IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MOVIE RECOMMENDATION SYSTEM USING MACHINE LEARNING

Nakeertha Rajkumar[1], Dr. G. Venkata Rami Reddy[2]

[*1]Student, M.Tech (Computer Science), School of Information Technology, JNTU Hyderabad, Hyderabad, India

[*2]Professor of IT, School of Information Technology, JNTU Hyderabad, Hyderabad, India

ABSTRACT: As part of an information filtering system, recommender systems are used to anticipate a user's bias/rating is likely to give an item. As a result of their effectiveness, CF techniques are very prevalent among several ways to approval. These traditional CF systems extremely well and produce create standardized advice for complex problems. For item-based on the preferences of their neighbour better suggestions are generated by collaborative-filtering techniques than by others. Other techniques, such as content-based prediction, suffer from subpar accuracy and scalability, data sparsity, and large-error estimate. I used an item-based CF approach to find these possibilities. In this item-based collaborative-filtering technique, I first examine user item rating matrix to identify relationships around various items, and then I use these relationships to determine user's recommendations.

Keywords – Movie Recommendation; Collaborative Filtering Based System; Content Based Filtering System; K-Means Algorithm; Naïve Bayes Algorithm; K-Nearest Neighbors Algorithm.

1. INTRODUCTION

In many aspects of real social networks, such as e-commerce services such as Amazon.com, movie rating website IMDB. This project focuses on multi-level link prediction for movie recommendation. Data on movie ratings is now widely available on the Internet. Most rating systems are designed with five score options ranging from 1 to 5. However, there isn't much of a difference between scores 4 and 5 when it comes to finding a potentially interesting movie for a specific user. It's a classification problem in which the user must predict the category of a movie that has not yet been watched. So instead of a specific score, I'm more concerned with the genre of the movie, and the user input the code will present the end user with the desirable results. Another important motivation for this project is to build a social recommendation system in the future. The idea comes from recommendations we get from our friends in everyday life. It is quite common that we tend to value more about reviews given by other strangers more before watching. In this project, I utilized a set of data that encompasses multiple genre ratings and hundreds of saved movie names. So, after the user selects three fields, all of which are genres, the recommendation system will show the user the best results that match all of the genres with the finest ratings in descending order.

A recommendation system is a type of information filtering system that attempts to predict a user's preferences and makes recommendations based on these preferences. There will be a explanatory applications for recommendation systems. These have grown in popularity in recent years and are now used in the majority of online platforms that we use. Such platforms' content ranges from movies, music, books, and videos to friends and stories on social media platforms, products on e-commerce websites, people on professional and dating websites, and Google search results. Users now expect good recommendations due to advances in recommender systems. They have a low threshold for services that are unable to make appropriate recommendations. If a music streaming app cannot predict and play music the user enjoys, the user will simply stop using it. As a result, tech companies have placed a strong emphasis on improving their recommendation systems. However, the problem is more complicated than it appears.

2. LITERATURE REVIEW

There are numerous types of collaborative filtering systems, but many common systems can be reduced to two steps:

1. Look for users who have similar rating patterns to the active user (the user whom the prediction is for).

2. Calculate a prediction for the active user using the ratings from those like-minded users found in step 1.

Content-based Filtering Systems (CBF based systems):

Items are recommended in content-based filtering based on comparisons between item profile and user profile. A user profile is content discovered to be relevant to the user in the form of keywords (or features). A user profile can be thought of as a set of assigned keywords (terms, features) gathered by an algorithm from items found relevant (or interesting) by the user. The Item profile is a collection of an item's keywords (or features). Consider the following scenario: A person goes to a pastry shop to purchase his favourite cake 'X.' Unfortunately, cake 'X' has been sold out, so the shopkeeper suggests that the person purchase cake 'Y,' which is made with similar ingredients to cake 'X.' This is an instance of content-based filtering.

Collaborative filtering based systems (CF based systems):

A collaborative filtering system recommends items based on measures of similarity between users and/or items. The system suggests items that similar users have liked. This is based on the scenario in which a person asks his friends with similar tastes to recommend some movies to him.

3. SYSTEM ANALYSIS

The term "collaborative filtering" first appeared in the context of Tapestry, the first commercial recommender system designed to recommend documents drawn from newsgroups to a group of users. The goal was to use social collaboration to keep users from being overwhelmed by a large volume of streaming documents. Collaborative filtering, which analyses usage data across users to find well-matched user-item pairs, has since been pitted against the older content filtering methodology, which has its origins in information retrieval. Recommendations in content filtering are not "collaborative" in the sense that suggestions made to a user do not explicitly use information from the entire user-base. The Group Lens system was one of the first collaborative filtering successes in related domains.

Existing System:

The main issue with recommender systems is that they require a large amount of data to make effective recommendations. It's no coincidence that the companies most associated with providing excellent recommendations are those with a large amount of consumer user data: Google, Amazon, and Netflix. The more item and user data a recommender system must work with, the better the chances of getting good recommendations. However, in order to get good recommendations, you must have a large number of users so that you can collect a large amount of data for the recommendations. Another issue with search engines is rapidly changing data. Clearly, an algorithmic approach will find it difficult, if not impossible, to keep up with fashion trends. Finally, the search engine is unpredictable because it cannot provide a perfect match.

