Movie recommendation system using clustering mining with Python

Jayakumar Sadhasivam1*, Juan Manuel Cera2, R Deepa3, K Satheshkumar4, V Muthukumaran5, S Satheesh Kumar6 and M Angeline Kavitha7

1School of Information Technology and Engineering (SITE), Vellore Institute of Technology, Vellore, India.
2Universidad de la Costa, Barranquilla, Colombia.
3Chaitanya Bharathi Institute of Technology (A), Hyderabad, Telangana, India.
4School of Computing, Kalasalingam Academy of Research and Education, Tamilnadu – 626128, India.
5Department of Mathematics, School of Applied Science, REVA University, Bengaluru –64, India.
6Department of Computer Science, School of Applied Science, REVA University, Bengaluru – 64, India.
7Voorhees College, Vellore, Tamilnadu, India.
E-mail: jayakumarsvit@gmail.com*

Abstract. The research we are pursuing here is about building a system which we will use and mine a movie review database using concepts of statistics like Euclidean Distance algorithm. This will help us determine similar review according to the ratings. What we plan on achieving through this approach is to analyze the reviews to suggest a suitable critic according to one’s preferences and one’s taste. We will do this according to the review we select and our algorithm which we use upon it. This algorithm will be the key to find those similar reviews. We are going to use Python programming to program this experiment. We have chosen to pursue this because of not the reviewers being incorrect or correct but merely different according to their preferences and their lean towards a certain genre and we know that a reviewer should also match the taste of the person who want to feel benefited of the review itself. So, we plan to make system that could help with that.

1. Introduction
We are going to pursue this analysis with few angles here. These models would be –

- **Content based filtering** – This filtering we will be analyzing upon the information like data about the movies such as the given summary, tags, themes, and any data that acts as a representation of factual entities being part of the movies in the relation.

- **Popularity based filtering** – This filtering we will be analyzing upon the data telling about popularity, that is basically the ratings

- **Collaborative filtering** – This filtering with use the powerful algorithms like SVD and python libraries for the statistical analysis like ‘surprise’ aiming at less overall errors.
In the end we will use of these techniques to combine together a hybrid filtering technique which will aim reducing the limitations of all these techniques.

The dataset we are going to use will be a TMDB dataset with separate full and small datasets. This data however not as huge as dataset of other websites is quite filtered and mined and perfectly suitable for our analytics. Also given its nature it will be possible to analyze this data at a quite moderate computing power.

2. Methodology
The first we will go with a simple recommendation analysis which will simply make the use of the notion that a popular and critically acclaimed movie will be more likely chosen by the audience. Although as you will see later this system will be a key part in the user-based filtering we are aiming at.

Here we will use the IMDb rating algorithm made public
\[
\left( \frac{v}{v+m} \ast R \right) + \left( \frac{m}{v+m} \ast C \right)
\]
where,
\( v \) is the vote count
\( m \) is for minimum votes required to be in the database
\( R \) is the mean rating
\( C \) is the average vote count.

And as we go by the limitation we decide we can filter the movies according to the data attributes.

From here we move to the Content based filtering. As we told about for the attributes mentioned above being very basic, in our analysis; last filtering was quite useful as the it basically gave us the movies which were similar since the crew usually doesn’t change among sequels and the whole franchises were readily predicted well.

Here we used the movie description attributes and used cosine similarity to judge.

Then comes the Collaborative filtering where we used the ‘surprise’ python library with SVD algorithm which we used to calculate the error rate which when seemed fit we proceed to train our dataset with results.

Now this training worked as the key measure to make our final filtering method where we take the input user id and title of the movie and give the movies for the suited taste.

Now since this dataset will use the maps and links of the links dataset and the all the filtering methods’ result trained data, this data gives the personalized recommendation which are different for each instance.

3. Proposed method
The techniques mentioned above like Content based filtering and collaborative filtering are used in studies like these users. Some systems which base their recommendations upon preferences are referred [1]. Also we came across systems which rated implicitly or explicitly [2]. Similarly, we came across a system which suggested according to the past behavior of customer [3]. Such as LIBRA [4] recommended books based on their descriptions. Also Amazon [5] analyses a pattern to user’s purchases.

