International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6617
***Article Received**: 28 February 2023 **Revised**: 30 March 2023 **Accepted**: 28 April 2023*

Automotive Movie Recommendation System based on Natural Language Processing

Dr. Pooja Bagane[1,*] Dr. Sudhanshu Gonge[2], Dr. Rahul Joshi[3], Obsa Amenu Jebessa[4]
[1]Department of CSE,
Symbiosis Institute of Technology, Constituent of Symbiosis International (Deemed University),
Pune, India
e-mail: poojabagane@gmail.com
[2]Department of CSE,
Symbiosis Institute of Technology, Constituent of Symbiosis International (Deemed University),
Pune, India
e-mail: sudhanshu.gonge@sitpune.edu.in
[3]Department of CSE,
Symbiosis Institute of Technology, Constituent of Symbiosis International (Deemed University),
Pune, India
e-mail: rahulj@sitpune.edu.in
[4]Faculty of Computing and Informatics
Jimma Institute of Technology,
Jimma, Oromia, Ethiopia
e-mail: obsa.amenu@ju.edu.et

Abstract—People are puzzled about which movie to watch these days because there are so many movies available on various OTT platforms. A recommender system would solve this problem by recommending the best movie to the user based on his genre, actor, director, and rating preferences. The cosine similarity principle would be used to guide the recommendation system. Apart from that, we will use the Tfidftransformer and count vectorizer from the sci-kit-learn library in Python in this work. In this study work, all of the approaches' constraints have been described. All of this work was done using datasets from several OTT platforms that were available on Kaggle.

Keywords- Cosine similarity; TfidfTransformer; count vectorizer; scikit-learn; Kaggle.

I. INTRODUCTION

The Recommendation Algorithm is a powerful and effective resource for those who want to make educated selections. It's a system that helps consumers access details that's relevant to them among a big amount of data. We use machines to manage projections based on a variety of computational data in order to help us get better outcomes. Even if a task isn't specifically planned to be done, this could be accomplished [1]. The process of obtaining the essential data gets time-consuming.

Web browsers help to tackle this type of issue, but they really do not resolve the issue of personalization. The Recommender Systems design is crucial in today's modern internet browsing, regardless of whether making a purchase out of a shopping site or watching a picture at something like an on-request facility [2].

Within everyday living, we very much depend on recommendations from other people, either it is by hearsay or responses to general questionnaires. People regularly use browser recommendation systems to make buying choices for items that are relevant to their tastes [3]. Suggestion expert systems are computer tools and procedures with the goal of making relevant and sensible suggestions for things or commodities which might be of importance to a bunch of people [4]. Alternatively, they are a sort of data filtering process that attempts to predict a product "liked" or "rated." In the recommender system, three techniques are typically utilized. Content-based screening is creating a user profile depending on the sort of content someone likes, and afterward proposing products based on that data [5]. Another method is CF, which includes categorizing people that are similar and then utilizing that information to make suggestions [6]. Hybrid systems utilize both of the aforementioned approaches to managing operational information in a more succinct way. Our objective is to produce precise suggestions with as little computing work as possible [7].

II. RELATIVE WORK

The following are some characteristics of typical recommendation systems approaches:

1. Content-Based
2. Collaborative Filtering (CF)
3. Hybrid Filtering

A. Content-Based

This method sorts things according to the viewer's preference. It displays individual comparisons of whether the viewer has evaluated it or not. Vector-Space-Model (VSM) is indeed the methodology implemented to describe the procedure. It introduces the notion of TF-IDF [14], which determines the item's resemblance out on its descriptions.

Tf (t)=
$$\frac{\text{frequency occurrence of term t in document}}{total\ number\ of\ terms\ in\ document}$$
If (t)=
$$log10\,\frac{\text{total number of document}}{number\ of\ documents\ containing\ term\ t}$$

Different approaches may be used to measure the correlation between element vectors:

1. Cosine Similarity Matrix
2. Euclidean Distance Method
3. Pearson’s Correlation Method

B. Collaborative Filtering (CF) Method

It is built on people who share common interests & produces an outcome for everyone. User-based: It is anticipated that somehow a viewer will love things that are appreciated by many other users with similar interests when using user-based CF. Item-based: This CF differs from other types of CF, in that it wants users to like products that really are related to products they have found compelling.

