ETL Project:
Movie Box Office Revenue vs Oscars Won 1939-2019

The Code Bots

Ashton Smith
Carolina Segovia
Claudia Canamas-Donnelly
Margaret Wharton

September 11, 2021

Extract: The Original Data Sources

We used Kaggle to find two datasets containing data about various movies. The first, 'All Time Worldwide Box Office' contains movies, ranked by their box office performance, as well as the box office revenue separated into worldwide, domestic, and international columns. This dataset contained about 700 rows to start. The second dataset is called 'The Oscar Award, 1929-2020.' This dataset contains all Oscar nominated movies, with the year of the ceremony, the year the movie was made, the nomination category, and whether or not the movie won the award. This dataset contained around 10,000 rows to start. Both of these datasets were in CSV format.

Links to data sources are as follows:
https://www.kaggle.com/kkhandekar/all-time-worldwide-box-office
https://www.kaggle.com/unanimad/the-oscar-award

Transform: Cleaning the Data

To prepare for cleaning our data, we loaded both CSV files into a Jupyter Notebook file using Pandas. Cleaning up the datasets took several steps as follows:

- Standardize all movie titles, since we will merge on this column later, by changing all to lowercase
- Determine locations of rows with null values and remove them
- We noticed that the Box Office dataset has a date range of 1939 to 2021 and the Oscars dataset has a date range of 1927 to 2019. In order to standardize the data, we dropped all rows containing dates prior to 1939 in the Oscars dataset and dates after 2019 in the Box Office dataset
- Merge the dataframes on 'film' column
- Drop unnecessary columns from the merged dataset: 'year_film', 'year_ceremony' and 'ceremony'
- Add new column 'Nomination Count' in order to more effectively show how many Oscar nominations a movie has received
- Create a second 'Movies' dataframe with a unique ID for each movie - since there is a possibility of two or more distinct movies having the same title, we create a unique ID by concatenating film title and year and assigning a movie ID to each
- We dropped 'film' from our Film Performance table and kept Movie ID because Movie ID is unique, and so that our Movies table would have a purpose (looking up film title)
- Gave each nomination row a Nomination ID for primary key of larger table

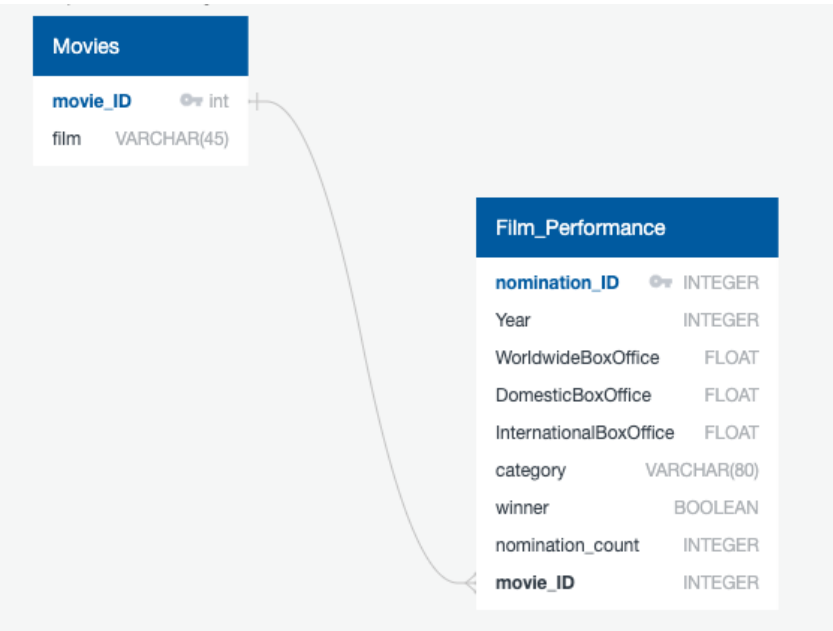
- Export two final CSV files: ‘Movies’ dataframe and ‘Film Performance’ dataframe

Load: The Final Database

For our final database, we created a structured database using SQL. We chose a structured database because we wanted it to be easy for users to run queries and find the movies they are looking for.

We used a .sql file to define the schema in pgAdmin, then used SQLAlchemy from a Jupyter Notebook to fill in the tables from our cleaned dataframes.

Our database includes two tables: Movies and Film Performance. The movies table includes a list of every movie in the database that can be easily searched by name in order to quickly determine if data for a particular movie is in the database. This table also includes a movie_id column, which is also used in the Movie Performance table as a foreign key. This will make performing queries to obtain data on a specific movie easy for users of this database.



ERD of our database

Uses for Our Database

A database containing box office data as well as Oscar award winners could be useful to anyone looking to analyze the relationship between a movie's box office and award show performance, since both metrics are highly valued in the film industry, but may not necessarily be correlated indicators.

We thought it would be interesting as a potential future project to perform analysis of popular movies and determine if box office revenue and/or Oscar nominations are actually indicators of a movie's mainstream success.