Movie Recommender Markov Chain Project

Ishaan Sathaye

December 10, 2020

Contents

1 Real-World Phenomenon

Recommendations and preferences in technology services are very important to companies and people, as it can help companies build profiles and help people find their favorites. Many services like Netflix, Youtube, and Amazon Prime Video depend on their recommendation systems to engage their customers. Personalizing a user's preferences on movies can help movie theaters and advertisements to be more focused. Using genres and movie ratings in a recommendation chain program can provide users with similar movies, based on what they have watched or liked. Movies that users tend to enjoy, usually have a genre in common and they all come with a rating. Using a combination of these aspects would allow a program to recommend users to watch certain movies over others.

2 Derivation and Assumptions

2.1 Derivation

The Markov model was derived by finding characteristics of movies that can be used to recommend the user more movies.

Genres(19): Adventure, Animation, Children, Comedy, Fantasy, Romance, Drama, Action, Crime, Thriller, Horror, Mystery, SciFi, IMAX, Documentary, War, Musical, Western, and Film-Noir
Ratings: On a scale from 1.0 to 5.0

The initial state matrix would entail the movie name and the rating of the user. This data would be organized with the name following the rating in the program. This matrix below could be a sample of what the user has watched or liked.

$$S_0 = \begin{bmatrix} ToyStory : 3.5 \\ Jumanji : 2.0 \\ Akira : 4.5 \\ Rampage : 3.1 \\ AvengersInfinityWar : 4.2 \end{bmatrix}$$

The program would then search the dataset for these movies and find all their genres. The transition matrix would include the quantitative data values consisting of a weighted average between genres and ratings for all the movies in the dataset. This data will be organized in a tabular format, and then converted to a list to find the preferences. Each position shows the probability of watching one movie over another, as shown in the matrix below. The matrix is calculated and continued for all movies in the dataset.

$$T = \begin{bmatrix} 1 & 0.1 & 0 & 0 & ... \\ 0 & 0.1 & 0 & 0.1 & ... \\ 0 & 0.5 & 0.3 & 0 & ... \\ 0 & 0.3 & 0.7 & 0 & ... \\ ... & ... & ... & ... & ... \end{bmatrix}$$

2.2 Assumptions

Assumptions need to be made in order to simplify the created recommendation system. Outside recommendations will not be considered, therefore the program will only take in the current user's preferences. This makes the system less complex and easier to track the sole user's preferences. Another assumption would be that there will be only two characteristics that the preferences will be based. The movie dataset will only include ratings and genres. Also, the assumption will be made that well-known and popular movies will be considered as a recommendation, so this may exclude foreign movies.

3 Data and Parameters

Data of movies, ratings, and genres will be taken from GroupLens and IBM. Using `!wget -O`, the program will download and unzip the dataset from an IBM API that contains the GroupLens movie data. All types of data will be first put into a CSV file, then into a data frame, and preprocessed.

GroupLens Datasets: http://grouplens.org/datasets/movielens/

The movie dataset will be then put into a data frame and preprocessed to include the movieID, title, and the genres only. The table below shows the first five rows of the move and genre data frame:

-	movieID	title	genres
0	1	Toy Story (1995)	Adventure,Animation,Children,Comedy,Fantasy
1	2	Jumanji (1995)	Adventure,Children,Fantasy
2	3	Grumpier Old Men (1995)	Comedy—Romance
3	4	Waiting to Exhale (1995)	Comedy—Drama—Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...

The same thing will apply to the ratings dataset, which will be first be in a CSV file then put into a data frame to be preprocessed. The table below shows the first five rows the ratings dataset that are organized by the movieID, which correspond to the movie and genre data frame:

-	userID	movieID	rating
0	1	169	2.5
1	1	2471 (1995)	3.0
2	1	48516	5.0
3	2	2571	3.5
4	2	109487	4.0
...	...	...	...

