TECNOLÓGICO DE MONTERREY

TRILOGY
EDUCATION SERVICES

ITESM & Trilogy

Anacondas team:
Daniela Villarreal
Raúl Flores Palacios
Michelle García
Juan Ramón Félix

Professors:
Mr. Alejandro Estrella Gabilondo
Mr. Jorge Luis Ramos Zavaleta

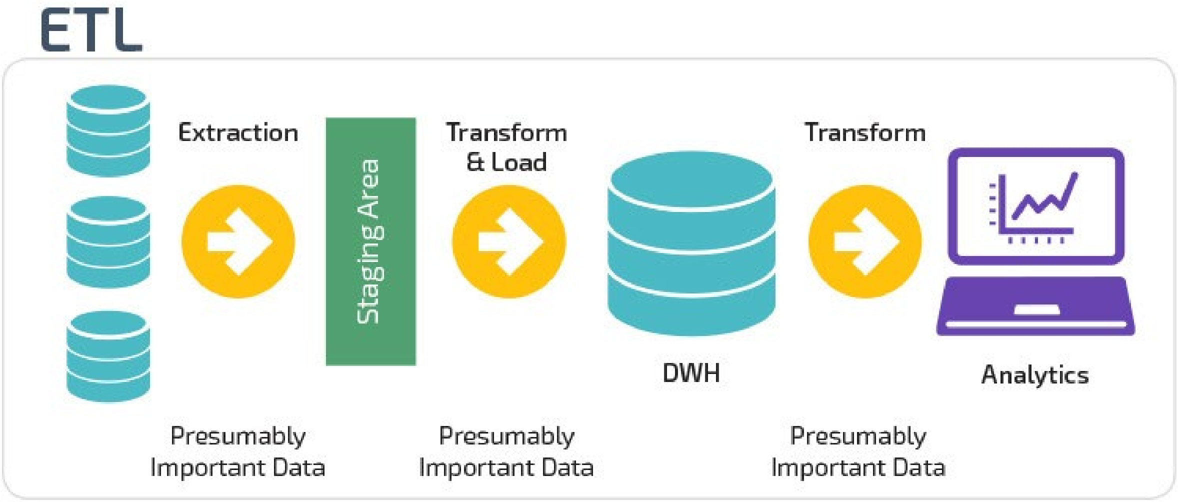
Project ETL

Data Analytics Bootcamp

18 de Julio de 2020

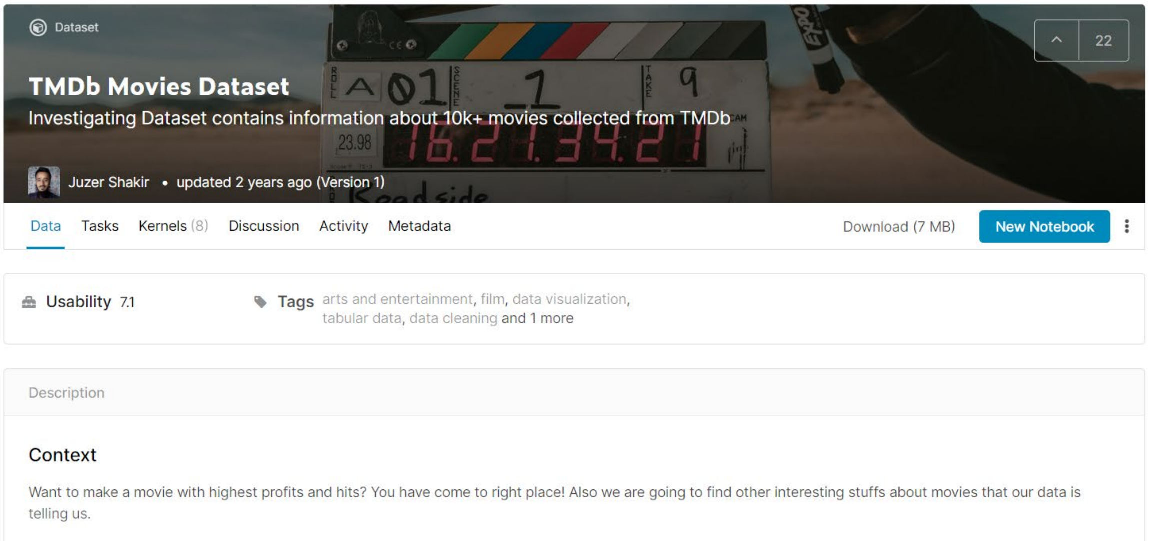
Introduction

For this project we have decided to work with information from movies as is something that we like and have interest in, and also because currently we have allot available spare time to watch a movie, so to make a better selection we have created this database. We were interested to have the ratings of the movies, that is why we use the database of reviews and qualifications of IMBD and Rotten Tomatoes, also we wanted to have a section where we can show in which streaming platform these movies were available. We follow the ETL process to acquire, transform and display our data, starting from the data extraction from databases available, then transforming our data with cleansing and adjustments and then saving this data in more readable and accessible format.



Extraction:

