# TOPIC: MOVIE RECOMMENDATION SYSTEM

**ABSTRACT**

The rise of OTT and online streaming services has allowed for not only for the creation of new avenues for movies, but has also created its own following. The ever-increasing numbers of the films and other media formats on such platforms has made obtaining movies for users, using existing technology, very difficult. Due for this reason, users and platform developers have been looking for a program that can efficiently and correctly recommend movies to its users, based on their previous choices in the same. However, many recommendation systems use shared browsing methods to predict user requirements as a result of this method provide a very accurate prediction. Today, many researchers are focused on developing a number of ways to improve accuracy instead rather than using collaborative browsing methods. Therefore, in order to improve the accuracy of recommendation program, we present the recommender system, k-clique method and KNN algorithm. The performance results show that the proposed methods improve more accuracy of the movie recommendation system than any other methods used in this experiment.

## INTRODUCTION

Personalized recommendations try to find out user features and preferences by collecting and analyzing the behavior and history of the user on the platform. This includes what kind of behavior the user has, what the type of items the user likes to share and more importantly, what user features and preferences based on platform rules and recommend information and assets interest the user.

The personalized complimentary system is a type of information filtering technology. It is an integrated program which is a combination of a variety of data mining algorithms and user-related information, meeting interests orcapabilities user interests.

The standard recommendation system is classified as a content-based recommendation program, a collaborative filtering system, as well as a hybrid recommendation system. The algorithm has a wide range of applications and usage conditions, with results on the use of a different recommendation algorithm for recommendations for similar information. In the real app of the complimentary system, the system is usually a hybrid recommendation system. That is, mixing profits algorithm for each recommendation in the recommended process successfully develop a recommendation result.

In this paper, the key content of the survey is to help users find a movie that interests the user automatically by size movie information data using the KNN algorithm, recommender systems and k clique method

**Recommender Systems** generate recommendations. The user can accept them according to his or her choice and can provide, immediately or in the next section, a clear or transparent response. User actions and responses can be stored on a recommended website and may be used to generate new recommendations for subsequent user system interactions. The economic potential of theses recommender systems has led some of the biggest e-commerce websites (like Amazon.com, snapdeal.com) and the online movie rental company Netflix to make these systems a salient part of their websites. Personalized high-quality recommendations add another dimension to user experience. Web-based complimentary programs have recently been used to provide a variety of customized information to their relevant users. These applications can be used on a variety of applications and are very common.

We can classify the recommender systems in two broad categories:

1. **Collaborative filtering approach**
2. **Content-based filtering approach**

For example,

1. Last.fm creates a "channel" of recommended songs based on whichindividual bands and songs the user listens to regularly and compares those that contradict the behavior of other users. Last.fm will play tracks that do not appear in the user library, but are often played by other users with similar interests. As this approach improves User Conduct, it is an example of a shared filtering strategy.

2. Pandora uses the features of a song or artist (a subset of 400 attributes provided by the Music Genome Project) to produce a "channel" that plays music with similar features. User feedback isused to improve channel results, emphasizing specific attributes when the user "dislikes" a particular song and emphasizing other attributes when the user "likes" a song. This is an example of a content-based approach.

Recommendation systems are a useful way to search algorithms because they help users find items they may not have found otherwise. Notably, recommendation programs are often implemented using search engines that index non-traditional data.

Another method which can be used is based on the **Invisible High Clique method.** It was the first time that this way of communicating with people analysis was used to present in the movie recommendation program, and found that work very well. The k-cliques, which are graphs partially fully connected tok-vertices and the most effective way to form groups in the analysis of social networks proposed. In the proposed method, a similarity of the cosine equation is used to estimate similarities between users. Tenth, use k-clique to create collections and present a movie in the same groups using the shared browsing method. Proposed delivery solutions advanced k-clique methods of working more effectively than existing partnerships filtering and great clique. The result of each test depends on the value of k.

In order to find a better solution, an improved k-clique will be provided.

Improve The k-cliques method is used to determine whether the value of k is the total value of k-clique method, which provides high accuracy with the system of recommendation. The best value of k in k-clique is the value that leads to the middle percentage the error of being a small value. To test performance, use MovieLens data, which is general information on movie recommendation systems. To evaluate the effectiveness of Movie Lens data set, broken down into the most widely used test and evaluation data in

practical wisdom.
Comparison of collaborative browsing methods using the nearest k neighbor, major clique method, k-clique method, and k-clique development to test performance.

