Building a Movie Recommendation System

Introduction

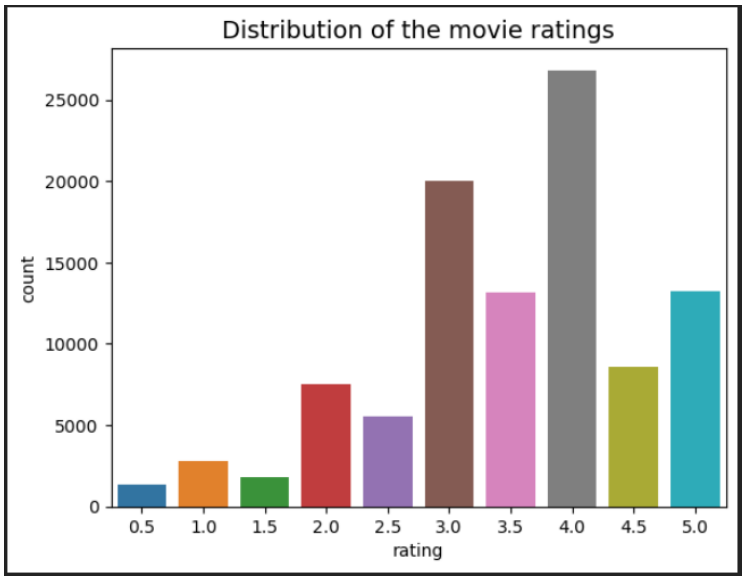
Recommendation systems are like prediction systems designed to predict what the users may like when there are various options to choose from. Some classic examples are item recommendations from Amazon and movie recommendations from Netflix, among others. Recommendation systems are essential for businesses as they can keep customers on the platform longer than originally intended. This can be very relatable while shopping on Amazon. How the customers get lured into buying more when they see recommendations such as "Customers who bought this also bought this."

Source of the Data:

In this project, a movie recommendation system is built using the datasets from the Movie Lens website (MovieLens, 2013). As there are different flavors of datasets available on the website, to keep things simple, a small version of the dataset is used. Two datasets are used in this project, one being “movies.csv” which contains the details of movies such as the movie ID, title, and genre. The other file is "ratings.csv", as the name suggests, includes the movie ratings' details and the user ID of the users that rated the movies.

Data Exploration and Feature extraction:

As an initial step of exploring the dataset, exploratory data analysis is performed to understand the data distribution better. Some of the analyses performed in this step include finding the unique number of movies in the ratings and the movie datasets, the unique number of users, the average number of ratings for the movies, the most popular genre of movies, etc.



A histogram of the distribution of the movies by ratings and a Bar plot of the movie distribution by Genres are also plotted to understand the data visually.

Figure 1: Distribution of the movie Ratings

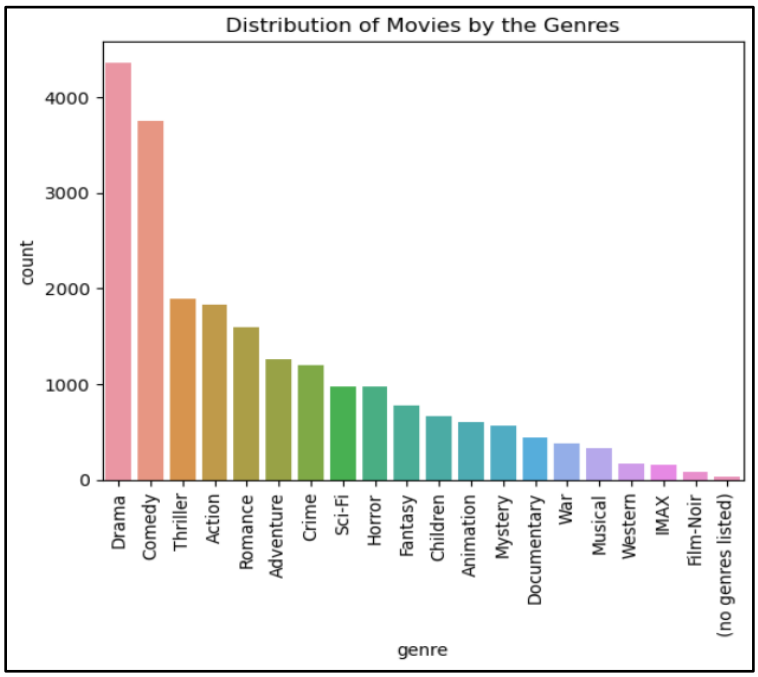


Figure 2: Distribution of the movies by Genre

Further analysis is done by joining the two datasets to find the movies that are rated the highest and lowest. While doing this analysis, it is found that merely calculating the mean ratings of the movies may not be enough as some movies may have one or two 5.0 ratings and can be rated highest, while some popular movies with 200 ratings with an average of rating of 4.0 can be rated as less popular. To avoid this problem, the Bayesian average method is employed to find the average rating using the model weights.

Recommendation system

Two methods were used to build the movie recommendation system in this project as discussed in this section.

Pairwise Correlation Method:

The first one was with the simpler pairwise correlation method, where the ratings and the movie dataset were merged, and the resulting dataset was reshaped to contain the user ID and movie titles with the user ratings. Then pairwise correlation was then computed for the movie of interest and after removing nulls, it was merged with the dataset that contains the count of movie ratings. The movies with more ratings and with higher correlations were chosen and displayed to the user. One big downside of this approach is that the outcome is based on the number of reviews that the movie received, though on the positive side, it is easier to deploy. (Nair, 2019)

K-Nearest Neighbors algorithm using Cosine distance Method:

In the second approach, first a sparse matrix is calculated on the user ratings of the ratings dataset. Then a function is built using K-nearest neighbors to find the similarity using the cosine distance method. The function sorts the results based on the similarity distance and returns the

top 10 movies. Though this method can be slightly complicated, it can be more efficient with the results compared to the first method. (says, 2020)

Observations and Conclusion

In this project, the recommendation system was built using two different methods. For illustration purposes, the algorithms were tested with the horror movie “The Shining” released in 1980. The recommendation results are more satisfactory with the KNN method compared to the correlation method. However, as this was built only with a small subset of movies, the project can be expanded further by testing on bigger datasets to compare the performance.

References:

MovieLens. (2013, September 6). GroupLens. https://grouplens.org/datasets/movielens

Nair, A. (2019, September 25). How To Build Your First Recommender System Using Python & MovieLens Dataset. Analytics India Magazine. https://analyticsindiamag.com/how-to-build-your-first-recommender-system-using-python-movielens-dataset/

says, D. S. (2020, November 9). Build A Movie Recommendation System on Your Own. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/