We have decided to use Kaggle as our starting point as is a coarse and useful web pages where we can find multiple Data bases of various topics. We search from its database collection on movies related, and we have found this to databases, that match the best to what we were looking for.



Dataset

22

TMDb Movies Dataset

Investigating Dataset contains information about 10k+ movies collected from TMDb

Juzer Shakir • updated 2 years ago (Version 1)

Usability 7.1

Tags arts and entertainment, film, data visualization, tabular data, data cleaning and 1 more

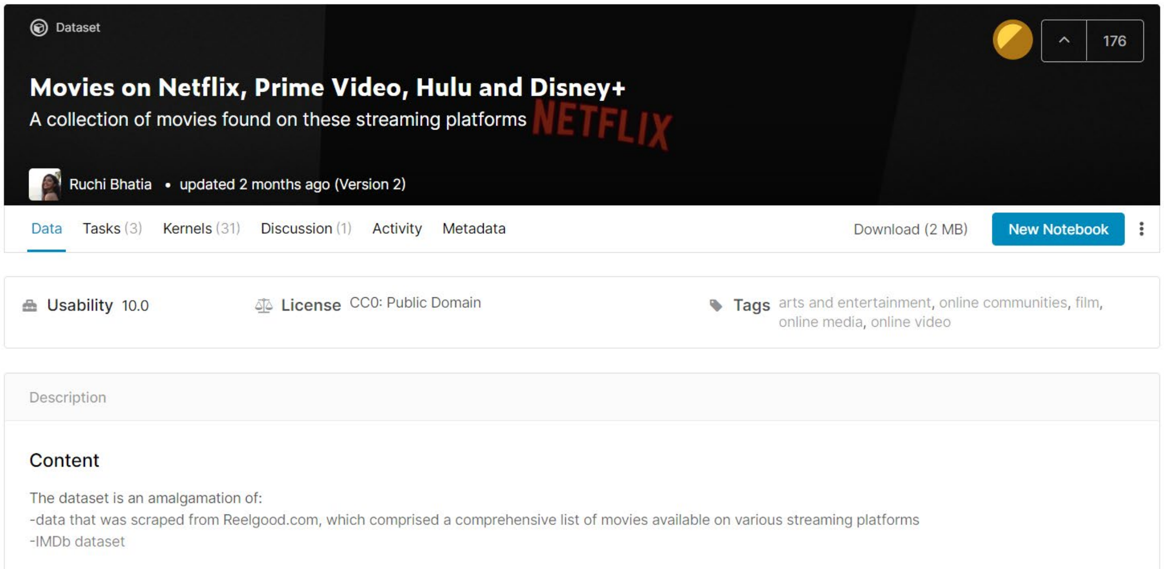
Description

Context

Want to make a movie with highest profits and hits? You have come to right place! Also we are going to find other interesting stuffs about movies that our data is telling us.

https://www.kaggle.com/juzershakir/tmdb-movies-dataset

https://www.themoviedb.org/?language=es



Dataset

176

Movies on Netflix, Prime Video, Hulu and Disney+

A collection of movies found on these streaming platforms

Ruchi Bhatia • updated 2 months ago (Version 2)

Usability 10.0

License CC0: Public Domain

Tags arts and entertainment, online communities, film, online media, online video

