FilmFinder360

Shruti Pathak(E23MCAG0033) Aryan Sood(E23MCAG0047)

Yatharth Aggarwal(E23MCAG0051) Shreyanshi Mukesh(E23MCAG0014)

Bennett University

Plot Nos 8, 11, TechZone 2, Greater Noida,

Uttar Pradesh 201310

# Abstract

The expansion of digital content platforms all around the globe has drastically increased the need for efficient and personalized recommendation systems, specifically in the movie industry. 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.[1] This research paper explores the application of Collaborative filtering techniques to improve movie recommendation systems. Collaborative Filtering uses the concept of user preferences and their behaviour to provide them with the most accurate movie suggestions.

This study examines two primary types of Collaborative Filtering: User-based and Item-based. By analysing their past preferences and historical data, the system studies different patterns and similarities among users or items, thus allowing to read user preferences for unrated movies. This research emphasizes the significance of overcoming various hurdles such as the cold start problem and data sparsity to ensure the rigidness and effectiveness of the recommendation system.

Moreover, the paper focuses on the convergence of various Machine Learning Algorithms to improve the accuracy of Collaborative Filtering models. Different metrics to evaluate such as precision, recall, and Mean Squared Error (MSE) are applied to evaluate the performance of the proposed system.

The conclusion of this research paper aims to contribute to the ongoing evolution of various recommendation systems, providing them with meaningful insights. Since the digital world is continuously evolving, the proposed approach can become a valuable asset to improve user experience, engagement, and satisfaction in the wide landscape of online movie platforms.

**Keywords:** Content-based Filtering, Collaborative Filtering, Movie Recommendation Systems, Digital Content Platforms, User Preferences, Machine Learning Algorithms, User Experience

# INTRODUCTION Reading the local TV guide, watching CDs and DVDs, watching tapes or slides... these are a thing of the past. The world's largest movie library has been digitized and sent to online streaming services such as Netflix, HBO, or YouTube. These platforms used by smart devices can now help us with perhaps the most difficult part of choosing a movie. It's time for machine learning to show its potential in today's art world. Data scientists are ready to study our behavioral patterns and the behavioral patterns of movies to create a better prediction system for true fans. Movie Recommendation System Project is a new solution designed to keep users interested and involved in the ever-expanding world of entertainment. In the age of multiple content options, the project aims to provide movie recommendations based on personal preferences.  The project is based on two main methods: collaborative filtering and content-based filtering. Collaborative filters gain insights from user behaviour and preferences by analysing patterns to make recommendations based on the choices of similar users. Content-based filtering searches for features of movies such as genre, actor, and director to show similar content that the user has liked in the past. The main objective of our system is to recommend movies to our users based on their viewing history and ratings they provide.[2] The system uses machine learning to create user profiles, learning from past interactions and feedback. It also uses the cosine similarity matrix to evaluate the similarity between videos, ensuring that recommendations are not only accurate but also diverse. The project solves the challenge of the cold start problem with rare user data through a combination of collaborative and content-based approaches. With this movie recommendation, users can expect an increasingly exciting movie experience. As the system continuously learns and adapts to changing preferences, it ensures that video recommendations remain relevant and engaging, making it a smart solution for self-discovery.

# Objectives

The primary concept on which a movie recommendation system is based is quite simple and includes two major elements – users and items. The recommendation system generates movie predictions for the clients or the users, while items are the movies themselves.

The foremost goal of the recommendation system is to filter and predict only those movies that a resultant user is most expected to want to watch. It includes working on a machine-learning model where these recommendation systems use the data from the system’s database. This data helps in predicting the future performance of the machine-learning model which eventually predicts the future behavior of the corresponding user based on the information from the past.

Our project on the Movie Recommendation System has some defined objectives –

It provides movie recommendations to a corresponding user based on their preferences, ensuring a more likable and satisfying viewing experience. It helps users discover and explore new movies that are in sync with their interests, and expand their viewing choices beyond mainstream movies. Movie Recommendation System encourages users to engage on the platform as they offer appealing and relevant suggestions based on their past preferences and history. One of the advantages of the recommendation system is that it reduces the search time spent on finding movies that resonate with the user’s preferences.

We have also used the collaborative filtering technique so that users will be able to explore a diverse and wide range of movie content rather than being confined to a limited set of genres, titles, and overviews. Moreover, we have also introduced an element of surprise as sometimes the recommendation system recommends movies that the user may not have expected or considered before but end up liking and enjoying enhancing the viewing experience. The constructive objective of our recommendation system is to generate movie recommendations in real time when users interact with the platform.

# Scope

The scope of the project primarily lies in enhancing user experience and saving time by generating the best possible movie recommendations according to the user’s history and information from the past based on the user’s preferences. In our movie recommendation system, the systematic collection of comprehensive data is involved including user preferences, ratings, vote counts, and movie metadata.

