Revolutionizing the User Experience Using Content-Based Recommendation System in Digital Platforms

[1]Tabish Rao, [2]Atika Gupta, [3]Yashvi Sharma, [4]Shiv Ashish Dhondiyal

[1, 2, 3]Graphic Era Hill University, Dehradun, Uttarkhand, India.
[4]Graphic Era Deemed to be University, Dehradun, Uttarkhand, India.

[1]tabishrao0609@gmail.com, [2]atika04591@gmail.com, [3]Yashvisharma2345@gmail.com
[4]Shivahsish1234@gmail.com

Abstract

In the current digital era, we are all bombarded with tons of data. Content-based recommender systems cut through the noise by suggesting personalized guidance based on user preferences, behavior, and previous engagement with content. Recommender systems play an important role in help users discover new and relevant content. Content-based systems try to help users broaden their exploration space by looking for similar items to the ones a user has interacted with and promoting those ads. These types of technologies are often found in e-commerce sites to recommend products based on users' prior purchase, clicks or viewed items. This improves the experience for customers while potentially increasing sales via targeted products. Content-based recommendations on sites like Netflix, Spotify and YouTube recommends movies, songs or videos to you based on your prior consumption.

Content-Based Recommender Systems are still an important part of creating personalized experiences, improving user engagement and solving use cases in multiple domains. This is advantageous for both users and institutions as the technology keeps progressing and the digital content is constantly increasing. Personalized suggestions contribute to boosting user engagement by delivering content that is relevant to the individual. This can lead to longer sessions, happier users, and increased retention rates for an application or platform.

Keywords: Bag of Words, Python, Count Vectorizer, Streamlit, NLP, Recommender System, Cosine Similarity.

1. Introduction

In an era where digital content libraries exist at all scales, it has never been more important to have effective algorithms for recommending movies. Recommender systems enhance the user experience by providing personalized suggestions based on user preference. Recommender systems evaluate user behaviour and interest in order to provide personalized dynamic suggestions, enhancing the user experience, and increasing satisfaction and engagement for movies products or information.

This research focuses on developing content-based movie recommender system using content-based filtering and natural language processing (NLP) techniques. Implementation Steps — Merging, Cleaning, and Process Textual Data This textual information — the movie overviews, genres, keywords, and cast — is parsed and converted into a standard set of ‘tags’ for each movie. In order to conduct text normalization, including lowercase, stemming, and so on, we use these techniques to make the features consistent and better quality.

The script measures the similarity between movies using cosine similarity on Bag-of-Words from movie tags.

Using the scikit-learn CountVectorizer to generate the numeric representation of the textual input that will be used for computing the similarity between both the input and the references/output. The result of this is a recommendation function that takes in a movie title and outputs films that have been ranked according to their cosine similarity score. The script also pickles the processed data and similarity matrix, which helps to efficiently store and reuse the recommendation model later on.

At last, deployed the recommender with Streamlit framework! However, with the help of Streamlit, it just takes the project to the next level and makes it an interactive web application. Streamlit makes it easy to convert data scripts into shareable web apps, here it enables the user to just input the movies they like and get instant recommendations.

2. Objective

The objective of this work is the design and programming of an interactive content-based movie recommendation system. Abstract Movie recommendation system plays a vital role in movie identifying process so that user can have a pleasant experience during watching movies. This system uses the content-based filtering criteria and Natural Language Processing techniques to provide the user the details of movie on the basis of user. The approach preprocesses and converts textual data about movies like summaries and then processes the cosine similarity among a Bag-of-Words representation of movie tags. The system also stores similarity matrices and serialized processed data for efficient storage.

By deploying the recommender system as a Streamline web application, the recommender can be made available to the users to automatically, quickly, easily, and on an, as-needed basis, provide for searching and recommending individualized and personalized selections of movies, creating for the users can easily exploitable and interactive movie search and discover interface [8], wherein the users can input some of their past movie watching experience and can be instantly provided back with specific movie recommendations.

One of the most important features of the research is a recommendation function that receives a user-specified movie as input and outputs a list of

3. Background and Related Work

With the increase in technology from time to time, so many researchers have developed movie recommendation systems using different methods and algorithms. These many initiatives, however, inspired me to try building a system of my own. Using the machine learning technique K-Means Clustering, the researchers suggested movies that should be watched in the year 2019. [1] Liu [2] Introduced another approach that can be used which is content-based filtering where the curator suggests movies based on the attributes of the movies. Nakhli [17] in 2019 studied a simple approach based on filtering algorithms and the like-to-view ratio for the movies. Then, in 2018, Ifada [3] proposed a solution to the scalability and sparsity issues of collaborative filtering approach. Based on relative weighting between similarities between the rating and same genre genres, the method combines those values together to estimate the number of rating entries while ensuring sparsity. In this standard, fuzzy c-means clustering is also applied to deal with the problem of scalability. Using recurrent neural networks, Kim [4] was able to group consumers with similar likes for movies in 2019. To achieve this, we used a pearson correlation coefficient. Films based on the Markovian Factorization of Matrix Processes [6]. Zhang [6] also proposed that this framework might be adapted to different kinds of collaborative

