MINI-PROJECT-REPORT

ON

"Movie Recommendation System"

Submitted by

BATCHU SATHVICK-19BEC009
INJETI PAVITRA-19BEC017
PANCHIREDDY MOHAN BABU-19BEC031
POLISETTY SRIRAM-19BEC033

Submitted to

Prof. Dr. Rajendra Hegadi
Associate Professor, Data Science and Intelligent Systems

Prof. Dr. Manjunath K V
Assistant Professor, Data Science and Intelligent Systems

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DHARWAD

CERTIFICATE

This is to certify that the Project Work entitled— "**MOVIE RECOMMENDATION SYSTEM**" is a bona fide work carried out by,

Batchu Sathvick-19bec009, Injeti Pavitra-19bec017, Panchireddy Mohan Babu-19bec031, Polisetty Sriram-19bec033, has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree, in fulfilment for the Mini Project of Bachelor of Technology in Computer Science & Engineering of the Indian Institute of Information Technology Dharwad during the year 2021-2022. The Project Report has been approved as it satisfies the academics prescribed for the Bachelor of Technology degree.

Signature of Supervisor(s)

Name(s)

Department(s)

(Month, Year)

ABSTRACT

The main motive of this project is to construct a recommendation system for "Movies" using recommendation algorithms which uses the user data or entire user database, find the patterns in the items or datasets and recommend the content other than which is already in the playlist or watch list. The general pipeline of the recommendation system starts with data extraction followed by data processing and silting through the items based on their similarities and finally recommending. There are three types of recommendation system algorithms, "Content-based", "Collaborative", and "Hybrid" models. The Hybrid model is focused in this report. This algorithm can be applied in OTT platforms, social media and in other applications. Movie Recommendation system 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.

Chapters

1.INTRODUCTION

In today's world the internet has become a major part of our daily life, users often face the problem of too many choices. Right from looking for a good watch to looking for good investment options, there is too many options available. To help the users cope with this sea of information, companies have developed recommendation systems to guide their users. Recommendation systems have been an asset and added to the profits of some major companies like Amazon, Netflix which have made these algorithms a major part of their websites.

NETFLIX	2/3rd of movies watched are recommended
AMAZON	Recommendations generates 38% more clicks-troughs
GOOGLE NEWS	38% sales from recommendations

High quality recommendation systems add a notch to the user experience and satisfaction. Recommendation systems usually classified into two broad categories:

- Content based filtering system
- Collaborative filtering system

1.1 Content-based filtering

Content-based filtering, usually builds a model which uses the user's past datasets and recommends the similar content matching the content in the dataset. In content-based filtering, to describe items we use keywords apart from user's profile to indicate user's preferred liked or dislikes. In other words, content-based filtering algorithms recommend those items or similar to those items that were liked in the past. It examines previously rated contents and recommends best matching content. For example, a user is given the option to select his choices from a set of attributes which include actor, director, genre, year and rating etc. We predict the user's choices based on the choices of the previous visited history of users.

1.2 Collaborative filtering

The collaborative filtering approach is very different, instead of recommending the content the user has liked in the past, this system recommends content preferences of other similar users. The similarity between the users is computed instead of the similarity between the content. A pure collaborative filtering system is that which does no analysis on the content at all, Collaborative filtering system can be stated as content-independent that is It only depends on the connections. It provides agreeable recommendations since, other users feedback influences what is recommended.

2. Methodology

2.1 Content-Based Recommendation

As we have seen above the basic idea of content-based filtering system is to recommend the content similar to what the user's previous preferences were, so the problem at hand is to calculate the similarity between the content. There are many methods to model this problem, but the commonly used is the Vector Space Model. This model extracts the keywords/tags of the item(content) and calculate the weight by TF-IDF. For example, suppose k_i is the *i*th keyword/tag of item d_j, w_{ij} is the weight of k_i for keyword of item d_j, then the content of d_j can be defined as:

$$\text{Content } (d_j) = \{w_{1j}, w_{2j}, \ldots..\}$$

As we stated before, content-based recommendation system recommends content that are similar to what the user liked before. So, the preferences of the user can be modelled with respect to the user history. Consider Content-based profile (u) as the preferences vector of the user u, the definition is defined as:

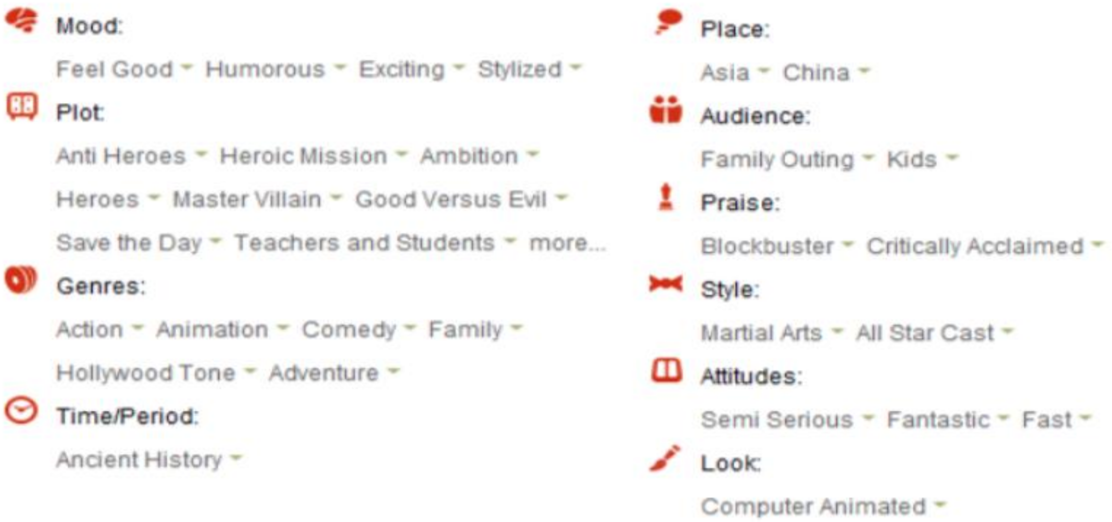
$$\text{Content-Based Profile(u)} = \frac{1}{|N(u)|}\, sum_{d\in N(u)}\text{Content(d)}$$

N(u) is the preference history of the user. After computing the content vector Content(.) and content preference vector Content Based Profile(.) of all users, suppose any user u and an item d, how the user liked the item(content) is given as the similarity between Content Based Profile (u) and Content (d):

$$\text{p (u, d) = sim (Content Based Profile (u), Content(d))}$$

Using keywords or tags to model item is an important step for most of the recommendation systems. Also, extracting keywords of an item is also a significant problem, especially in the case of media, because it is a complex process to extract text keywords from a video. There are two suitable solutions for this kind of problem, one is getting an expert to tag the items and the other way is to let the user tag the items. Some of the expert tagging systems are Pandora for music and jinni for movies. Taking jinni as an example, the researchers at jinni defined 900 and more tags as movie genre, these tags are made for them by movie experts. These tags account for different categories such as movie genre, plot, time, location and cast.

This is a figure from jinni, which are the tags for the movie Kung Fu Panda, as the figure shows, the tags for the movie are divided into ten categories in total, Mood, Place, Plot, Audience, Genres, Praise, Time, Style, Attitudes and Look. These tags contain the all aspects of the movie in total, which can accurately describe a movie

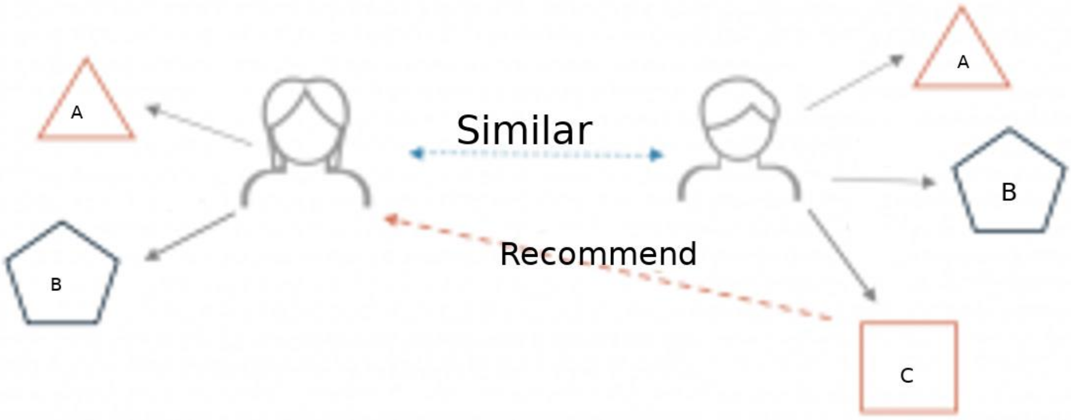


In comparison to the expert-tagged system the user-tagged system is widely used. The websites are Flickr and Delicious. The user-tagged systems are more diverse when compared to the expert-tagged system, but the catch is the quality of tags is lower, there is a lot of misinformation. The two main problems in the user-tagged system are, firstly the tag recommendation, which means when a user tags an item, the system can recommend some relative tags for him to choose. The first purpose is to be convincing to the user and to improve the tag quality, the other question arises how to recommend items based on tags. After tagging the items, the easiest way to recommend is by using tags as keywords of the items, and recommend using the content-based algorithm.

