Netflix Movie Recommendation Using Cosine Similarity

CH. VASISHTA[1,]B. CHRISSIE[2.]

1. *Student in the Department of Computer ScienceKLEF,GreenFields,Vaddeswaram-522502,Guntur (Dist),AndhraPradesh,India.*
2. *Student in the Department of Computer Science KLEF,GreenFields,Vaddeswaram-522502,Guntur (Dist),AndhraPradesh,India.*

ABSTRACT:*A movie recommendation system is a way to recommend films to people based on their preferences, tastes, and history. When movie recommendation systems are developed, they usually use data from users' ratings of what they like, how often they watch movies of a particular genre, and how much more likely they are going to watch a movie like the movies they have already watched. There are many algorithms that can help you to build a movie recommendation system but among them, the most popular algorithm is the cosine algorithm. The cosine algorithm is an algorithm that is used to generate these ratings. This algorithm is based on the concept of cosine similarity. The cosine algorithm is a well-known mathematical algorithm used to generate recommendations for items in each set. The cosine algorithm has been found to be a suitable algorithm for movie recommendation systems as it is able to generate recommendations that are both accurate and relevant to the user based on their data's history. The cosine similarity algorithm generates a single value that represents the similarity between two items. The algorithm uses mathematical formulas to recommend movies that are likely to interest you based on two parameters: the rating of the user and the rating of the movie.*

Keywords*- Movie recommendation, cosine similarity, Netflix*

Date of Submission: 10-03-2023 Date of acceptance: 23-03-2023

I. Introduction

A movie recommendation system is a way to recommend films to people based on their preferences, tastes, and history. When movie recommendation systems are developed, they usually use data from users' ratings of what they like, how often they watch movies of a particular genre, and how much more likely they are going to watch a movie similar to the movies they have already watched. There are many algorithms that can help you to build a movie recommendation system but among them, the most popular algorithm is the cosine algorithm. The cosine algorithm is an algorithm that is used to generate these ratings. This algorithm is based on the concept of cosine similarity. The cosine algorithm is a well-known mathematical algorithm used to generate recommendations for items in a given set. The cosine algorithm has been found to be a suitable algorithm for movie recommendation systems as it is able to generate recommendations that are both accurate and relevant to the user based on their data's history. The cosine similarity algorithm generates a single value that represents the similarity between two items. The algorithm uses mathematical formulas to recommend movies that are likely to interest you based on two parameters: the rating of the user and the rating of the movie The cosine algorithm is based on the cosine function, which is a mathematical function that is used to find the angle between two vectors. The algorithm works by finding the angle between the movies you have watched and a vector that represents the movies that you like. It then calculates a cosine score for each movie. The movies with the highest cosine scores are the ones that are recommended to you. The cosine similarity algorithm is widely used because it is fast and produces good results. That's the reason why we chose the cosine similarity algorithm to build a movie recommendation system. This research paper is going to be all about how to implement a cosine similarity algorithm and code a model for movie recommendation system.

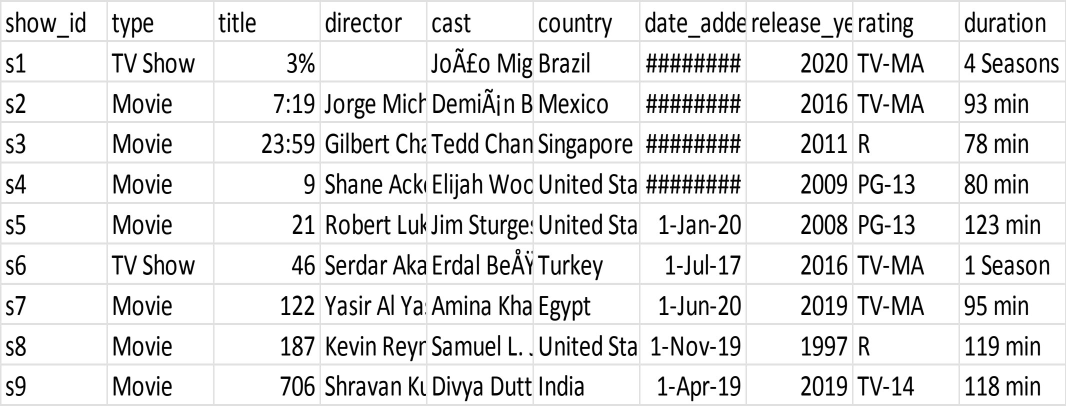
Theoretical Analysis

Dataset Development

In this module sample data of Netflix movies and tv shows is collected from Kaggle. Data set contains features as data show id, type, title, director, cast, country, date added, release year, rating, duration.