Another method is **the KNN algorithm is called K nearest neighbour algorithm**. The main concept of the KNN algorithm is: if most k neighbors who are very similar to sample in feature area belongs to a particular category. The algorithm steps are user-matched statistics, KNN selection of anearby neighbor and predict points to count.

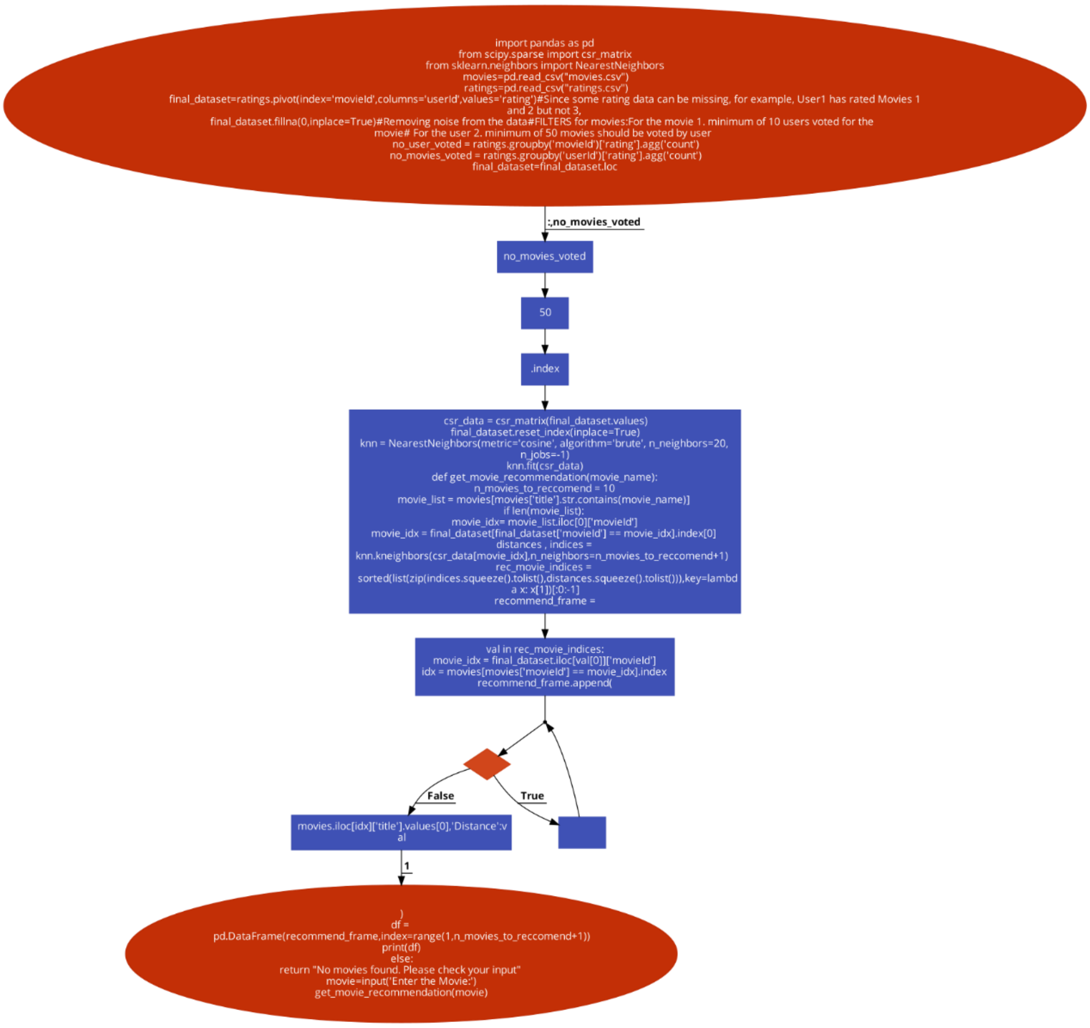
## LITERATURE REVIEW

| REFERENCES | METHOD | METRICS | LIMITATIONS |
|---|---|---|---|
| 1 | A personalised movie recommendation system based on collaborative filtering | In this Item based collaborative filtering technique, firstly the User item rating matrix examines and identifies the relationships among various items, and then we use these relationships in order to compute the recommendations for the user. | Collaborative filtering systems are basedon the action of available data from similar users. If we are building a movie recommendation system, you would have no user data to start with or may have improper data |
| 2 | Content-based filteringfor recommendation systems using multiattribute networks | In this paper, the recommendation system has been built on the type of genres that the user might prefer to watch. The approach adopted to do so is content-based filtering using genre correlation. The dataset used for the system is Movie Lens dataset. The data analysis tool used is R. | Since the feature representation of theitems are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.<br>The model can only make recommendations based on existing interests of the user. In other words, themodel has limited ability to expand on the users' existing interests. |
| 3 | A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System | It will work in a hybrid format The system uses a mix of content as well as collaborative filtering algorithm. The context of the movies is also considered while recommending. The user - user relationship as well as user - item relationship plays a role in the recommendation. | As it is the combination of the abovetwo methods ,it does not have any major limitations. |