C. Hybrid Filtering

Hybrid filtering is a term that refers to a combination of different types of filtering. Filtering by collaboration and filtering by content is the case. Today's most prevalent and popular technique. It stays away from every single recommended technique that has a flaw. Several problems with these systems are:

Cold Start Issue: When a customer initially registers up for a service, he has not yet viewed any movies. As a consequence, the recommendation engine seems unable to generate findings for any film [8]. This would be referred to as the "cold-start" problem [9]. The recommendation engine experiences such a problem due to the absence of an old record. For every new customer, this happens once.

Data sparsity problem: This problem happens when a person only rates just a few items, making it very hard for the predictive algorithm to offer accurate results. The outcomes of this task aren't even similar to what was expected. Poor suggestions are generated whenever the mechanism refuses to deliver adequate outcomes. In addition, data insufficiency often results in coverage difficulties [10].

Scalability: Another difficulty that comes up when it comes to suggestions is scalability. Item encoding is done in a linear method in this situation. The technique works effectively whenever the set of data is limited in quantity. The recommendation engine struggles to deliver credible suggestions based on diverse genres as even the set of data develops. The scalability challenge can be solved using singular value decomposition, a dimensionality reduction approach (SVD). It also cuts down on the time it takes to produce a result while also ensuring that it is correct and efficient.

Synonym: When a large number of words have the same meaning, they’re referred to as synonyms. In this instance, the When a system can't distinguish among two identical words, it can't provide the desired outcome [11]. The terms movies and films, for example, have the very same sense but are treated differently by the recommendation system. As more than just a result, it is unable to generate reliable findings. Latent Value Decomposition (LVD) and Singular Value Decomposition (SVD), Synonymy can be resolved using semantic indexing. Other types of recommendation systems include: Ratings and genres are used to provide suggestions. As a result of other users' actions, it became tough to give precise outcomes. Collaborative filtering is the term for this. This is a typical recommendation system approach [12]. To make things easier for users and overcome this flaw, we utilized a different recommendation mechanism called content-based filtering. This method makes use of the user's own ratings and genres. It makes recommendations depending on the movies the user has previously watched. So because rating and themes are supplied purely by the viewer and are based exclusively on the user's choices, not even on the interests of everyone else, this helps us provide suitable recommendations. Due to their reliance on other users, previous studies lacked the accuracy of real suggestions. It is vital to note that recommendations can be tailored to a specific user and are guaranteed to be accurate because they are based on that user's interests and concerns. Our suggested system has the ability to hold a large amount of data while also producing relevant findings. Our recommendation engine looks for it and provides the finest films based on genres that really are similar to those we've seen.

III. Methodology

COSINE SIMILARITY MATRIX

Similarity measure relation between these two items is used to calculate the degree of cosine over them. It compares different items using a standardized metric. Computing a dot product between those 2 attributes is one way to do it [13].

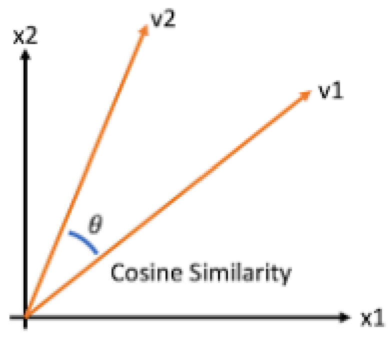


Figure 1. Cosine Similarity.

As shown in the Figure 1, the inclination among v1 & v2 is. The narrower the angle seen between vector fields, the stronger their resemblance. It indicates that even if the angle formed between 2 variables is minimal, then they are almost similar, but if the angle gets wide, the variables are very different.

