Movie Recommendation System

Mariyam Begom, Raihan Sultana, Sabrina Doha, Md. Tahmidur Rahman

Department of Computer Science and Engineering, Brac University

{mariam.rahman30.mr, raihansultana1418, psabrinadoha96, tahmidullas}@gmail.com

*Abstract*— Nowadays recommendation systems are in great use in order to find out a preference of a user. The most popular areas where this system is used is to recommend books, news, music, movies etc. In this paper we are proposing a personalized movie recommender system. We used user- based collaborating filtering method, where interest of a user is found using preference taken from many other users. In our system list of movies are recommended to a particular user, and the list is sorted out according to the number of ratings given by previous users. Top 25 names of movies with highest ratings will be suggested with their respective predicted rating. This helps the users to browse movies of their choice easily with less time.

Keywords—recommendation system; personalized movie recommender; collaborating filtering

# INTRODUCTION

# In today's world, high technology led to availability of huge number of choices for us. We often get lost in the information explosion and unable to choose the right option. Choosing which movie to watch next is such kind of problem. With a large number of movies available in sites, it is not possible for a user to decide which movies they may like. Also people no more have time to go through the reviews of all movies before selecting one. Hence recommending system model was first proposed to solve this kind of problems. It is a method of filtering out choices based on the user's interest which it learned from other user's rating style. A huge number of research is done on his topic to make it more efficient. Moreover, websites like Netflix made these systems a salient part of their site.

# Our system basically suggests a list of movies to a user according to their choice. Later the action and the feedback of the user will be stored in the database, which will further be used to generate new recommendation in the next user-system interaction. Nowadays a web based recommendation model is applied to provide different type of customized information to their user. These systems are implemented in different fields and applied in different websites to make day to day choices easier in less time.

There are different approaches to build our project, such as content based filtering and collaborative filtering. We implemented our project using collaborating filtering algorithm using spark's alternating least square method. Collaborating filtering itself is of two type, user based and item based. We chose user based for our implementation. Here users are categorized based on their rating, and users giving similar rating to similar movies are assumed to be like wise thinking users, hence movies will be suggested to user who did not watch the movie yet but liked by similar user.

# LITERATURE REVIEW

Initially, we went through several papers and websites in order to understand how the already existing recommender systems work and how they have been implemented. In one paper, from the Apache spark, MovieLens dataset was used to make a movie recommender using collaborative filtering with Spark’s Alternating Least Square implementation. It consisted two parts, in the first part movies and ratings data was collected and parsed into Spark RDDs, and in the second part they worked in building and using the system for later use. This paper was our main help source.

# PROPOSED METHODOLOGY

First of all, we built a popularity based model, i.e. where all the users have same recommendation based on the most popular choices. For this, we have used GraphLab recommender functions ‘popularity\_recommender’. For comparison, we built another model using the ‘Pearson’ similarity metrics from collaborative filtering model. This ‘Pearson’ similarity metric is also supported by GraphLab. We used the ‘spyder’ package from Anaconda and the language ‘python’ for executing this particular recommendation system.

## Datasets

The dataset was collected from the MovieLens website which contained:

* **100,000 ratings** (1-5) from 943 users on 1682 movies.
* Each user has rated **at least 20 movies.**
* Simple demographic info for the users (age, gender, occupation, zip)
* Genre information of movies

## Training and testing

## In order to make the models, train and set datasets were required. The MovieLens already provides with pre-divided data wherein the test data has 10 ratings for each user. Therefore, we loaded the pre-divided data and then converted them into SFrames (a scalable, out-of-core dataframe, which allows you to work with datasets that are larger than the amount of RAM on your system).

