ISSN: 2366-1313

MOVIE RECOMMENDER SYSTEM USING MATRIX FACTORIZATION METHOD

[1]Ms. Y. NAGA LAVANYA, [2]B.NISHITHA, [3]M.DURGAKSHARA, [4]M.VENISRI

[1]Assistant Professor, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad, ynagalavanya85@gmail.com

[2]BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

bojjanishitha@gmail.com

[3]BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

akshara1723@gmail.com

[4]BTech student, Dept.of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

manthinivenisri7@gmail.com

***Abstract**: Recently recommender system is emerging as a promising way of communication between user and massive amount of data. Recommendation system organizes the data in large amount to determine the concentration of analyst and make the information retrieval process easier for the users. For this purpose and criteria, large scale recommendation systems are available such as shopping website, movie, music etc. Many researchers have developed several algorithms like collaborative filtering, content-based filtering using Hadoop technology etc. The project introduces movie recommender system using both content-based and collaborative filtering approaches. This is designed and implemented our system based on the matrix factorization-based model.*

***Keywords**: Recommendation Systems; Matrix Factorization; Collaborative Filtering, Content-based filtering.*

I. INTRODUCTION

A recommendation system uses big data to suggest or recommend additional content to users. Recommendation system provides the facility to understand a person's taste and find new, desirable content for them automatically based on the pattern between their likes and rating of different movies. The era now living is called "era of abundance". For any given product, there are sometimes thousands of options to choose from. Think of the examples: streaming videos, social networking, online shopping; the list goes on. Recommender system helps to personalize a platform and help the user find something they like. The easiest and simplest way to do this is to recommend the most popular items. However, to really enhance the user experience through personalized recommendations, we need dedicated recommender systems.

Recommendation systems has been rapidly growing due to huge data available. We need a personalized systems that can be match to us, based on what we read recently, our last activities, and how its relevance to us. Recommendation systems become more popular today and has been used in various fields. Collaborative filtering has been the mostly used in recommendation systems nowadays. This method was able to give recommendation to user based on their similarity to others. On the other hand, matrix factorization has also been familiar since Netflix Grand Prize [1]. Generally, recommendation system can be divided into two parts, the Content-based Filtering and Collaborative Filtering. Content-based filtering based on the similarity of the items to the object that the user liked in the past [2]. Meanwhile, Collaborative filtering is the process of evaluation using the information of user behavior or the behavior of other users [3]. Collaborative filtering grouped by memory-based and model-based. Some of the central issues in the collaborative filtering is sparsity, content analysis, overspecialization and new user issue (cold start problem) [4]. Hybrid method was used in order to solve these problems. Hybrid is a combination between several methods. This method was also a key that brought Bellkor team as the winner in Netflix Prize in 2009.

This recommendation strategy is intuitive and easy to implement, and is suitable for various fields, such as books and movies. In recent years, there have been many algorithmic debates, which have been widely used. However, it remains a challenge to deal with the increasingly sparse user project scoring matrix and the cold start problem, which occurs when a user or project is newly added to a system with a small number of ratings. In order to solve these problems, researchers pay extensive attention to efficient feature extraction methods and auxiliary information, and propose various hybrid CF methods combining CF methods with various additional information sources. Among different CF methods, matrix factorization (MF) has become the most original and popular CF algorithm with its expansibility and flexibility . The traditional matrix decomposition model is considered to use edge information to improve the recommendation accuracy.

ISSN: 2366-1313

However, due to the limited learning ability and sparsity of side information, these methods are difficult to capture the complex relationship between users and items, and it is difficult to learn effective potential factors from side information.

Motivation

The purpose of "MOVIE RECOMMENDER SYSTEM" is to suggest relevant movies to users. To achieve this task, there exist two major categories of methods: collaborative filtering methods and content -based methods.

Significance of the Project

The need of this project is to help the users to get similar recommendations based on matrix factorization method. Latent features, the association between users and movies matrices, are determined to find similarity and make a prediction based on both movie and user entities.

