Age and Context Sensitive Entertainment Recommendation System for Families

Abhinn Yadav 2020013 | Dhruv Mishra 2020296 | Hunar Kaur 2020303 | Sadhvi Bhan 2020325

Abstract

With the hectic lifestyle, the time a family spends together has decreased in today's world. Rather than spending quantity family time, quality family time has become essential. Hence, many people have started with family movie night rituals in their houses. However, families have to spend much time finding movies that all family members prefer and that are age-appropriate. Our project aims to create a movie recommendation system that considers the family members' preferences, past activities, and age and recommends an age-appropriate movie for the family to watch.

Introduction

As fun and easy as it is to binge your favorite shows with your loved ones, choosing what to watch is challenging. With so many genres to choose from, different preferences among individuals, and choosing age-appropriate content, narrowing down on the top movies and shows can be complicated.

Our Ml model is a movie recommendation system for a group of individuals which recommends the top movies based on the above criteria. Our dataset contains information about the users and the ratings given by them to make suggestions.

Literature Survey

Broadly, recommendation-based systems involve three different major types of Filtering. Content-based filtering metadata about the show presents a list of similar movies. Collaborative Filtering, on the other hand, uses the ratings given by the user to find other individuals who share similar interests. It recommends movies based on this match. Hybrid Filtering combines the power of the two.

1. User Based vs. Item-Based Filtering [1]

In user-based Filtering, users' interests were compared to find similar ones. Pearson Correlation Coefficient was used to determine the similarity between ratings. The higher the coefficient, the more correlated were the two users. An algorithm that found N nearest neighbors and made clusters of neighbors was used for user-based Filtering. Since ratings change from time to time, this method is not static. In Item Based Filtering, LogLikelihood similarity was used. User ratings, once given, do not change. Hence values of similarity would remain constant.

2. Movie recommendation system using collaborative learning [2]

This paper uses a collaborative approach to prescribing movies to others with similar tastes, allowing users to explore more. A web application is implemented that will enable users to rate movies and recommend appropriate movies based on others' ratings. This paper concluded that the content-based recommendation systems work on individual users' ratings, limiting users from exploring more. However, the collaborative learning approach computes the connection between different users and, relying upon their ratings, recommends movies to others with similar tastes, allowing users to explore more.

Dataset

https://drive.google.com/drive/folders/1W6GibRSrN6HujoaYbA2zk_BEQ4vL0-Ef?usp=sharing

Dataset Description For Content and Demographic Filtering:

Our dataset contains the following attributes:

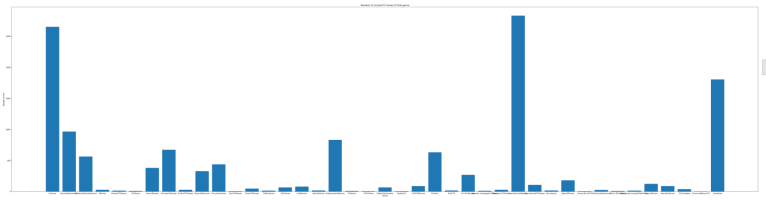
- type: If it is a movie or a TV show.
- title: The current title of the movie.
- director: The director(s) of the movie/TV show.
- cast: The actor(s)/actress(es) of the movie/TV show.
- country: The country in which the movie was released.
- date added: The date on which the movie/TV show was released.
- age_rating: The age rating given to the movie/TV show.
- duration: Total duration of the movie/TV show
- listed_in: Genre(s) of the movie/TV show.
- description: Sentence(s) describing the movie/TV show.
- IMDB: The IMDB rating of the movies/TV show
- TMBD: The tmdb rating of the movies/TV show

Dataset Description For Collaborative Filtering:

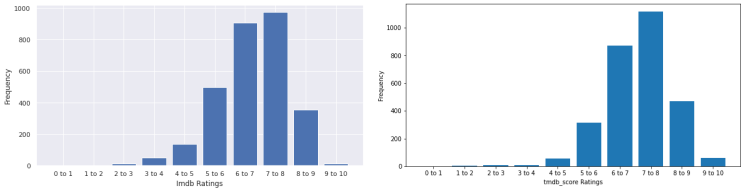
Our dataset contains the following attributes:

- title: The current title of the movie.
- user id: Uniquely identifies the user
- movie id: Uniquely identifies the movie.
- rating: The rating given by the user to a specific movie.

