International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 07 | July - 2021 ISSN: 2582-3930

Movie Recommendation System using Content-based Filtering with a Hybrid Approach

Yeole Madhavi B.[1], Rokade Monika D.[2] Khatal Sunil S.[3]

[1]*Department of Computer Engineering, SPCOE Dumbarwadi (Otur), Maharashtra, India*
[2]*Department of Computer Engineering, SPCOE Dumbarwadi (Otur), Maharashtra, India*
[3]*Department of Computer Engineering, SPCOE Dumbarwadi (Otur), Maharashtra, India*

---***---

Abstract -There is already enough content available on the movie recommendation system. Showing the movie recommendations is essential so that the user need not waste a lot of time searching for the content which he/she might like. Thus, movie recommendation system plays a vital role to get user personalized movie recommendations. After searching a lot on the internet and referring to a lot of research papers, we got to know that the recommendations made using Content-based Filtering are using a single text to vector conversion technique and a single technique to find the similarity between the vectors. In this research work, we have used multiple text to vector conversion techniques and manipulated the results of the multiple algorithms to get the final recommendation list. You can think of it as a hybrid approach using the Content-based Filtering technique only.

***Key Words*:**Movie Recommendations, Content-based Filtering, Text to vector, Vector similarity, Hybrid approach

1.INTRODUCTION

Due to abundance of information collected till 21st century and the increasing rate of information flowing over the internet, there is a lot of confusion related to what to consume and what not to consume. Even on YouTube, when you want to watch a video of a particular concept, generally, there are a lot of videos available out there for you. Now, since the results are ranked appropriately, there may not be much issue but what if the results were not ranked appropriately? Well, in that case, we would probably spend a lot of time to find the best possible video which suits us and satisfies our need. This recommendation results are when you search something on a website. Next time, when you visit a particular website, without even searching, sometimes the system is able to show you recommendations which you might like. Isn't this an interesting feature? So, basically, the job of a recommender system is to suggest the most relevant items to the user. Recommendation systems are used in YouTube for video recommendation, Amazon and Flipkart for product recommendation, Netflix and Amazon Prime for movie recommendation, and so on. Whatever you do on such websites, there is a system which see your behavior and then ultimately suggest things / items with which you are highly likely to engage. This research paper deals with movie recommendations and logic behind movie recommendation system, traditional movie recommendation systems, issues related to traditional movie recommendation systems, and a proposed solution for Artificial Intelligence based personalized movie recommendation system. A lot of famous movie recommendation related datasets are already available on Kaggle and other websites. Some of the famous datasets include Movielens dataset, TMDB Movie Dataset, and the dataset by Netflix itself. Websites like Netflix, Amazon Prime, etc. use movie recommendation to increase their revenue or profits by ultimately improving the user experience. In fact, there was a competition conducted by Netflix in the year 2009 with a prize money of nearly 1 million dollars ($1M) for making at least 10% improvement in the existing system.

As dealt earlier, we have a lot of data available at our exposure and we need to filter the data in order to consume it because generally we are not interested in each and everything available to us. In order to filter the data, we need some filtering techniques. There are different types of filtering techniques or movie recommendation algorithms over which a recommendation system can be based upon.

Major filtering techniques or movie recommendation algorithms are as follows:

1. Content Based Filtering
2. Collaborative Filtering
3. Hybrid Filtering

Some of these techniques can be further broken into subparts

2.LITERATURE REVIEW

Sang-Min Choi, et. al. [1] mentioned about the shortcomings of collaborative filtering approach like sparsity problem or the cold-start problem. In order to avoid this issue, the authors have proposed a solution to use category information. The authors have proposed a movie recommendation system which is based on genre correlations. The authors stated that the category information is present for the newly created content. Thus, even if the new content does not have enough ratings or enough views, still it can pop up in the recommendations list with the help of category or genre information. The proposed solution is unbiased over the highly rated most watched content and new content which is not watched a lot. Hence, even a new movie can be recommended by the recommendation system.

George Lekakos, et. al. [2] proposed a solution of movie recommendation using hybrid approach. The authors stated that Content based filtering and Collaborative filtering have their own shortcomings are can be used in a specific situation. Hence, the authors have come up with a hybrid approach which takes into consideration both content-based filtering as well as collaborative filtering. The solution is implemented in 'MoRe' which is a movie recommendation system. For the sake of pure collaborative filtering, Pearson correlation coefficient has not been used. Instead, a new formula has been used. But this formula has an issue of 'divide by zero' error. This error occurs when the users have given same rating to the movies. Hence, the authors have ignored such users. In case of pure content-based recommendation system, the authors have used cosine similarity by taking into consideration movie writers, cast, directors, producers and the movie genre. The authors have implemented a hybrid recommendation method by using 2 variations - 'substitute' and 'switching'. Both of these approaches show results based on collaborative filtering and show recommendations based on content-based filtering when a certain criterion is met. Hence, the authors use collaborative filtering technique as their main approach.