3.1 Proposed Algorithm

Our application represents the similarity between two separate items in a numerical value between 0 and 1 by utilizing the notion of Cosine Similarity, a straightforward but wonderful methodology.
Start with the Input Data Preprocessing i.e. removing unnecessary data and then all the relevant input tags are converted into a Bag of Words for further processing. Our model is based on the concept of Similarity Cosine where similarity between two movies is lies in the range 0-1 where 0 means the entities are absolutely different and1 means both are identical

The formula for Cosine Similarity is as given below:

$$\cos(t,e) = \frac{te}{||t||\,||e||} = \frac{\sum_{i=1}^{n} t_i e_i}{\sqrt{\sum_{i=1}^{n}(t_i)^2}\sqrt{\sum_{i=1}^{n}(e_i)^2}} \qquad (1)$$

Equation 1. Cosine Similarity Formula

The dataset used is the TMDb dataset from Kaggle and this dataset includes two different files namely movies, which contains 5000 movies, with the detailed information of each movie like its title, genre, overview, popularity, along with some other information like its original language, budget, tagline, and each movie is given its unique id and the other file is credits, containing the names of its cast and crew along with its id and title.

Firstly, the relevant columns are selected and cleaned by removing punctuation and extra spaces and finally the bag of words model is designed by concatenating actors, genre and overview in a single column and then this column is stemmed using PorterStemmer.

Table 1: Bag of Word Model

Spider-Man 3	the seemingli invinc spider-man goe
Tangled	when the kingdom' most wanted
Avengers: Age of Ultron	when toni stark tri to jumpstart a dormant peacekeep program , thing go
Harry Potter and the Half-Blood Prince	as harri begin his sixth year at hogwarts, he discov an old bo ok mark as 'properti of the half-blood prince', and
Batman v Superman: Dawn of Justice	fear the action of a god-lik super hero left unchecked, gotham city' own formidable, forc vigilant take on

Following the creation of the model table mentioned above, a cosine similarity matrix is created. A cosine similarity matrix compares the frequency of each word in two movies and assigns each one a score between 0 and 1, which essentially represents how similar the two films are. So, when a title is passed to the recommendation model, it finds its index in the similarity matrix and sorts all the movies in descending order then recommends movies from the top of the sorted list. The frontend of the application is deployed using python and Streamlit library. Streamlit is compatible with other major python libraries like scikit-learn, numpy, matplotlib, etc and enables us to display descriptive text and model outputs, visualize data and model performance and modify model inputs through the UI using sidebars.

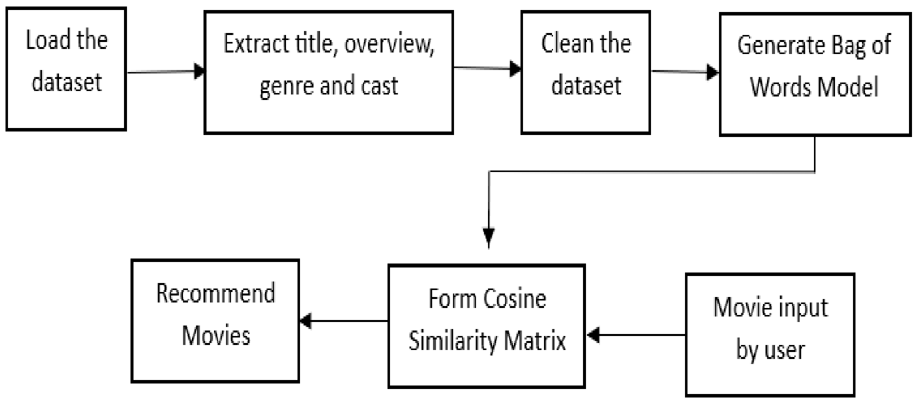

Figure 1. Flow of Model

4. Challenges Faced

The following challenges were faced during the model development:

- Having a system that is easy to use and comprehend is important.
- To locate a dataset that may satisfy users from various geographic regions and contains all pertinent information.
- To assign relative importance to certain traits.

4.1 Overcome the Problem

- A limited group of people have tested the proposed approach, and their feedback has been good.
- Our method has been maintained straightforward and user-friendly.
- We have chosen a dataset that includes interesting films from nearly every region in the world.
- We have utilized a cosine similarity matrix to provide the consumer with reliable movie recommendations.

5. Conclusion

Studies show that the time you spend on someone choosing a film to watch from a huge number of options is enough to watch more than half of a full-length movie. In turn, the average consumer was spending that hour he was saving each week doing something more productive. User interface has been constructed with the combination of buildings where user can interact with application. The user interface is short and simple, so frontend changes can be done in future easily. For future work, this constraint of the IMDB dataset should potentially be removed. This was a restriction that would become a hindrance if the product was ever to be shipped out to customers, as only the movies present within the dataset could be evaluated.

