TO SOLVE COLD START PROBLEM IN COLLABORATIVE FILTERING RECOMMENDER SYSTEM

Sreenivas Mekala
Sreenidhi Institute of Science & Technology

Ch Chaithanya Reddy
Sreenidhi Institute of Science & Technology

TO SOLVE COLD START PROBLEM IN COLLABORATIVE FILTERING RECOMMENDER SYSTEM

[1]Dr. Sreenivas Mekala, [2]Ch. Chaithanya Reddy, [2]V. Shashi Kumar, [2]D. Adarsh Reddy
[1]Associate Professor, Information technology, Sreenidhi institute of science and technology
[2]UG Scholar, Information technology, Sreenidhi institute of science and technology

Abstract: It is crucial that customers have access to useful information in light of the plethora of data already available online. Here is where computer-generated advice proves useful. A subclass of information filtering systems, recommendation systems offers customized recommendations to end users. It's a tool that aims to facilitate the discovery of desired resources. Several companies employ recommender systems to better serve their clientele. They use a sizable database to determine which products and services will best meet the needs of each individual customer. Given their versatility, recommendation systems are at the cutting edge of a number of emerging industries. One popular technique for recommendation systems is called collaborative filtering, and it involves analyzing customers' past interactions with products. Despite the success of this strategy, the problem of cold start must still be taken into account. The "cold start problem" encapsulates the difficulties encountered while making recommendations to new users. When no historical information is available about the user, it is more difficult to make educated guesses about their preferences and provide useful recommendations. The proposed method uses a movie dataset to address this problem in the recommender system.

Keywords: Machine Learning, Python, Recommendation system, Movie.

I. INTRODUCTION

An information filtering system, a recommendation system shows users material they might be interested in. Recommender systems are computer programmes that suggest goods and services to users. To determine which goods will be of most interest to a certain user or client, it employs machine learning techniques. Amazon and Netflix, two of the biggest names in the internet industry, are using similar methods to draw in more customers and encourage them to use their products. While there is a lot of data accessible online, it may take some time to zero in on what you need. This is another way in which recommender systems save us time. Movies, publications, products, etc. are just few of the many areas where they are currently being employed as filters.

Due to the exponential growth of online data, users are finding it difficult to keep up. Finding a way to filter out the irrelevant data from the ocean of data is therefore essential. Many people in the academic, entertainment, and business communities see the recommender system as a potent tool for satisfying this need.

Since we all rely on information and expertise to make choices that are in our best interests, Recommendation Systems are ubiquitous. It helps when we want to learn more about related topics to enrich our viewing experience. Creating a system that meets the needs of its users is of paramount importance. If the user is unhappy with the information shown to him, he may lose interest in continuing to utilize the system.

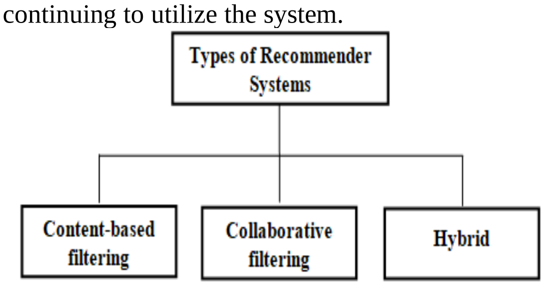

Fig 1: Types of recommender systems

It is more efficient to focus on the content rather than the form. It depends on how well the characteristics of the two products align with one another. When evaluating a movie system, for example, we look for recurring elements like the genre, hero, director, and heroine of the films in the system. Because of this, the primary focus that is maintained throughout the design process of a content-based recommender system is on doing an analysis of the features and qualities of the item being recommended.

Tools/Technologies used

Python is often used for ML due to its friendliness, adaptability, and abundance of ML-friendly tools and frameworks. It is possible to write this Python code in a Jupyter notebook. It's a platform for combining augmented text, code execution, graphs, and mathematics in a dynamic and interactive setting. It's easy to use and has great features, both of which improve the efficiency with which you can write code.