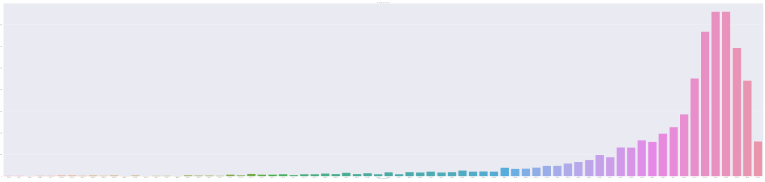
Exploratory Data Analysis:



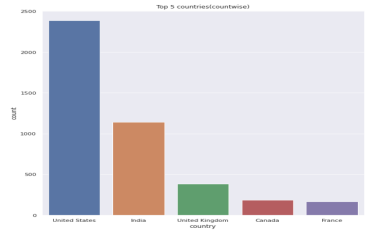
Genres of movies and TV shows. The highest genre comes out to be international movies, the second highest genre is Drama.

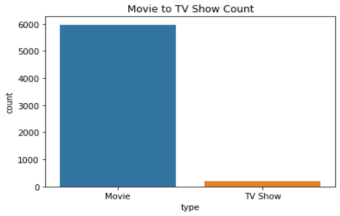


Frequency of the imdb and tmdb ratings of the movies and TV shows

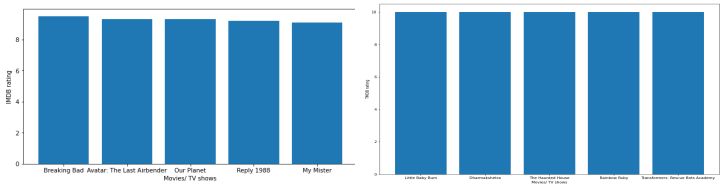


Number of movies released in a year with max in 2016 and 2017

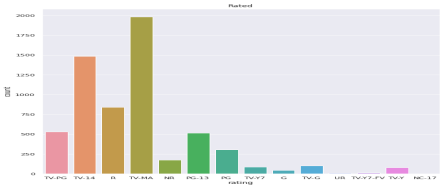




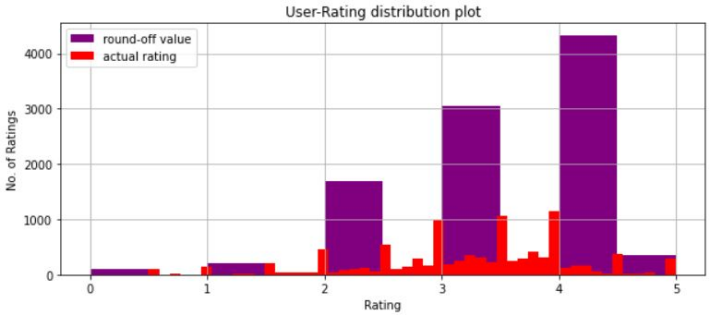
Countries that made the most movies/TV shows. The USA is the highest among all the countries(left graph). The count of movies and TV shows in the dataset. There are more movies than TV shows in our dataset(right graph).



