CAPSTONE PROJECT

Netflix

RECOMMENDATION SYSTEM

Sekhar Chandra Padhi

Table of Contents :

PROBLEM STATEMENT:

Customer Behaviour and its prediction lies at the core of every Business Model. From Stock Exchange, e-Commerce and Automobile to even Presidential Elections, predictions serve a great purpose. Most of these predictions are based on the data available about a person's activity either online or in-person.

Recommendation Engines are the much-needed manifestations of the desired Predictability of User Activity. Recommendation Engines move one step further and not only give information but put forth strategies to further increase users' interaction with the platform.

In today's world OTT platform and Streaming Services have taken up a big chunk in the Retail and Entertainment industry. Organizations like Netflix, Amazon etc. analyse User Activity Pattern's and suggest products that better suit the user needs and choices.

For the purpose of this Project we will be creating one such Recommendation Engine from the ground-up, where every single user, based on there area of interest and ratings, would be recommended a list of movies that are best suited for them.

PROJECT OBJECTIVE:

We are building a tailored Recommendation Engine from scratch. This engine will suggest a personalized list of movies to each user, carefully selected based on their interests and past ratings.

DATA DESCRIPTION:

PROJECT CONSIST OF 2 FILES

1. “combined.txt’ file which contains:-

- Customer ID : It consists of Unique ID of the Customer.
- Ratings : It tells what where the ratings given by customer to the movie. Ratings are on a five-star (integral) scale from 1 to 5.
- Movie ID : It consists of Unique ID of each movie.
- This file consists of 24058263 rows and 2 columns

2. “MOVIE TITLES “ file contains:-

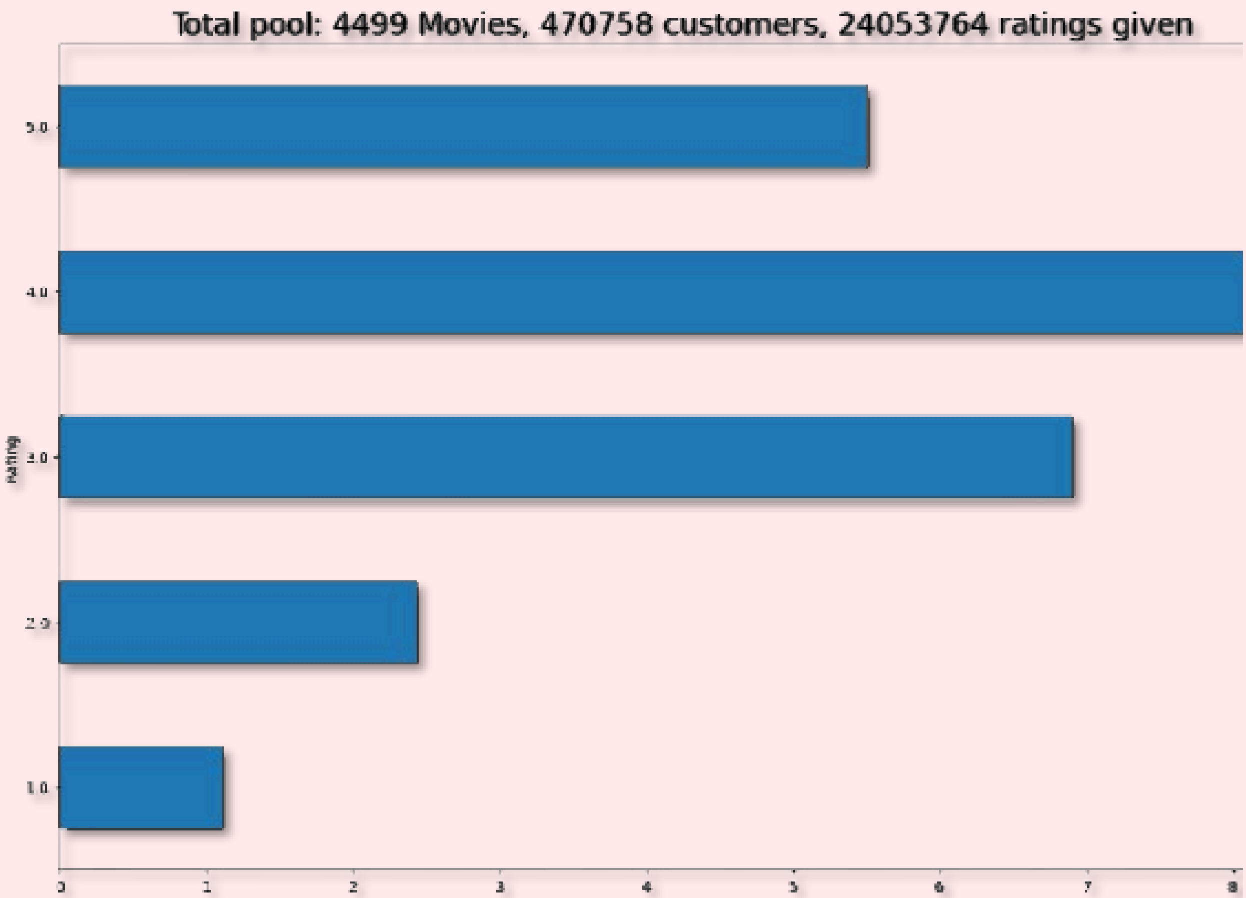
- MOVIE ID: It consists of Unique ID of each movie.
- MOVIE NAME: IT consist of Names of The Movie.
- YEAR OF RELEASE: Consist of Year of release for each year.
- This file consists of 17770 rows and 3 columns