Pandas is a library for doing things like that. It offers data structures for efficiently handling large datasets and efficient ways for transforming, aggregating, and cleaning data. Series and DataFrame objects are pandas' two primary data types. Series objects store data in a single dimension, while DataFrames store multiple Series objects in a two-dimensional table-like structure. In addition to these basic operations, Pandas also includes more advanced features for working with data, including as pivoting, grouping, merging, and filtering.

Operations like removing rows or columns with blank values, or inserting missing data, are available in Pandas to help with missing data. Pandas integrates with Matplotlib to provide flexible and powerful visualisation tools for analysing and plotting data. Commonly employed in data science and machine learning applications, the toolset for data analysis and manipulation is powerful and adaptable.

II. RELATED WORKS

This study suggests a novel method of using feature selection and prediction to address the cold start problem in collaborative recommender systems. In this study, they propose a technique based on selecting the most important traits, sometimes called relevant features. The system attempts to ascertain which components are most relevant to the user's aim in order to reduce the number of questions asked and improve the precision of predictions and recommendations. Once the most linked characteristics have been identified as the relevant features, the algorithm will attempt to collect the attribute values from the cold user. In order to gauge the potential interest of a cold user, the proposed technique employs a regressor to predict the value of the system's objective feature using data collected for those features of interest. The strongest correlations between attributes were used to determine which ones to use as predictors. The regressor's expected output stands in for the objective quality of the cold start user's target value. The proposed method creates suggestions by using a similarity metric to discover warm users who are similar to active users. When making suggestions, the top N users are those with the highest similarity values.

[2] It was proposed to implement a recommendation system for tourist destinations using collaborative filtering. The primary objective of this study was to identify the best model for recommending a specific location to a new target user based on their preferences and past actions. Data acquired in an intentional manner about user travel was essential for this purpose. Previous TRSs that recommend travel itineraries have mostly focused on different ways to infer travel-related information from users' implicit actions, such as through collecting user data from many web sources. Using, for instance, information about when and where users have checked in. However, the validity of extrapolating user preferences is a major issue with the implicit method. Users may be more likely to provide reliable responses to a questionnaire than they would when using the explicit technique, which depends on direct questioning of the user. Therefore, some clever RSs opted for option two and put it to good use. Following the similarity metric application, they chose their neighbours using the best k nearest method. We utilised the top-k method (where k is the total number of users) to determine our neighbourhood. Using the idea of the Elbow technique, they designed an experiment to find the optimal value for top k-nearest neighbours using several types of similarity metrics. Starting with 10 neighbours, the k-neighbor experiment then added 5 more neighbours for every 100 additional neighbours.

[3] In this research, a recommender system was created using a Singular Value Decomposition technique and a collaborative filtering strategy. This work proposes an incorporation-based recommendation strategy to solve the sparsity problem in SVD-based algorithms. At first, we find connections between users and the products they utilise. After values are correlated, data can be generated. Based on the new information, the SVD framework is revised. To improve the performance of SVD-based recommendation, this research tackles the issue of data sparsity. The SVD matrix factorization technique is used to reduce the number of space dimensions from N to K, where K N is the number of features in the data collection. Collaborative filtering recommender systems make suggestions based on a user's expressed interest in a certain product or service. The user's rating of the product is indicative of how much he values it. The options are 1, 2, 3, 4, and 5. Few products remain unrated, which makes it difficult to administer CF recommender systems. SVD is among the most powerful MF methods. Initial ratings matrix generation is based on data collected from MovieLens. Similarity calculations make it possible to locate users and goods with shared characteristics. After a similarity calculation is performed, good neighbours are identified for both the users and the goods. Using this newly created data, the SVD can predict the rating. In general, this approach succeeds in guiding users in the right direction.