After getting the data we will preprocess the data as it has so many errors or missing data. For example the data of some cast is missing as well as some directors names and we will have occasional mistakes for duration we will do preprocess and then we will use that data to do some data visualization on the data after processing it.

Table 1-Dataset Description

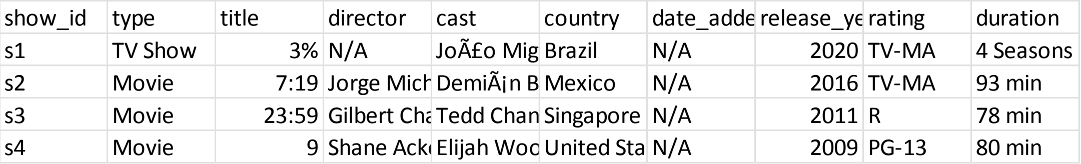


show_id	type	title	director	cast	country	date_adde	release_ye	rating	duration
s1	TV Show	3%		JoÃ£o Mig	Brazil	########	2020	TV-MA	4 Seasons
s2	Movie	7:19	Jorge Mich	DemiÃ¡n B	Mexico	########	2016	TV-MA	93 min
s3	Movie	23:59	Gilbert Cha	Tedd Chan	Singapore	########	2011	R	78 min
s4	Movie	9	Shane Ack	Elijah Woo	United Sta	########	2009	PG-13	80 min
s5	Movie	21	Robert Luk	Jim Sturge	United Sta	1-Jan-20	2008	PG-13	123 min
s6	TV Show	46	Serdar Aka	Erdal BeÅŸ	Turkey	1-Jul-17	2016	TV-MA	1 Season
s7	Movie	122	Yasir Al Ya	Amina Kha	Egypt	1-Jun-20	2019	TV-MA	95 min
s8	Movie	187	Kevin Reyn	Samuel L. .	United Sta	1-Nov-19	1997	R	119 min
s9	Movie	706	Shravan Ku	Divya Dutt	India	1-Apr-19	2019	TV-14	118 min

Developing Features

Here after preprocessing we will have data that is empty and in those places we will put N/A if they are supposed to be words, if they are numbers we will add mean number of that column. And we have so much data we have to do pre process for reviews even for description we have to do this so that when executing the data we will not have any errors.

Table 2: - data after preprocessing



show_id	type	title	director	cast	country	date_adde	release_ye	rating	duration
s1	TV Show	3%	N/A	JoÃ£o Mig	Brazil	N/A	2020	TV-MA	4 Seasons
s2	Movie	7:19	Jorge Mich	DemiÃ¡n B	Mexico	N/A	2016	TV-MA	93 min
s3	Movie	23:59	Gilbert Cha	Tedd Chan	Singapore	N/A	2011	R	78 min
s4	Movie	9	Shane Ack	Elijah Woo	United Sta	N/A	2009	PG-13	80 min

Cosine similarity

The cosine similarity algorithm is implemented by comparing the angles of the two vectors. The angle between the two vectors is calculated by calculating the dot product of the two vectors. The dot product of two vectors A and B is calculated by multiplying the two vectors together and then adding the absolute value of the result. The cosine similarity algorithm is implemented by calculating the dot product of A and B and taking the cosine of the result. The cosine of 0.5 is 0.5, so the cosine similarity algorithm calculates the dot product of A and B, calculates the cosine of the result, and compares the cosine with 0. 5. If A and B are very similar, the cosine similarity algorithm will return a very small value. If A and B are very dissimilar, the cosine similarity algorithm will return a large value.

Formula

As I mentioned earlier, the cosine algorithm is the most popular algorithm for finding the similarity between two sets of objects. It calculates the cosine of the angle between the two sets of objects. The cosine of 0 degrees is 0, and the cosine of 180 degrees is 1. The closer the angle is to 0, the greater the similarity between the two vectors. The cosine algorithm can be implemented in Python using the following formula: The cosine similarity of the two movies is calculated with the following equation: S = cos (θ) Where θ is the angle between the two movies. The cosine similarity of the two movies is calculated with the following equation: S = cos (θ) Where θ is the angle between the two movies.