Description

Content

The dataset is an amalgamation of:
-data that was scraped from Reelgood.com, which comprised a comprehensive list of movies available on various streaming platforms
-IMDb dataset

https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney

https://reelgood.com/

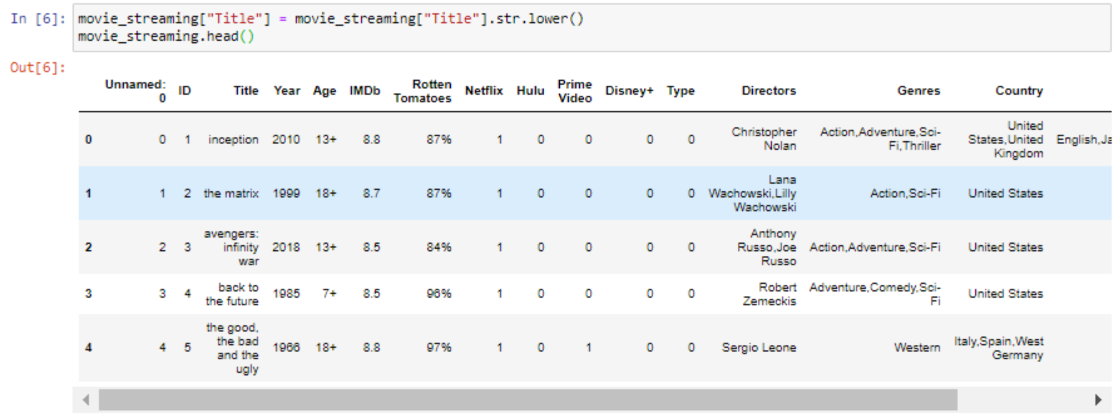
This tow data bases where available in a CSV documents. That is handily as is easier to use for extracting the data.

Transformation:

This was the most complex part of the whole project as we required to do many modifications to the databases in order to adjust them to our preferences and make it more easy for surf through database and let us see only what we wanted. Some of the modification that we made are the following.

Modification and cleansing 1:

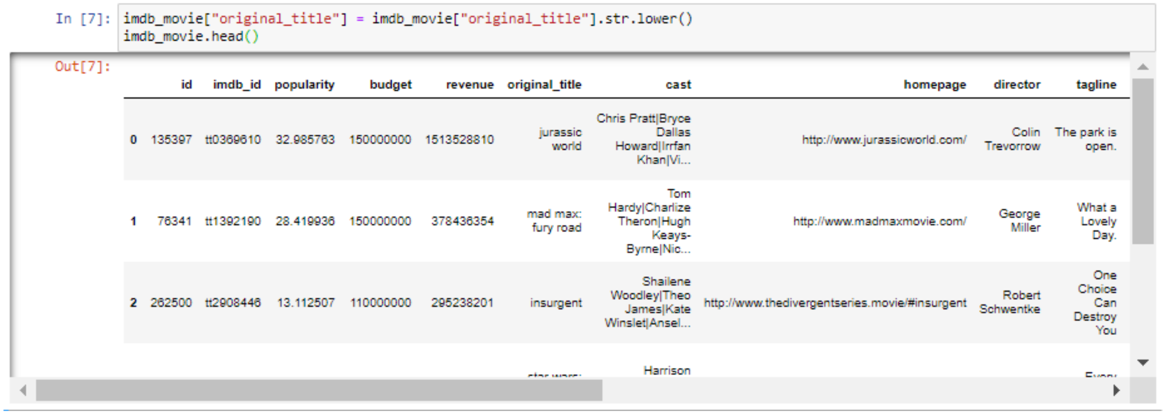
The first modification that we encounter that we required was to adjust the titles or movies names in our CSV files, so they match in format and in name. This way is will be easier and without problems in the merge of the datasets, as is will find more titles that match. One of this modification was to change all the titles to lower case in both datasets.