II. LITERATURE SURVEY

In this section we briefly review the main works in the context. The list of references is not exhaustive due to the page limit. Our main focus was in collaborative filtering which has been successfully applied to several real-world problems, such as Netflix's movie recommendation.

Collaborative Filtering (CF) is basically based on evaluating and leveraging the relationships and interactions between a large number of users and items to make recommendations to other users or items, effectively identifying the preferred items of a particular user. However, the performance of the CF approach in the case of data sparsity and cold start is often challenged. In this case, many researchers suggest using user and item-related information, also known as auxiliary information, to improve recommendation performance.

In [5] The approach to develop Movie recommender system using the technique collaborative filtering. This approach is based on cosine similarity using k-nearest neighbour with the help of a collaborative filtering technique, at the same time removing the drawbacks of the content-based filtering. Although using Euclidean distance is preferred, cosine similarity is used as the accuracy of cosine angle and the equidistance of movies remain almost the same.

In [6] In this movie recommender system it helps users to find the movies of their choices based on movie experience. It is based on collaborative filtering approach

ISSN: 2366-1313

that makes use of the information provided by users, analyses them and then recommends the movies that is best suited to the user at that time. The recommended movie list is sorted according to the ratings given to these movies by previous users and it uses K-means algorithm for this purpose. MOVREC also help users to find the movies of their choices based on the movie experience of other users in efficient and effective manner without wasting much time in useless browsing.

In [7] This approach is based on how far the machine learning techniques can be applied on movie recommender system. Recommender systems are a useful alternative to search algorithms since they help users discover items they might not have found otherwise. Of note, recommender systems are often implemented using search engines indexing non-traditional data.

In [8] Designed the movie recommender system based on cuckoo search. In this research article, a novel recommender system has been discussed which makes use of k-means clustering by adopting cuckoo search optimization algorithm applied on the Movielens dataset. Our approach has been explained systematically, and the subsequent results have been discussed. It is also compared with existing approaches, and the results have been analyzed and interpreted. Evaluation metrics such as mean absolute error (MAE), standard deviation (SD), root mean square error (RMSE) and t-value for the movie recommender system delivers better results as our approach offers lesser value of the mean absolute error, standard deviation, and root mean square error.

In [9] This paper includes movie recommendation along with sentiment analysis by approaching machine learning algorithms. To enhance the user experience, this system performs sentiment analysis on the reviews of the movie chosen using machine learning. Two of the machine learning algorithms Naïve Bayes (NB) Classifier and support vector machine (SVM) Classifier are used to increase the accuracy and efficiency. This paper also gives a comparison between NB and SVM on the basis of parameters like Accuracy, Precision, Recall and F1 Score.

III. PROPOSED SYSTEM

There were several things that need to be done before the recommendation can be made. Firstly, it needs to extract the dataset, in order to process recommendations. This section will explain how the data will be used, a hybrid recommendation techniques and evaluation methods.

Models

The objective of the development phase is to convert the deliverables of the design phase into a complete system. The activities in the development phase translate the system design produced in the design phase into a working system. The development phase includes activities for developing the system, testing the system, and to ensure the system functions satisfy the functional process requirements. At the end of this phase, the system will be ready for the activities of the testing phase.

For our recommender system, we'll use both of the techniques mentioned above: content-based and collaborative filtering. To find the similarity between movies for our content-based method, we'll use a cosine similarity function. For our collaborative filtering method, we'll use a matrix factorization technique

CONTENT-BASED MOVIE RECOMMENDER SYSTEM

Content-based methods are based on the similarity of movie attributes. Using this type of recommender system, if a user watches one movie, similar movies are recommended. For example, if a user watches a comedy movie starring Adam Sandler, the system will recommend them movies in the same genre or starring the same actor, or both. With this in mind, the input for building a content-based recommender system is movie attributes.

