Extraction

We used a number of different datasets from the platform Kaggle. The data in the in files mainly included the following information:

**The Movies Dataset:** https://www.kaggle.com/rounakbanik/the-movies-dataset/code?datasetId=3405&searchQuery=genres

Movie\_name, imdb\_id, country, genre, imdb\_rating, user\_id, user\_gender, …

**User data:**

CSV files related to IMDb movie rating by users were obtained from Kaggle.

These CSV’s included data on user\_id, user\_occupation, movie\_ratings and movie\_names.

We did Web scraping from the IMDB website to get the top-rated movies along with their ratings and saved them in a CSV file which is saved as the jupyter notebook named web\_scraping.ipynb.

Text

Description automatically generated

Links:

<https://www.imdb.com/chart/top/>

From Kaggle, we used a CSV related to Netflix movies and the country of origin.

Our data is mainly originated from Kaggle metadata and IMDB rating data and we performed the ETL process by adding our data to the PostgresSQL database called ETL\_Project.

Our fields of interested include:

Movie names

User information - gender and occupation

Movies Country of origin

Movies Genres

Transformation:

To transform the public data and use it in our study we performed the following:

Cleaned the data by removing various text symbols and characters

Removed the duplicates

Dropped the NaN and none values

           Redefined columns and dropped/removed various columns that were not relevant to our study.

The main part of our project was to deal with the movie genres as each movie were set to be in a variable list of genres as depicted below:

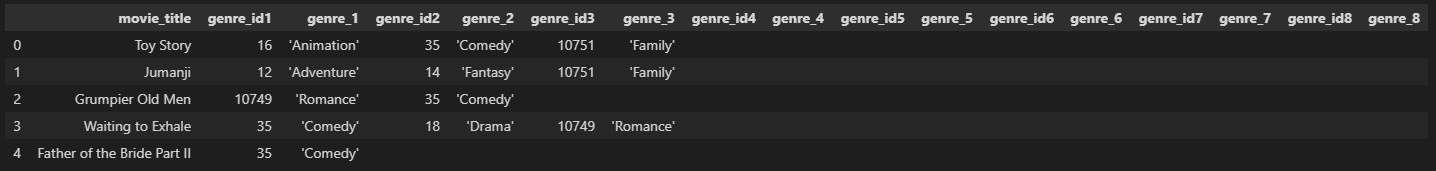
Graphical user interface

Description automatically generated

Using the replace function the genre information was cleaned

Text

Description automatically generated



In order to create a data-frame with a unique set of movie\_title & movie\_genre, we created 8 different tables out of the above, and then used the append function to obtain a single table with the following pattern:

A picture containing text, scoreboard, screenshot

Description automatically generated

Load:

We used the quick database website to create the schema that got loaded into the Postgres database that generated our tables. After the data transformation, we loaded the tables with only the relevant information and connected them to the database.

Graphical user interface, application

Description automatically generated