$\mathbf{cosine(x,y)=(x.y^{T})/(||x||.||y||)}$

Steps to be followed for Cosine similarity

To implement the cosine algorithm in Python, you will need the following libraries:

1. NumPy: A Python library for mathematical operations
2. SciPy: A library of scientific computing tools
3. matplotlib: A library for data visualization
4. pandas: A library for data analysis
5. IPython: A powerful interactive shell for Python

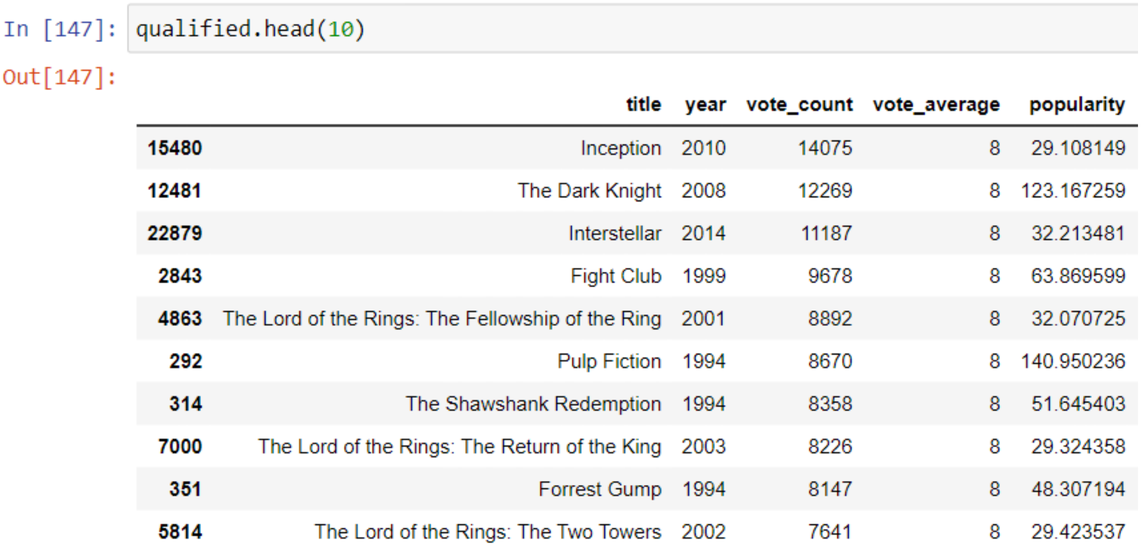
Once you have these libraries installed, you can install the cosine algorithm module by entering the following command:

pip install cosine

Once the cosine algorithm module is installed, you can initialize it by entering the following code:
from cosine import Cosine
First, we will create a model to predict the rating of a movie. We will use the following data:
Name: movie_id
Rating: 1, 2, 3, 4, 5 Next, you can use the cosine algorithm to recommend a movie. To do this, you will first need to create a movie object. This object will contain the ratings and viewing history for the movies that the cosine algorithm will use to recommend a movie to the user. You can create this object by entering the following code:
movie = Cosine. From_ratings_and_viewing_history(ratings=['The Shawshank Redemption'], viewing_history=['The Godfather'],)
After you create the movie object, you can use the cosine algorithm to recommend a movie to the user. To do this, you will first need to create a recommendation object. This recommendation object will contain the movie object and the user's rating. You can create this object by entering the following code: recommendation = Cosine.from_ movie(movie, rating=user.rating)
Next, you can use the recommendation object to recommend a movie to the user.