Algorithms based on both approaches which are Content-Based Filtering and Collaborative Filtering have been applied based on information from the dataset considering historical movie preferences, feedback, and viewing habits of the user.

The system is designed to provide real-time movie suggestions when users will be interacting with the platform. Feature engineering includes extracting relevant features from the data, such as movie genres, user preferences, and collaborative filtering features. Instinctive User Interface has been designed for users to interact with the recommendation system. Machine Learning Model has been integrated with the backend ensuring efficient communication between the components. Detailed and comprehensive documentation including references and insights has been presented for the understanding of model architecture, system constituents, and data sources. The study also includes the comparison between content-based and collaborative filtering based on the results including generated movie recommendations by both the machine learning models on the diverse datasets.

# Related Work

## Content Based Filtering

Content-based filtering has been regarded as one of the most preferable techniques in various movie recommendation systems, working on the suggestions according to different user's preferences. Content based filtering algorithms are based on the contents of an item/product and a list of the user’s preferences which shows kinds of items user may like.[3] It excels in attenuating the cold start problem by recommending users according to their past preferences. In this system, different movie characteristics like genres, actors, overview, and actors are assessed, building a user profile based on their past choices. The core of content-based filtering is building a movie profile and giving more emphasis on features that are based on user specifications for precise understanding.   
The Natural Language Processing (NLP) technique extracts different insights from the textual data, helping in creating diverse specifications. Various machine learning algorithms can then later be used to understand and analyse user preferences and predict which movies are most likely to be preferred by the user. For example, a decision tree or a neural network can be trained on historical user data to make predictions about the likelihood of a user enjoying a particular movie.

### Advantages of Content-Based Filtering:

* **Customization:** Content-based filtering provides recommendations according to individual user preferences, providing a specific and individualized user experience.
* **Independence from User Behaviour:** Content-based filtering does not rely on the behaviour of other users. This makes him effective in the case of dealing with sparse user data or cold start problems for new users.
* **Transparency and clarity:** Recommendations are based on various features like genre, director, overview, actors, and tagline, making the system’s decisions transparent and understandable for the user.
* **Less Dependency on Community Developments:** Content-based systems are less dependent on community developments within a user community, making sure that all the recommendations are completely focused on a particular user’s tastes.
* **Reduction of Cold Start Problem:** Content-based filtering can provide valuable and accurate recommendations even while having limited information about a particular user, thereby solving the issue of Cold Start Problem effectively.

### Disadvantages of Content-based Filtering

* **Limited Providence:** The system often recommends movies that have been watched in the past, thereby limiting exposure to diverse content and minimizing the element of coincidence in discoveries.
* **Hard to capture updating preferences:** Content-based filtering may find it difficult to modify itself according to the continuous change in the preference of the user as it works on the historical data.
* **Dependency on Quality Metadata:** The efficiency of content-based filtering is variable on the availability and quality of metadata for items. Inaccurate or inadequate metadata can often lead to unsatisfactory recommendations.
* **Tough to handle partisanship:** Various features like genres and sentiments obtained from user reviews might be opinionated and might not consider the preferences of the user accurately.
* **A challenge in Recommending New or Niche items:** Content-based systems might face challenges in recommending new or niche items that have inadequate historical data for precise profiling.

## Collaborative Filtering

Collaborative filtering in movie recommendations works on the user preferences to suggest movies accordingly. Collaborative filtering is also called social filtering in which other user's recommendation is used to filter information.[3] User-based collaborative filtering identifies similar users, providing them with customized recommendations. There are three major types of collaborative filtering:

### User Based Collaborative Filtering

#### Advantages:

* **Personalization:** It recommends items based on similar user preferences, providing a customized experience.
* **Zero dependencies on movie attributes:** It doesn’t require any information regarding items, and focuses completely on user interactions.
* **Happenstance:** This can provide users with unexpected movies that they haven’t watched but are preferred by similar users.

#### Disadvantages

* **Cold Start Problem:** Defiance might occur when dealing with new users or users with very little interaction history.
* **Scalability:** Operationally difficult as the user base expands.
* **Scantiness:** Difficult to find a resembling user base when data is sparse.

### Item-Based Collaborative Filtering

#### Advantages

* **No user Cold Start issue:** It is very effective in situations with new users as it does not face the issue of cold start.
* **Extensibility:** More expandable than user-based filtering as item homogeneity is anticipated.
* **Robustness:** Robust when the number of users fluctuates.

#### Disadvantages

* **Item Cold Start:** Faces challenges when a new movie is added which has very little interaction.
* **Subjugation on Item Features:** Requires a lot of information about every item to evaluate similarities.
* **Less Fortuity:** Recommendations might be less provident compared to user-based filtering.