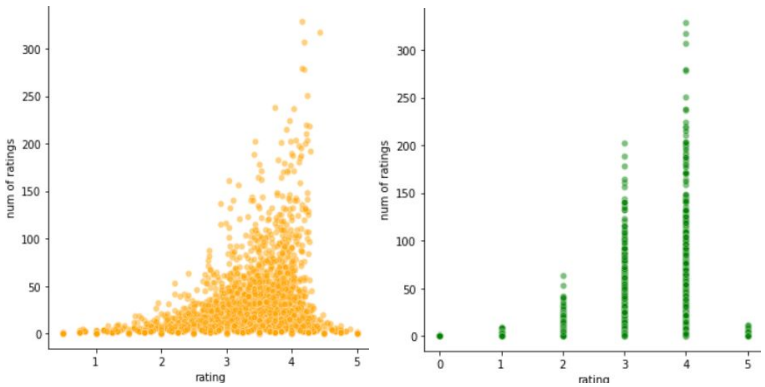
The Top 5 movies/Tv shows with the highest IMDb rating(left) and tmdb rating(right)



Count of the age certification ratings of the movies/TV shows of the dataset. TV shows with a rating of TV-MA(Mature Audience) have the highest count.



A bell shaped plot depicting the ratings and round-off rating given by the users.



(a) (b)

(a)Scatter plot depicting for each movie its average rating(x) vs the number of ratings given by the user (b) Scatter plot after rounding the ratings as in (a)

Preprocessing on Dataset:

Creation of Dataset:

We considered multiple datasets consisting of essential features like movie genres, age rating, and IMDB rating of the movie/ Tv Show apart from other metadata. We observed that after removing null values from these datasets, we needed more data to train our model. Hence we combined two similar datasets based on common columns using outer join to obtain a merged dataset.

On this newly merged dataset, it was observed for the same movies/show. Multiple rows could exist. A new row was created if a particular show/movie existed in both datasets with any column value missing or differing. To correct this, we found the number of occurrences of the same title and their row indices in the merged dataset. We would keep only the first row out of these and drop the rest. If the row lacked some column value, we iterated through the other rows and filled the row to be kept with those values. We later dropped all the duplicate rows.

This gave us a dataset with unique values (for the title column) without any empty cells.

Dataset Modification:

A Show/Movie can have multiple genres. In our dataset, we had 40+ different genres in total. We need to convert string data into numerical form to use it. We applied one-hot encoding on the genres. New columns were created for each genre where column value = 1 indicated the movie was to be listed in that genre, and column value = 0 indicated the movie did not belong.

We also vectorized the description based on which we have predicted movies/TV shows. This is all done so the data is in a numerical format so we can find correlations, as discussed later.

Methodology

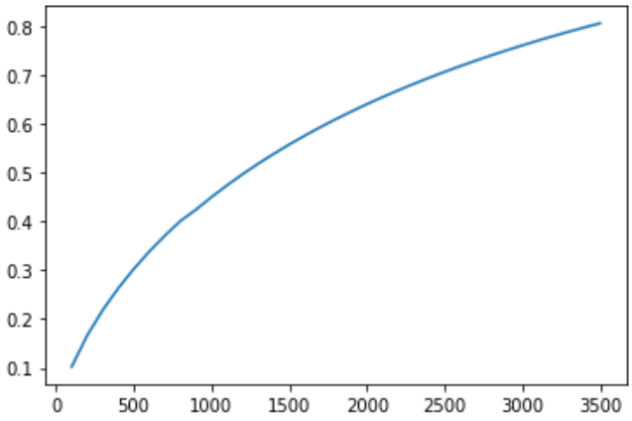
1. Variance vs. PCA Dimensionality

Our initial vector, after conversion of description into a TFIDF matrix, came out to be of size ~ 8809 X 18000. We can not reduce the dimensionality of the 8k rows as they represent the movies in our dataset. We applied PCA on the 18k cols to reduce the dimensionality of the column vector. We tried various values of n ranging till ~3500 and compared the results.