This project demonstrates a recommendation system that uses natural language processing and collaborative filtering to build a movie recommender system. Using a cosine similarity makes the system search for and match the movie titles textually. With collaborative filtering through user ratings, your recommendation is further improved as you know not only the users, but similar users and their likes also. Retaining accuracy as well as providing diversity in the movie suggestions to users, this technique is used to make sure that users get the right movie types according to the interest of the users. The integration of interactive widgets further improves the user experience, providing real-time recommendations based on user input. In general, this recommender system portrays an excellent blend of content based and collaborative filtering-based methods to provide the user with high quality personalized movie recommendation.

5.1 Future Scope

This current implementation of the recommendation system is a strong base for moving forward for enhancements for movie recommendations. Our open-ended future work includes for example applying deep learning methods – which could be very beneficial to many classes of recommender systems, e.g. neural collaborative filtering or RNNs – to further enhance the recommendation accuracy. These approaches can capture non-linear relations between the users and movies and hence provide more accurate recommendations.Mixing this data with user demography and contextual information (e.g., viewing time, location and view history) should help lead to more context-adaptive recommendations relevant to user current mood. However, increasing the dataset to include a more diverse range of genres will also help add more versatility to the system. It would make the system more versatile, as well as catering to a broader range of user-types and preferences. Also, external data sources of NZIR like social media activity or online reviews can be used provide enhancement to the recommendation process by considering user sentiments and social influence factors.Allow for user ratings and instant feedback on the recommendations → which will in turn enable dynamic self-tuning of the recommendation algorithms based on continuous learning. This feedback loop can actually help a lot in making the systems more flexible and responsive to changing choices by the user.In addition, using hybrid representation models which integrate collaborative filtering, content-based filtering and knowledge-based systems may deliver a more complete form of recommendation. Using these models, one can avoid the drawbacks of individual techniques and use their strengths to provide more accurate and varied suggestions.

We could also create a more instinctive and interactive End-user interface that can enhance user interactivity and happiness. Enhanced user experience can be achieved by incorporating features like personalized recomendation, dashboards, advanced search filters and social sharing.Consequently, these upcoming improvements can really give some speed to the RF system, enabling it to be more adaptive, personalized and user centric, thus improving the user experience as a whole by rendering a more enjoyable and engaging experience.

Reference

1. Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 263-268). IEEE.
2. Liu, G., & Wu, X. (2019, March). Using Collaborative Filtering Algorithms Combined with Doc2Vec for Movie Recommendation. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 1461-1464). IEEE.
3. Ifada, N., & Prasetyo, E. H. (2018, November). Employing Sparsity Removal Approach and Fuzzy CMeans Clustering Technique on a Movie Recommendation System. In 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM) (pp. 329-334). IEEE.
4. Kim, M., Jeon, S., Shin, H., Choi, W., Chung, H., & Nah, Y. (2019, June). Movie Recommendation based on User Similarity of Consumption Pattern Change. In 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (pp. 317-319). IEEE. [5] Hossain, M. A., & Uddin, M. N. (2018, September). A Neural Engine for Movie Recommendation System. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT) (pp. 443-448). IEEE.
5. Zhang, R., & Mao, Y. (2019). Movie Recommendation via Markovian Factorization of Matrix Processes. IEEE Access, 7, 13189-13199.
6. Zhao, W., Wang, B., Yang, M., Ye, J., Zhao, Z., Chen, X., & Shen, Y. (2019). Leveraging Long and Short-Term Information in Content-Aware Movie Recommendation via Adversarial Training. IEEE transactions on cybernetics.
7. Chen, Xiaojie, Pengpeng Zhao, Yanchi Liu, Lei Zhao, Junhua Fang, Victor S. Sheng, and Zhiming Cui. "Exploiting Aesthetic Features in Visual Contents for movie recommendations." IEEE Access 7(2019):49813- 49821
8. For deployment purposes, Streamlit documentation was referred: API Reference - Streamlit Docs
9. Dhamecha, M., Dobaria, K., & Patalia, T. (2019, March). A Survey on Recommendation System for Bigdata using MapReduce Technology. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (pp. 54-58). IEEE.
10. Haruna, K., Ismail, M. A., Suyanto, M., Gabralla, L. A., Bichi, A. B., Danjuma, S., ... & Herawan, T. (2019). A Soft Set Approach for Handling Conflict Situation on Movie Selection. IEEE Access.
11. Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., ... & Oh, P. (2019, January). T-RECSYS: A Novel Music Recommendation System Using Deep Learning. In 2019 IEEE International Conference on Consumer Electronics (ICCE) (pp. 1-6). IEEE.
12. He, M., Wang, B., & Du, X. (2019). HI2Rec: Exploring Knowledge in Heterogeneous Information for Movie Recommendation. IEEE Access, 7, 30276-30284.
13. Zhang, J., Wang, Y., Yuan, Z., & Jin, Q. (2019). Personalized real-time movie recommendation system:Practical prototype and evaluation. Tsinghua Science and Technology, 25(2), 180-191
14. Nakhli, R. E., Moradi, H., & Sadeghi, M. A. Movie Recommender System Based on Percentage of View. In 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI) (pp. 656-660). IEEE