Also the SVD method of surprise library in python plays a major and easy tool in this research. The idea of Hybrid recommender cam with [6] the categorization in Burke’s with mentions to different types of Hybrid filtering. Usage of demographic information [7] pointed us toward using genres here. Also using relational algebra across databases came from the idea of LOD from [8]. Similarly, the
matching of synonyms and NLP was represented in [9]. For countering this problem, ontologies [10] and SVD [11] were used.

One main thing we looked upon also were fake ratings [12]. To deal with such attacks we had to give preference to item-based techniques. Also, we came to know about the information attacker needs for this like knowledge and size of data [13] and shilling attacks [14]. Also randomizing techniques were used [15] to protect identities. Also, for preventing the leak of information, ontologies and NLP techniques were used as told above [16]. Also, discreteness among the items was vital as well was randomness. This is where we integrated techniques for this result and used a different type of clustering for a better performance.

Although we referred the major milestones in this study like Netflix prize etc. the heavy data of that research can become a future scope of this one. As we know something like movie recommendation cannot be easily classified as accurate or not, we used error metrics to evaluate performance. We divided the data in folds for that. Also, we made our system context aware and used content-based algorithms and multiway algorithms. Similarly, we used clustering differently by combining content analysis and clustering and also used slopes with weights to determine ratings.

Latency is a key issue with these systems, so we combined category based and user-based approaches for that with clustering offline giving scalability. The information about MAE and Precision were there in. Also information about contextual precision and its effect on recommenders was valuable. Also, the future scope for analysis expression and signal analysis.

4. Results
The results of this study have been consistent for the model for having the same training and dataset, even for various inputs.

The mean square error root was found to be **0.92** and for a specific input we got the prediction metric of 2.7.
5. Conclusion
These recommendation methods although robust on their own combine together to form an engine combining abilities, advantages and reducing the disadvantages. Focusing on the metrics of different methods but not all at once and combining those methods in the sequence gave us this result. Also, this goes on to show that this methodology can be further used with other metrics such as metadata and data processed from movies.

References
[1] Adomavicius G and Tuzhilin A 2003 IEEE Trans. Knowl. Data Eng.
[2] Linden G, Smith B and York J 2003 IEEE Internet Comput.
[3] Muthukumaran V and Ezhilmaran D 2020 International Journal of Information Technology and Web Engineering 15(4) pp 18-36.
[4] Ganesh Gopal Deverajan, Muthukumaran V, Ching-Hsien Hsu, Marimuthu Karuppiah, Yeh-Ching Chung and Ying-Huei Chen 2021 Transactions on Emerging Telecommunications Technologies.
[5] Celma O and Herrera P 2008 RecSys ’08: Proceedings of the 2008 ACM conference on Recommender systems.
[6] Ziegler C N, McNee S M, Konstan J A and Lausen G 2005 Proceedings of the 14th International Conference on World Wide Web.
[7] Katarya R and Verma O P 2016 Phys A Stat Mech Appl. pp 182–90.
[8] Isinkaye F O, Folajimi Y O and Ojokoh Ba 2015 Egypt Inform J.
[9] Su X and Khosghofigaar TM 2009 Adv Artif. Intell.
[10] Shi YUE, Larson M and Hanjalic A 2014 ACM Comput Surv. pp 1–45.
[11] Fahad A, Alshatri N, Tari Z, Alamri a, Khalil I and Zomaya A 2014 IEEE Trans Emerg Top Comput.
[12] Hartigan J A and Wong M A 1979 J R Stat Soc.
[13] Jain A K, Murty M N and Flynn PJ 1999 ACM Comput Surv. pp 264–323.
[14] Mukhopadhyay A, Maulik U and Bandyopadhyay S A 2015 ACM Comput Surv.
[15] Kanungo T, Mount D M, Netanyahu N S, Piatko C D, Silverman R and Wu A Y 2002 IEEE Trans Pattern Anal Mach Intel.
[16] Yang X S and Deb S 2013 Comput Oper Res.