| | | | |
|---|---|---|---|
| 4 | An improved collaborative movie recommendation system using computational intelligence | The proposed approach can provide high performance in terms of accuracy, and generate more reliable and personalized movie recommendations when compared with the existing methods. | It is high is cost |
| 5 | An Efficient movie recommendation algorithm based on improved k-clique | The improved k-cliques method is used to determine that the value of k is the optimal value of the k-clique method, which provides the maximum accuracy with the recommender system. | Although accuracy higher but complex to implement |
| 6 | Movie Recommendation System Based on Movie Swarm | It generates movie swarms not only convenient for movie producer to plan a new movie but also useful for movie recommendation. Experimental studies on the real data reveal the efficiency and effectiveness of the proposed system. | It is not much scalable |
| 7 | Movie Recommendation System Using Semi-Supervised Learning | The proposed approach can provide high performance in terms of accuracy, and generate more reliable and personalized movie recommendations when compared with the existing methods. | Higher cost |
| 8 | An improved approach for movie recommendation system | This approach helps to get the advantages from both the approaches as well as tries to eliminate the drawbacks of both methods. | As it is the combination of the above two methods ,it does not have any major limitations. |

## MACHINE LEARNING MODEL

The Model is based on the KNN(K-Nearest Neighbours) Algorithm.




**Program:**

```
import pandas as pd
from scipy.sparse import csr_matrix
from sklearn.neighbors import NearestNeighbors
movies=pd.read_csv("movies.csv")
ratings=pd.read_csv("ratings.csv")
final_dataset=ratings.pivot(index='movieId',columns='userId',values='rating')
#Since some rating data can be missing, for example, User1 has rated Movies 1
and 2 but not 3,
final_dataset.fillna(0,inplace=True)
#Removing noise from the data
#FILTERS for movies:For the movie 1. minimum of 10 users voted for the
movie
#                  For the user 2. minimum of 50 movies should be voted by user
no_user_voted = ratings.groupby('movieId')['rating'].agg('count')
no_movies_voted = ratings.groupby('userId')['rating'].agg('count')
```

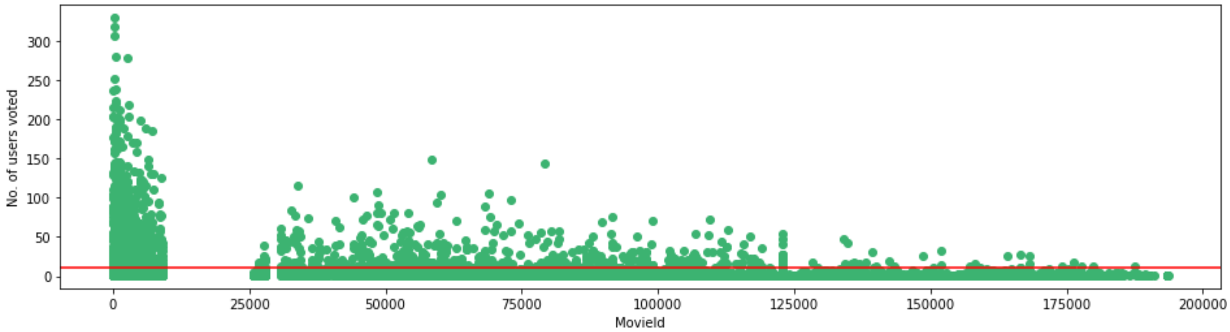
```
final_dataset=final_dataset.loc[:,no_movies_voted[no_movies_voted >
50].index]
#Removing Sparsity
csr_data = csr_matrix(final_dataset.values)
final_dataset.reset_index(inplace=True)
knn = NearestNeighbors(metric='cosine', algorithm='brute', n_neighbors=20,
n_jobs=-1)
knn.fit(csr_data)

def get_movie_recommendation(movie_name):
    n_movies_to_reccomend = 10
    movie_list = movies[movies['title'].str.contains(movie_name)]
    if len(movie_list):
        movie_idx= movie_list.iloc[0]['movieId']
        movie_idx = final_dataset[final_dataset['movieId'] == movie_idx].index[0]
        distances , indices =
knn.kneighbors(csr_data[movie_idx],n_neighbors=n_movies_to_reccomend+1)
        rec_movie_indices =
sorted(list(zip(indices.squeeze().tolist(),distances.squeeze().tolist())),key=lambd
a x: x[1])[:0:-1]
        recommend_frame = []
        for val in rec_movie_indices:
            movie_idx = final_dataset.iloc[val[0]]['movieId']
            idx = movies[movies['movieId'] == movie_idx].index

recommend_frame.append({'Title':movies.iloc[idx]['title'].values[0],'Distance':v
al[1]})
        df =
pd.DataFrame(recommend_frame,index=range(1,n_movies_to_reccomend+1))
        print(df)
    else:
        return "No movies found. Please check your input"

movie=input('Enter the Movie:')
get_movie_recommendation(movie)
```