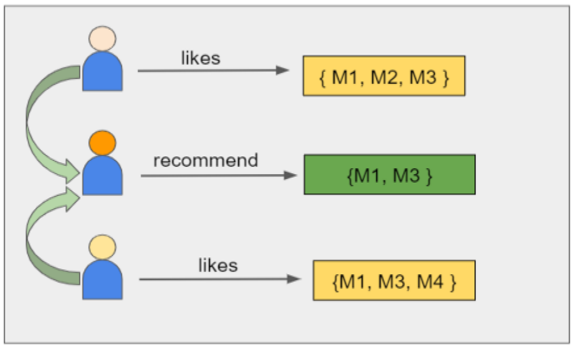


Fig.1 Content based movie recommendation system

Collaborative filtering

Collaborative filtering was one of the most widely used techniques for recommendation systems. Companies like Google, Amazon using the recommendation, at least in part based on collaborative filtering. CF was a popular recommendation algorithm that bases its predictions and recommendations on ratings or opinions of other users in the system. Typically, predictions about user interests were obtained by collecting taste information from many other similar users. The fundamental assumption of this method was that the tastes of other users can be selected and assembled in such a way to give a reasonable prediction of the active user's preference. CF requires a large amount of information to be processed, including large-scale data sets such as the ecommerce and web applications. Over the past few decades CF has been continuously improved and has become one of the most prominent personalization techniques in the field of recommendation systems.

Matrix Factorization

Collaborative filtering recommendation method has been widely used in the nearest neighbour (NNH) algorithm combined with the Pearson correlation or cosine similarity to calculate the predicted values. In terms of sparse rating, NNH approach usually experience difficulties in finding the right match and consequently produce a weak recommendation with accuracy. Furthermore, NNH calculation algorithm complexity tends to increase with number of users and the number of items, which means the recommendation system will have serious problems in scalability. To overcome this problem, a matrix factorization approach has been proven to be an efficient method for rating-based recommendation

In proposed system we discuss about modern recommender systems by combining both approaches content-based and collaborative filtering systems.

The basic process will look like this:

Step-1: Build a matrix factorization-based model

Step-2: Create handcrafted features

Step-3: Implement the final model

SYSTEM ARCHITECTURE

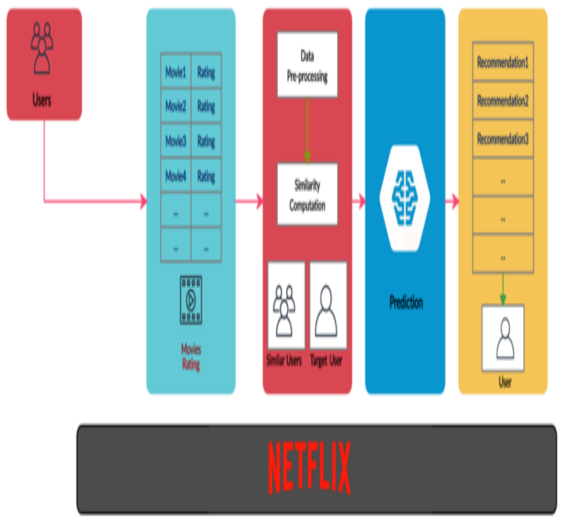


Fig.2 System architecture

The system generates movie predictions for its users, while items are the movies themselves. The primary goal of movie recommendation systems is to filter and predict only those movies that a corresponding user is most likely to want to watch. The basic concept behind a movie recommendation system is quite simple. In particular, there are two main elements in every recommender system: users and items.

IV. IMPLEMENTATION

Dataset

ISSN: 2366-1313

For our own system, we'll use the open-source datasets from Kaggle. This dataset contains 5K data points of various movies and credits.

We will use two datasets namely:

tmdb_5000_movies

tmdb_5000_credits

Vectorization

By performing a vector search, I will input one movie into the dataset. As a result, the software will find what the closest 5 points in space are, each one representing a similar movie: this is what we call a recommendation system.

To perform the vector search, I can use the sklearn API that provides me with the nearest neighbour algorithm.