Debashis Das, et. al. [3] wrote about the different types of recommendation systems and their general information. This was a survey paper on recommendation systems. The authors mentioned about Personalized recommendation systems as well as non-personalized systems. User based collaborative filtering and item based collaborative filtering was explained with a very good example. The authors have also mentioned about the merits and demerits of different recommendation systems.

Jiang Zhang, et. al. [4] proposed a collaborative filtering approach for movie recommendation and they named their approach as 'Weighted KM-Slope-VU'. The authors divided the users into clusters of similar users with the help of K-means clustering. Later, they selected a virtual opinion leader from each cluster which represents the all the users in that particular cluster. Now, instead of processing complete user-item rating matrix, the authors processed virtual opinion leader-item matrix which is of small size. Later, this smaller matrix is processed by the unique algorithm proposed by the authors. This way, the time taken to get recommendations is reduced.

S. Rajarajeswari, et. al. [5] discussed about Simple Recommender System, Content-based Recommender System, Collaborative Filtering based Recommender System and finally proposed a solution consisting of Hybrid Recommendation System. The authors have taken into consideration cosine similarity and SVD. Their system gets 30 movie recommendations using cosine similarity. Later, they filter these movies based on SVD and user ratings. The system takes into consideration only the recent movie which the user has watched because the authors have proposed a solution which takes as input only one movie.

Muyeed Ahmed, et. al. [6] proposed a solution using K-means clustering algorithm. Authors have separated similar users by using clusters. Later, the authors have created a neural network for each cluster for recommendation purpose. The proposed system consists of steps like Data Preprocessing, Principal Component Analysis, Clustering, Data Preprocessing for Neural Network, and Building Neural Network. User rating, user preference, and user consumption ratio have been taken into consideration. After clustering phase, for the purpose of predicting the ratings which the user might give to the unwatched movies, the authors have used neural network. Finally, recommendations are made with the help of predicted high ratings.

Gaurav Arora, et. al. [7] have proposed a solution of movie recommendation which is based on users' similarity. The research paper is very general in the sense that the authors have not mentioned the internal working details. In the Methodology section, the authors have mentioned about City Block Distance and Euclidean Distance but have not mentioned anything about cosine similarity or other techniques. The authors stated that the recommendation system is based on hybrid approach using context based filtering and collaborative filtering but neither they have stated about the parameters used, not they have stated about the internal working details.

V. Subramaniyaswamy, et. al. [8] have proposed a solution of personalized movie recommendation which uses collaborative filtering technique. Euclidean distance metric has been used in oder to find out the most similar user. The user with least value of Euclidean distance is found. Finally, movie recommendation is based on what that particular user has best rated. The authors have even claimed that the recommendations are varied as per the time so that the system performs better with the changing taste of the user with time.

Harper, et. al. [9] mentioned the details about the MovieLens Dataset in their research paper. This dataset is widely used especially for movie recommendation purpose. There are different versions of dataset available like MovieLens 100K / 1M / 10M / 20M / 25M /1B Dataset. The dataset consists of features like user id, item id / movie id, rating, timestamp, movie title, IMDb URL, release date, etc. along with the movie genre information.

According to R. Lavanya, et. al. [10], in order to tackle the information explosion problem, recommendation systems are helpful. Authors mentioned about the problems of data sparsity, cold start problem, scalability, etc. Authors have done a literature review of nearly 15 research papers related to movie recommendation system. After reviewing all these papers, they observed that most of the authors have used collaborative filtering rather than content-based filtering. Also, the authors noticed that a lot of authors have used hybrid-based approach. Even though a lot of research has been done on recommendation systems, there is always a scope for doing more in order to solve the existing drawbacks.

Ms. NeeharikaImmaneni, et. al. [11] proposed a hybrid recommendation technique which takes into consideration both content-based filtering approach as well as collaborative filtering approach in a hierarchial manner in order to show a personalized movie recommendation to the users. The most unique thing about this research work is that the authors have made movie recommendations using a proper sequence of images which actually describe the movie story plot. This actually helps for better visuals. The author have also described the graph based recommendation system, content-based approaches, hybrid recommender systems, collaborative filtering systems, genere correlations based recommender system, etc. The proposed algorithm has 4 major phases. Initially, social networking website like Facebook is used to know the user interest. Later, the movie reviews needs to be analysed and the recommendations needs to be made. Finally, story plot needs to be generated for better visuals.

Md. Akter Hossain, et. al. [12] proposed NERS which is an acronym for neural engine-based recommender system. The authors have done a successful interaction between 2 datasets carefully. Moreover, the authors stated that the results of their system are better than the existing systems because they have incorporated the usage of general dataset as well as the behaviour-based dataset in their system. The authors have used 3 different estimators in order to evaluate their system against the existing systems.