## Evaluating the proposed Models

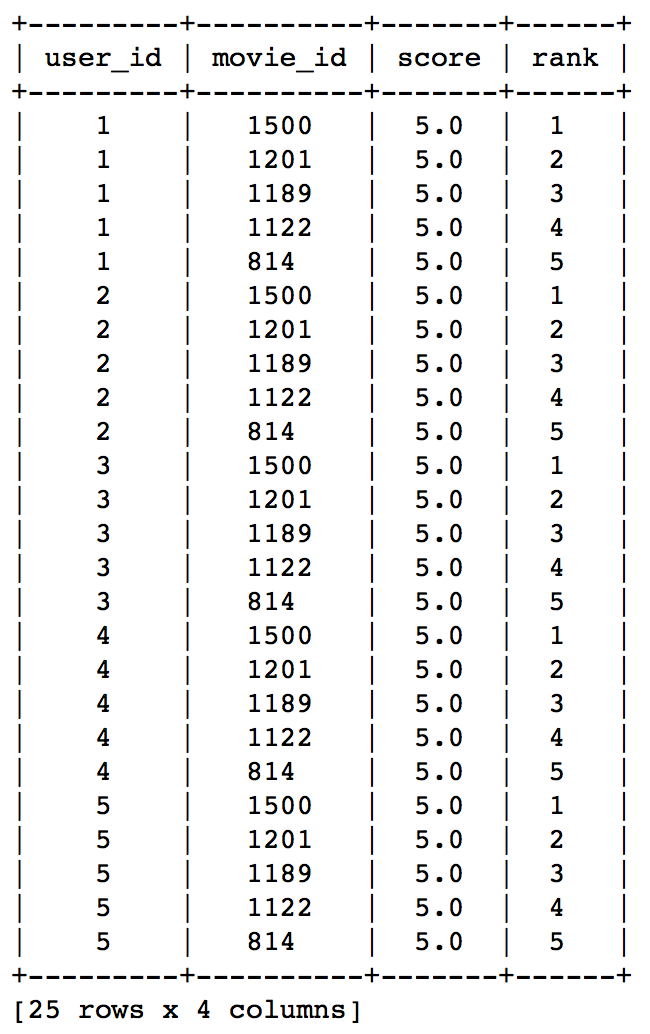
## For evaluation of the recommendation engines, we used the concept of precision-recall.

* **Recall:**
  + What ratio of items that a user likes were actually recommended.
  + If a user likes say 5 items and the recommendation decided to show 3 of them, then the recall is 0.6.
* **Precision**
  + Out of all the recommended items, how many the user actually liked?
  + If 5 items were recommended to the user out of which he liked say 4 of them, then precision is 0.8.

An idea recommender system is the one which only recommends the items which user likes. So in this case precision=recall=1. This is an optimal recommender and we should try and get as close as possible. Thus, our aim was to maximize both precision and recall.

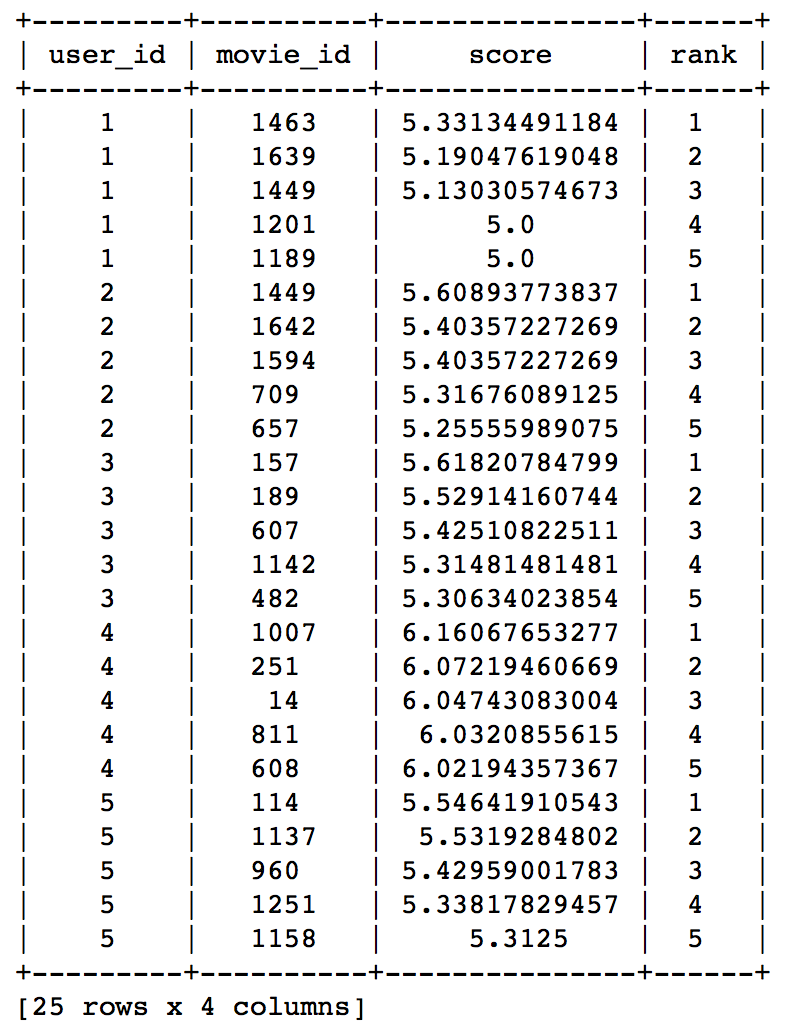
# RESULT

Starting with the recommendation results from popularuity based model, we recommended 5 highly rated movies to each user with particular ranking. The results are shown below:



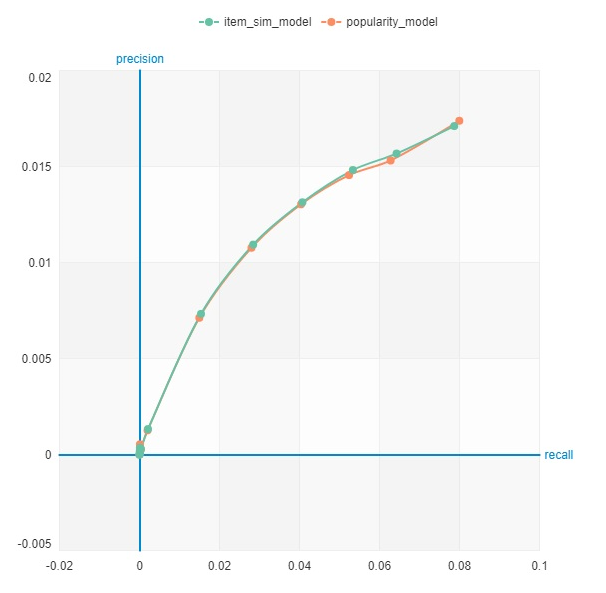
Here, we can see that same movies of Id 1500,1201,1189,1122 and 814 has been recommended to the users because these are the most popular movies and are highly rated by every other user.

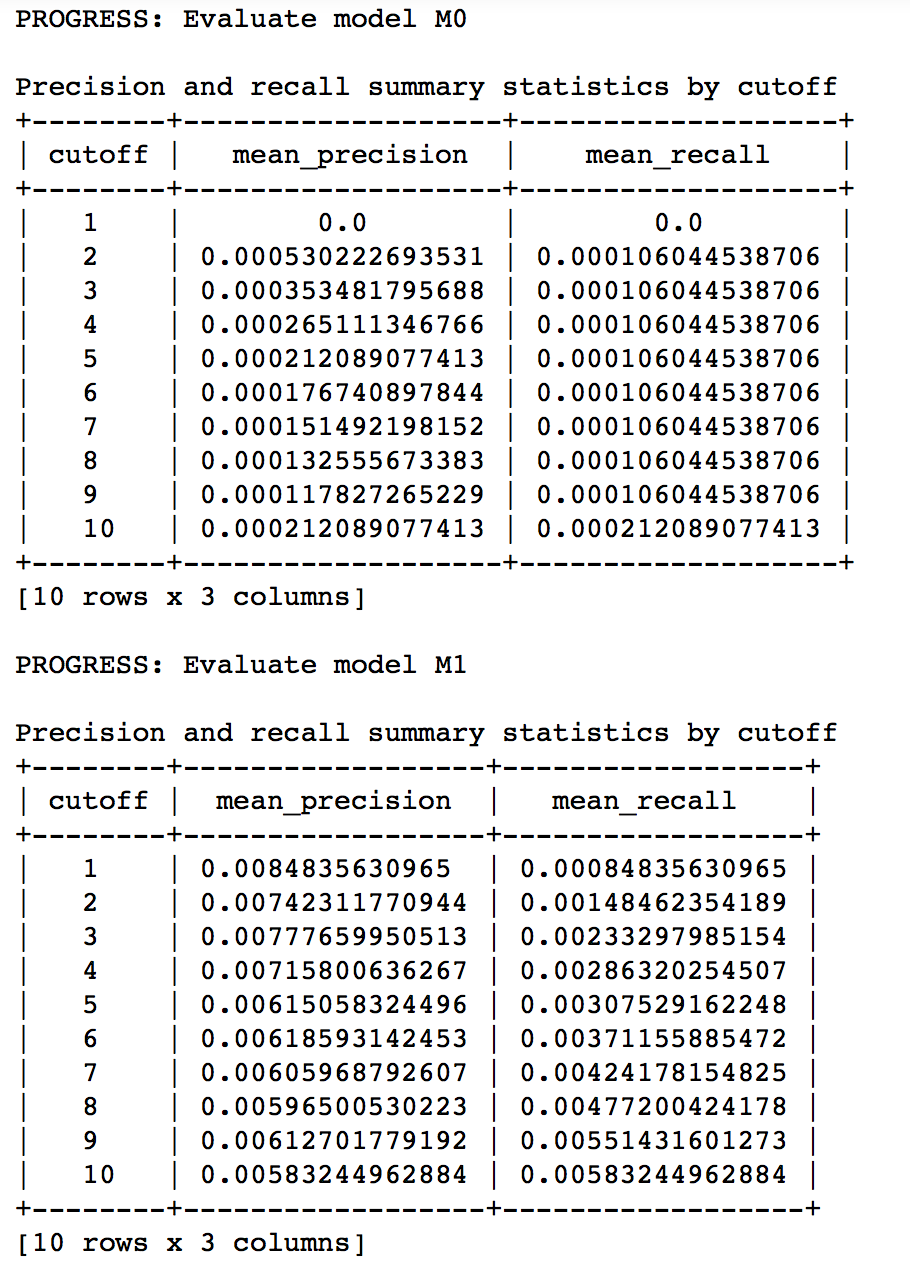
Coming to the similarity based model, we again recommended 5 movies but this time the recommendations were made as per the movies the user already liked. The results are shown below:



Here, we can see the recommendations are different for each user. Therefore, personalization exists.

Lastly, we see the results of evaluation of the proposed models below:





From the result tables, it can be easily observed that similarity based model is definitely better than the popularity based model since the precision and recall values are closer to the required values compared to those of the popularity model ones.

# CONCLUSION

Collaborative filtering is the most successful and popular algorithm in the recommender system's field. It helps users to make a better decision by recommending interesting items.

Our system's target is to provide the best possible recommendations to the users using the appropriate algorithms. The recommendation system helps the users to browse movies in a faster way.

##### VI. REFERENCES

[1] Bobadilla, J., Ortega, F. and Hernando, A. (2012), “A collaborative filtering similarity measure based on singularities”, Information Processing & Management, Vol. 48 No. 2, pp. 204-217. [[Google Scholar]](http://scholar.google.com/scholar_lookup?hl=en&publication_year=2012&pages=204-217&issue=2&author=J.+Bobadilla&author=F.+Ortega&author=A.+Hernando&title=A+collaborative+filtering+similarity+measure+based+on+singularities&) [[Crossref]](https://www.emeraldinsight.com/servlet/linkout?suffix=ref003&dbid=16&doi=10.1108%2FIJCS-08-2017-0021&key=10.1016%2Fj.ipm.2011.03.007) [[Infotrieve]](https://www.contentscm.com/vlib/order/OpenURLReceive.aspx?clientid=8710&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.volume=48&rft.issue=2&rft.atitle=A+collaborative+filtering+similarity+measure+based+on+singularities&rft.jtitle=Information+Processing+%26+Management&rft.date=2012)

[2] Feng, Z.J., Xian, T.A.N.G.A. and Feng, G.J. (2004), “An optimized collaborative filtering recommendation algorithm”, Journal of Computer Research and Development, Vol. 41, pp. 1842-1847. [[Google Scholar]](http://scholar.google.com/scholar_lookup?hl=en&publication_year=2004&pages=1842-1847&author=Z.J.+Feng&author=T.A.N.G.A.+Xian&author=G.J.+Feng&title=An+optimized+collaborative+filtering+recommendation+algorithm&)

[3 ]kumar Bokde D, Girase S, Mukhopadhyay D. Role of Matrix Factorization Model in Collaborative Filtering Algorithm:A Survey. International Journal of Advance Foundation and Research in Computer (IJAFRC). 2014;1(6):6. Available from: <http://arxiv.org/abs/1503.07475>.

* [View Article](http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0155739)
* [Google Scholar](http://scholar.google.com/scholar?q=Role+of+Matrix+Factorization+Model+in+Collaborative+Filtering+Algorithm%3AA+Survey+kumar+Bokde+2014)

[4] Huang, C.G., Yin, J., Wang, J., Liu, Y.B. and Wang, J.H. (2010), “Uncertain neighbors’collaborative filtering recommendation algorithm”, Chinese Journal of Computers, Vol. 33 No. 8, pp. 1369-1377. [[Google Scholar]](http://scholar.google.com/scholar_lookup?hl=en&publication_year=2010&pages=1369-1377&issue=8&author=C.G.+Huang&author=J.+Yin&author=J.+Wang&author=Y.B.+Liu&author=J.H.+Wang&title=Uncertain+neighbors%E2%80%99collaborative+filtering+recommendation+algorithm&) [[Crossref]](https://www.emeraldinsight.com/servlet/linkout?suffix=ref011&dbid=16&doi=10.1108%2FIJCS-08-2017-0021&key=10.3724%2FSP.J.1016.2010.01369) [[Infotrieve]](https://www.contentscm.com/vlib/order/OpenURLReceive.aspx?clientid=8710&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.volume=33&rft.issue=8&rft.atitle=Uncertain+neighbors%E2%80%99collaborative+filtering+recommendation+algorithm&rft.jtitle=Chinese+Journal+of+Computers&rft.date=2010)