**Output:**

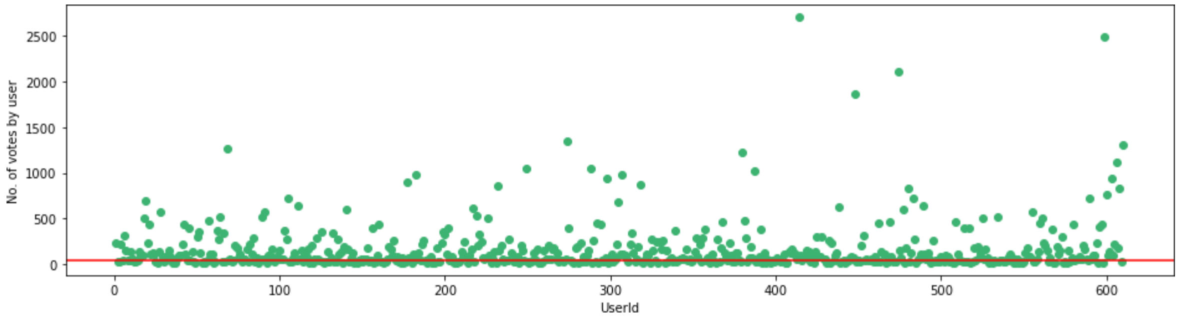
```
In [38]: runfile('C:/Users/saahi/OneDrive/Deskto
saahi/OneDrive/Desktop/Movie Rec')

Enter the Movie:Iron Man
                            Title  Distance
1                       Up (2009)  0.368857
2   Guardians of the Galaxy (2014)  0.368758
3                 Watchmen (2009)  0.368558
4                Star Trek (2009)  0.366029
5            Batman Begins (2005)  0.362759
6                   Avatar (2009)  0.310893
7               Iron Man 2 (2010)  0.307492
8                   WALL·E (2008)  0.298138
9         Dark Knight, The (2008)  0.285835
10           Avengers, The (2012)  0.285319

In [39]: runfile('C:/Users/saahi/OneDrive/Deskto
saahi/OneDrive/Desktop/Movie Rec')

Enter the Movie:Inception
                           Title  Distance
1          Hangover, The (2009)  0.369214
2               Iron Man (2008)  0.369175
3             Fight Club (1999)  0.367898
4        Sherlock Holmes (2009)  0.366418
5       Django Unchained (2012)  0.362976
6         Shutter Island (2010)  0.345888
7          Avengers, The (2012)  0.340302
8   Dark Knight Rises, The (2012)  0.335075
9     Inglourious Basterds (2009)  0.305288
10      Dark Knight, The (2008)  0.213876

In [40]:
```

Data Visualization:

Let’s visualize the number of users who voted with our threshold of 10.




Let’s visualize the number of votes by each user with our threshold of 50.