3. PROPOSED METHODOLGY

We need to perform preprocessing on the dataset and combine the relevant features into a single feature. Later, we need to convert the text from that particular feature into vectors. Later, we need to find the similarity between the vectors. Finally, get the recommendations as per the system architecture mentioned below.

1.1. ARCHITECTURE

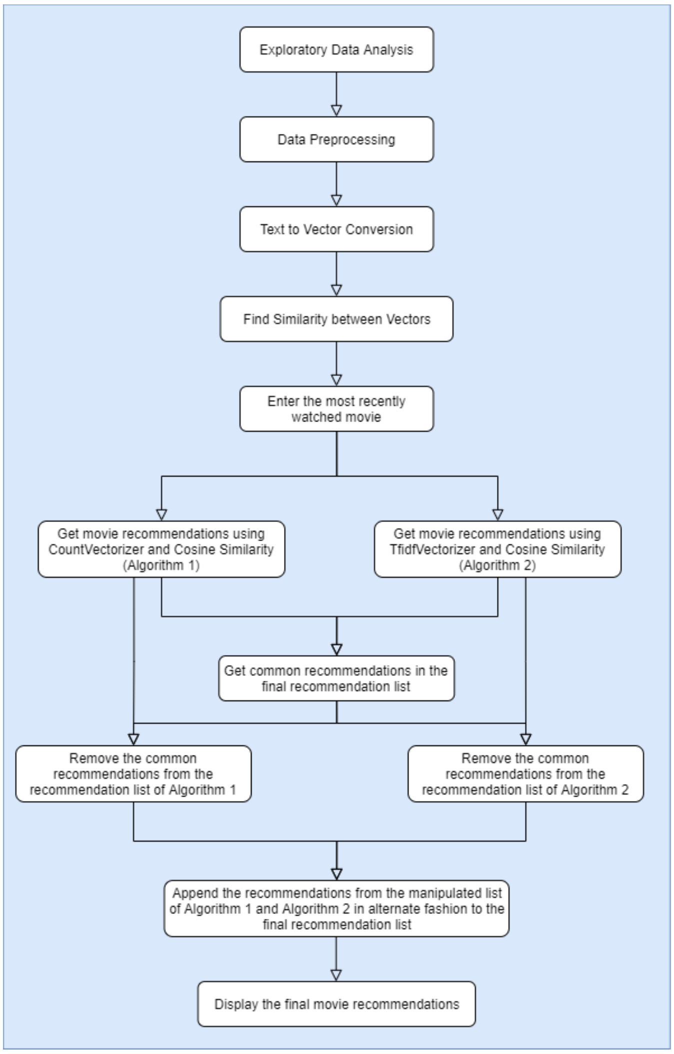

Fig -1: System Architecture

1.2. DATASET, EXPLORATORY DATA ANALYSIS & PREPROCESSING

The ‘TMDB 5000 Movie Dataset’ is taken into consideration for movie recommendation purpose in this research work. This dataset is available on kaggle.com. The dataset is composed of 2 CSV files - ‘tmdb_5000_movies.csv’ and ‘tmdb_5000_credits.csv’

The ‘tmdb_5000_movies.csv’ dataset consists of the following attributes:

- ‘budget’: It indicates the budget of the movie.
- ‘genres’: It indicates the genres of the movie like Action, Documentary, etc.
- A movie can have multiple genres.
- ‘homepage’: It indicates the homepage of the movie. It is basically a website link.
- ‘id’: It indicates movie ID.
- ‘keywords’: It indicates the keywords of the movie. Apart from the title of the movie, keywords give a quick information about the movie.
- ‘original_language’: It indicates whether the movie is originally created in English or other language.
- ‘original_title’: It is nothing but the movie title.
- ‘overview’: It is a short description of the movie.
- ‘popularity’: It is a metric which indicates popularity.
- ‘production_companies’: It consists of the names of companies which has produced the movie.
- ‘production_countries’: It consists of the names of the countries in which the movie production took place.
- ‘release_date’: It consists of the release date of the movie. The format used is yyyy-mm-dd where ‘yyyy’ indicates year of release, ‘mm’ indicates the month of release, and ‘dd’ indicates the day of release.
- ‘revenue’: It indicates the revenue earned by the movie.
- ‘runtime’: It indicates the runtime of a movie. Runtime basically means the length of the movie.
- ‘spoken_languages’: It consists of the languades spoken in the movie.
- ‘status’: It indicates the status of the movie. For example, a movie can be released or not released which basically indicates the status of that movie.
- ‘tagline’: It consists of the tagline of the movie.
- ‘title’: It consists of the title of the movie.
- ‘vote_average’: It indicates the average of the votes.
- ‘vote_count’: It indicates the vote count.

