IMDb Top 250 Movies ETL Project: Analytics Write-up

SMU Data Analytics Bootcamp | Project 2

Group 3: Alyssa DiFurio, Garrett Kidd, Vanessa Martinez, and Japhet Mwamba

December 28, 2022

I. Introduction

Movies and films have been around for over 100 years and are a part of almost everyone's lives. Many people go to IMDb to look up different aspects of movies as well as contribute their own personal ratings and reviews. IMDb is an online database of information related to films, television series, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews. For this ETL project the top 250 movies according to IMDb were scraped to conduct an analysis regarding genres, Motion Picture Association of America ratings, language and countries of origin. The majority of the work for this project was completed using Python Pandas and Jupyter Notebook.

II. ETL

Extract

The first part of our analysis involved the extraction of data. We chose to web scrape the IMDb page for the Top 250 movies of all time, according to reviewers and critics. This page included a lot of information that would be useful for us, but in order to streamline our extraction process, we elected to web scrape only the IMDb Movie ID for each of the top 250 movies. Within the Jupyter Notebook, we wrote a few lines of code that used Selenium to connect to a headless browser, and BeautifulSoup to identify the Movie ID objects on the page. We wrote a For Loop in Python to read through the table on the web page, and compiled a list and then a data frame of the top 250 movies with titles and Movie ID.

Next, we utilized the OMDB API to gather the rest of the data associated with each of the movies, such as run time, director, IMDb Rating, MPAA rating, year released, etc. Using the list of movie ID's and another For Loop in our Notebook, we were able to construct another data frame that contained all of the data pertinent to our analysis. With this data in hand, and unedited, we then proceeded to the Transform step in the ETL process.

Transform

The data frame we constructed with all of the additional data included all columns in the Object data type. Because of this, the first step in our data transformation required us to cast the data into more appropriate data types. Columns such as Year, Metascore, IMDb Rating, IMDb Votes, and Box Office (revenue) were transformed into either integer or float data types. Some additional cleaning involved stripping the figures of characters such as dollar signs and commas. Next, we identified several columns that weren't pertinent to our analysis, and dropped these from the data frame altogether. Additionally, we replaced all null or "N/A" values with zeros (0) to allow for the entire columns to be cast to a single data type (some of the movies were very old and did not have figures for box office revenue). All of these transformations were accomplished with Pandas.

Further data transformations included simplifying some of the columns. Data for Language, Country, and Genre included multiple values (English, French for Language, United

States, France for Country, Drama, Romance for Genre). We elected to strip the secondary data in these columns and simplify these attributes to only the first listed variation. We also did some additional research to combine some of the attributes within single columns, for example MPAA ratings included the typical R, PG-13, PG, and G, but also TV-MA, GP, “Approved”, and more. We found these to be either antiquated movie ratings for older movies or ratings that were synonymous with the current MPAA ratings scale. These data types were replaced with their appropriate version. One of the final steps in transforming the main data frame with all of our movie data required us to rename the column titles so that they aligned with how they were referred to in the Postgres Database.

With our main data frame cleaned and transformed, we began the process of breaking out four new data frames that would serve as separate primary key tables in our database. We created simple data frames for Genre, MPAA Rating, Country, and Language. The process was relatively straightforward, involving several Pandas data transformations such as identifying only the unique values per category, and then casting these to separate data frames. The next step involved loading these single category data frames into our database with the .to_sql() function, and then pulling them back into our Notebook using the .read_sql() function. Doing this would allow us to have data frames with single categories and corresponding unique ID values for each category attribute.

The final part of the Transform process required us to merge our primary key / unique category data frames with our main data frame, and also assign the new foreign key ID value to the appropriate movie, representative of its original “spelled-out” value (ex. Genre type Drama =