Experimental Results

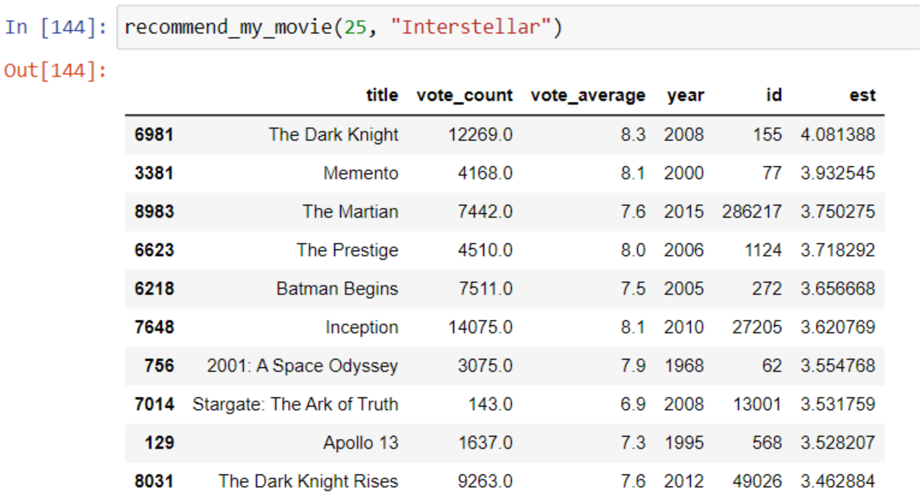
The method used to calculate how similar two films are based on how closely they share several characteristics. It displays the cosine of the angle of two mathematically. projected vectors in a multidimensional environment. Finding related things is made easier because to the cosine similarity.



```
In [147]: qualified.head(10)
Out[147]:
```

	title	year	vote_count	vote_average	popularity
15480	Inception	2010	14075	8	29.108149
12481	The Dark Knight	2008	12269	8	123.167259
22879	Interstellar	2014	11187	8	32.213481
2843	Fight Club	1999	9678	8	63.869599
4863	The Lord of the Rings: The Fellowship of the Ring	2001	8892	8	32.070725
292	Pulp Fiction	1994	8670	8	140.950236
314	The Shawshank Redemption	1994	8358	8	51.645403
7000	The Lord of the Rings: The Return of the King	2003	8226	8	29.324358
351	Forrest Gump	1994	8147	8	48.307194
5814	The Lord of the Rings: The Two Towers	2002	7641	8	29.423537

Fig: 1: - Top 10 movies based on IMDB movie weightage formula.



```
In [144]: recommend_my_movie(25, "Interstellar")
Out[144]:
```

	title	vote_count	vote_average	year	id	est
6981	The Dark Knight	12269.0	8.3	2008	155	4.081388
3381	Memento	4168.0	8.1	2000	77	3.932545
8983	The Martian	7442.0	7.6	2015	286217	3.750275
6623	The Prestige	4510.0	8.0	2006	1124	3.718292
6218	Batman Begins	7511.0	7.5	2005	272	3.656668
7648	Inception	14075.0	8.1	2010	27205	3.620769
756	2001: A Space Odyssey	3075.0	7.9	1968	62	3.554768
7014	Stargate: The Ark of Truth	143.0	6.9	2008	13001	3.531759
129	Apollo 13	1637.0	7.3	1995	568	3.528207
8031	The Dark Knight Rises	9263.0	7.6	2012	49026	3.462884

Fig: 2: - Top 10 movies similar to Interstellar using cosine.

II. Results and Discussion

We have done some data visualization and also created a model that will recommend movies based on a number of parameters. We can see that the country that produces the most movies/tv shows is USA. The directors who directed the most movies are Raul Campos and Jan Suter with 17 movies to each name. By using cosine, we came to know that the movies that are similar to interstellar are The dark knight followed by memento and The Martian in that order.

