of lost information is has crucial has never before. This report intends to document the process of creation of a good database and an information processing and retrieval tool to be used in that same database in the context of the 2021/22 Project for Information Processing and Retrieval course.

KEYWORDS

Information Processing and Retrieval, M.EIC, datasets, data, IMDb, movie's data, statistics

1 INTRODUCTION

There is a considerably large amount of data out there on the web, data which can be accessed by any person, from anywhere, but most of the time this data is found in a very raw format, which can lead to a difficult understanding. Our purpose for this report is to document how we handled movie (and movie teams) data from IMDb.

We have selected IMDb movies for our project's theme because the movie industry is an area rather old, and it has been accumulating data for decades now, and it's not something that will stop accumulating data for the foreseeable future.

2 DATASET

2.1 Dataset Choice

Since the moment we have decided to work with movie data from IMDb, we've immediately started looking for datasets for the project. On *IMDb - Interfaces* we could find an extensive dataset, or rather datasets, containing data regarding movies, personal, ratings, reviews and votes. We recall that on just the movie dataset there were 8 million rows worth of data, which can be challenging and exciting to work with, but we have also noticed there was not enough textual data to work with, namely, the movie synopsis for example was missing.

Our first approach to the missing textual values was to find an API for us to retrieve that information, unfortunately, there's no free API available which can process 8 million requests, the best we could find could only offer 100 requests per day. After failing with the API approach we've decided to try to scrape the data from the IMDb website directly, and despite being possible we came to the conclusion it would take forever to scrape the data we needed.

The solution for this problem was to abandon the IMDb provided datasets and find someone who have collected the data we needed. We've found a Kaggle Dataset (link below this section) containing exactly what we've been looking for, the only downside was it only had around 86 thousand movies, which can be a large number but it is nothing compared to 8 million movies worth of data.

In conclusion, we've selected datasets with data from IMDb but collected by someone else, with the data we needed in a smaller factor.

Datasets - Kaggle:
https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset

2.2 Dataset Content

The datasets we've chosen, while not as large as the IMDb provided ones, contains the following:

- The movies dataset includes 85,855 movies with attributes such as movie description, average rating, number of votes, genre, etc.
- The ratings dataset includes 85,855 rating details from demographic perspective.
- The names dataset includes 297,705 cast members with personal attributes such as birth details, death details, height, spouses, children, etc.
- The title principals dataset includes 835,513 cast members roles in movies with attributes such as IMDb title ID, IMDb name ID, order of importance in the movie, role, and characters played.

2.3 Data Quality and Source

Kaggle is a well-known community for data analysts and researchers, the post author states the data was scraped directly from the IMDb public website, and the post has over 400 votes, with this information we can infer the source is reliable.

Regarding data quality, we'd say it met our criteria, the data we've needed was there and there wasn't any unpredicted values or formats found while doing a brief analysis of the data.

3 PIPELINE

Our Data Preparation Pipeline is built almost entirely on python scripts, the *pandas* and *matplotlib* libraries were extremely helpful for the data handling and manipulation, which lead to simple yet powerful scripts to clean and organize the data.

Our main goal was to have clean data in a structured data format such as an SQL database system.

3.1 Data Refinement

When it comes to the Data Refinement process, the first step was to exclude all columns with too many missing values, more than 1/3, with the most incomplete columns having more than 4/5 missing values. After further investigation within the dataset, we found some columns repeated in more than one table and other redundant

Supervised by Prof. Sara Fernandes.

Project in Computer Science (DAT620), IDE, UiS
2018.

information which were deleted. In the last step of the refinement, some other columns with irrelevant information for our project were identified and also excluded.

3.2 Data Analysis

To get a better understanding of the data in our hands, we developed a python script with several functions to obtain general information about the dataset. The more information we have, better the decisions we will have to make throughout the development of our project. By taking a quick look at the plots below, some interesting conclusions can be easily drawn.

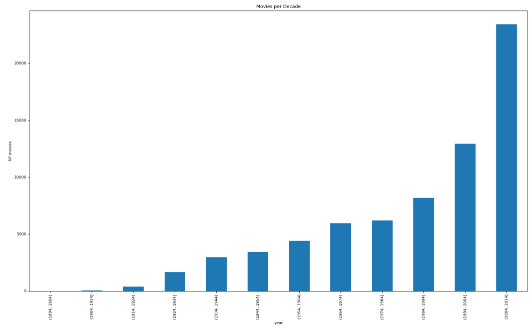

Figure 1: Movies Produced per Decade

First, the number of movies produced per decade has been increasing since the earlier decades to the present day, which means that the information about the most recent years and decades is much more rich and detailed, just as it implies more movies are being produced every decade, of course. The bar corresponding to the earliest decade has been taken out of this plot on purpose due to how big it is compared to the remaining bars, as it would make the distinction of the sizes of the smaller bars very hard.