	budget	id	popularity	revenue	runtime	vote_average	vote_count
count	4.803000e+03	4803.000000	4803.000000	4.803000e+03	4801.000000	4803.000000	4803.000000
mean	2.904504e+07	57165.484281	21.492301	8.226064e+07	106.875859	6.092172	690.217989
std	4.072239e+07	88694.614033	31.816650	1.628571e+08	22.611935	1.194612	1234.585891
min	0.000000e+00	5.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000
25%	7.900000e+05	9014.500000	4.668070	0.000000e+00	94.000000	5.600000	54.000000
50%	1.500000e+07	14629.000000	12.921594	1.917000e+07	103.000000	6.200000	235.000000
75%	4.000000e+07	58610.500000	28.313505	9.291719e+07	118.000000	6.800000	737.000000
max	3.800000e+08	459488.000000	875.581305	2.787965e+09	338.000000	10.000000	13752.000000

Fig -2: Statistical data about ‘tmdb_5000_movies.csv’ dataset using pandas Dataframe.describe() method

```
movies.iloc[25]

budget                                                          200000000
genres                                     ['Drama', 'Romance', 'Thriller']
homepage                                      http://www.titanicmovie.com
id                                                                    597
keywords                  ['shipwreck', 'iceberg', 'ship', 'panic', 'tit...
original_language                                                      en
original_title                                                    Titanic
overview                  84 years later, a 101-year-old woman named Ros...
popularity                                                        100.026
production_companies      ['Paramount Pictures', 'Twentieth Century Fox ...
production_countries      [{"iso_3166_1": "US", "name": "United States o...
release_date                                                   1997-11-18
revenue                                                        1845034188
runtime                                                               194
spoken_languages          [{"iso_639_1": "en", "name": "English"}, {"iso...
status                                                           Released
tagline                           Nothing on Earth could come between them.
title                                                             Titanic
vote_average                                                          7.5
vote_count                                                           7562
Name: 25, dtype: object
```

Fig -3: Glimpse of the ‘tmdb_5000_movies.csv’ dataset using ‘Titanic’ movie

The ‘tmdb_5000_credits.csv’ dataset consists of the following attributes:

- ‘movie_id’: It indicates the movie ID.
- ‘title’: It indicates the title of the movie.
- ‘cast’: It consists of the cast of the movie. Cast implies the actors and actresses who appear in the movie.

- 'crew': It consists of those people who are concerned with the production of the movie.

	movie_id
count	4803.000000
mean	57165.484281
std	88694.614033
min	5.000000
25%	9014.500000
50%	14629.000000
75%	58610.500000
max	459488.000000

Fig -4: Statistical data about 'tmdb_5000_credits.csv' dataset using pandas Dataframe.describe() method

```
credits.iloc[25]

movie_id                                                    597
title                                                   Titanic
cast        ['Kate Winslet', 'Leonardo DiCaprio', 'Frances...
director                                          James Cameron
Name: 25, dtype: object
```

Fig -5: Glimpse of the 'tmdb_5000_credits.csv' dataset using 'Titanic' movie

The Exploratory Data Analysis (EDA) has been inspired by HeeralDedhia's blog on medium.com.

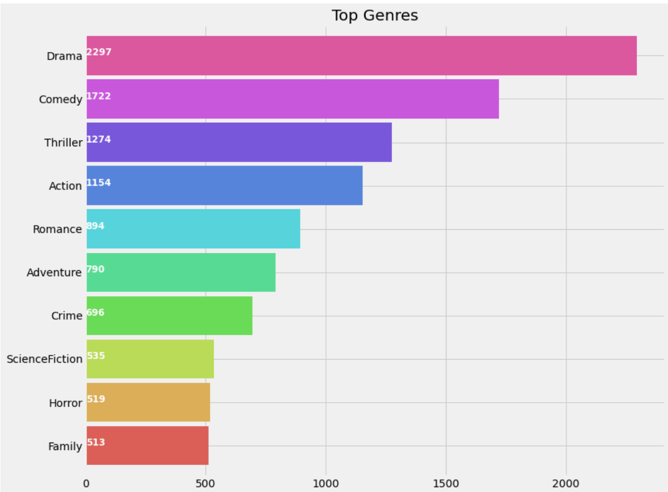

Fig -6: Top Genres

Movies having the genre as Drama are maximum in number as compared to Family movies and Horror movies. A movie might have multipe genres.

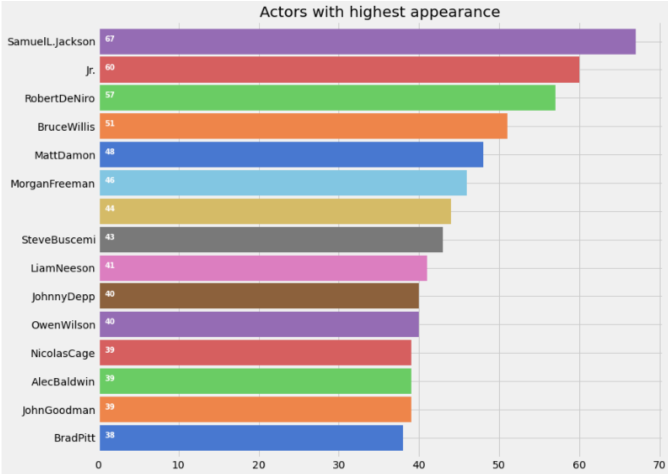

Fig -7: Actor with highest appearance

The above figure indicates the actors with the highest appearance in the decreasing order.