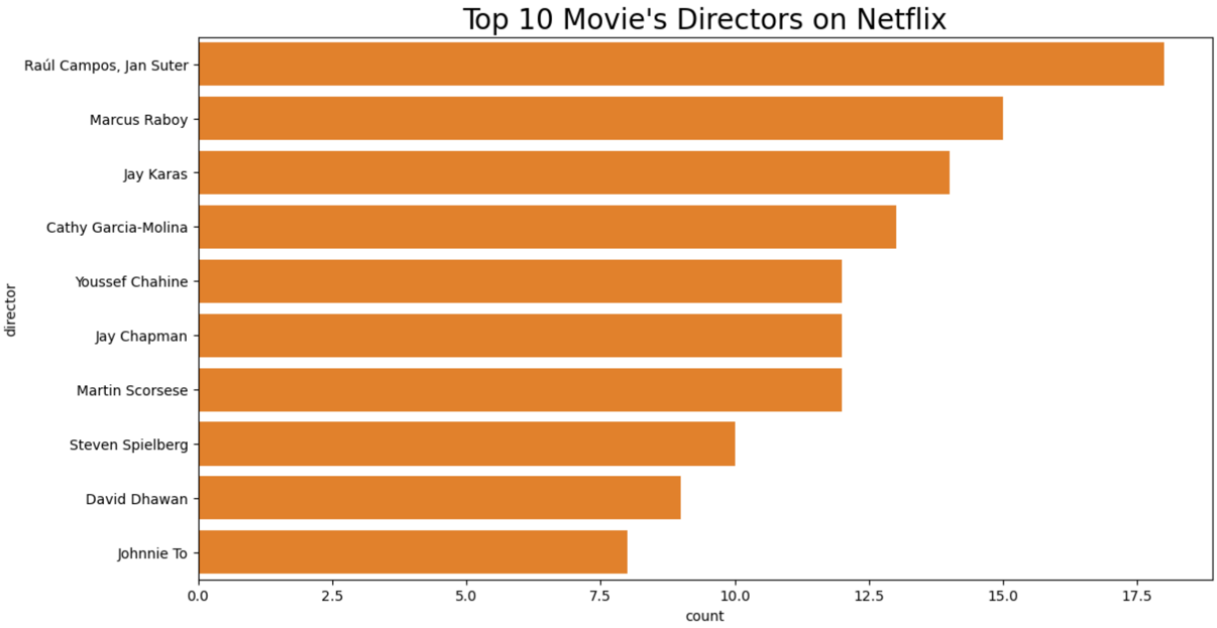


Fig: 3: - Top 10 movie directors.

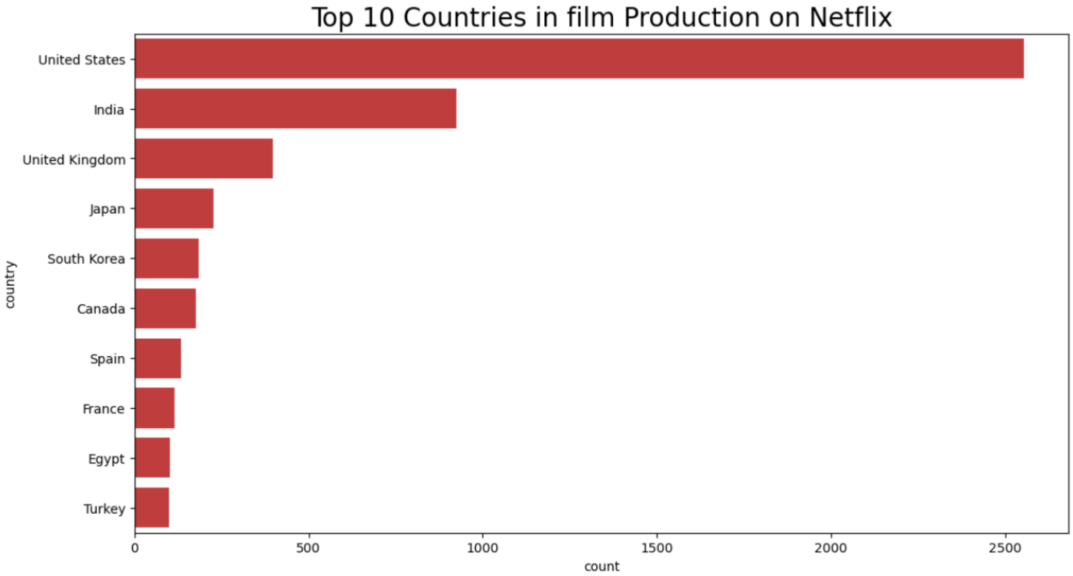
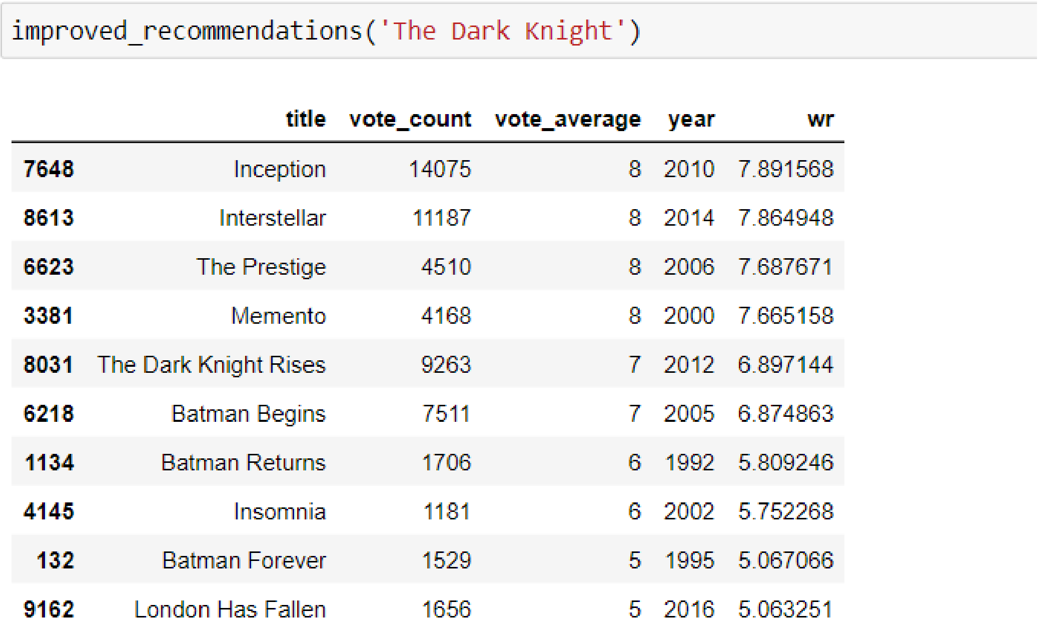