We plotted the variance for different values of n(after PCA). Application of PCA on the data changes that data, but the domain remains unchanged. Variance indicates the amount of information retained after PCA. As expected, the value of variance increases as the value of n increases. From the plot, we have chosen n = 3400 as it preserves 80 % of the information.

```
Components =  3000 ;
Total explained variance =  0.76102
Components =  3100 ;
Total explained variance =  0.77073
Components =  3200 ;
Total explained variance =  0.78012
Components =  3300 ;
Total explained variance =  0.78914
Components =  3400 ;
Total explained variance =  0.79787
```

Value of variance increases as components increase



N components vs Variance Plot

Results comparison after dimensionality reduction

```
Enter the Total Number of Users:1
Enter Your Age:20
Enter Your Movie Preference:Death of Me
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows...
Here's What We Think You'll Like:
        1 Age of Tanks
        2 Nature's Great Events: Diaries
        3 Classic Legends
        4 After Porn Ends 3
        5 Mortified Nation
```

Results for n = 2. Recommendations are not close

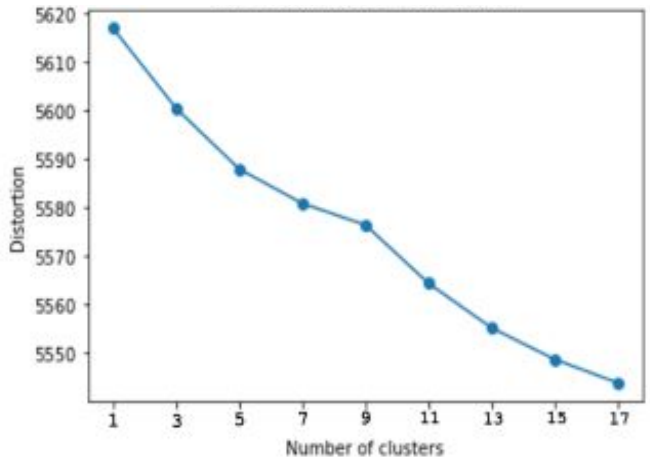
```
Enter the Total Number of Users:1
Enter Your Age:20
Enter Your Movie Preference:Transformers: Cyberverse
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows...
Here's What We Think You'll Like:
         1 Black Mirror
         2 Til Death Do Us Part
         3 Inception
         4 Abby Sen
         5 Apollo 18
```

```
Enter the Total Number of Users:2
Enter Your Age:20
Enter Your Movie Preference:Death of Me
Enter Your Age:20
Enter Your Movie Preference:Shutter
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows...
Here's What We Think You'll Like:
         1 Unsolved Mysteries
         2 The Unborn Child
         3 Malevolent
         4 13 Reasons Why
         5 Intrusion
```

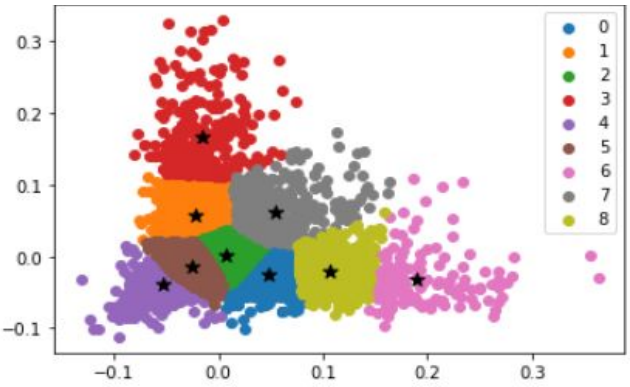
Results for n = 3400. Recommendations are close