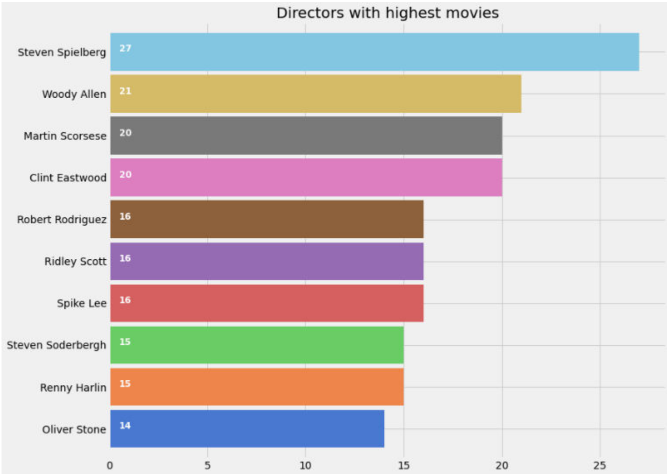

Fig -8: Directors with highest movies

The above figure indicates the directors with the highest appearance in the decreasing order.

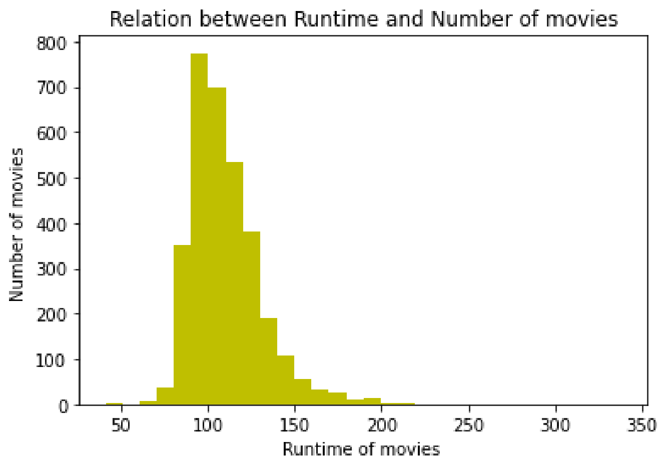

Fig -9: Runtime versus Number of movies

As the runtime increases, number of movies are increasing. After certain point, as the runtime increases, the number of movies decreases. There are some exceptions.

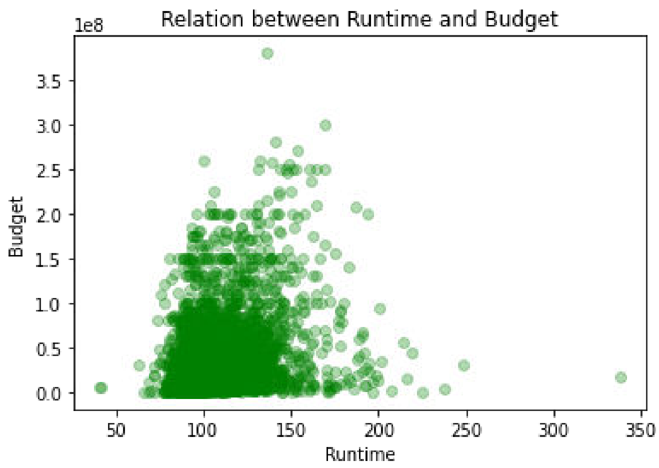

Fig-10: Runtime versus Budget

There are a lot of movies with lower budget and falling in the range of runtime 70 to runtime 150.

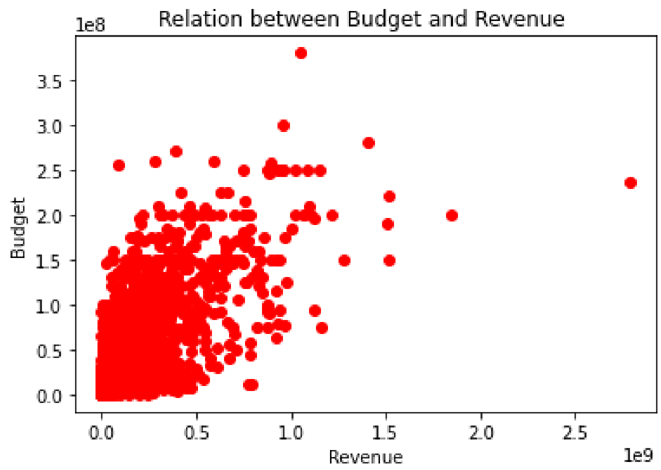

Fig -11: Revenue versus Budget

It can be seen from the above figure that low budget movies have low revenue in general.