Parameters for the initial state matrix will include the user's watched or liked movies that will be inputted. The Pandas data frame of the input will look simple with only the title of the movies that the user likes and the ratings of those movies:

-	title	rating
0	Breakfast Club, The	5.0
1	Toy Story	3.5
2	Jumanji	2.0
3	Pulp Fiction	5.0
4	Akira	4.5

4 Benchmarking

By inputting people's movies, the program was tested on how well it recommended movies that were similar to the input. The first way that the program was benchmarked was by having a person pick 6 movies that they like and are very similar to each other. One person was told to pick 6 movies that were very similar to each other so he or she picked 6 Jackie Chan movies, and 5 of them were inputted into the program acting as the initial state matrix. Then the recommendation was run and checked to see if the 6th movie, which in this case was *Rumble in the Bronx*, appeared. In fact, the program did excellent as it recommended the movie in 2nd place out of 50 movies. This method was repeated: another person was asked to input any 6 movies that he or she likes this time. The sixth movie that was left out was Journey to the Center of the Earth and the other movies had adventurous genres common with this. When the program was run, the recommendation did not show the sixth movie at the top, rather it was 48th with *Super Mario Bros* and *Space Jam*. Although this method did not show the consistency of the program, it made recommendations that users have not watched and showed what genres led to the recommendation.

5 Results

The recommendation program operated successfully by creating recommendations for a user, based on the movies given, a movie dataset, and their ratings/genres. By inputting movies that a user watched, the program outputted

a list of 50 movies that the program recommends the user watches.
Below is a recommendation table (showing only top 5) that is generated:

movieId	title	genres	year
546	Super Mario Bros.	[Action, Adventure, Children, Comedy, Fantasy,...	1993
673	Space Jam	[Adventure, Animation, Children, Comedy, Fanta...	1996
4781	Megiddo: The Omega Code 2	[Action, Adventure, Fantasy, Sci-Fi, Thriller]	2001
5882	Treasure Planet	[Adventure, Animation, Children, Sci-Fi, IMAX]	2002
6350	Laputa: Castle in the Sky (Tenkû no shiro Rapy...	[Action, Adventure, Animation, Children, Fanta...	1986
6365	Matrix Reloaded, The	[Action, Adventure, Sci-Fi, Thriller, IMAX]	2003
...(45)	...(45)	...(45)	...(45)

6 Discussion

The model was able to recommend movies in the end using the initial state matrix of the user's movies and ratings. The program made a profile for the user based on the ratings given and the genres of the movies from the input and the entire Movie Dataset:

Adventure	8.8
Animation	3.5
Children	5.5
Comedy	3.5
Fantasy	8.8
Romance	0.0
Drama	0.0
Action	6.8
Crime	0.0
Thriller	3.5
Horror	0.0
Mystery	0.0
Sci-Fi	7.8
IMAX	11.1
Documentary	0.0
...(5)	...(5)

These values of the profile were then used to generate the transition matrix which contains the likelihood of watching each movie in the dataset:

movieId	-
32031	0.826307
85261	0.792580
103042	0.730185
54278	0.694772
47124	0.694772
...	...

Then the matrix is applied to the original dataset where the movies that have the best weighted values are sorted to the top, resulting in the recommendation table. To test this model, 6 movies were picked by random people to see if the 6th movie not inputted would make it to the recommendation table. 1 person was told to pick 6 movies that he or she liked to watch and were very similar to each other. The other person was told to pick any 6 movies that he or she loved. The results reveal that with more similar movies, the program outputted a similar movie at the top, while with random and not closely related movies the 6th movies was far down the table. To improve this model, we can use more factors like actors, year, language, and topic that can make the recommendations more precise and personalized. Therefore improving the amount of variables and factors, in addition to expanding the movie dataset can further refine the recommendation model.

7 Code

Jupyter notebook code or Python script in separate files.
Github Link: Markov Movie Recommender Program