[4] Suppose a new consumer signed up for an online store and created an account. The

recommender system would use the user's information to find other users whose profiles were most like their own. The system would then extrapolate the interests of these parallel users and assume that the new user shared them. While the recommender system in this instance did a good job of taking users' actual preferences into consideration, When creating an account, a user is prompted to enter personal details such as name, date of birth, gender, country of residence, and so on. The user's interests would be extracted based on his rates, of course. These passions are recorded in a separate table from the user profile information called the ProfileInterest Table, where each row represents a user and each column represents a characteristic of the user's profile. The final column in the table displays the user's preferences. After the table of profile-interest data has been compiled, a classifier is utilised to glean insights from it and identify patterns. Patterns can be used to infer information about users, such as their age. When a new user signs up and fills up his own profile, the trained classifier is used to predict that person's interests and provide personalised recommendations.

III. METHODOLOGY

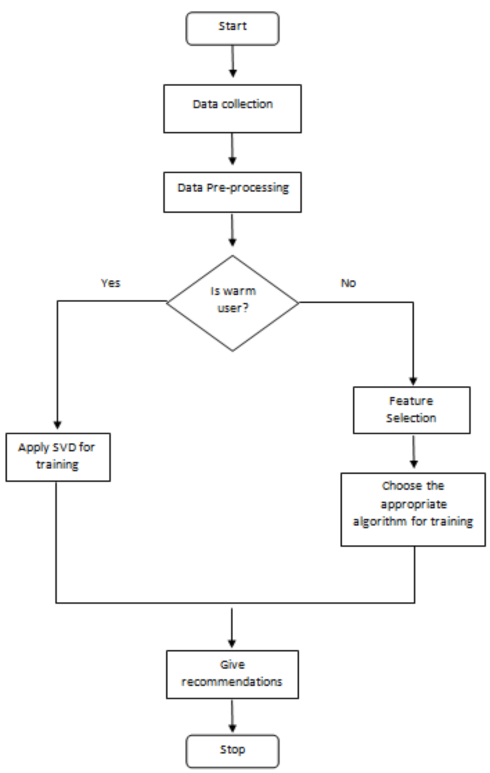

Fig 2: Flowchart

The fundamental function of a recommendation system is to provide suggestions that will appeal to the consumers. Since the collaborative filtering method is effective for constructing such systems, we will employ it when creating our own and also address the system's cold start issue. Here are the measures to be taken:

Loading the Dataset

Finding and importing the dataset is the first step.Our approach makes use of the MovieLens dataset.

Pre-processing

The entire process of tidying up the dataset occurs here. To begin, we cherry-pick the most important features and an impartial one from the dataset. In addition, we can pick out of the dataset only the records that are suitable for our purposes. To do this, we must first determine a minimum required number of users, and then get rid of everything that has fewer ratings than that. This procedure will ensure that we have only the necessary materials. The ratings' numerical values are then analysed, and scale is applied if necessary.

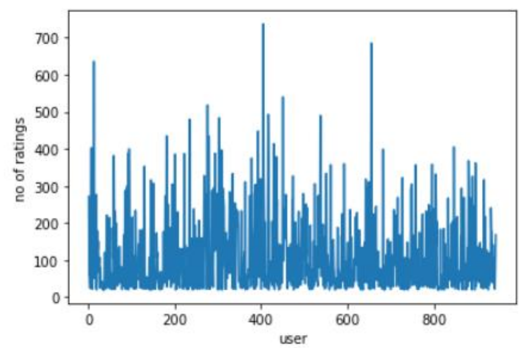

Fig 3: User vs Number of ratings

Model

The model is trained using the data after the dataset has been prepared. Many different machine learning algorithms can be used to teach it.

When a user launches the app, they'll see personalized suggestions tailored to their preferences. Users can be categorized as either "warm" or "cold." Warm users are long-time customers whose information can be used to tailor suggestions. Making recommendations to users who have never interacted with the system before, known as "cold users," presents a challenge for collaborative filtering, and we want to address this issue.