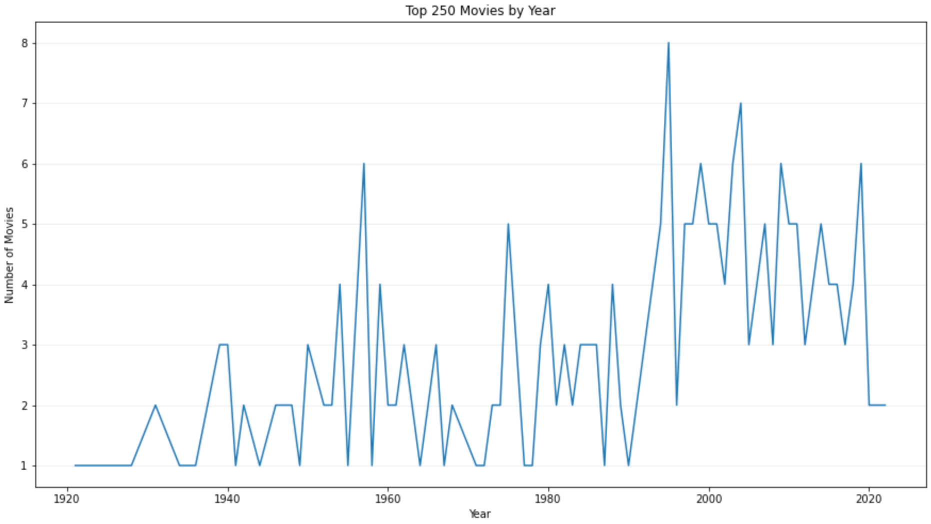
1). Finally, with the four foreign key types merged and replacing the original data values in our main data frame, we were able to load this table to our Postgres database.

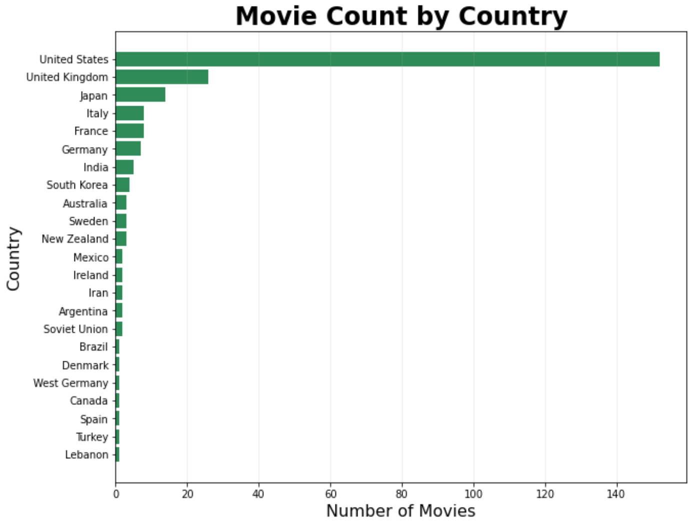
Load

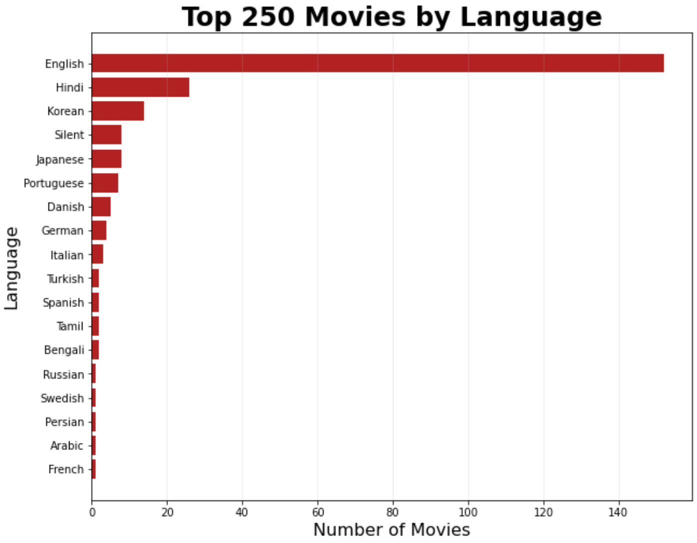
To create the tables and schema for our database, we used https://www.quickdatabasediagrams.com. Five tables were made which included movie information, MPAA rating, genre, language, and country. Movie information was the largest table and consisted of columns including movie title, year it was released, directors, how much revenue was generated at the box office, the length of movie in minutes, actors/actresses in the movie, genre, rating, language, country, IMDb rank, IMDb rating, IMDb votes, and Meta score. Each table was given an id and last updated column, and foreign keys were assigned to the film rating, genre, language, and country lookup tables. In order to successfully load all the tables into Postgres, id's were created in our notebook for genre, language, country, and film ratings which were then merged into the main data frame table. We created a connection to the local Postgres database, inspected the tables and columns with an inspector engine, and added the completed tables into Postgres using the .to_sql function. From there we were able to run queries and further analyze the movie data.