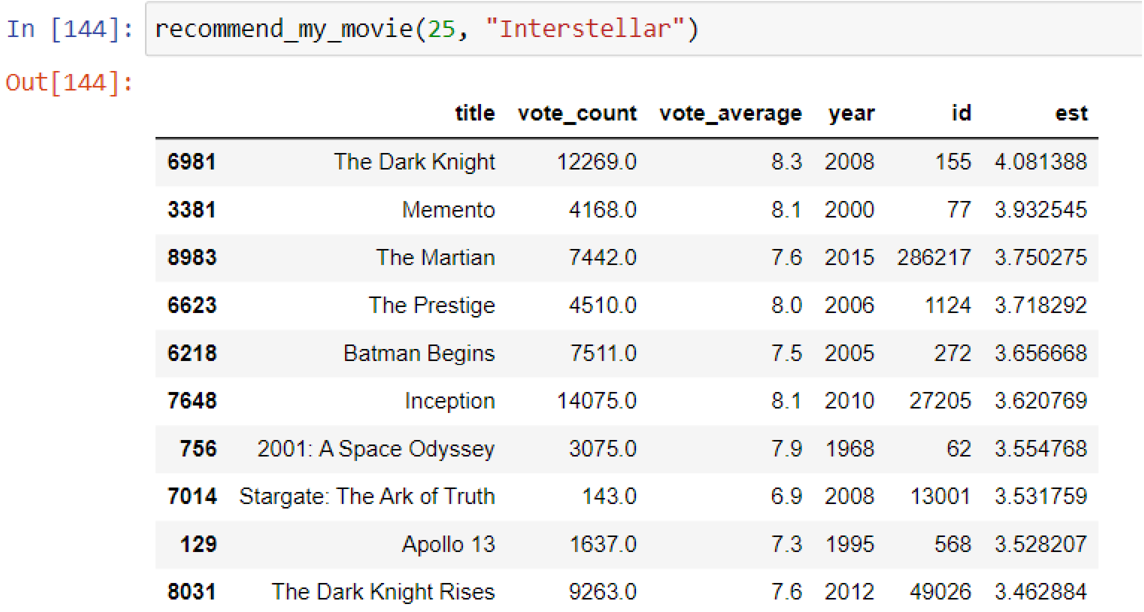
Fig: 4: - Top 10 countries who produce more content.



```
improved_recommendations('The Dark Knight')
```

	title	vote_count	vote_average	year	wr
7648	Inception	14075	8	2010	7.891568
8613	Interstellar	11187	8	2014	7.864948
6623	The Prestige	4510	8	2006	7.687671
3381	Memento	4168	8	2000	7.665158
8031	The Dark Knight Rises	9263	7	2012	6.897144
6218	Batman Begins	7511	7	2005	6.874863
1134	Batman Returns	1706	6	1992	5.809246
4145	Insomnia	1181	6	2002	5.752268
132	Batman Forever	1529	5	1995	5.067066
9162	London Has Fallen	1656	5	2016	5.063251