DATA PRE-PROCESSING STEPS AND INSPIRATION:

- **Decoding Data:** Since the data is encoded, our first step is to decode it.
- **Pre-processing:** Before any further steps, the data needs thorough pre-processing.
- **Assign Column Names:** Initially, the columns lack names, so we will assign appropriate names to them.
- **Identify Null Ratings:** Beneath each movie ID is a customer ID who has rated the movie. We notice that the ratings column shows null wherever there's a movie ID.
- **Custom Loop for Null Values:** We can create a custom loop to utilize the null values in the ratings column, identifying the range of movie IDs for which customers have given ratings.
- **Set Cutoff Values:** Establish cutoff values for the dataset, such as a minimum rating count of 908 for movie IDs and a minimum of 36 ratings from customers, to ensure only relevant data is considered for the model.
- **Change Data Types:** Convert the data types of certain columns from object to float for better processing.
- **Visualization:** To gain insights from the data, we used bar plots and pie charts to visualize various aspects of the dataset.

DATA INSIGHTS:

- Data consist of 4499 movies in the data set and total Customer count is 470758.
- Total number ratings given by customer 24053764.



CHOOSING THE ALGORITHM FOR THE PROJECT:

- The Singular Value Decomposition algorithm was chosen because it is very effective in collaborative filtering.

- Collaborative filtering depends on user-item interactions to make recommendations and SVD specially is well-suited for this task.

- It factorises the user-item interaction matrix into three matrices, it also captures latent features of users and items.

MOTIVATION AND REASONS FOR CHOOSING THE ALGORITHM:

- SVD is effective for managing missing data, which is a common issue in recommendation systems. Its ability to reduce dimensionality makes it an ideal technique for large-scale recommendation systems.

- Additionally, SVD excels at preserving critical information while simplifying complexity. Consequently, it is widely adopted in real-world recommendation systems and collaborative filtering applications.

ASSUMPTION:

- Although SVD decomposition can be efficiently computed, in practical scenarios, computing the full SVD decomposition might become computationally expensive.
- SVD-based models assume that the latent features extracted are meaningful, even though these features might not always have clear significance.
- User preferences are assumed to remain constant over time, but in reality, these preferences can change.

MODEL EVALUATION AND TECHNIQUES:

- The model's evaluation results using Root Mean Squared Error (RMSE) for the Singular Value Decomposition (SVD) algorithm over 3 folds are presented.
- The RMSE values range from approximately 0.966 to 0.969. Metrics suggest that the SVD algorithm is making reasonably accurate predictions, given the relatively low error rates.

INFERENCES FROM THE SAME:

- The SVD model's prediction performance was excellent, as indicated by the RMSE results.

- The training time was somewhat slow, so I have taken a sample of the data to expedite the process. Despite this, the model shows potential for handling large datasets with further optimization.

- There was variability in the testing process duration, indicating that further strategies could be explored.

FUTURE POSSIBILITIES:

- To further enhance the system, exploring alternative algorithms could be beneficial.
- Incorporating user demographics and genre preferences as features might also improve recommendation accuracy.

CONCLUSION:

- This project successfully implemented a movie recommendation system for the Netflix dataset using Singular Value Decomposition (SVD). The model achieved a reasonable RMSE score of [insert RMSE value] during cross-validation. The recommendations generated for a sample user (ID 1331154) appeared relevant and aligned with their past ratings.

- Deploying this system would involve integrating it with a user interface and a database of movie information. Real-time updates and handling large-scale user data would be key challenges.

REFERENCES :

Dataset-
Netflix project/Copy of combined_data_1.txt.zip,
Netflix project/Copy of movie_titles.csv