2.2 Collaborative Recommendation

The collaborative filtering system is based totally on the past preferences and not on the content. More specifically, it is based on the similarity between the preferences, choices and tastes of two users. It generalizes the similarity between the tastes and preferences of two users and makes recommendations based on that. Collaborative recommendation system is the work-horse of recommendation systems. The algorithm has a very fascinating property of being able to do feature learning by itself, which means that it can start learning for itself what features to use.

For suppose, two users buy a same product ‘X’ and ‘Y’ from an e-commerce website or store. After the purchase the similarity index between the two users is computed. Based on the result/score the recommendation system can recommend item/product ‘Z’ to the other user because it finds that the two users are same in terms of the products they purchased.



2.3 Hybrid Recommendation

Hybrid recommendation system is very popular and most of the recommendation systems use this approach. Combining the approach of content-based and collaborative filtering can be more effective, this can be done simply by taking the predictions of both the filtering systems and combining them, adding the best of both methods and getting a more precise recommendation system. These hybrid models are useful to overcome some common recommendation system problems like cold start, sparsity problem etc., following are some of the hybridization methods:

- Weighted: Adding scores of multiple recommendation components.
- Switching: Switching between multiple recommendation components to choose methods.
- Mixed: Show recommendation result from multiple systems.
- Features Combination: Extracting features from multiple sources and compiling them as a single input.
- Feature Augmentation: Compute features by one recommendation system and feed the result to the next step.
- Cascade: Generate some rough results by a recommendation technique and recommend on top of the previous result.
- Meta-level: Use the model outcome of one recommender which is used as the input to another recommender technique.

In spite of having several combinations theoretically, it might not always be efficient for a specific problem. The most important point of having a hybrid recommendation is to avoid the limitations of every other recommendation technique.

3. Results

When user watches a movie, our model recommends top 10 movies with highest hybrid cosine similarity score for instance the user has selected or watched Toy story which is an animated movie similar animated movies like Toy Story 2, Bugs Life, Monsters, Inc., Finding nemo etc., are recommended.

	content	**Collaborative**	**Hybrid**
Toy Story 2 (1995)	0.963633	0.736437	0.850035
Bug's Life,A (1998)	0.910092	0.640285	0.775189
Monsters,Inc(2001)	0.888226	0.613719	0.750973
Finding Nemo (2003)	0.874042	0.591571	0.732807
Incredibles,The (2004)	0.797007	0.565039	0.681023
Ice Age (2002)	0.875240	0.479267	0.677254
Ratatouille (2007)	0.903943	0.425328	0.664635
Antz (1998)	0.740393	0.566646	0.653519
Toy Story 3(2010)	0.878544	0.407997	0.643271
Sherk (2001)	0.639776	0.620458	0.630117
Cars (2006)	0.742309	0.409924	0.576117

After calculating cosine similarity between a movie vector with respect to all other movies. Weighted average is used to suggest top N movies to the user. The weighted average is basically the hybrid model score.

Weighted average = Cosine score(content) + cosine score(collaborative)/2

A good recommendation system is essential for better predicting the user's habit. Hybrid recommendation system is one of the most effective ways to improve our model. By combining different approaches above, we can better customize our model to fit the recommendation condition and dataset requirements.

4. Challenges

4.1 Challenges Faced

While developing any system the main goal is to satisfy the end user for which the system is being developed. We had to face certain complication while developing our system. The following are some of the challenges faced:

- To generate a dataset that has all the appropriate information about a particular movie or content
- The greatest challenge was to have a relevant recommended list.
- To make our system more diverse, so that it can satisfy users of different locations.

4.2 Overcome the Challenges

- For data collection, free online databases were intensively searched and relevant data-sets were extracted which were useful for our proposed system.
- For accurate movie recommendation to the user, Cosine similarity and SVD algorithm were applied with a pre-filter.
- In our database movies were included irrespective of the geographic location or language, so that users from all over the world can use our system.