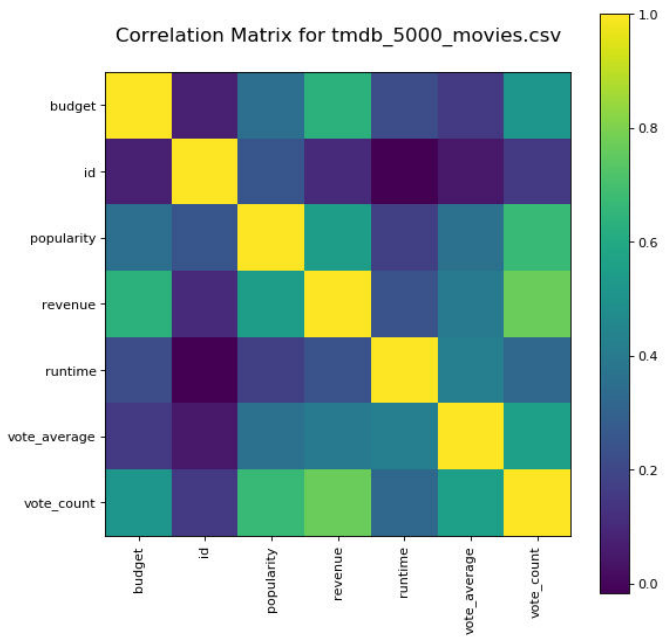

Fig -12: Correlation Matrix for 'tmdb_5000_movies.csv' dataset

From the above correlation matrix, it can be seen that the diagonal is yellow colored because similarity of something with itself is always 1.0, i.e., maximum. Moreover, it can be seen that revenue and vote count have more similarity as compared to budget and vote count.

Preprocessing steps include removing stopwords, combining the first name and the last name into a single name, removing punctuation marks, lowercasing the text, etc.

	title	combine_feature
0	Avatar	cultureclash future spacewar samworthington zo...
1	Pirates of the Caribbean: At World's End	ocean drugabuse exoticisland johnnydepp orland...
2	Spectre	spy basedonnovel secretagent danielcraig chris...
3	The Dark Knight Rises	dccomics crimefighter terrorist christianbale ...
4	John Carter	basedonnovel mars medallion taylorkitsch lynnc...
...	...	...
4798	El Mariachi	unitedstates–mexicobarrier legs arms carlosgal...
4799	Newlyweds	edwardburns kerrybishé marshadietlein edwardb...
4800	Signed, Sealed, Delivered	date loveatfirstsight narration ericmabius kri...
4801	Shanghai Calling	danielhenney elizacoupe billpaxton danielhsia
4802	My Date with Drew	obsession camcorder crush drewbarrymore brianh...

4803 rows × 2 columns

Fig -13: Director, Keywords, Cast and Genres of a movie are combined into a single feature titled 'combine_feature'

The 'combine_feature' attribute needs to be further processed by using some algorithms.

1.3. ALGORITHMS

We can use CountVectorizer or TfidfVectorizer or Glove or Word2Vec in order to create vectors from the text. After converting the text into vectors, we need to find the similarity between the vectors. Cosine Similarity or sigmoid_kernel or some other technique can be used to find the similarity between the vectors.

1. Algorithm 1: Content-based Recommendation using CountVectorizer and Cosine Similarity

In this case, we will use CountVectorizer in order to create vectors from the preprocessed text mentioned in the 'combine_feature' attribute.

After getting the vectors, we will find the similarity between the vectors using Cosine Similarity.

2. Algorithm 2: Content-based Recommendation using TfidfVectorizer and Cosine Similarity

In this case, we will use TfidfVectorizer in order to create vectors from the preprocessed text mentioned in the 'combine_feature' attribute.

After getting the vectors, we will find the similarity between the vectors using Cosine Similarity.

After getting the recommendations using Algorithm 1 and Algorithm 2, get the common movies from both the recommendations initially. Later, append the remaining movies to the common movies in an alternate fashion.

4. RESULT AND ANALYSIS

```
get_recommendations('The Dark Knight', cosine_similarity_cv)

3          The Dark Knight Rises
119               Batman Begins
4638     Amidst the Devil's Wings
2398                      Hitman
1720                    Kick-Ass
1740                  Kick-Ass 2
3326               Black November
1503                      Takers
1986                      Faster
303                     Catwoman
747               Gangster Squad
1253               Kiss of Death
1278                  The Gunman
2154                Street Kings
2793         The Killer Inside Me
Name: title, dtype: object
```

```
get_recommendations('The Dark Knight', cosine_similarity_tv)

3          The Dark Knight Rises
119               Batman Begins
1196                The Prestige
1740                  Kick-Ass 2
1720                    Kick-Ass
163                     Watchmen
4638     Amidst the Devil's Wings
2793         The Killer Inside Me
1033                    Insomnia
95                  Interstellar
2398                      Hitman
739             London Has Fallen
96                     Inception
3819                    Defendor
2060           Out of the Furnace
Name: title, dtype: object
```

