**Netflix Vs Disney +**

**Databoot Camp Project 2**

**Team Members:**

Jesse Tan

Kailyn Yong

Jazzlyn Phan

Darren McMurtrie

**Project Overview**

The purpose of this project is to create a database of movies where users can search for a movie and all its information, and determine if it is available on Netflix or Disney+.

By merging the 2 datasets we can clearly find out which platform the movie available on. We can also see the number of contents that is accessible in each streaming services. This could be a big factor for customers when choosing the services if they have the same pricing. By using the combined data set we can see whether they have content catering to adults or children as well which could be the decision-maker for the parents.

Introducing IMDB into the merged dataset will enrich the data by adding the rating of each movie that is available on the streaming platform.

The 2 datasets we have used are Netflix (https://www.kaggle.com/shivamb/netflix-shows) and Disney+ (https://www.kaggle.com/unanimad/disney-plus-shows) which were sourced from Kaggle. In addition, we incorporated them with the dataset of IMDB (https://www.imdb.com/interfaces/) to standardise the rating for all movies and create our database to help users search for movies (e.g., by title or region) that are readily available on either streaming platform.

**Data Source**

Source 1: IMDB datasets were directly downloaded via<https://www.imdb.com/interfaces/>, we need the following 4 relational files

1. <https://datasets.imdbws.com/title.basics.tsv.gz>
2. <https://datasets.imdbws.com/title.ratings.tsv.gz>
3. <https://datasets.imdbws.com/title.crew.tsv.gz>
4. <https://datasets.imdbws.com/name.basics.tsv.gz>

|  |  |
| --- | --- |
| File | Description |
| **title.basics.tsv**  **(7659194 records)**  **(rename to title.tsv)** | Contains the following information for titles:   * tconst (string) - alphanumeric unique identifier of the title * titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc) * primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release * originalTitle (string) - original title, in the original language * isAdult (boolean) - 0: non-adult title; 1: adult title * startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year * endYear (YYYY) – TV Series end year. ‘\N’ for all other title types * runtimeMinutes – primary runtime of the title, in minutes * genres (string array) – includes up to three genres associated with the title |
| **title.ratings.tsv**  **(1126355 records)**  **(rename to rating.tsv)** | Contains the IMDb rating and votes information for titles   * tconst (string) - alphanumeric unique identifier of the title * averageRating – weighted average of all the individual user ratings * numVotes - number of votes the title has received |
| **title.crew.tsv**  **(7659194 records)**  **(rename to crew.tsv)** | Contains the director and writer information for all the titles in IMDb. Fields include:   * tconst (string) - alphanumeric unique identifier of the title * directors (array of nconsts) - director(s) of the given title * writers (array of nconsts) – writer(s) of the given title |
| **name.basics.tsv**  **(10749850 records)**  **(rename to name.tsv)** | Contains the following information for names:   * nconst (string) - alphanumeric unique identifier of the name/person * primaryName (string)– name by which the person is most often credited * birthYear – in YYYY format * deathYear – in YYYY format if applicable, else '\N' * primaryProfession (array of strings)– the top-3 professions of the person * knownForTitles (array of tconsts) – titles the person is known for |

Source 2: Netflix and Disney+ datasets were downloaded from kaggle.com using kaggle api, they were 2 separated files:

1. <https://www.kaggle.com/unanimad/disney-plus-shows>
2. <https://www.kaggle.com/shivamb/netflix-shows>

|  |  |
| --- | --- |
| File | Description |
| **disney-plus-shows.csv**  **(894 records)** | Contains the following information for Disney+ shows:   * imdb\_id - Internet Movie Database ID * title - show's title * plot - show’s plot * type - show’s type * rated - age-rated * year - publish year * released\_at - release date * added\_at - when the title was added to the platform * runtime - runtime * genre - genre list * director - directors list * writer - writer list * actors - actors list * language - available languages list * country - country available list * awards - awards list * metascore - metascore * imdb\_rating - Internet Movie Database rate * imdb\_votes - Internet Movie Database votes |
| **netflix-shows.csv**  **(7787 records)** | Contains the following information for Netflix shows:   * show\_id - unique id for every movie / TV show * type - identifier - a movie or TV show * title - Title of the movie / TV show * director - director list of the movie * cast - actors involved in the movie/show * country - the country where the movie/show was produced * date\_added - date it was added on Netflix * release\_year - actual release year of the movie/show * rating - age rating of the movie/show * duration - total duration - in minutes or number of seasons * listed\_in - genre * description - the summary description |