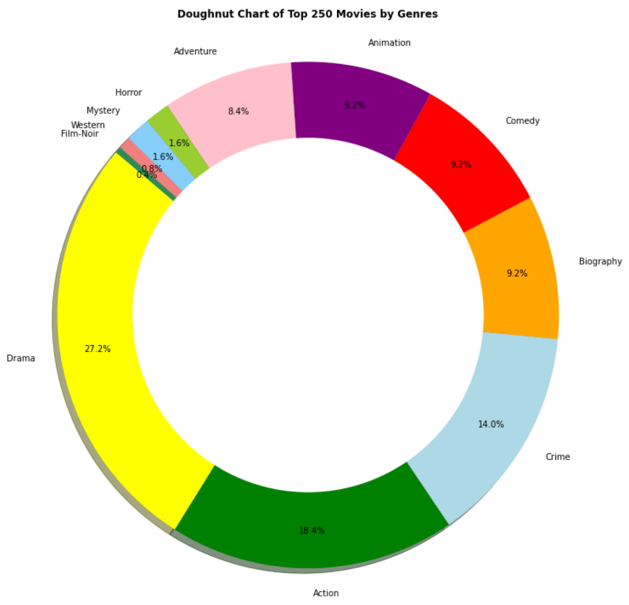
III. Visualization

After uploading the pandas tables into our Postgres database we were able to run a handful of queries to gather comparisons of our data. Our queries were designed to compare the main statistics of the top 250 data including the genre division, runtime division, country of origin, count of directors - tables that would allow us to compare primary features enjoyed by critics and viewers. Once we wrote out the queries in our Jupyter notebook, we were able to utilize pandas, seaborn, and matplotlib to create charts of each query. We utilized bar graphs, donuts, line charts, and a violin plot to show our various queries. This allows an outside viewer looking at our initial data the ability to look at our visualizations to provide shorter stories and answers related to the data.