Cosine Similarity can be defined as (Taking “A” and “B” as “Genre of Movie 1” and “Genre of Movie 2”):

$$\text{Similarity} = \cos(\theta) = A.B\frac{A.B}{||A||||B||}$$

The similarity function will calculate the angle between the 2 movies, the more the angle tends towards 1, the more the movies are similar.

We first created a data frame for storing the genre of the movies and then we apply the TFIDF vectorizer function to convert text to the vector form and thereafter, applied a cosine similarity function that will calculate the similarities between the movies. Now this result is stored in a variable called similar movies. By using the lambda function, we sorted it in descending order.

Now, whenever the user inputs the movie name, a similarity matrix is created according to the movie, and the top 10 movies are recommended to the user. In our case “Stranger Things” was entered as an input, giving the results - Chilling Adventures of Sabrina, Helix, Nightflyers, The Mystic River, Typewriter, Ju-On Origins, Goedam, Reality Z, Spectros, Anjaan: Special Crime Unit.

IV. RESULTS AND DISCUSSION

We employed Cosine Similarity & Content-based methods in the supervised learning architecture recommender systems to anticipate our results and propose films to viewers by applying algorithms in a Programming language, Python, utilizing NumPy & Pandas toolkits.

The method for determining how close two films are is dependent on the commonalities in many characteristics. It depicts the cos of an angle between vector fields reflected inside a multi-dimensional space quantitatively. This similarity measure is particularly useful since it aids in the discovery of related things.

The equation for recommending films using the Cosine similarity method.

$$CosSim(x,y) = \frac{\Sigma ixiyi}{\sqrt{\Sigma ixi^2}\sqrt{\Sigma iyi^2}}$$

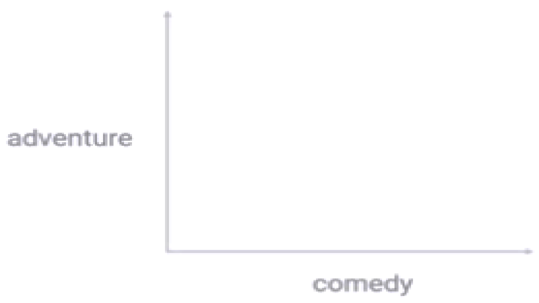


Figure 2. To demonstrate Cosine resemblance, we will use two films from distinct genres: adventurous and humor.

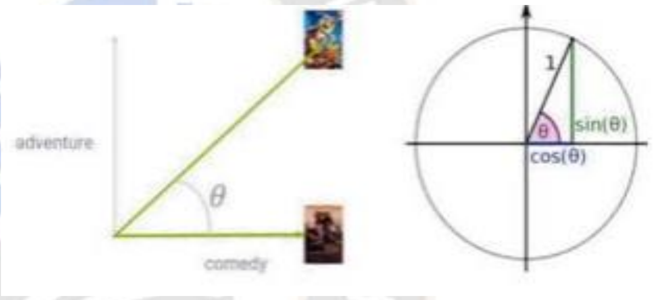
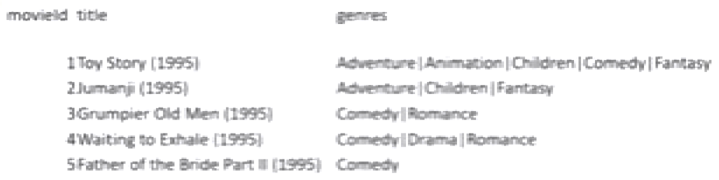


Figure 3. Cosine Similarity for combination of 2 genres.

The resemblance of these two films is determined by the angle theta between them. Theta is a number that spans from 0 to 1. It is most comparable if the theta value is close to 1. If it's close to 0, it's the least similar. The film's release date has yet to be determined. If it's near to 1, it's advised; otherwise, there's no use. They have a lot in common. It will suggest the best films based on the user's genre similarity. To obtain our distance calculation function, we used a standardized acceptance rating based on cosine similarity.



movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy

Figure 4. Films of various genres are selected, then their resemblance is determined using cosine similarity.