Proposed System:

Recommendation for a Film A collaborative filter based on users is used by Search Engine. The user-based CF approach is similar to the item-based CF approach. However, in this case, similarity is calculated between users rather than items. The fundamental assumption of the collaborative filtering approach is that if a person A and a person B have the same opinion on an issue, A is more likely to have B's opinion on a different issue than a randomly chosen person. The search engine will take the user's genre and movie name input and recommend a list of ten movies that he would also enjoy. To divide the objects in the dataset into clusters, the search engine employs the k means clustering method. A class label will be attached to the clusters for matching based on the user input. A hard code on the search engine is that the user’s input for movie name should be in the order of 5,4,3 ratings since the search engine would return a movie based on the hard code.

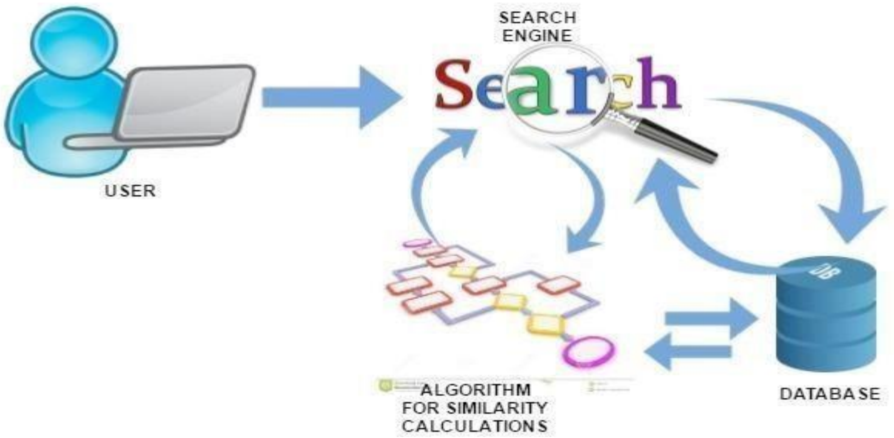


Fig-1: System Architecture

4. ALGORITHMS

K-MEANS CLUSTERING ALGORITHM

It is an unsupervised learning algorithmic program that's used to solve the cluster issues in machine learning or data science. It permits us to cluster the information into completely different groups and a convenient way to discover the classes of groups within the untagged dataset on its own without the requirement for any coaching. It's a centroid-based algorithmic program, wherever every cluster is related to a centre of mass. The most aim of this algorithmic program is to reduce the total of distance between the data point and their corresponding clusters.

K-NEAREST NEIGHBORS ALGORITHM

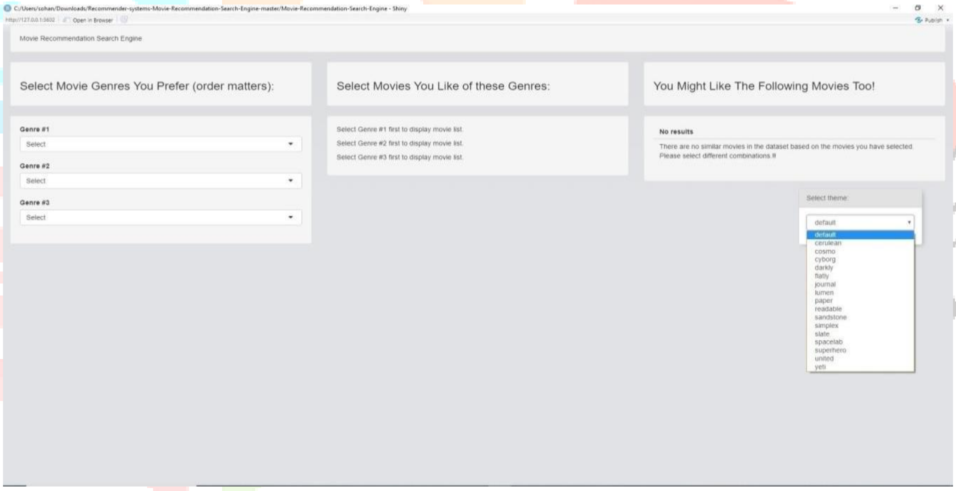
This is an unsupervised learning algorithm known as Nearest Neighbours. Since this algorithm calculates distance between two points we will pivot our dataset into an item user matrix and an empty cell with 0. Now once a movie name is given a s input we'd like to seek out any such movie present in our dataset or not. It's not present then we have a tendency to can't recommend anything. Thus for string matching we have a tendency to use fuzzy matching, a list of recommendations are going to be generated. The function can come from the movie_id of the movie title that best matches the input string. It additionally prints all matches.

NAÏVE BAYES ALGORITHM

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which help in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

5. EXPERIMENTAL RESULTS

When we launch the app, we can select the three genres and related movies to get the recommended movies.