In [6]:
```
movie_streaming["Title"] = movie_streaming["Title"].str.lower()
movie_streaming.head()
```

Out[6]:

	Unnamed: 0	ID	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Type	Directors	Genres	Country	
0	0	1	inception	2010	13+	8.8	87%	1	0	0	0	0	Christopher Nolan	Action,Adventure,Sci-Fi,Thriller	United States,United Kingdom	English,Ja
1	1	2	the matrix	1999	18+	8.7	87%	1	0	0	0	0	Lana Wachowski,Lilly Wachowski	Action,Sci-Fi	United States	
2	2	3	avengers: infinity war	2018	13+	8.5	84%	1	0	0	0	0	Anthony Russo,Joe Russo	Action,Adventure,Sci-Fi	United States	
3	3	4	back to the future	1985	7+	8.5	96%	1	0	0	0	0	Robert Zemeckis	Adventure,Comedy,Sci-Fi	United States	
4	4	5	the good, the bad and the ugly	1966	18+	8.8	97%	1	0	1	0	0	Sergio Leone	Western	Italy,Spain,West Germany	



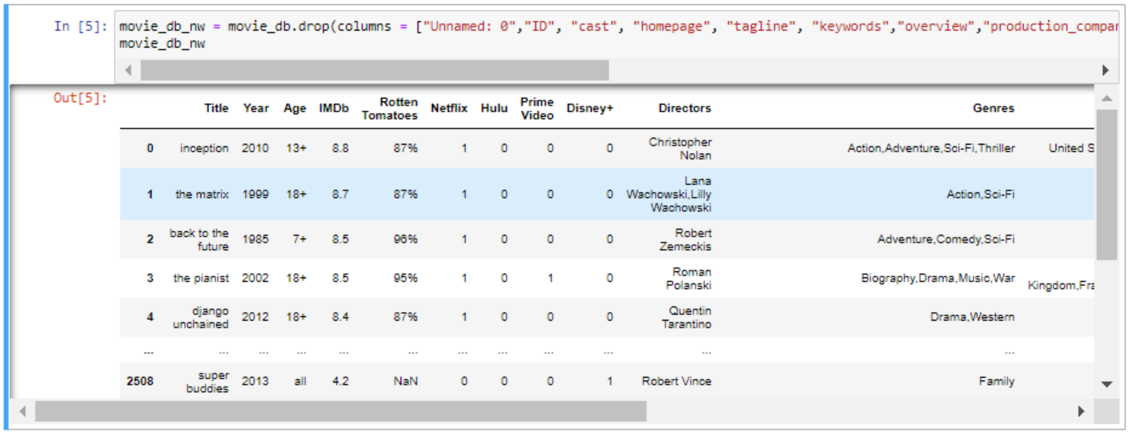
In [7]:
```
imdb_movie["original_title"] = imdb_movie["original_title"].str.lower()
imdb_movie.head()
```

Out[7]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline
0	135397	tt0369610	32.985763	150000000	1513528810	jurassic world	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Colin Trevorrow	The park is open.
1	76341	tt1392190	28.419936	150000000	378436354	mad max: fury road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/	George Miller	What a Lovely Day.
2	262500	tt2908446	13.112507	110000000	295238201	insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent	Robert Schwentke	One Choice Can Destroy You
						star wars:	Harrison			Every

Modification and cleansing 2:

Once we have merged our datasets the we have deleted all the columns that were not interesting for our research Cast, home page, tagline, key words, over view, production company, etc.…, also we have delated the columns that were repetitive. We have as well save some interesting columns that we wanted to have in our data base as are director name, run time, categories, genres, country of production, etc.…



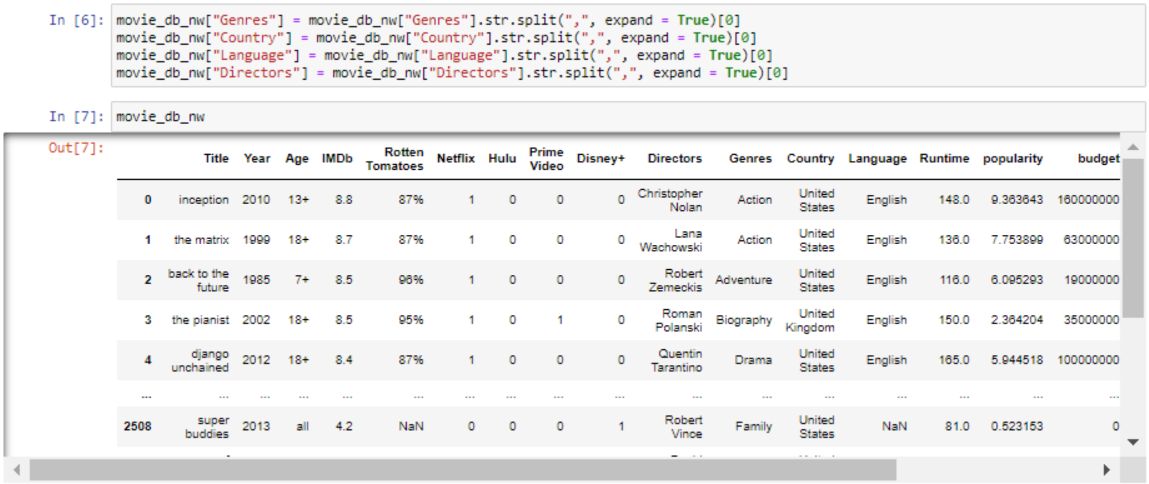
In [5]:
```
movie_db_nw = movie_db.drop(columns = ["Unnamed: 0","ID", "cast", "homepage", "tagline", "keywords","overview","production_compar
movie_db_nw
```