2. K-means clustering

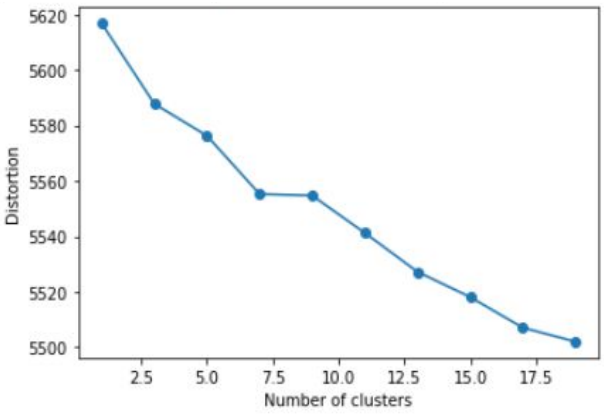
We analyzed the results of k-means clustering on our dataset. We performed for k-means clustering for values of n. We plotted the elbow curve and determined the optimal number of clusters. Plotting the clusters for n = 2 we observed that data is linearly separable in 2D. However when we plotted the clusters for n = 3400, the data was not separable in 2D. However, the data could be separable in higher dimensions. The corresponding elbow curves and cluster plots are given below.



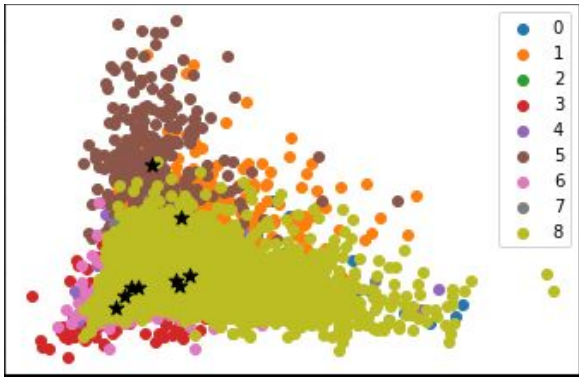
For n = 2 value of k as observed from plot= 9



The corresponding k cluster plot which is separable.



N = 3400, k = 9



Data is not separable if observed in two dimensions using K-Means Clustering.

3. Age filters

As depicted by EDA Graph above, we can see the vast number of genres present among the movies and TV-Shows. We have different categories like TV-MA: strictly for adults, TV-14: for 14 and above, and TV-PG: for 14 and above but with parental guidance. Following this, we updated our dataset to remove ratings unsuitable for the given age group. We decide on suitable ratings by the min and max age present in the group. Min-age is calculated to remove adult movies, while max-age is calculated so PG movies(parental guidance) can be allowed when an adult accompanies the children.

4. Demographic Filtering

Demographic refers to the statistical characteristics of the human population. When a new user joins the system, the model is unaware of his past choices. This situation is termed a 'Cold Start'. In such situations, we can predict the top-rated movies of all time to the user. In our dataset, we had access to both the IMDB and TMDB ratings of a movie. The Pearson correlation between them was found to be 0.57. This is a moderate-high positive correlation, i.e., if one increases, the other is also expected to increase. We defined a new metric to score the movies, which is as follows:

$$=\alpha(\text{IMDB rating}) + \beta(\text{TMDB rating})/(\alpha+\beta)$$

Where α and β are hyperparameters that can be tuned as more data on user preference is calculated

IMDb ratings are more popular hence Imdb ratings were given a higher priority in the new_score calculation.

5. Content-Based Filtering

The idea behind content filtering is that, given a movie, the program must find and recommend a similar movie to the user. To compare similarities between the plots of the movies, we planned to use the movie description column of our dataset.

For the algorithmic computations, we first converted our string description into a vector. We did this by calculating the Term Frequency-Inverse Document Frequency (TF-IDF) vectors of the description. By doing this for all the movies, we got the TFID matrix(Matrix of TFID vectors) for the entire dataset.

However, before calculating the TDIF matrix, we enforced a strict age rating filter on the dataset and filtered out all inappropriate entries. We did this to ensure that the content is family-friendly and no viewer is exposed to potentially inappropriate content.

For quantifying similarity, we compared different similarity metrics(Cosine, Pearson, and Euclidean) and found relatively indifferent rankings across the three. We finally settled for Cosine Similarity as it is independent of magnitude and is much faster to calculate than the other two metrics without significantly affecting our final results.