Fig -14: Recommendations similar to 'The Dark Knight' movie using Algorithm 1 and Algorithm 2

```
get_final_recommendations('The Dark Knight Rises')[0:15]

['The Dark Knight',
 'Batman Begins',
 "Amidst the Devil's Wings",
 'The Prestige',
 'Romeo Is Bleeding',
 'The Killer Inside Me',
 'Black November',
 'Insomnia',
 'Takers',
 'Interstellar',
 'Faster',
 'The Statement',
 'Catwoman',
 'Inception',
 'Gangster Squad']
```

Fig -15: Final Recommendations similar to 'The Dark Knight' movie

```
get_recommendations('Avatar', cosine_similarity_cv)

206                              Clash of the Titans
71             The Mummy: Tomb of the Dragon Emperor
786                                The Monkey King 2
103                          The Sorcerer's Apprentice
131                                          G-Force
215              Fantastic 4: Rise of the Silver Surfer
466                                 The Time Machine
715                                The Scorpion King
1        Pirates of the Caribbean: At World's End
5                                      Spider-Man 3
9               Batman v Superman: Dawn of Justice
10                                 Superman Returns
12      Pirates of the Caribbean: Dead Man's Chest
14                                     Man of Steel
17     Pirates of the Caribbean: On Stranger Tides
Name: title, dtype: object
```

```
get_recommendations('Avatar', cosine_similarity_tv)

2403                        Aliens
206            Clash of the Titans
587                      The Abyss
43            Terminator Salvation
132            Wrath of the Titans
282                      True Lies
1448                      Sabotage
47           Star Trek Into Darkness
3439                 The Terminator
3184                The Ice Pirates
4114                        Subway
2827                    Crossroads
812                     Pocahontas
94         Guardians of the Galaxy
279      Terminator 2: Judgment Day
Name: title, dtype: object
```

Fig -16: Recommendations similar to 'Avatar' movie using Algorithm 1 and Algorithm 2

```
get_final_recommendations('Avatar')[0:15]

['Clash of the Titans',
 'The Mummy: Tomb of the Dragon Emperor',
 'Aliens',
 'The Monkey King 2',
 'The Abyss',
 "The Sorcerer's Apprentice",
 'Terminator Salvation',
 'G-Force',
 'Wrath of the Titans',
 'Fantastic 4: Rise of the Silver Surfer',
 'True Lies',
 'The Time Machine',
 'Sabotage',
 'The Scorpion King',
 'Star Trek Into Darkness']
```

Fig -17: Final Recommendations similar to 'Avatar' movie

```
get_recommendations('The Godfather', cosine_similarity_cv)

867       The Godfather: Part III
2731       The Godfather: Part II
4638     Amidst the Devil's Wings
2649              The Son of No One
1525                 Apocalypse Now
1018                The Cotton Club
1170      The Talented Mr. Ripley
1209                 The Rainmaker
1394                 Donnie Brasco
1850                      Scarface
2280                   Sea of Love
2792            Glengarry Glen Ross
3012                 The Outsiders
3450              West Side Story
4124             This Thing of Ours
Name: title, dtype: object
```

```
get_recommendations('The Godfather', cosine_similarity_tv)

867     The Godfather: Part III
1525              Apocalypse Now
2731     The Godfather: Part II
4124          This Thing of Ours
4147          Small Apartments
2649           The Son of No One
1170    The Talented Mr. Ripley
512                       Wanted
1225            Mickey Blue Eyes
1209               The Rainmaker
2280                 Sea of Love
613                    The Score
1018             The Cotton Club
4209            The Conversation
4432           On the Waterfront
Name: title, dtype: object
```

Fig -18: Recommendations similar to 'The Godfather' movie using Algorithm 1 and Algorithm 2

```
get_final_recommendations('The Godfather')[0:15]

['The Godfather: Part III',
 'The Godfather: Part II',
 'The Son of No One',
 'Apocalypse Now',
 'The Cotton Club',
 'The Talented Mr. Ripley',
 'The Rainmaker',
 'Sea of Love',
 'This Thing of Ours',
 "Amidst the Devil's Wings",
 'Small Apartments',
 'Donnie Brasco',
 'Wanted',
 'Scarface',
 'Mickey Blue Eyes']
```