Singular Value Decomposition (SVD) is a matrix factorization that produces three new matrices. It's a method of technological application for making things smaller in scale. Important geometrical and theoretical insights into the linear transformations are conveyed, and it also has numerous interesting algebraic properties. It also has some very important applications in data science. SVD is used in movie recommendation systems to analyse user ratings of films and anticipate how users will rate other films based on those ratings. In data analysis and machine learning, SVD is often used for dimensionality reduction, data compression, image and signal processing, and collaborative filtering. SVD can be applied to a data matrix to extract its essential features, thereby reducing the data's dimensionality while keeping what's most important. Predicting user tastes by collaborative filtering using SVD requires a matrix of user-item ratings. When given a matrix A, SVD breaks it down into its component parts.

$$A_{[m \times n]} = U_{[m \times r]}\Sigma_{[r \times r]}(V_{[n \times r]})^T$$

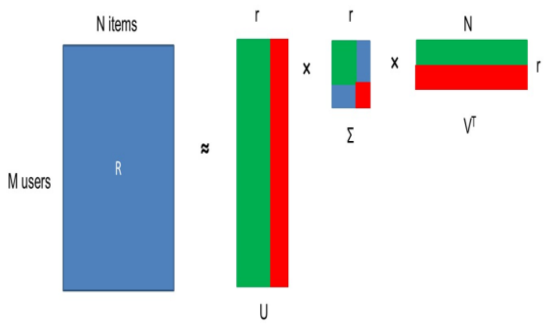

Fig 4: Singular Value Decomposition

Gradient boosting is a machine learning technique used for regression and classification. It provides a prediction model in the form of a collection of decision trees, each of which is a relatively weak prediction model. When a weak learner is provided with a decision tree, the resulting method is known as a gradient-boosted tree.

It's a powerful method for improving performance that has helped many students who were struggling before. Gradient descent is used to train each new model so that it minimises some loss function—typically mean square error or cross entropy—that was used to train the previous model. In each iteration, the algorithm computes the gradient of the loss function relative to the predictions of the current ensemble, and then trains a new, less robust model in an effort to minimise this gradient.Predictions from the revised model are added to the ensemble, and this process is repeated until a stopping requirement is met.

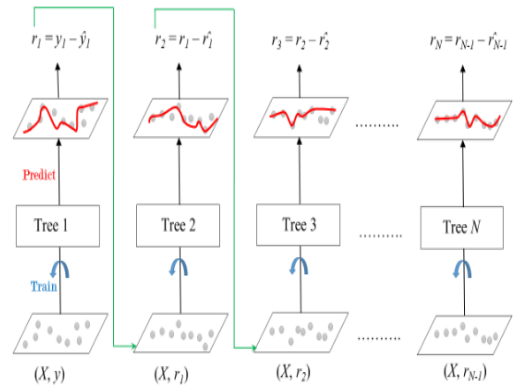

Fig 5: Gradient Boosting

Dataset

Each row in a dataset represents a single record in the dataset, and each column in the dataset represents a separate piece of information. It is used in the process of educating models.

On the MovieLens dataset, we have successfully deployed. 943 people are represented, along with 1681 films and 99999 ratings.

The data in this set is organised into three separate tables labelled "Movies," "Users," and "Ratings."

Table.1: Movies data

movie_id	movie_title	genre
1	Toy Story	Animation/Children/Comedy
2	GoldenEye	Action/Adventure/Thriller
3	Four Rooms	Thriller
4	Get Shorty	Action/Comedy/Drama

Users table:

This table contains data about the users and it has the following features: user_id, age, gender, occupation, zip_code. The important features are user_id, age and gender

Table 2: Users data

user_id	age	gender
1	24	M
2	53	F
3	23	M
4	24	M
5	33	F

Ratings table

This table contains data about the rating of a movie given a particular user. It has the following features: user_id, movie_id, rating, timestamp.