5. Future Scope and Applications

Recommendation Systems have been developed for so many years, has a significant role and never entered a dull phase, the progress made in machine-learning, large scale network and high-performance computing is pushing this field towards development. The following are some of the aspects taken into consideration for future works:

- Introducing Machine-Learning applications:

  For future study, dynamic parameters will be added into the recommendation system, machine learning can be used to alter the weights of each feature automatically and most suitable weights are considered

- Introducing user-dislike movie list:

  The user data is very resourceful in movie recommendation system. In future more user data can be collected and generate a user dislike movie list. We can feed the dislike movie list into the recommendation system and shall be able to generate scores which can be added to previous results. Through this process we can improve the results of the recommendation system

- Introduce more accurate and proper features of the movie:

  Generally, a collaborative filtering recommendation use the rating instead of object features. In future extraction of features such as subtitles and colour from movies might be able to provide more accurate information for the movie

- Preferring collaborative filtering recommendation:

  After sufficient user data collection, collaborative filtering recommendation can be introduced, as we discussed collaborative filtering is based on the social information of users, which can be analysed in future research

6. Conclusion

The recommendation system has become more and more important because of the information overload, since this system is based on a hybrid approach which is a compilation of content-based and collaborative filtering, it will give explicit and serendipitous outcomes contrasted with different systems that are based on only the content-based or collaborative approach. In contrast, content-based systems are limited to the individuals and cannot recommend content from out of the box, whereas the collaborative filtering also has a limitation of cold start, that is the system has a difficulty in recommending when either the user or the content is new. While our model being a hybrid, uses both the filtering systems to deliver the best recommendation, subsequently allowing the user to explore more

We have also introduced the cosine similarity which is commonly used in the industry. For the weight features, we introduced TF-IIDF-DC which improves the representation of the movie, also the new approach for setting weight for the features, the movie can be represented more accurately by the TF-IIDF-DC which is the key point of our project. At the end of the project, we use K-NN and various metrics to evaluate the improvement of the new approach. It is illustrated that the new approach contributes positively according to the evaluation.

7. REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. Modern information retrieval, volume 463. ACM press New York, 1999.

[3] Shumeet Baluja, Rohan Seth, D Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In Proceedings of the 17th international conference on World Wide Web, pages 895–904. ACM, 2008. [4] Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 95–104. ACM, 2007. [5] Suvir Bhargav. Efficient features for movie recommendation systems. 2014.

[6] Robin Burke. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4):331–370, 2002.

[7] Robin Burke. Hybrid web recommender systems. In The adaptive web, pages 377–408. Springer, 2007.

[8] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems, pages 293–296. ACM, 2010.

[9] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. JAsIs, 41(6):391–407, 1990.

[10] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. Knowledge and data engineering, ieee transactions on, 19(3):355–369, 2007.

[11] Xu Hailing, Wu xiao, Li Xiaodong, and Yan Baoping. Comparison study of internet recommendation system. Journal of Software, 20(2):350–362, 2009.

[12] Anne Håkansson. Portal of research methods and methodologies for research projects and degree projects. In The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP 2013; Las Vegas, Nevada, USA, 22-25 July, pages 67–73. CSREA Press USA, 2013.

[13] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In IJCAI, volume 99, pages 688–693, 1999.
[14] Anna Huang. Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, pages 49–56, 2008.
[15] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. Recommender systems: an introduction. Cambridge University Press, 2010.
[16] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In Knowledge Discovery in Databases: PKDD 2007, pages 506–514. Springer, 2007.
[17] Joseph A Konstan. Introduction to recommender systems: Algorithms and evaluation. ACM Transactions on Information Systems (TOIS), 22(1):1–4, 2004.
[18] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434. ACM, 2008.
[19] Daniel Z Lieberman, Suena H Massey, Vilmaris Quiñones Cardona, and Kenneth P Williams. Predicting content preference: Applying lessons learned from the commercial web to therapeutic software. Cyberpsychology, 2(2), 2008.
[20] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 7(1):76–80, 2003.
[21] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Recommender systems handbook, pages 73–105. Springer, 2011.
[22] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. Introduction to information retrieval, volume 1. Cambridge university press Cambridge, 2008
[23] Christopher D Manning and Hinrich Schütze. Foundations of statistical natural language processing. MIT press, 1999.
[24] Bradley N Miller, Istvan Albert, Shyong K Lam, Joseph A Konstan, and John Riedl. Movielens unplugged: experiences with an occasionally connected recommender system. In Proceedings of the 8th international conference on Intelligent user interfaces, pages 263–266. ACM, 2003.
[25] Jiangang Yang Mingyong Liu. An improvement of tfidf weighting in text categorization. 2012.