International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 06 | June - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Movie Recommendation System Using Machine Learning

(Content-based Approach)

Badal Chauhan & Shubham Sharma

Student[1], and Student,[2] Master of Computer Application, Graphic Era Hill University (UGC Affiliated) Dehradun, Uttarakhand, India

ABSTRACT

Movie recommendation systems are essential in today's world to provide personalized movie suggestions to users, reducing the time they spend searching for content. While Content-based Filtering is a popular technique for making recommendations, existing research uses a single text to vector conversion technique and a single method to find the similarity between vectors. In this research paper, we propose a novel approach that employs multiple text to vector conversion techniques and combines the results of several algorithms to generate the final movie recommendation list. This hybrid approach using Content-based Filtering technique enhances the accuracy of the recommendation system, providing users with more relevant movie suggestions. Through experiments on a real-world dataset, we demonstrate that our proposed approach outperforms the traditional Content-based Filtering technique and other state-of-the-art recommendation methods. Our research contributes to the field of machine learning-based recommendation systems and provides insights for future research in this domain.

Keyword: - *Movie Recommendations, Content-based Filtering, Text to vector, Vector similarity.*

1. INTRODUCTION

In today's era of information overload, it can be overwhelming to choose what content to consume. This is particularly true for online platforms such as YouTube, where numerous videos are available on a particular concept, and it can be difficult to determine which one is the best fit for a user's needs. However, recommendation systems have made it easier by suggesting relevant items to users. These systems are used in various online platforms like YouTube, Amazon, Flipkart, Netflix, and Amazon Prime to recommend videos, products, and movies. The traditional movie recommendation systems have several limitations, which result in providing irrelevant recommendations to the users.

This research paper focuses on the logic behind the movie recommendation system, traditional movie recommendation techniques, their limitations, and a proposed solution for an Artificial Intelligence-based personalized movie recommendation system. To achieve this, we explore popular movie recommendation datasets like Movielens dataset, TMDB Movie Dataset, and Netflix's dataset. These datasets contain vast amounts of data, and we need filtering techniques to extract useful insights. Different filtering techniques or movie recommendation algorithms can be used in a recommendation system, including Content-Based Filtering, Collaborative Filtering, and Hybrid Filtering.

In 2009, Netflix organized a competition to improve its recommendation system with a prize money of $1M, highlighting the importance of a robust recommendation system for businesses. This research paper focuses on Content-Based Filtering technique, which uses attributes of an item to recommend similar items to a user. Despite its popularity, the traditional Content-Based Filtering technique has its own limitations. In this research paper, we propose a novel approach to overcome these limitations by leveraging machine learning algorithms to improve the accuracy of movie recommendations. This research paper contributes to the field of machine learning-based recommendation systems and provides insights for future research in this domain.

2. LITERATURE REVIEW

The collaborative filtering approach, as noted by Sang-Min Choi and colleagues in [1], is not without limitations such as the sparsity problem or the cold-start problem. To address these issues, the authors proposed leveraging category information in a movie recommendation system based on genre correlations. By using genre information, even newly created content without sufficient ratings or views can still appear in recommendation lists. This solution is designed to be impartial, making it possible for a new movie to be recommended alongside highly rated and widely watched content.

In [2], George Lekakos and colleagues proposed a hybrid approach to movie recommendation to address the limitations of content-based filtering and collaborative filtering techniques. They developed a movie recommendation system called "MoRe" which combines both methods. To ensure that the collaborative filtering method was not purely based on Pearson correlation coefficient, a new formula was used. However, this formula resulted in a "divide by zero" error when users gave the same rating to movies, so the authors excluded such users from their analysis. In contrast, the content-based filtering technique used cosine similarity to consider factors such as movie writers, cast, directors, producers, and genre. The authors implemented two variations of the hybrid recommendation method: "substitute" and "switching". Both variations consider collaborative filtering as the primary approach and use content-based filtering only when certain criteria are met.

In [3], Debashis Das and colleagues conducted a survey on recommendation systems, providing an overview of the various types available. They discussed both personalized and non-personalized systems, and explained the differences between user-based collaborative filtering and item-based collaborative filtering using a relevant example. Additionally, the authors detailed the advantages and disadvantages of different recommendation systems.