Out[5]:

	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Directors	Genres	
0	inception	2010	13+	8.8	87%	1	0	0	0	Christopher Nolan	Action,Adventure,Sci-Fi,Thriller	United S
1	the matrix	1999	18+	8.7	87%	1	0	0	0	Lana Wachowski,Lilly Wachowski	Action,Sci-Fi	
2	back to the future	1985	7+	8.5	96%	1	0	0	0	Robert Zemeckis	Adventure,Comedy,Sci-Fi	
3	the pianist	2002	18+	8.5	95%	1	0	1	0	Roman Polanski	Biography,Drama,Music,War	Kingdom,Fra
4	django unchained	2012	18+	8.4	87%	1	0	0	0	Quentin Tarantino	Drama,Western	
...	...	...	...	...	...	...	...	...	...	...	...	
2508	super buddies	2013	all	4.2	NaN	0	0	0	1	Robert Vince	Family	

Modification and cleansing 3:

We found that some values in the columns have a list per line, there for we have taken the decision that we were only to use the first value of the list in order to facilitate the searching of the directories. There for if the movie have the genre as action, adventure, sci-fi and thriller, we have only taken the value "action" or if the film was produced in the United States and United Kingdom and Germany have only saved the United States as the country of origin.



In [6]:
```
movie_db_nw["Genres"] = movie_db_nw["Genres"].str.split(",", expand = True)[0]
movie_db_nw["Country"] = movie_db_nw["Country"].str.split(",", expand = True)[0]
movie_db_nw["Language"] = movie_db_nw["Language"].str.split(",", expand = True)[0]
movie_db_nw["Directors"] = movie_db_nw["Directors"].str.split(",", expand = True)[0]
```