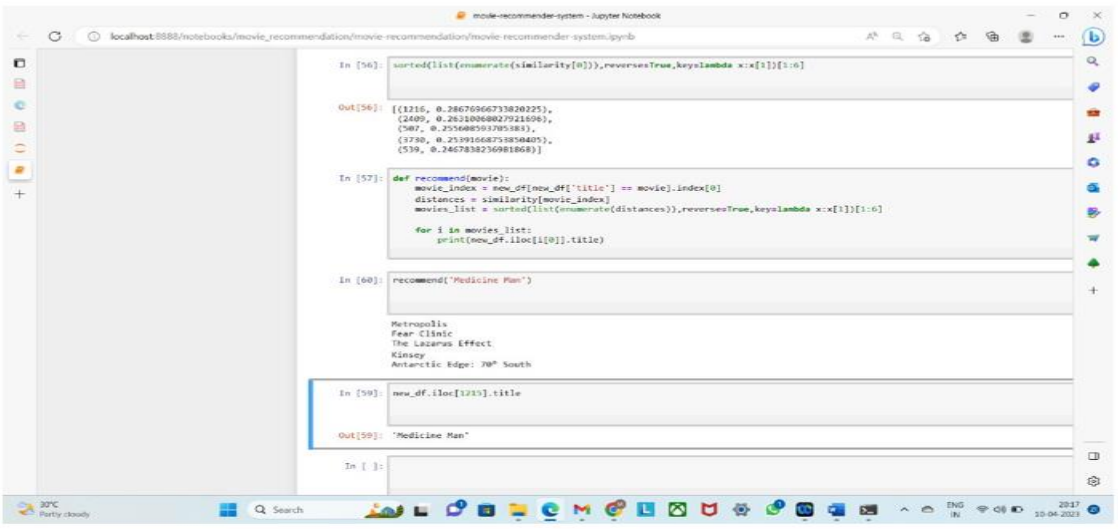
Cosine similarity

The Cosine Similarity from Sklearn, as the metric used to compute the similarity between two movies. Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.

Specific recommendations

Based on the user's input after performing vectorization and cosine similarity some specific recommendations are going to made to the use

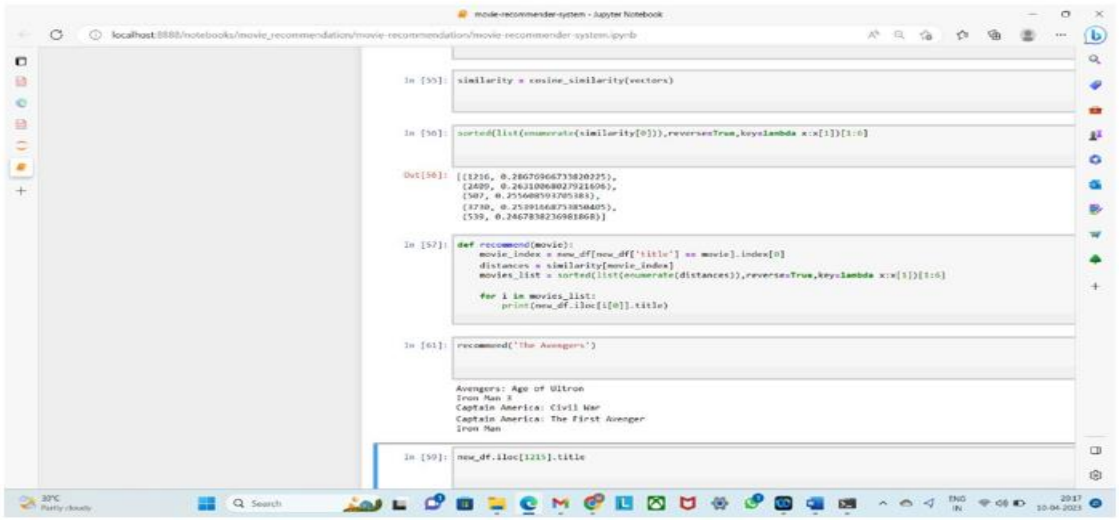
V. RESULTS



```
In [56]: sorted(list(enumerate(similarity[0])),reverse=True,key=lambda x:x[1])[1:6]

Out[56]: [(1216, 0.28676966733820225),
          (2409, 0.26310068027921696),
          (507, 0.255608593705383),
          (3730, 0.25391668753850405),
          (539, 0.2467838236981868)]

In [57]: def recommend(movie):
             movie_index = new_df[new_df['title'] == movie].index[0]
             distances = similarity[movie_index]
             movies_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]

             for i in movies_list:
                 print(new_df.iloc[i[0]].title)

In [60]: recommend('Medicine Man')

         Metropolis
         Fear Clinic
         The Lazarus Effect
         Kinsey
         Antarctic Edge: 70° South

In [59]: new_df.iloc[1215].title

Out[59]: 'Medicine Man'

In [ ]:
```

