A picture containing clock, knife

Description automatically generated

Netflix ETL Data Analysis Report

Gredel, CJ, and Angie

The purpose of this report is to use data analysis to allow our business to make the best decisions about which movie offerings to invest in. Our profit depends on our subscription numbers. One vital component to increase subscriptions is to provide a strong reason for the consumer to pay a monthly fee to have access to our offerings. More information about the movies that we currently offer will help us make better predictions about potential movies and choices for which movies to acquire in the future. This initial data analysis will focus on movies produced in the United States.

Extract, Transform, Load

The current Netflix database (source: csv file) was filtered to movies produced in the United States. Two popular databases; OMDb API and TMDB were used to extract movie data. We wrote an API call in Jupyter Notebook for both data bases and used the ‘title’ of each movie in our csv file as the search criteria. This ensured that we only get data for movies that Netflix currently offers to our subscribers.

A screenshot of a computer

Description automatically generatedWe prepared a database in PgAdmin called movie\_db to store the data extracted from the API calls. The relational database was designed with the “title” of the movie as the primary key to allow for the data to be queried, updated, or dropped easily from PostGres in the future *(Figure 1).* Three tables were created for the Netflix, OMDb, and TMDB data in PostgresSQL using PgAdmin4. After inspection of the data sets, only columns of interest were defined for the database, and datatypes were chosen based on data from each source. From Jupyter Notebook the data was cleaned further before loading to the SQL database. Columns for each dataframe were renamed appropriately to match the destination tables in the database. The box office revenue column in the OMDb dataframe was converted from a string into a decimal which was required for use in data analysis. After data cleaning of the dataframes in Jupyter Notebook the dataframes were loaded into the prepared SQL database.

Figure 1. Relational Database Schema.

Data Analysis

We then cleaned the data and prepared each column to determine if there are any correlations between the viewer rating, votes, vote average, popularity, box office revenues, and the metascore of each movie. Using python (with the following libraries: pandas, sqlaclhemy, scipy.stats, numpy, sklearn, matplotlib.pyplot) we transformed/cleaned the data such that scatter plots, subsequent linear regression analysis and r-squared values could be performed. The following is a step-by-step outline of the data preparation and clean-up performed for each scatter plot:

1. Performed an outer merge of all three dataframes on the title column using pd.merge.
2. Upon examination of the data we then dropped duplicates using drop\_duplicates on the “title” column.
3. NaN values were dropped for Box Office Revenue and IMDB Rating columns and a plotted.
4. NaN values were then dropped for the TMDB Vote Count and plotted against Box Office Revenue.
5. The IMDB Rating and TMDB Popularity were then compared in a scatter plot.
6. The Metascore vs. IMDB Rating and the Metascore vs. TMDB Vote average were both plotted as well.
7. Linear regression and r-squared values were calculated for all plots.

Results

The relational database diagram is shown in Figure 2.

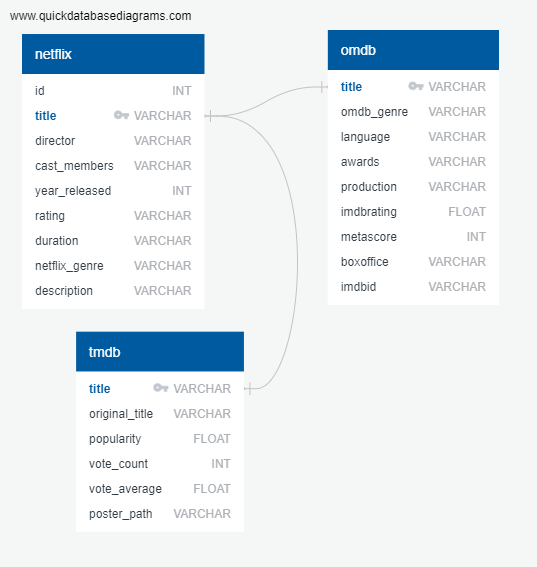
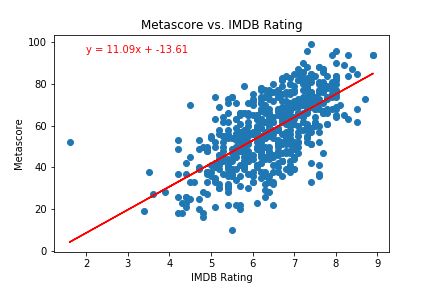
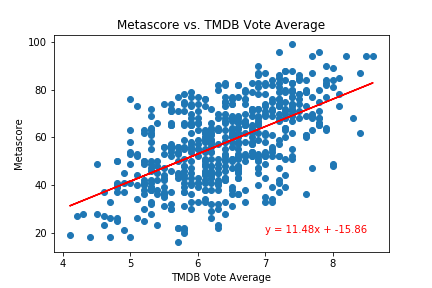
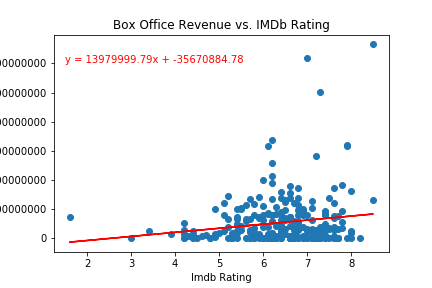
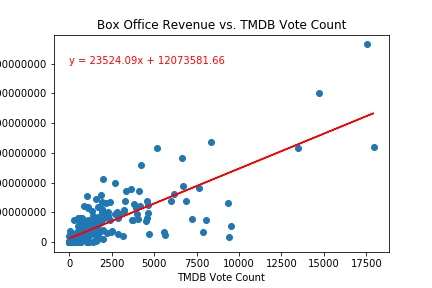
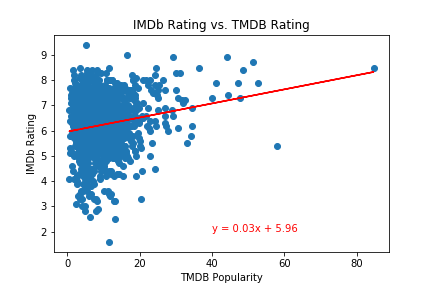


Figure 2. Relational Database Schema Diagram.

Below are the scatter plots and regression analysis for each area of interest:







r2 Values:

Metascore vs. TMDB Rating r2=0.331

Metascore vs. IMDB Rating r2=0.412

IMDB vs. TMDB r2=0.032

Box Office vs. TMDB r2=0.619

Box Office vs. IMDB r2=0.027

Sources:

<https://www.kaggle.com/shivamb/netflix-shows> (csv download)

<http://www.omdbapi.com/>

<https://www.themoviedb.org>