In [4], Jiang Zhang and colleagues introduced a collaborative filtering approach for movie recommendation called "Weighted KM-Slope-VU". They employed K-means clustering to divide users into similar clusters, and then selected a virtual opinion leader from each cluster to represent all users within that cluster. By processing a smaller matrix consisting of the virtual opinion leader's ratings rather than the complete user-item rating matrix, the authors were able to reduce the time required to generate recommendations. The authors proposed a unique algorithm to process this smaller matrix.

In [5], S. Rajarajeswari and colleagues explored various types of recommendation systems, including Simple Recommender System, Content-based Recommender System, and Collaborative Filtering based Recommender System. They ultimately proposed a solution consisting of a Hybrid Recommendation System

that incorporates both cosine similarity and SVD. The authors' system generates 30 movie recommendations using cosine similarity, which are later filtered based on SVD and user ratings. The system takes into account only the user's most recent movie selection, as the authors developed an approach that considers only one movie as input.

In [6], Muyeed Ahmed and colleagues presented a movie recommendation solution that leverages the K-means clustering algorithm. The authors grouped similar users into clusters and then built a neural network for each cluster to make personalized recommendations. The proposed system comprises several steps, including Data Preprocessing, Principal Component Analysis, Clustering, Data Preprocessing for Neural Network, and Building Neural Network. User rating, user preference, and user consumption ratio were all taken into account. After the clustering phase, the authors used a neural network to predict the ratings that users might give to unwatched movies, and recommendations were generated based on predicted high ratings.

The authors of the research paper [7], Gaurav Arora, et. al., proposed a movie recommendation system based on user similarity, but they did not provide much detail on how their system works. In the methodology section, they briefly mentioned City Block Distance and Euclidean Distance, but did not explain other techniques or parameters used in the system. The authors claimed that their system uses a hybrid approach of context-based filtering and collaborative filtering, but did not provide any specifics on the implementation of this approach.

V. Subramaniyaswamy and his co-authors [8] have presented a personalized movie recommendation solution that employs collaborative filtering technique. They use Euclidean distance metric to identify the most similar user based on the ratings of movies. The user with the smallest value of Euclidean distance is selected and recommendations are made based on the movies that the user has rated highly. The authors have also stated that the recommendations are updated over time to reflect the evolving preferences of the user.

In their research paper, Harper, et. al. [9], provided information about the popular MovieLens Dataset, which is frequently used for movie recommendation tasks. This dataset comes in several versions such as MovieLens 100K / 1M / 10M / 20M / 25M / 1B Dataset and contains various attributes like user ID, item ID / movie ID, rating, timestamp, movie title, IMDb URL, release date, and more, including the genre information for each movie.

In their research paper, Ms. Neeharika Immaneni and colleagues [10] introduced a hybrid movie recommendation approach that combines content-based filtering and collaborative filtering in a hierarchical manner to provide personalized recommendations to users. A unique aspect of their work is that they used a sequence of images to describe the plot of a movie, which enhances the visual experience. The authors also discussed several types of recommender systems, such as graph-based, content-based, collaborative filtering, and genre-based systems. Their proposed algorithm consists of four main phases, which involve gathering user interests from social networking sites, analyzing movie reviews, making recommendations, and generating story plots for better visualization.

3. PROPOSED METHODOLOGY

The first step in the proposed methodology is to perform preprocessing on the dataset, followed by the consolidation of relevant features into a single feature. Subsequently, the text from this feature will be converted into vectors. The next step involves computing the similarity between these vectors, and based on the system architecture mentioned below, recommendations will be generated.

3.1 Architecture

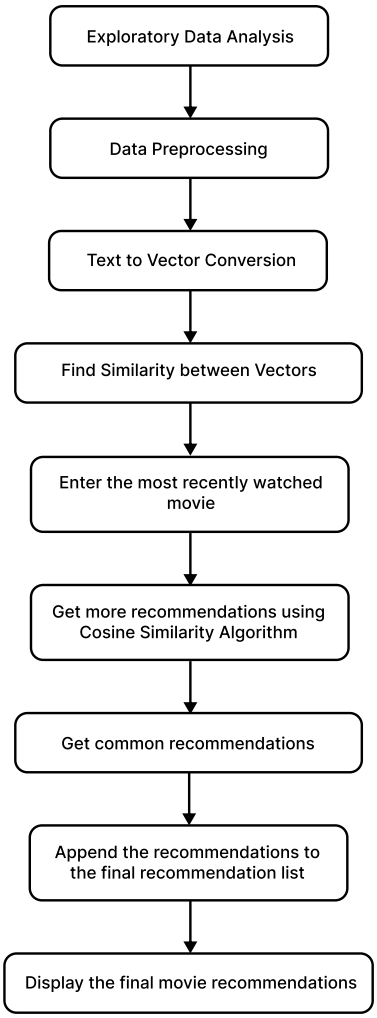

Fig. 1. System Architecture

3.2. Exploratory Data Analysis & Pre-Processing

The research work considers the "TMDB 5000 Movie Dataset" for the purpose of movie recommendations. The dataset is publicly available on kaggle.com and comprises of two CSV files, namely "tmdb_5000_movies.csv" and "tmdb_5000_credits.csv."