In [7]:
```
movie_db_nw
```

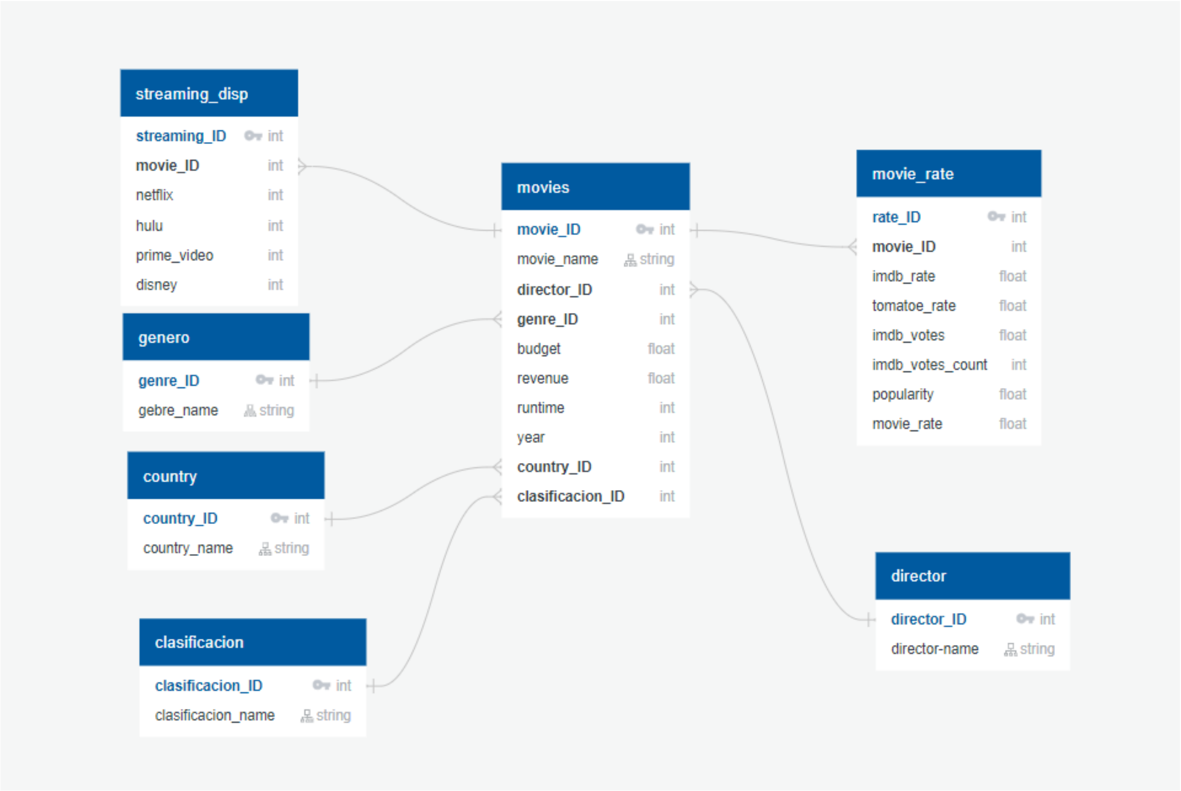
Out[7]:

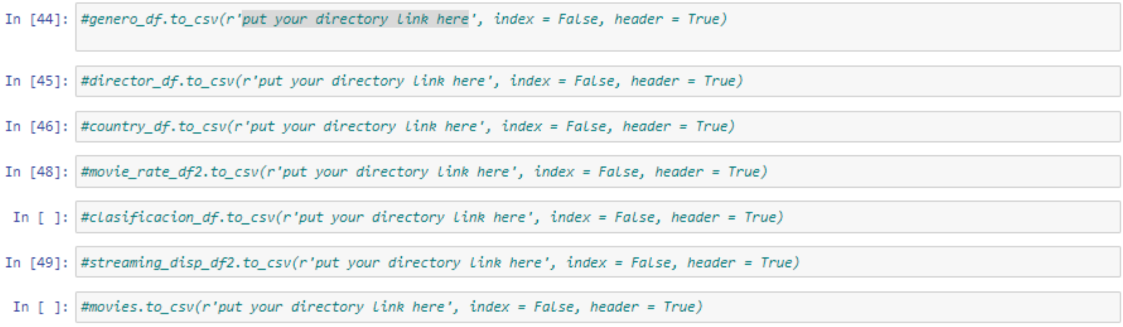
	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Directors	Genres	Country	Language	Runtime	popularity	budget
0	inception	2010	13+	8.8	87%	1	0	0	0	Christopher Nolan	Action	United States	English	148.0	9.363643	160000000
1	the matrix	1999	18+	8.7	87%	1	0	0	0	Lana Wachowski	Action	United States	English	136.0	7.753899	63000000
2	back to the future	1985	7+	8.5	96%	1	0	0	0	Robert Zemeckis	Adventure	United States	English	116.0	6.095293	19000000
3	the pianist	2002	18+	8.5	95%	1	0	1	0	Roman Polanski	Biography	United Kingdom	English	150.0	2.364204	35000000
4	django unchained	2012	18+	8.4	87%	1	0	0	0	Quentin Tarantino	Drama	United States	English	165.0	5.944518	100000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2508	super buddies	2013	all	4.2	NaN	0	0	0	1	Robert Vince	Family	United States	NaN	81.0	0.523153	0

Modification and cleansing 4:

Once we have our main data set clean, we have processed then to divide it in directories to make it in smaller tables that are easier to relate and to review. And in other to use them in our SQL we have make CSV´s of this tables.