The formal, mathematical definition of cosine similarity is as follows:

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

After this, we sorted the list containing the pairwise cosine similarity of all the movies with the movie we chose. In the case of multiple movies of choice(when multiple users are present), we took the total sum of pairwise cosine similarity of each movie in the dataset with each movie of choice as our final similarity metric. We then recommended the movies in decreasing order of value of this metric.

6. Collaborative-Based Filtering

In the collaborative filtering approach, we recommend the movies to a user based on other users having similar interests. We look for users with similar tastes and preferences to the current user. Then we recommend the movies favored by the matched users to the current user. We don't require movie metadata to recommend movies for this technique. Instead, it requires the ratings the user gave to the movies. The dataset we used had the movie id, user id, and the ratings assigned by the user id to the specific movie id.

We had one dataset containing movie id, movie name, and movie genres, and a second dataset containing user id, movie id, and the rating given by the user id to the corresponding movie id. We joined both of these datasets using movie id as the primary key. We then count the ratings per movie and append the rating count column corresponding to each movie id in the new dataset created. Further, we make a pivot table with index value as the user id, column value as the movie title and the feature whose statistical summary is to be seen is ratings. Now we input a movie name to which the current user wants similar recommendations. We calculate the Pearson correlation of the input movie title column vector with all other column vectors having ratings of more than 20.

$$\text{Pearson correlation coefficient} = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Where,
x = Values in the first set of data
y = Values in the second set of data
n = Total number of values.

Now we get the pairwise Pearson Correlation value of every movie with the input movie. Now, we recommend five movies whose pairwise Pearson Correlation is the highest. In the case of multiple user inputs, we can recommend movies based on whose sum of the pairwise Pearson Correlation values associated with each input movie is the highest.

7. Hybrid Filtering

The hybrid filtering technique aims to recommend movies using the combination of the content-based filtering technique and collaborative filtering method. The Hybrid filtering approach can be performed in many ways. The contest-based filtering and collaborative-based filtering can be used separately, and then their predictions can be combined. Or we can add the functionalities from the contest-based methods to a collaborative-based approach. We seek to implement the latter approach to make our current model more effective and accurate. Due to discrepancies and missing data entries in the datasets used for content-based and collaborative filtering, we could not implement hybrid filtering.

Model Performance Analysis through CTR

Our model is based on unsupervised learning. We can not define loss w.r.t the current dataset. Loss is based on user feedback, that is, whether the user liked a recommendation or not. So our model cannot reduce the loss unless we get real time feedback. In real life performance is measured through click-through rate(CTR). CTR is the number of clicks a movie recommendation receives divided by the number of times that recommendation is shown. For example, if a movie recommendation were clicked five times after being shown 100 times as a recommendation, its CTR would be 5%. If our model is deployed on real-world data, we can analyze the feedback from users and improve the CTR of our recommendations.

Results and Analysis

Demographic Filtering: As expected, the top movies and shows based on our new metric are shown.

	title	new_score
36	Breaking Bad	9.263333
869	Our Planet	9.133333
49	Avatar: The Last Airbender	9.100000
521	My Mister	9.033333
317	Reply 1988	9.033333
1280	Arcane	9.027000
34	Okupas	8.933333
126	Hunter x Hunter	8.933333
548	Numberblocks	8.900000
54	DEATH NOTE	8.888000

The output of demographic filtering: Top 10 movies recommended based on IMDb and tmdb rating

Content Based: As can be seen, in content-based filtering, when users entered their movie preferences, similar recommendations based on description were made. All adult-rated movies were filtered out when the group consisted of users below 18. For a group of all 18+ users, the output consisted of adult-rated movies as well.

```
Enter the Total Number of Users:2
Enter Your Age:15
Enter Your Movie Preference:#realityhigh
Enter Your Age:16
Enter Your Movie Preference:Care of Kancharapalem
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows...
Here's What We Think You'll Like:
         1 Big Stone Gap
         2 Kocan Kadar Konus
         3 Just Friends
         4 Sakaling Maging Tayo
         5 Roswell, New Mexico
```