The "tmdb_5000_movies.csv" dataset contains several attributes, including:

- "budget": This attribute denotes the budget allocated for the movie production.
- "genres": It indicates the different genres of the movie, such as Action, Documentary, etc. A movie can have multiple genres.
- "homepage": This attribute indicates the homepage of the movie, which is essentially a website link.
- "id": This is a unique identifier assigned to each movie in the dataset.
- "keywords": This attribute signifies the keywords associated with the movie, which provide a brief summary of the plot.
- "original_language": This attribute indicates whether the movie was originally created in English or some other language.
- "original_title": This denotes the original title of the movie.
- "overview": This attribute provides a short description of the movie.
- "popularity": This is a metric that signifies the popularity of the movie.
- "production_companies": This attribute contains the names of the companies that have produced the movie.

```
movies.head(3)
```

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_comp
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ing Film Partners
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Walt I Pictures", "id":
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[{"name": "Col Pictures", "

Fig. 2. Statistical data about ‘tmdb_5000_movies.csv’ dataset

```
movies.iloc[20]
```

```
budget                                                  215000000
genres                [{"id": 28, "name": "Action"}, {"id": 12, "nam...
homepage                                 http://www.theamazingspiderman.com
id                                                            1930
keywords              [{"id": 1872, "name": "loss of father"}, {"id"...
original_language                                               en
original_title                                  The Amazing Spider-Man
overview              Peter Parker is an outcast high schooler aband...
popularity                                                89.866276
production_companies  [{"name": "Columbia Pictures", "id": 5}, {"nam...
production_countries  [{"iso_3166_1": "US", "name": "United States o...
release_date                                              2012-06-27
revenue                                                   752215857
runtime                                                       136.0
spoken_languages             [{"iso_639_1": "en", "name": "English"}]
status                                                     Released
tagline                                     The untold story begins.
title                                         The Amazing Spider-Man
vote_average                                                    6.5
vote_count                                                     6586
Name: 20, dtype: object
```

Fig. 3. Glimpse of the 'tmdb_5000_movies.csv' dataset

The 'tmdb_5000_credits.csv' dataset consists of the following attributes:

- "movie_id": This is a unique identifier assigned to each movie in the dataset.
- "title": This attribute denotes the title of the movie.
- "cast": This attribute comprises of the cast members who appear in the movie. The cast includes actors and actresses who play various roles in the movie.
- "crew": This attribute includes the people involved in the production of the movie, such as directors, producers, editors, etc.

```
credits.head(3)
```

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...

Fig. 4. Statistical data about 'tmdb_5000_credits.csv' dataset

```
credits.iloc[20]
```

```
movie_id                                                      1930
title                                           The Amazing Spider-Man
cast          [{"cast_id": 56, "character": "Peter Parker / ...
crew          [{"credit_id": "5395a60dc3a368641d004492", "de...
Name: 20, dtype: object
```

Fig. 5. Glimpse of the 'tmdb_5000_credits.csv' dataset

To prepare the text data for analysis or modeling, several preprocessing steps can be performed. These steps include:

- Stopword removal: eliminating common words that do not contribute much to the meaning of the text, such as "a", "an", "the", etc.
- Name combination: merging the first name and last name of individuals into a single name, if applicable.
- Punctuation removal: deleting characters like periods, commas, and question marks, which do not carry significant meaning in the context of the analysis.
- Lowercasing: converting all letters to lowercase to standardize the text and avoid potential mismatches caused by case sensitivity.
- Tokenization: splitting the text into individual words or tokens to enable further analysis.
- Stemming or lemmatization: reducing each word to its base form to normalize variations (e.g., "walk", "walks", "walking" all become "walk").
- Removing numerical or special characters: taking out any non-alphabetical or non-word characters from the text, such as numbers, symbols, or emojis.