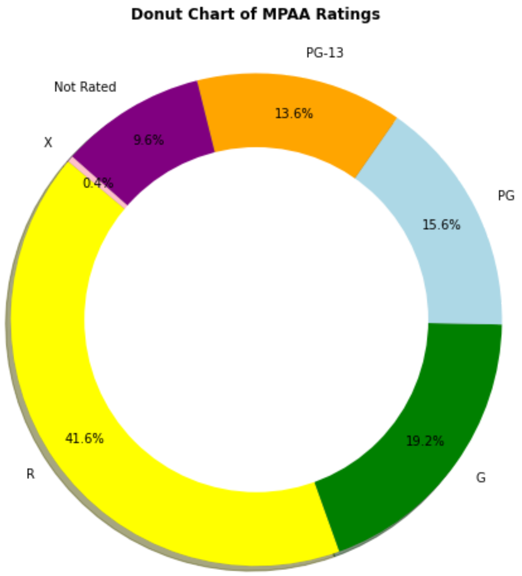




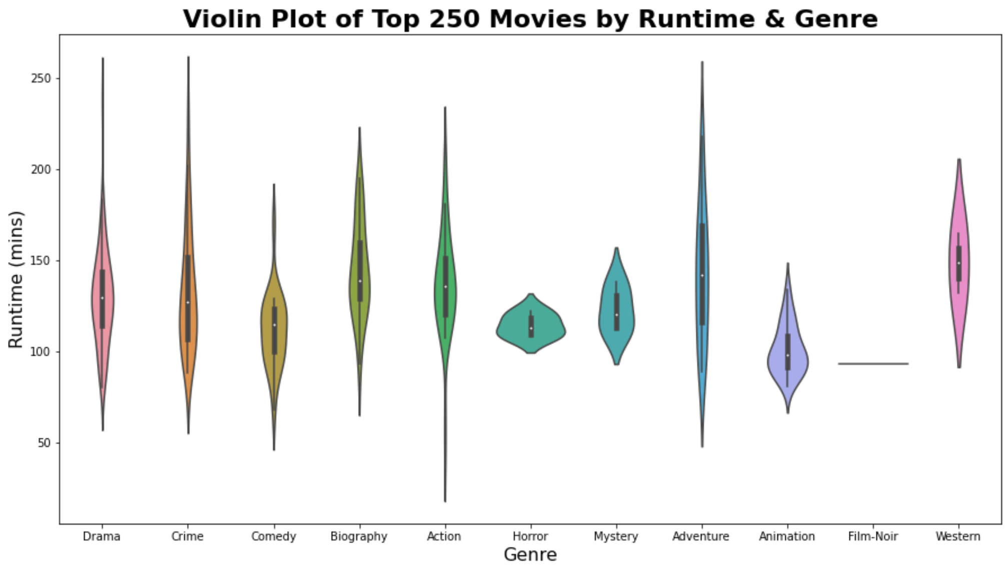




Doughnut Chart of Top 250 Movies by Genres



Donut Chart of MPAA Ratings



Violin Plot of Top 250 Movies by Runtime & Genre

IV. Analysis

Our analysis of the IMDb Top 250 movies included several questions posed to understand what the data could tell us. First we examined the top movies making the list by year over time. Our line plot showed that since around 1920, there have been many peaks and valleys for movies making the list, and that there is a slight trend of newer movies since 1980 being more likely to make the list. The late 1990's saw the highest number of movies make the top 250. A second question we posed was what were the most popular genres that made the list. We found that Drama, Action, and Crime movies were the top three most popular genres in that order. Perhaps movies that are of the Drama genre are more likely to make the top 250 because they are naturally more serious movies, and more likely to be rated highly by critics. A third question we asked was, what is the breakdown by country of origin for the top 250? Overwhelmingly, the United States was the country of origin for many of the top 250. Interestingly though, there were movies that came from 22 other countries, with the United Kingdom and Japan taking the second and third spot, respectively. Also relatedly and unsurprisingly, English was far and away the most common language for movies in the top 250. However, somewhat surprisingly, Hindi and Korean were the second and third most common languages of movies.

Our group also examined the breakdown of MPAA ratings for the top 250. "R" rated movies were number one, with over 40% of the top 250, but then "G" rated movies were the second most common rating. This could suggest that the most highly rated movies commonly fall into two buckets: one for more "adult" and serious movies, and another for more "kid-friendly" and perhaps "feel-good" style of movie. Finally we constructed a Violin Plot of the Top 250 movies by runtime and genre. This plot visualized that Drama, Crime, Comedy, Biography, Action, and Adventure all varied widely in their runtime. The plot also showed that Horror, Mystery, Animation, and Western type movies had much smaller variation in runtime. Overall, many types of movies are well represented within the IMDb Top 250 list. But, it appears that "R" rated Dramas from the United States, spoken in English and running in excess of two hours are the most likely type of movie to find itself in this top 250 list.

V. Limitations and Further Research

Most of the movies in the IMDb top 250 list are considered to be some of the greatest movies ever made, although some might be on the list due to IMDb user votes. Limitations include not using more websites to see different movie rankings or movie information. Other movie websites, such as *Rotten Tomatoes,* also have lists of top rated movies. However they are categorized more specifically and don't have a general "best movies of all time" list comparable to the IMDb top 250. Further research might include scraping multiple movie websites to collect more data on films, to compare and analyze with the data we have already collected. This information could include data regarding films from specific time periods, specified film genres, or other film reviews/rankings.

VI. Conclusion

At the end of this project we were able to utilize our scraped data and our query tables to visualize different information. The top 250 list contained a large variety of movies in terms of style, year, director, and runtime, showing no significant way to track movies that would make it into this list. A cross comparison of other sites and other critic or public rating systems would have been an interesting way to compare the information a little more. Overall, our code ran smoothly and our notebooks and queries came error free, allowing us to easily pull and display our data into visuals for analysis.

VII. Works Cited

IMDb Top 250 Movies List - https://www.imdb.com/chart/top/

OMDB API website - https://www.omdbapi.com/

Quick Database Diagrams website - https://www.quickdatabasediagrams.com/

IMDb Wiki - https://en.wikipedia.org/wiki/IMDb

Rotten Tomatoes website - https://www.rottentomatoes.com/