DBD diagram





```
In [44]: #genero_df.to_csv(r'put your directory link here', index = False, header = True)

In [45]: #director_df.to_csv(r'put your directory link here', index = False, header = True)

In [46]: #country_df.to_csv(r'put your directory link here', index = False, header = True)

In [48]: #movie_rate_df2.to_csv(r'put your directory link here', index = False, header = True)

In [ ]: #clasificacion_df.to_csv(r'put your directory link here', index = False, header = True)

In [49]: #streaming_disp_df2.to_csv(r'put your directory link here', index = False, header = True)

In [ ]: #movies.to_csv(r'put your directory link here', index = False, header = True)
```

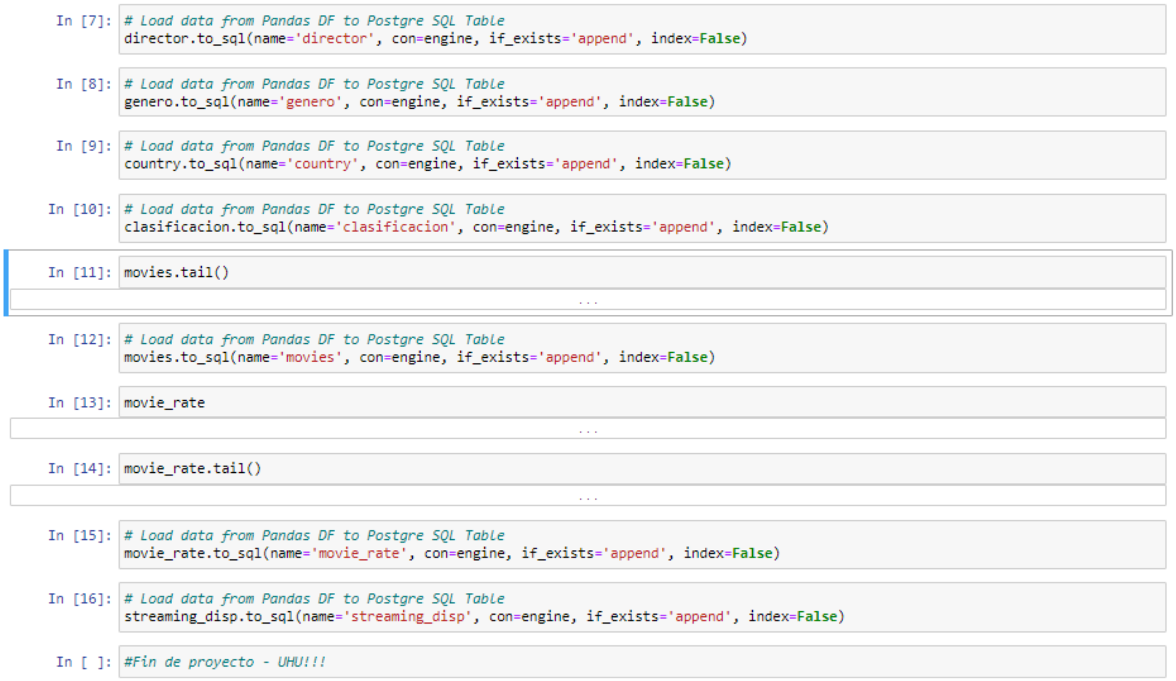
Loading:

We have use PostgreSQL to make a new database with our new tables and their relation references this way you can check the tables of our data base more easy and you can read only the information that you want using queries.