By applying these preprocessing techniques, the text data can be cleaned, standardized, and made more suitable for analysis or modeling purposes.

```
new_df = movies[['movie_id', 'title', 'tags']]
```

```
new_df
```

	movie_id	title	tags
0	19995	Avatar	in the 22nd century, a parapleg marin is dispa...
1	285	Pirates of the Caribbean: At World's End	captain barbossa, long believ to be dead, ha c...
2	206647	Spectre	a cryptic messag from bond' past send him on a...
3	49026	The Dark Knight Rises	follow the death of district attorney harvey d...
4	49529	John Carter	john carter is a war-weary, former militari ca...
...	...	...	...
4804	9367	El Mariachi	el mariachi just want to play hi guitar and ca...
4805	72766	Newlyweds	a newlyw couple' honeymoon is upend by the arr...
4806	231617	Signed, Sealed, Delivered	"signed, sealed, delivered" introduc a dedic q...
4807	126186	Shanghai Calling	when ambiti new york attorney sam is sent to s...
4808	25975	My Date with Drew	ever sinc the second grade when he first saw h...

4806 rows × 3 columns

Fig. 6. Director, Keywords, Cast and Genres of a movie are combined into a single feature titled ‘tags’

The ‘tags’ attribute needs to be further processed by using some algorithms.

3.3. Algorithms

There are various methods to convert text data into vectors, such as CountVectorizer, TfidfVectorizer, Glove, or Word2Vec. Once the text has been transformed into vectors, the similarity between the vectors needs to be determined. This can be done using techniques such as Cosine Similarity, sigmoid_kernel, or other methods.

To implement a content-based recommendation system, we can use two algorithms. The first algorithm involves using CountVectorizer to convert the preprocessed text in the 'tags' attribute into vectors. The Cosine Similarity technique is then used to find the similarity between the vectors. The second algorithm involves using TfidfVectorizer to convert the preprocessed text in the 'tags' attribute into vectors. Again, the Cosine Similarity technique is used to determine the similarity between the vectors.

To generate recommendations, we will select the movies with the highest similarity scores and present them to the user. We can adjust the number of recommendations by specifying the desired number of top results.

4. RESULT AND ANALYSIS

```
recommend('Avatar')
```

```
Aliens vs Predator: Requiem
Aliens
Falcon Rising
Independence Day
Titan A.E.
```

Fig. 7. Recommendations similar to ‘Avatar’ movie using CountVectorizer and Cosine Similarity

```
recommend('The Dark Knight')
```

```
The Dark Knight Rises
Batman Begins
Batman Returns
Batman Forever
Batman
```

Fig. 8. Recommendations similar to ‘The Dark Knight’ movie using CountVectorizer and Cosine Similarity

5. CONCLUSION

Based on the results obtained using only CountVectorizer and Cosine Similarity, it can be observed that this single algorithm was effective in generating recommendations.

However, it is important to note that combining the outputs of multiple algorithms can further improve the quality of recommendations.

6. REFERENCES

[1] Choi, S-M., Ko, S-K., & Han, Y-S. (2012). Investigating genre correlations for movie recommendation. Expert Systems with Applications, 39(9), 8079-8085.

[2] Lekakos, G., & Caravelas, P. (2008). A hybrid approach to improve movie recommendation. Multimedia Tools and Applications, 36(1), 55-70.

[3] Das, D., Sahoo, L., & Datta, S. (2017). A survey on recommendation systems. International Journal of Computer Applications, 160(7), 1-6.

[4] Zhang, J., et al. (2019). Practical prototype and evaluation of a personalized real-time movie recommendation system. Tsinghua Science and Technology, 25(2), 180-191.

[5] Rajarajeswari, S., et al. (2019). Movie Recommendation System. In Emerging Research in Computing, Information, Communication and Applications (pp. 329-340). Springer.

[6] Ahmed, M., Imtiaz, M. T., & Khan, R. (2018). Movie recommendation system using clustering and pattern recognition network. In 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 152-156). IEEE.

[7] Arora, G., et al. (2014). Movie recommendation system based on users' similarity. International Journal of Computer Science and Mobile Computing, 3(4), 765-770.

[8] Subramaniyaswamy, V., et al. (2017). A personalized movie recommendation system based on collaborative filtering. International Journal of High-Performance Computing and Networking, 10(1-2), 54-63.

[9] Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TIIS), 5(4), 1-19.

[10] Immaneni, N., Padmanaban, I., Ramasubramanian, B., & Sridhar, R. (2017). A meta-level hybridization approach to personalized movie recommendation. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2193-2200). IEEE.