Content-Based Filtering for groups of teenagers

```
Enter the Total Number of Users:3
Enter Your Age:19
Enter Your Movie Preference:Shutter
Enter Your Age:29
Enter Your Movie Preference:Phobia 2
Enter Your Age:23
Enter Your Movie Preference:Death of Me
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows...
Here's What We Think You'll Like:
         1 Unsolved Mysteries
         2 Fear Files... Har Mod Pe Darr
         3 The Unborn Child
         4 Malevolent
         5 13 Reasons Why
```

Content-Based Filtering for an average urban household

Collaborative Filtering: Two users with similar choices are found, and a movie recommendation is based on this.

```
Initializing Recommender...
Recommender Initialized

Now Cleaning The Dataset...
Dataset Cleaned...

Now Ready To Recommend...
Successfully Added 2 New Users
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows...
Here's What We Think You'll Like:
         1 Sardaar ji
         2 Insidious
         3 Vivah
         4 Night of Knots
         5 Death of Me

Others Also Watched:
         1 Flight of the Navigator (1986)
         2 My Fair Lady (1964)
         3 Driving Miss Daisy (1989)
         4 Inspector Gadget (1999)
         5 M*A*S*H (a.k.a. MASH) (1970)
```

Combined Output:
After combining results for demographic, content and collaborative filtering along with application of age based-filters, this is how the output is shown to users. Note that we first recommend content based results(here's what we think you'll like) followed by collaborative filtering results(others also watched) and lastly demographic filtering results(Don't like any of these? Start here)

Input:

```
r = Recommender('merged_genre.csv')
r.addUsers([["Hellboy",20],["Shutter",20]])
r.recommend()
```

Output:

```
Initializing Recommender...
Recommender Initialized

Now Cleaning The Dataset...
Dataset Cleaned...

Now Ready To Recommend...
Successfully Added 2 New Users
Recalculating the Entries To Suit Your Preferences
Now searching from a catalogue of over 8809 Movies and TV Shows...
Here's What We Think You'll Like:
     1 Sardaar ji
     2 Aaviri
     3 Insidious
     4 Vivah
     5 Death of Me

Others Also Watched:
     1 Flight of the Navigator (1986)
     2 My Fair Lady (1964)
     3 Driving Miss Daisy (1989)
     4 Inspector Gadget (1999)
     5 M*A*S*H (a.k.a. MASH) (1970)

Don't like any of these? Start Here:
     1 Breaking Bad
     2 Our Planet
     3 Avatar: The Last Airbender
     4 My Mister
     5 Reply 1988
     6 Arcane
     7 Okupas
     8 Hunter x Hunter
     9 Numberblocks
     10 DEATH NOTE
```

Conclusion

Our learning from the project:
Through this project, we learned about the three main filtering techniques for movie recommendation systems: content-based, collaborative, and demographic filtering. Content-based filtering provides recommendations similar to the user's preferences in the past. Demographic filtering utilizes information from the population to make recommendations where user preferences are unknown. Collaborative-based filtering provides recommendations by finding similar users. We learned about encoding string data into more computationally friendly formats while preserving the integrity of the data. We also learned about different similarity metrics like Pearson correlation and cosine similarity. We learned how to apply PCA and chose the number of components to reduce dimensionality. We applied K-means to analyze if our data could be clustered.

Each member's contribution:
We all have worked equally on this project.

References:

[1] Ching-Seh Mike Wu, Deepti Garg, and Unnathi Bhandary.*Movie Recommendation System Using Collaborative Filtering*
[2] F. Furtado, A, Singh, *Movie Recommendation System Using Machine Learning 2022*