**Prepare Files**

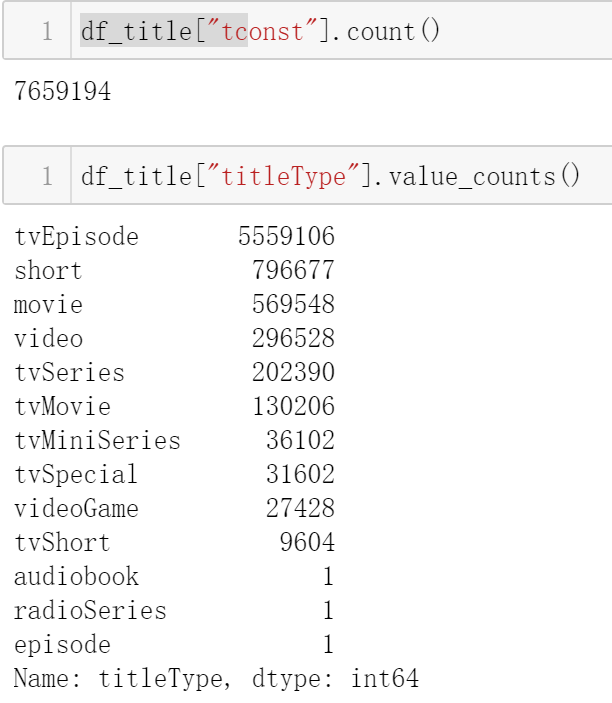
We use python libraries to download and extract the source files. For imdb datasets, we use *urllib.request.urlretrieve* function to download the files and for kaggle datasets, we follow [*https://www.kaggle.com/docs/api*](https://www.kaggle.com/docs/api) to insall the kaggle api and define a function *KaggleDownload* to implement this api.

After all the files been downloaded, we use *gzip* and *zipfile* libraries to unzip the files in gz format and zip format.

**Exploratory Data Analysis**

We load the tsv and csv files by pandas and print out some of the information:

1. For *title.tsv* file, we get the total rows count and the total number of different types of shows.



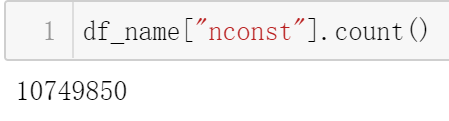
1. For *rating.tsv* file, we get the total rows count:



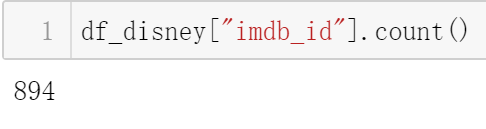
1. For *crew.tsv* file, we get the total rows count:



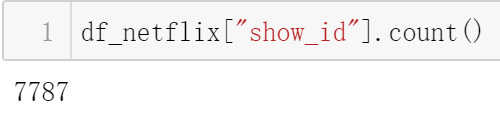
1. For *name.tsv* file, we get the total rows count:



1. For *disney\_plus\_shows.csv* file, we get the total rows count:



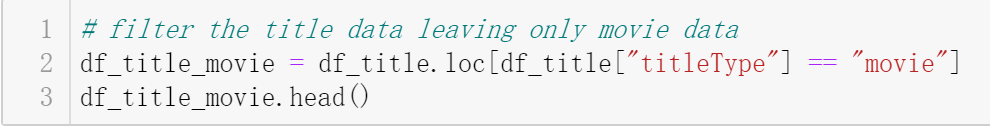
1. For *netflix\_titles.csv* file, we get the total rows count:



**Transformations**

1. **IMDB data cleaning**

As this project is focus on movies only, we will first filter out the shows with the other types in imdb dataframe, which will give us a much smaller and accurate dataframe for our later operations:



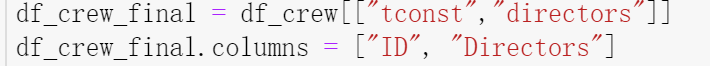
Later on, we get rid of the columns we don’t need and give a new column name to the columns stayed:

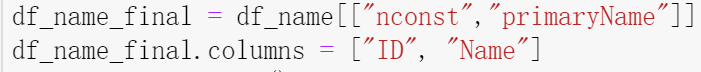


And also, we fill the cells with “np.Nan” for those have a string value “\N”, and change the “ReleaseYear” column to integer type as it was string type.

We do the column renaming to all the other imdb dataframe:







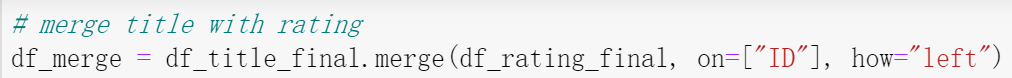
And we also check or replace if there are any “null” values for these dataframes:

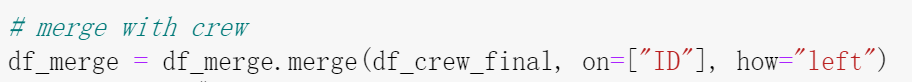


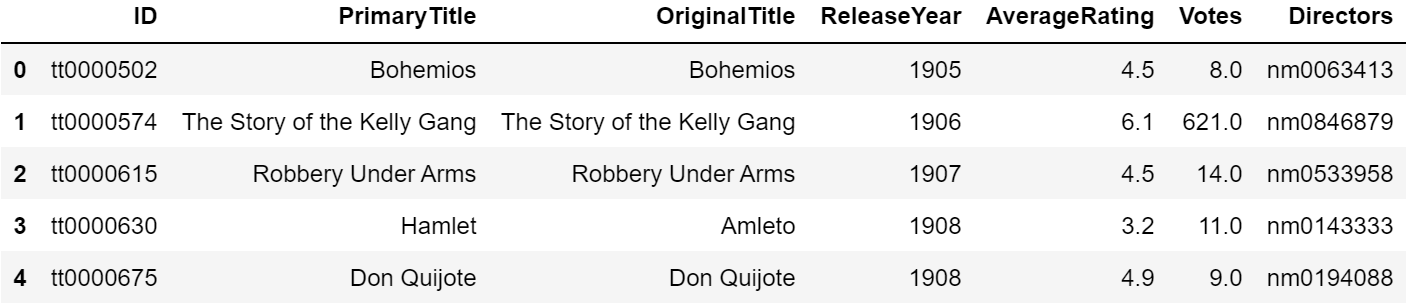


For the “df\_name” dataframe, it has only 2 columns and it is a 1-1 relation, so it won’t have any data contains null value.

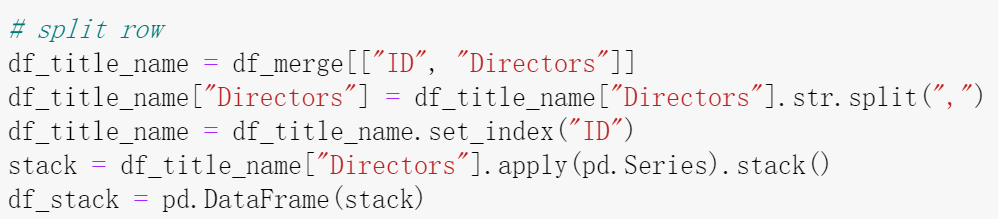
After those data cleaning processes, we can start to merge those dataframes:







Because the “Directors” column is a “csv” style value, we need to split those value into different rows to form a mapping dataframe which contains only two columns, “ID” and “Director”, a N-1 mapping:



As the dataframe is large, this operation will run for a while.