Table 3: Ratingsdata

user_id	movie_id	rating	timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

IV. RESULTS AND DISCUSSION

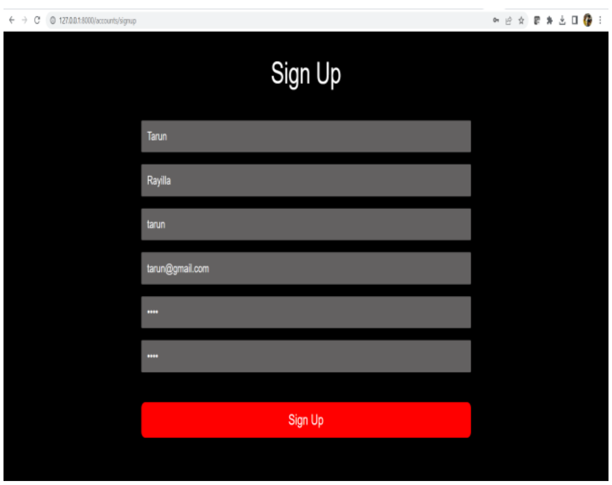

Fig 6: Sign Up page

This is the sign-up page for the users. The system asks for firstname, lastname, username, email, and password. After successfully entering the details, the user will be registered.

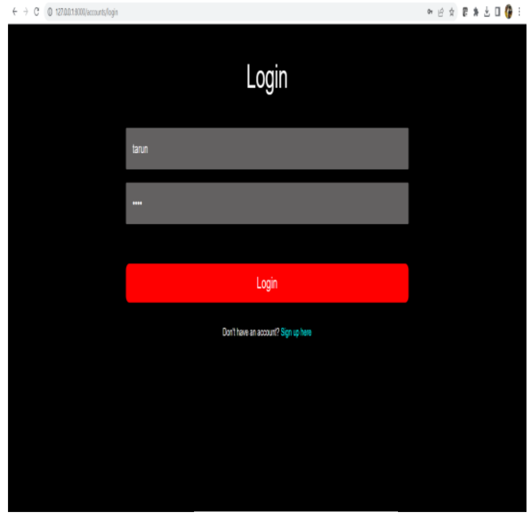

Fig 7: Login page

This is the Login page for the users. With the correct email and password, users can login into their accounts.

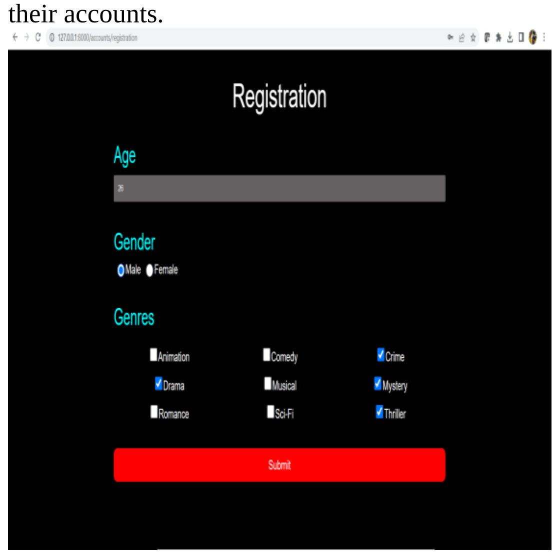

Fig 8: Registration page

This is the registration page for the users. The user gets to fill these details during the sign up.

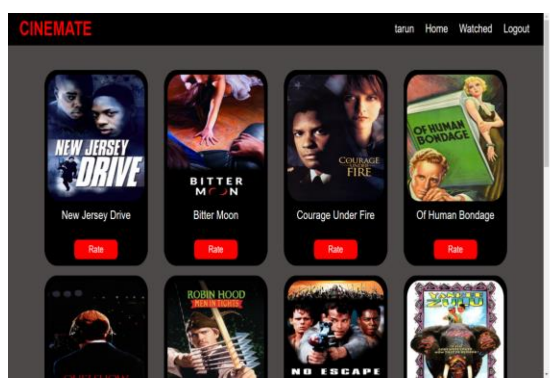

Figure 9: Home screen

This is the home screen after logging in. The user will be able to view the recommended movies to him/her in this screen. The user will be able to rate a movie by clicking on the rate button.