## REFERENCES


1. *A Movie Recommender System: MOVREC using Machine LearningTechniques (Ashrita Kashyap, Sunita. B, Sneh Srivastava, Aishwarya. PH, Anup JungShah Department of Computer Science & Engineering SAIT, Bengaluru, Karnataka, India)*

2. *A Movie Recommender System: MOVREC (Manoj Kumar, D.K. Yadav, Ankur Singh, Vijay Kr. Gupta)*

3. *An Efficient movie recommendation algorithm based on improvedk-clique (Phonexay Vilakone , Doo-Soon Park, Khamphaphone Xinchang and FeiHao)*

4. *Design and Implementation of Movie Recommendation System Based onKnn Collaborative Filtering Algorithm (Bei-Bei CUI)*

5. *Movie Recommendation System Using Machine Learning (F. Furtado, A. Singh)*

6. *Movie Recommender System Using Machine Learning Algorithms (Piyush Kumar, Shaik Golam Kibriya, Yuva Ajay and Ilampiray)*

7. *Personalized Movie Recommendation Method Based on Deep Learning (Jingdong Liu ,Won-Ho Choi and Jun Liu)*

8. *Movie Recommendation System using Content-based Filtering (Yeole Madhavi, Rokade Monika Khatal Sunil)*

9. *Machine Learning Model for Movie Recommendation System (M. Chenna Keshava, P. Narendra Reddy, S. Srinivasulu, B. Dinesh Naik)*

10. *An Efficient Deep Learning Approach for Collaborative Filtering Recommender System (Mohammed Fadhel Aljunid, Manjaiah DHb)*

11. *Viana P, Pinto JP (2017) A collaborative approach for semantic time-based video annotation using gamification. Hum Cent Comput Inf Sci 7:13*

*12.Desrosiers C, Karypis G (2011) A comprehensive survey of neighborhood-based recommendation methods. In: Ricci F, Rokach L, Shapira B, Kantor P (eds) Recommender systems handbook. Springer, Boston, pp 107–144*

*13.Bagci H, Karagoz P (2016) Context-aware location recommendation by using a random walk-based approach. Knowl Inf Syst 47:241–260*

*14.Wong F, Lee S, Wong Q, Lee S (2015) Points of interest recommendation based on context-aware. Int J Hybrid Inf Technol 8:55–62*

*15.Lee D (2015) Personalizing information using users' online social networks: a case study of CiteULike. J Inf Process Syst 11:1–21*

*16.Zou HT, He Y, Zheng S, Yu HL, Hu C (2018) Online group recommendation with local optimization. CMES: 217–231*

*17.Hui P, Crowcroft J (2008) Human mobility models and opportunistic communications system design. Phil Trans R Soc A. https://doi.org/10.1098/rsta.2008.0010:2005-2016*

*18.Hao F, Park DS, Pei Z (2018) When social computing meets soft computing: opportunities and insights. Hum Cent Comput Inf Sci 1:8*

*19.Hao F, Park DS, Min G, Jeong YS, Park JH (2016) k-cliques mining in dynamic social networks based on triadic formal concept analysis. Neurocomputing 209:57–66*

*20.Ryu PM (2018) Predicting the unemployment rate using social media analysis. J Inf Process Syst 14:904–915*

*21.Hao F, Sim DS, Park DS, Seo HS (2017) Similarity evolution between graphs: a formal concept analysis approach. J Inf Process Syst 13:1158–1167*

*22.Carlos P, Alberto CL, Juan BM, Alvaro TL, Enrique HV (2017) Fuzzy linguistic recommender systems for the selective diffusion of information in digital libraries. J Inf Process Syst 13:653–667*

*23.Hao F, Park DS, Pei Z (2017) Detecting bases of maximal cliques in social networks. MUE2017 1–1*

*24.Ricci F, Rokach L, Shapira B (2011) Introduction to recommender systems handbook, recommender systems handbook. Springer, Berlin, pp 1–35*

25. *Jafarkarimi H, Sim ATH, Saadatdoost R (2012) A naïve recommendation model for large databases. Int J Inf Educ Technol 2:216–219*

26. *Linden, Gregory D, Smith BR, Zada KN (2015) Automated detection and exposure of behavior-based relationships between browsable items U.S. Patent 9,070,156*

27. *Neil R, Mehdi E, Masashi S, Dain K (2016) Active learning in recommender systems handbook, 2nd edn. Springer, US. ISBN 978-1-4899-7637-6*

28. *Hao F, Min G, Pei Z, Park DS, Yang LT (2015) K-clique communities detection in social networks based on formal concept analysis. IEEE Syst J. https://doi.org/10.1109/jsyst.2433294*
29. *Gregori E, Lenzini L, Mainardi S (2013) Parallel (k)-clique community detection on large-scale networks. IEEE Trans Parallel Distrib Syst 24:1651–1660*

30. *Palla G, Derenyi I, Farksa I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435:814–818*

*(The code of this paper is available at: https://colab.research.google.com/drive/1gmrTQUg23RQ_bdK8CD95t5gEyleftjMG?usp=sharing)*