Making the shell scheme:

```
DROP TABLE IF EXISTS movies CASCADE;
DROP TABLE IF EXISTS movie_rate CASCADE;
DROP TABLE IF EXISTS streaming_disp CASCADE;
DROP TABLE IF EXISTS director CASCADE;
DROP TABLE IF EXISTS country CASCADE;
DROP TABLE IF EXISTS genero CASCADE;
DROP TABLE IF EXISTS clasificacion CASCADE;

CREATE TABLE director (
    director_ID SERIAL PRIMARY KEY,
    director_name character varying(250) NOT NULL
);

CREATE TABLE genero (
    genre_ID SERIAL PRIMARY KEY,
    genre_name character varying(250) NOT NULL
);

CREATE TABLE country (
    country_ID SERIAL PRIMARY KEY,
    country_name character varying(250) NOT NULL
);

CREATE TABLE clasificacion (
    clasificacion_ID SERIAL PRIMARY KEY,
    clasificacion_name character varying(250) NOT NULL
);

CREATE TABLE movies (
    movie_ID SERIAL PRIMARY KEY,
    movie_name character varying(250) NOT NULL,
    director_ID integer NOT NULL REFERENCES director(director_ID),
    genre_ID integer NOT NULL REFERENCES genero(genre_ID),
    budget FLOAT NOT NULL,
    revenue FLOAT NOT NULL,
    runtime integer NOT NULL,
    year integer NOT NULL,
    country_ID integer NOT NULL REFERENCES country(country_ID),
    clasificacion_ID integer NOT NULL REFERENCES clasificacion(clasificacion_ID)
);

CREATE TABLE movie_rate (
    rate_ID SERIAL PRIMARY KEY,
    movie_ID INT NOT NULL REFERENCES movies(movie_ID),
    imdb_rate FLOAT NOT NULL,
    tomatoe_rate FLOAT NOT NULL,
    imdb_votes FLOAT NOT NULL,
    imdb_votes_count integer NOT NULL,
    popularity FLOAT NOT NULL,
    movie_rate FLOAT NOT NULL
);

CREATE TABLE streaming_disp (
    streaming_ID SERIAL PRIMARY KEY,
    movie_ID INT NOT NULL REFERENCES movies(movie_ID),
    netflix integer NOT NULL,
    hulu integer NOT NULL,
    prime_video integer NOT NULL,
    disney integer NOT NULL
);
```

Loading the data:



```
In [7]:  # Load data from Pandas DF to Postgre SQL Table
         director.to_sql(name='director', con=engine, if_exists='append', index=False)

In [8]:  # Load data from Pandas DF to Postgre SQL Table
         genero.to_sql(name='genero', con=engine, if_exists='append', index=False)

In [9]:  # Load data from Pandas DF to Postgre SQL Table
         country.to_sql(name='country', con=engine, if_exists='append', index=False)

In [10]: # Load data from Pandas DF to Postgre SQL Table
         clasificacion.to_sql(name='clasificacion', con=engine, if_exists='append', index=False)

In [11]: movies.tail()
         ...

In [12]: # Load data from Pandas DF to Postgre SQL Table
         movies.to_sql(name='movies', con=engine, if_exists='append', index=False)

In [13]: movie_rate
         ...

In [14]: movie_rate.tail()
         ...

In [15]: # Load data from Pandas DF to Postgre SQL Table
         movie_rate.to_sql(name='movie_rate', con=engine, if_exists='append', index=False)

In [16]: # Load data from Pandas DF to Postgre SQL Table
         streaming_disp.to_sql(name='streaming_disp', con=engine, if_exists='append', index=False)

In [ ]:  #Fin de proyecto - UHU!!!
```

Challenges and conclusion

This was a fascinating project as it was our first encounter that we can do analysis and extraction from data in the real world and not just from academic content that is previously checked and reviewed for us the students. We think that things same really simple at the beginning but then you realize that you have make a mistake in the cleansing and transformation, that is a really common mistake that happens in the task of reviewing data. We found that we have need to make many adjustments to have the data according to our original design and that is not simple task as you need to clearly understand what you have, what you want and how to reference correctly the data to achieve that design that you have started with. Also, we found that there is much more extend to our analysis and that we can still contribute more to our database perhaps information from the directors, have more ratings from other sites etc... In conclusion ETL is a really powerful tool that we can use to analyze any data that we want from the web, but it only can be achieved if you put a lot of effort in understanding and cleaning your data.

References

Bhatia, R. (2020, May 22). Movies on Netflix, Prime Video, Hulu and Disney+. Retrieved July 19, 2020, from https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney

Regodee. (n.d.). Regodee: Where to Stream Movies &amp; TV Shows on Every Service. Retrieved July 19, 2020, from https://reelgood.com/

Shakir, J. (2018, March 24). TMDb Movies Dataset. Retrieved July 19, 2020, from https://www.kaggle.com/juzershakir/tmdb-movies-dataset

TMBD. (n.d.). The movie data base. Retrieved July 19, 2020, from https://www.themoviedb.org/?language=es