So now for imdb, we got two final dataframes:

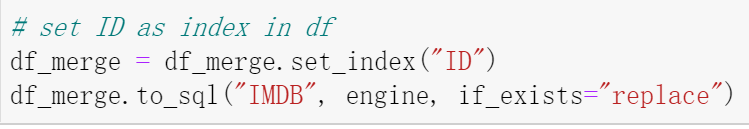
1. “Title” dataframe which contains “ID”, “PrimaryTitle”, “OriginalTitle”, “ReleaseYear”, “AverageRating”, “Votes” and “Directors” columns;
2. “Stack” dataframe which contains only “ID” and “Director” columns, for “ID” is represent the “ID” column in “Title” dataframe and “Director” column is represent the “ID” column in the “Name” dataframe.
3. **Netflix and Disney+ data clean**

The Netflix and Disney+ datasets are much smaller compare to the imdb datasets, and the cleaning steps is similar to what we have done to the imdb datasets:

1. Filter out the non-movie types;
2. Rename and retype columns.

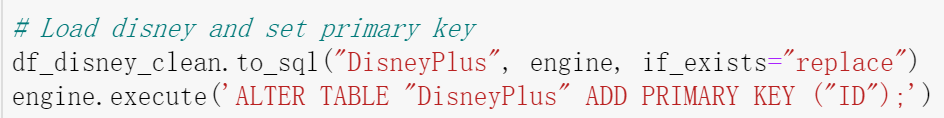
**Load**

After cleaning the data, we import the dataset into postgresql database by using pandas and sqlarchemy:



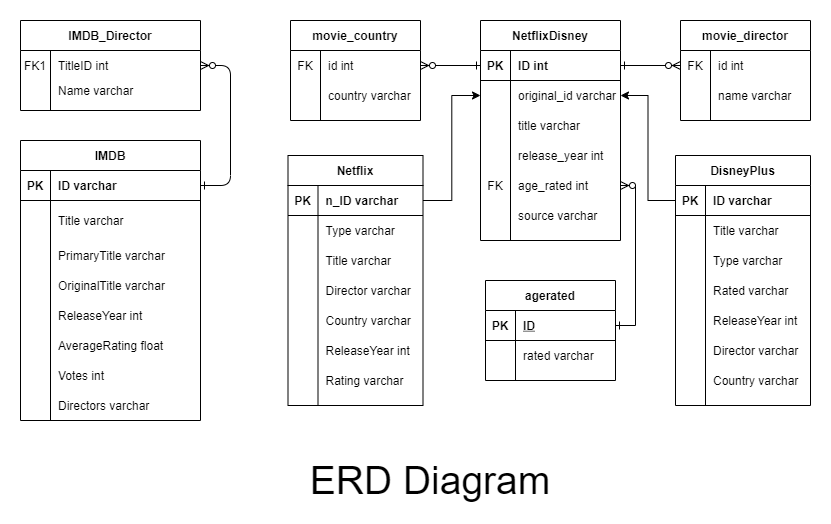








Inside the postgresql, the relationship between these tables as follow:



In this diagram, we have some extra tables that we created to be used later for some of our queries:

1. agerated table includes the age rated classification by different ages, according to the standard rated system such as “PG”,”M”;
2. NetflixDisney is a union table by Netflix and DisneyPlus tables;
3. movie\_country and moive\_director are two tables that split the column “Country” and “Director”.

We prepare IMDB\_Director and moive\_director table for complex joining for IMDB and NetflixDisney tables when some of the title for the same movie are not match such as “October 1” and “Oct 1”. We did not implement this as time not enough for doing this project. We have do some queries and joins as follow:

1. searching disneyplus movie that release in specify region;
2. searching movie with IMDB rating higher than 7;
3. searching movie in NetflisDisney with agerating 14+ and show its averagerating in imdb.

**Web Service**

We finally put everything into a web service by using Flask.