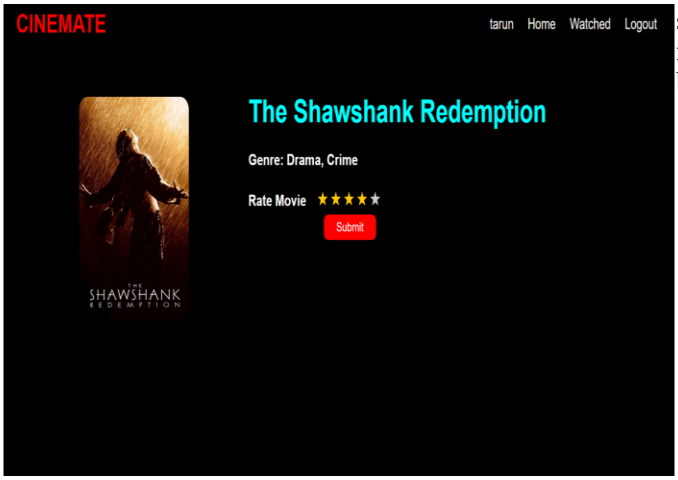

Fig 10: Rating page

The reviews can be found here. In this section, the user can rank the movie's appeal to him on a scale from 1 to 5. We have established that a correlation matrix can be used to identify influential aspects, and that their values can be retrieved by user interaction; now we must forecast the objective feature by feeding the objective features into a prediction algorithm. We have used three prediction algorithms—a Decision Tree, a Random Forest, and a Gradient Boosting Classifier. Table 4 displays the MAE used to evaluate the algorithms' performance.

Table 4 : Performance comparison of different algorithms

Algorithm	MAE
Decision Tree	1.10
Random Forest	0.92
Gradient Boosting Classifier	0.85

As can be seen in Table 4 our suggested system makes use of the Gradient Boosting Classifier because it outperforms the other two options.We calculated the MAE and RMSE of our system after putting the model into practise. According to the calculated values of MAE and RMSE, the quality of the recommendations is high.

V. CONCLUSION

Companies can use robust recommender systems to give clients individualized suggestions. Users can benefit greatly from recommender systems, which are powerful instruments for guiding users towards optimal choices and introducing users to new items. The proliferation of Internet access has resulted in a corresponding uptick in data storage requirements. Recommendation systems have been shown to be useful in facilitating the rapid discovery of desired information. We propose a mechanism to make useful suggestions to users in this work. To build this system, we adopted a technique known as "collaborative filtering." However, because this approach had the cold start problem, we tried to fix it so that we could provide better recommendations to new users, which would improve the interaction between them and the system.

REFERENCE

1. M. Kuanr, P. Mohapatra and M. Yesubabu, "Feature Selection Based Approach for Handling Cold Start Problem in Collaborative Recommender Systems," *First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR)*, 2022, pp. 1-7, doi: 10.1109/ICAITPR51569.2022.9844218.
2. "Personalized Tourist Attraction Recommendation System Using Collaborative Filtering on ToW. Supanich and S. Kulkarineethamurist Preferences," *19th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2022, pp. 1-6, doi: 10.1109/JCSSE54890.2022.9836255.
3. R. Barathy and P. Chitra, "Applying Matrix Factorization In Collaborative Filtering Recommender Systems," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 635-639, doi: 10.1109/ICACCS48705.2020.9074227.
4. B. Hawashin, S. Alzubi, A. Mughaid, F. Fotouhi and A. Abusukhon, "An Efficient Cold Start Solution for Recommender Systems Based on Machine Learning and User Interests," *Seventh International Conference on Software Defined Systems (SDS)*, 2020, pp. 220-225, doi: 10.1109/SDS49854.2020.9143953.
5. M. Gupta, A. Thakkar, Aashish, V. Gupta and D. P. S. Rathore, "Movie Recommender System Using Collaborative Filtering," International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 415-420, doi: 10.1109/ICESC48915.2020.9155879.
6. C. Sirikayon, P. Thusaranon and P. Pongtawevirat, "A collaborative filtering based library book recommendation system," *5th International Conference on Business and Industrial Research (ICBIR)*, 2018, pp. 106-109, doi: 10.1109/ICBIR.2018.8391175.