C:/Users/sohan/Downloads/Recommender-systems-Movie-Recommendation-Search-Engine-master/Movie-Recommendation-Search-Engine - Shiny

http://127.0.0.1:3602

Movie Recommendation Search Engine

Select Movie Genres You Prefer (order matters):

Genre #1: Select

Genre #2: Select

Genre #3: Select

Select Movies You Like of these Genres:

Select Genre #1 first to display movie list.

Select Genre #2 first to display movie list.

Select Genre #3 first to display movie list.

You Might Like The Following Movies Too!

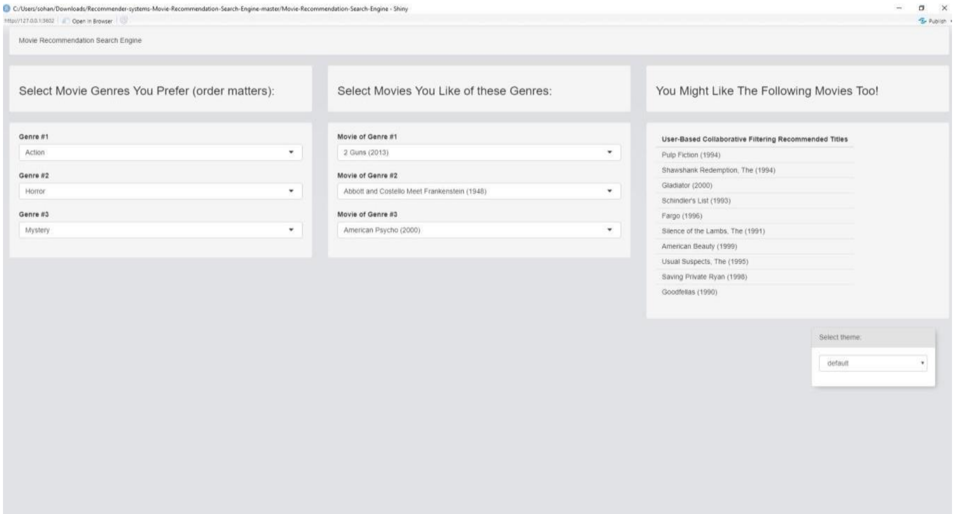
No results

There are no similar movies in the dataset based on the movies you have selected. Please select different combinations !!

Select theme: default

Figure-2: Output1

Select the genres and movies that are related to the genre and the input are sent to the recommender system, which displays the most recommended movies.



C:/Users/sohan/Downloads/Recommender-systems-Movie-Recommendation-Search-Engine-master/Movie-Recommendation-Search-Engine - Shiny

http://127.0.0.1:3602

Movie Recommendation Search Engine

Select Movie Genres You Prefer (order matters):

Genre #1: Action

Genre #2: Horror

Genre #3: Mystery

Select Movies You Like of these Genres:

Movie of Genre #1: 2 Guns (2013)

Movie of Genre #2: Abbott and Costello Meet Frankenstein (1948)

Movie of Genre #3: American Psycho (2000)

You Might Like The Following Movies Too!

User-Based Collaborative Filtering Recommended Titles

Pulp Fiction (1994)
Shawshank Redemption, The (1994)
Gladiator (2000)
Schindler's List (1993)
Fargo (1996)
Silence of the Lambs, The (1991)
American Beauty (1999)
Usual Suspects, The (1995)
Saving Private Ryan (1998)
Goodfellas (1990)

Select theme: default

Figure-3: Output2

6. CONCLUSION

The movie recommendation search engine returns a list of ten movies that users might be curious in as well. Item-based collaborative filtering was found to be less error-prone than user-based collaborative filtering. Furthermore, because it’s less-dynamic model was computed less frequently and stored in a smaller matrix, item-based systems outperformed user-based systems. Soon, I'd like to improve prediction accuracy by employing the Slope One algorithm. Slope One is a family of collaborative filtering algorithms introduced in a 2005 paper by Daniel Lemire and Anna Maclachlan. It is arguably the most basic form of non-trivial item-based collaborative filtering based on ratings. Their simplicity makes them particularly easy to implement efficiently, and their accuracy is frequently comparable to that of more complicated and computationally expensive algorithms. They were also deployed to improve other algorithms as building blocks. They are included in popular open-source libraries like Apache Mahout and Easyrec.

7. REFERENCES

1. C. M. Wu, D. Garg and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018, pp. 11-15, doi: 10.1109/ICSESS.2018.8663822.

2. https://rpubs.com/tarashnot/recommender_comparison.

3 https://shiny.rstudio.com/tutorial/

4. https://link.springer.com/article/10.1007/s10479-016-2367-1#Sec15

5 L. T. Ponnam, S. Deepak Punyasamudram, S. N. Nallagulla and S. Yellamati, "Movie recommender system using item based collaborative filtering technique," 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, 2016, pp. 1-5, doi: 10.1109/ICETETS.2016.7602983.