movie-recommender-system - Jupyter Notebook
localhost:8888/notebooks/movie_recommendation/movie-recommendation/movie-recommender-system.ipynb
30°C
Partly cloudy
Search
ENG
IN
20:17
10-04-2023

Fig.3 movie recommendation system loaded in jupyter notebook


movie-recommender-system - Jupyter Notebook
localhost:8888/notebooks/movie_recommendation/movie-recommendation/movie-recommender-system.ipynb
Search
10-04-2023

```
In [55]: similarity = cosine_similarity(vectors)

In [56]: sorted(list(enumerate(similarity[0])),reverse=True,key=lambda x:x[1])[1:6]

Out[56]: [(1216, 0.28676966733820225),
          (2409, 0.26310068027921606),
          (507, 0.2556083593705383),
          (3730, 0.25391668753850405),
          (539, 0.24678382369081868)]

In [57]: def recommend(movie):
             movie_index = new_df[new_df['title'] == movie].index[0]
             distances = similarity[movie_index]
             movies_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]

             for i in movies_list:
                 print(new_df.iloc[i[0]].title)

In [61]: recommend('The Avengers')

         Avengers: Age of Ultron
         Iron Man 3
         Captain America: Civil War
         Captain America: The First Avenger
         Iron Man

In [59]: new_df.iloc[1215].title
```

VI. CONCLUSION

Hence, we can conclude that the design phase of the movie recommender system gives us the information of all the processes used in the project and their relation.

The project helps to reduce complications, in handling cold-start problem. It is easy to recommend the movies compare to Hadoop technology. In this the project proposed a recommendation system for the large amount of data available on the web in the form of ratings, reviews, opinions, complaints, remarks, feedback, and comments about any movie using matrix factorization-based model. The project introduces movie recommender system using both content-based and collaborative filtering approaches. Here, we have designed and implemented our system based on the matrix factorization-based model.

REFERENCES

[1] Yehuda Koren. (2009) The BellKor Solution to the Netflix Grand Prize. [Online]. http://www.netflixprize.com/assets/GrandPrize2009_B PC_BellKor.pdf.

[2] Michael J Pazzani and Daniel Billsus, "Content-Based Recommendation Systems," The Adaptive Web Lecture Notes in Computer Science, vol. 4321, pp. 325-341, 2007.

[3] Michael D Ekstrand, John T Riedl, and Joseph A Konstan, "Collaborative Filtering Recommender Systems," Foundations and Trends in Human– Computer Interaction, vol. 4, no. 2, pp. 81 - 173, 2010.

[4] Gediminas Adomavicius and Alexander Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Transaction on

Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, 2005.

[5] Yehuda Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2008, pp. 426-434.

[6] Yehuda Koren, "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering," ACM Transactions on Knowledge Discovery from Data, pp. 1-24, 2010.

[7] Abhinandan Das, Mayur Datar, and Ashutosh Garg, "Google News Personalization: Scalable Online Collaborative Filtering.," in International World Wide Web Conference Committee (IW3C2), 2007, pp. 271- 280.

[8] Greg Linden, Brent Smith, and Jeremy York, "Amazon.com Recommendation: Item-To-Item Collaborative Filtering," 2003.

[9] Paolo Cremonesi, Roberto Turrin, and Fabio Airoldi, "Hybrid algorithms for recommending new items," in Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, New York, 2011, pp. 33-40.

[10] G Takács, I Pilászy, B Németh, and Domonkos Tikk, "Investigation of Various Matrix Factorization Methods for Large Recommender Systems," in IEEE International Conference on Data Mining Workshops, Pisa, 2008., pp. 553-562.