Proposal

Team members:

Chipampe Masongo

Nassim Zeinali

We will be extracting various sources of movie related data, and transforming it into genres, user ratings, general user details (age and gender).

Streaming services, investors and directors can use this information to identify target groups for their movies based on movie genres and/or user ratings.

Timeline:

Tuesday - finalise data sources

Data sources:

* Movies\_metadata.csv

# Oscar Best Movie Winners data

# MovieLens 100K Dataset

Thursday

* Transformation - cleaning data, creating database
* Loading Draft Data

Monday

* Project report.
* Loading final Database

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

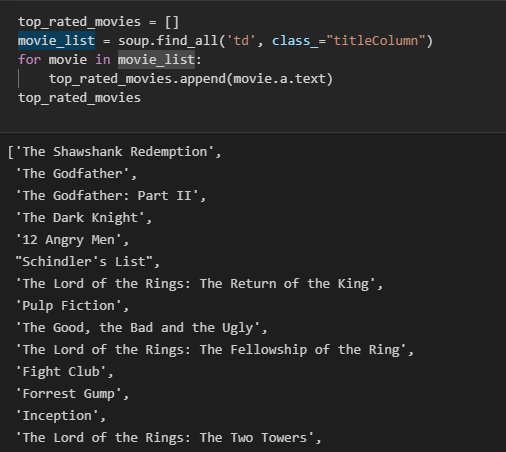
Movie\_name, imdb\_id, country, genre, imdb\_rating, user\_id, user\_gender, …

**Userdata:**

CSV files related to imdb movie rating by users were obtained form kaggle.

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

We did Web scraping from 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.



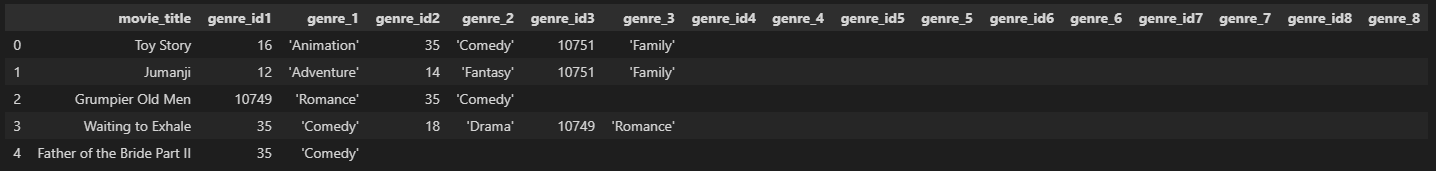
Links:

<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 ETL process by adding our data to PostgresSQL database called ETL\_Project.

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 movie genres as depicted below:



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



Transformation:

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

Cleaned the data by removing ‘ , “ characters

Removed the duplicates

Dropped the NaN values

Load:

After we pulled in the CSV files and loaded them into the data frames, we did an initial connection to the Postgres database using PG admin to store our original clean data sets. We used the quick database website to create the initial table schema that got loaded into the Postgres database that generated the first set of tables. After running the queries and creating the new tables with only the relevant information we reconnected to the database and generated additional tables for the data frames.