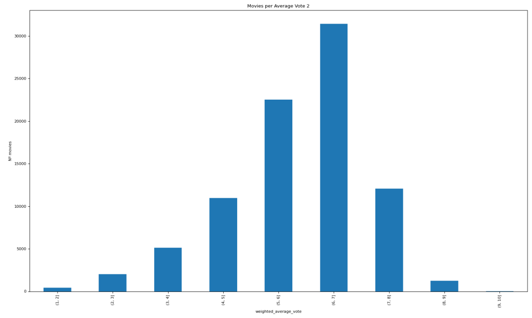

Figure 2: Movies Grouped by Rating

The second plot shows that movies tend to get a score from 5 to 7, with only a small minority of movies being able to get scores above 8 or below 3.

Last but not least, Drama and Comedy seem to be the most popular movie genres by a big margin, with Romance, Crime and Thriller completing the top 5. It is worth mentioning that all movie genres with less than 5000 movies were included in the "other" category to reduce the amount of slices in the chart and thus greatly improve readability.

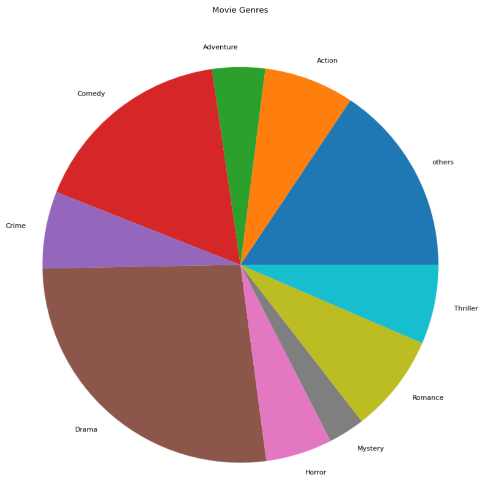

Figure 3: Movies Grouped by Genres

3.3 Database Creation

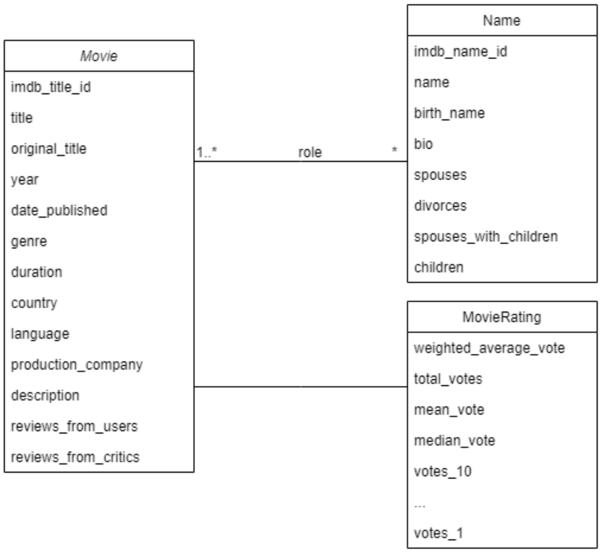

Figure 4: Database Model

The creation of the database was done with another python script that, considering all the analysis done, creates an SQL database accordingly using the refined datasets. The database consists of four tables. The Movie table stores the general information about every movie. The Movie Rating table stores the rating related information for every movie. The Name table contains data about people related to the movie industry. And a MoviePersonal table, every person is connected to a movie with the help of a role association that also serves the purpose of identifying, as the name suggests, the role played by that person in the creation of the movie.

3.4 Pipeline Conclusion

The following pipeline scheme summarizes the hole dataset cleaning, analysis and database creation process.

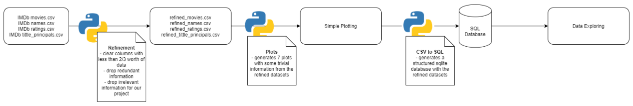
IMDb movies.csv
IMDb names.csv
IMDb ratings.csv
IMDb tittle_principals.csv
Refinement
- clear columns with less than 2/3 worth of data
- drop redundant information
- drop irrelevant information for our project
refined_movies.csv
refined_names.csv
refined_ratings.csv
refined_tittle_principals.csv
Plots
- generates 7 plots with some trivial information from the refined datasets
Simple Plotting
CSV to SQL
- generates a structured sqlite database with the refined datasets
SQL Database
Data Exploring

Figure 5: Pipeline Scheme

This scheme summarizes way how the initial 4 files of raw data were turned into a clean and functional sql database. First, the data was refined, then analysed and finally the database was created. In the future we intend to explore the data even further in order to create the best information retrieval tool possible for this particular database.