Fig -19: Final Recommendations similar to 'The Godfather' movie

```
get_recommendations('Titanic', cosine_similarity_cv)

1081              Revolutionary Road
4247           Me You and Five Bucks
49                   The Great Gatsby
872               All the King's Men
1311                      Angel Eyes
1492                      The Reader
2449            Sense and Sensibility
2661                  Romeo + Juliet
2701                 Little Children
2946     What's Eating Gilbert Grape
4589                          Fabled
297                    Blood Diamond
351                     The Departed
439                   Shutter Island
622                     Body of Lies
Name: title, dtype: object
```

```
get_recommendations('Titanic', cosine_similarity_tv)

1081              Revolutionary Road
609                      Escape Plan
282                        True Lies
3097                     Swept Away
3439                  The Terminator
818                 Captain Phillips
2403                          Aliens
984                    Into the Blue
3695                  The Blue Lagoon
587                        The Abyss
872               All the King's Men
279      Terminator 2: Judgment Day
49                  The Great Gatsby
622                     Body of Lies
351                     The Departed
Name: title, dtype: object
```

Fig -20: Recommendations similar to 'Titanic' movie using Algorithm 1 and Algorithm 2

```
get_final_recommendations('Titanic')[0:15]

['Revolutionary Road',
 'The Great Gatsby',
 "All the King's Men",
 'The Departed',
 'Body of Lies',
 'Me You and Five Bucks',
 'Escape Plan',
 'Angel Eyes',
 'True Lies',
 'The Reader',
 'Swept Away',
 'Sense and Sensibility',
 'The Terminator',
 'Romeo + Juliet',
 'Captain Phillips']
```

Fig 21: Final Recommendations similar to 'Titanic' movie

4. CONCLUSION

We can see from the results that the final recommendations are slightly better than the individual recommendations of Algorithm 1 and Algorithm 2 mentioned in this research work. Hence, it is always better to manipulate the results of different algorithms to get the final result which has the advantages of the individual algorithms.

REFERENCES

1. Choi, Sang-Min, Sang-Ki Ko, and Yo-Sub Han. "A movie recommendation algorithm based on genre correlations." Expert Systems with Applications 39.9 (2012): 8079-8085.
2. Lekakos, George, and Petros Caravelas. "A hybrid approach for movie recommendation." Multimedia tools and applications 36.1 (2008): 55-70.
3. Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." International Journal of Computer Applications 160.7 (2017).
4. Zhang, Jiang, et al. "Personalized real-time movie recommendation system: Practical prototype and evaluation." Tsinghua Science and Technology 25.2 (2019): 180-191.
5. Rajarajeswari, S., et al. "Movie Recommendation System." Emerging Research in Computing, Information, Communication and Applications. Springer, Singapore, 2019. 329-340.
6. Ahmed, Muyeed, Mir Tahsin Imtiaz, and Raiyan Khan. "Movie recommendation system using clustering and pattern recognition network." 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2018.
7. Arora, Gaurav, et al. "Movie recommendation system based on users' similarity." International Journal of Computer Science and Mobile Computing 3.4 (2014): 765-770.
8. Subramaniyaswamy, V., et al. "A personalised movie recommendation system based on collaborative filtering." International Journal of High Performance Computing and Networking 10.1-2 (2017): 54-63.
9. Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." Acm transactions on interactive intelligent systems (tiis) 5.4 (2015): 1-19.
10. R. Lavanya, U. Singh and V. Tyagi, "A Comprehensive Survey on Movie Recommendation Systems," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 532-536, doi: 10.1109/ICAIS50930.2021.9395759.
11. N. Immaneni, I. Padmanaban, B. Ramasubramanian and R. Sridhar, "A meta-level hybridization approach to personalized movie recommendation," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 2193-2200, doi: 10.1109/ICACCI.2017.8126171.
12. M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), 2018, pp. 443-448, doi: 10.1109/CEEICT.2018.8628128.
13. Monika D.Rokade ,Dr.YogeshkumarSharma,"Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic." IOSR Journal of Engineering (IOSR JEN), ISSN (e): 2250-3021, ISSN (p): 2278-8719
14. Monika D.Rokade ,Dr.YogeshkumarSharma"MLIDS: A Machine Learning Approach for Intrusion Detection for Real Time Network Dataset", 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE
15. Monika D.Rokade, Dr. Yogesh Kumar Sharma. (2020). Identification of Malicious Activity for Network Packet using Deep Learning. International Journal of Advanced Science and Technology, 29(9s), 2324 - 2331.
16. Sunil S.Khatal ,Dr.Yogeshkumar Sharma, "Health Care Patient Monitoring using IoT and Machine Learning.", IOSR Journal of Engineering (IOSR JEN), ISSN (e): 2250-3021, ISSN (p): 2278-8719
17. Sunil S.Khatal ,Dr.Yogeshkumar Sharma, "Data Hiding In Audio-Video Using Anti Forensics Technique ForAuthentication"; IJSRDV4I50349, Volume : 4, Issue : 5
18. Sunil S.Khatal Dr. Yogesh Kumar Sharma. (2020). Analyzing the role of Heart Disease Prediction System using IoT and Machine Learning. International Journal of Advanced Science and Technology, 29(9s), 2340 - 2346.