Fig: 5: - Top 10 movies similar to dark knight.



```
In [144]: recommend_my_movie(25, "Interstellar")
Out[144]:
```

	title	vote_count	vote_average	year	id	est
6981	The Dark Knight	12269.0	8.3	2008	155	4.081388
3381	Memento	4168.0	8.1	2000	77	3.932545
8983	The Martian	7442.0	7.6	2015	286217	3.750275
6623	The Prestige	4510.0	8.0	2006	1124	3.718292
6218	Batman Begins	7511.0	7.5	2005	272	3.656668
7648	Inception	14075.0	8.1	2010	27205	3.620769
756	2001: A Space Odyssey	3075.0	7.9	1968	62	3.554768
7014	Stargate: The Ark of Truth	143.0	6.9	2008	13001	3.531759
129	Apollo 13	1637.0	7.3	1995	568	3.528207
8031	The Dark Knight Rises	9263.0	7.6	2012	49026	3.462884

Fig: 6: - Top 10 movies similar to Interstellar.

III. Conclusion

A cosine algorithm is a versatile tool that can be used for a variety of purposes, such as movie recommendation, music recommendation, book recommendation system, product recommendation system, and so on. In this paper, we used a cosine algorithm to build a movie recommendation system, and the reason why we used the cosine algorithm among all the other algorithms to build a recommendation system is that cosine is easy to implement, fast, and produces accurate results compared to other algorithms. We can use this cosine because it is fast ad reliable it is used also for checking text documents or files. In this we use cosine similarity because most of recommendations don't take summery into account but to get the good recommendation, we take plot into consideration. Most of recommendations are good based on plot but bad on basis of quality of movie. So, we use IMDB data to calculate the IMDB weightage ranking to get a score for movie and recommend movies that are only acceptable only if they are in top percentile so that we can get more accurate results.

REFERENCES

[1]. Rahutomo, Faisal, TeruakiKitasuka, and Masayoshi Aritsugi. "Semantic cosine similarity." The 7th international student conference on advanced science and technology ICAST. Vol. 4. No. 1. 2012.

[2]. Tata, Sandeep, and Jignesh M. Patel. "Estimating the selectivity of tf-idf based cosine similarity predicates." ACM Sigmod Record 36.2 (2007): 7-12.

[3]. Beel J., Langer S., Genzmehr M., and Nürnberger A., "Introducing Docear's Research Paper Recommender System," in Proceedings of the 13th ACM/IEEE-CS Joint Conferenc

[4]. Han J., Kamber M., "Data Mining: Concepts and Techniques", Morgan Kaufmann (Elsevier), 2006

[5]. Badrul, Sarwar., George, Karypis., Joseph, Konstan., and John, Riedl. "Item-based collaborative filtering recommendation algorithms" Proceedings of the 10th international conference on World Wide Web. 2001.

[6]. S. Wei, N. Ye, S. Zhang, X. Huang and J. Zhu, "Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure," 2012 International Conference on Computer Science and Service System, Nanjing, China, 2012, pp. 2038-2041, doi: 10.1109/CSSS.2012.507.

[7]. Gomez-Uribe, C. and Neil Hunt. "The Netflix Recommender System: Algorithms, Business Value, and Innovation." ACM Trans. Manag. Inf. Syst. 6 (2016): 13:1-13:19.

[8]. L. E. Molina and S. Bhulai, Recommendation System for Netflix, 2018

[9]. Lekakos, George, and Petros Caravelas. "A hybrid approach for movie recommendation." Multimedia tools and applications 36 (2008): 55-70.

[10]. Reddy, Muppana Mahesh, R. SujithraKanmani, and B. Surendiran. "Analysis of Movie Recommendation Systems; with and without considering the low rated movies." 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE, 2020.