V. Conclusion

We were able to illustrate how to develop a movie recommendation engine by employing content-based screening. The concept metric cosine similarity is employed since it is more precise and also has a reduced degree of sophistication than some other different algorithms. Proposed methods have emerged as the greatest significant repository of valuable and reliable information available on the internet. Simple ones take just one or very few criteria into consideration, whereas increasingly complex stuff use several variables that refine the performance and make it highly user-friendly. Other methods like Neural Networks, could be used to create a robust movie recommendation engine.

Acknowledgment

The authors would like to thank Symbiosis International University, Jimma Institute of Technology, Ethiopia and Symbiosis Institute of Technology for their support and help in this research, and would also like to thank anonymous reviewers who made this study better with their valuable input.

References

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

[2] R. B. Yates, B. R.-Neto, et al. 'Modern information retrieval", volume 463. ACM Press New York, 1999.

[3] S. Baluja, R.Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. 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] X. Hailing, W. Xiao, L. Xiaodong, and Y. Baoping, "Comparison study of Internet recommendation system", Journal of Software, 20(2):350–362, 2009.

[5] T. E. D. Mining, "Enhancing teaching and learning through educational data mining and learning analytics: An issue brief," in Proceedings of a conference on advanced technology for education, 2012.

[6] Nakagawa and T. Ito, "An implementation of a knowledge recommendation system based on similarity among users profiles," in Proceedings of the Sice Conference, pp. 326–327 vol.1, 2002.

[7] T. K. Quan, I. Fuyuki, and H. Shinichi, "Improving the accuracy of recommender system by clustering items based on the stability of user similarity," in International Conference on Computational Intelligence for Modelling Control and Automation, 2006.

[8] M. Muozorganero, G. A. Ramezgonzlez, P. J. Muozmerino, andC. D Kloos, "A collaborative recommender system based on space-time similarities," vol. 9, no. 3, pp. 81–87, 2010.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web. ACM, 2001, pp. 285–295.

[10] G. Wang, "Survey of personalized recommendation system," Computer Engineering & Applications, 2012.

[11] Albadvi and M. Shahbazi, "A hybrid recommendation technique based on product category attributes," Expert Systems with Applications, vol. 36, no. 9, pp. 11 480–11 488, 2009.

[12] F. R. Hernandez and N. Y. G. Garcia, "Distributed processing using cosine similarity for mapping big data in Hadoop," IEEE Latin America Transactions, vol. 14, no. 6, pp. 2857–2861, 2016.

[13] A. Gupta, and Dr. B. K. Tripathy, "A generic hybrid recommender system based on neural networks", Advance Computing Conference (IACC), 21-22 February 2014.

[14] G. L. Giller, "The Statistical Properties of RandomBit-streams and the Sampling Distribution of Cosine Similarity", Giller Investments Research Notes, 2012.

[15] N. Rengaraj, C. M. Kavitha, S. Loganathan, and N.Muthurasu, "A study of existing systems of recommendation engines", National Conference of Emerging ComputingTechnologies and Applications, Vel Tech University, Avadi, Chennai. 05-06 April 2018.

[16] R. Bell, Y. Koren, and C. Volinsky, "Modelling relationships at multiple scales to improve the 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.

[17] S. Bhargav, "Efficient features for movie recommendation systems", 2014.

[18] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, "Recommender systems: an introduction", Cambridge University Press, 2010.

[19] J. A Konstan, "Introduction to recommender systems: Algorithms and evaluation", ACM Transactions on InformationSystems (TOIS), 22(1):1–4, 2004.

[20] P. Vilakone, D. Park, K. Xinchang, et al. "An Efficient movie recommendation algorithm based on improved k-clique", Hum.Cent. Comput. Inf. Sci. 8, 38 (2018).