### Model-Based Collaborative Filtering

#### Advantages

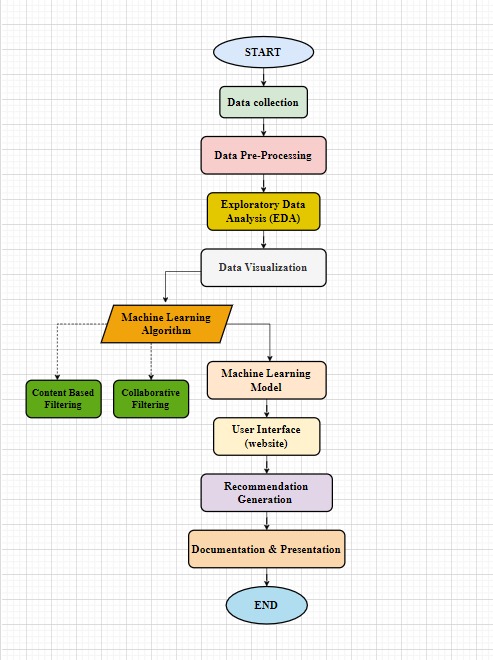
* **Expandability:** Model-based collaborative filtering is expected to be more expandable than memory-based approaches, especially as the total number of items and movies tends to increase with time.
* **Cold Start Problem Alleviation:** It can handle new users or items effectively by maximizing the underlying model, and working on the cold-start problem.
* **Utilizes Additional Information:** Model-based collaborative filtering can easily occupy additional user or item features, improving the recommendation system accuracy by considering diverse factors.
* **Less deficiency issues:** Model-based approaches can provide accurate data even in situations with very little data, where other memory-based methods might struggle.

#### Disadvantages

* **Complexity:** Building and creating a predictive model can also create complexity, requiring strong expertise in Machine learning and Data Science.
* **Data Validation:** Model-based collaborative fileting often calls for expansive and diverse datasets for accurate training, which might not be available everywhere.
* **Comprehensibility:** Models can often lack transparency, making it hard to evaluate the rationale behind specific recommendations, and reducing comprehensibility.
* **Cold Start for Items:** While efficiently dealing with the cold start problem for users, model-based methods might still face issues when new items lack enough interaction history.

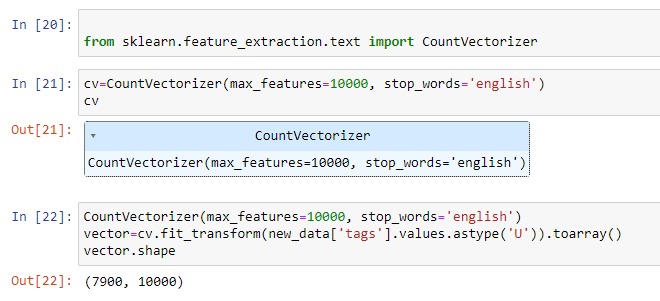
# Proposed Methodology

* **Data Collection-** Data is collected from various resources such as APIs, Kaggle, user reviews, etc.
* **Data pre-processing-** The Data set has been cleaned and transformed into the required format by removing and handling null and duplicated values.
* **Exploratory Data Analysis (EDA)-** Analysis has been done to draw useful insights from the dataset on which further visualization can be done.
* **Data Visualization**- Data visualization has been done by plotting graphs to gather useful information about the movies.
* **Machine Learning Model–** Content-based filtering and collaborative filtering have been applied to generate movie recommendations based on user preferences.
* **User interface –** A user interactive platform has been created for the user in the form of a website.
* **Documentation –** All the comprehensive study has been presented with references and insights about the machine learning models and results generated by them.



### Bag of Words

In a movie recommendation system taking advantage of the Bag of Words approach, textual information of movies is converted into a matrix of word frequencies, ignoring the grammar and the word order. This method streamlines the representation of information regarding different movies. With the help of various Python tools like sci-kit-learn, descriptions of all the movies become vectors in high-dimensional space, allowing for the calculation of semantic similarity between the user’s query and existing movie descriptions. This similarity matrix helps in identifying movies with specific content or themes. By using Bag of Words, the system gives recommendations based on the relevance of words within the textual data, thus providing a systematically efficient means of connecting user preference with different movie features. This approach plays a vital role in natural language processing for retrieving meaningful information from textual data in the context of movie recommendation systems



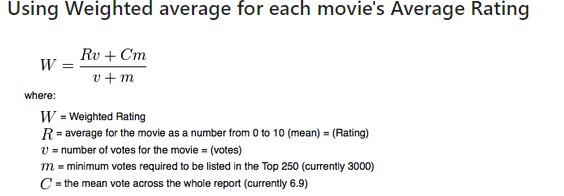
### Average Weighted Recommendation Engines using Python

To create an equally weighted recommendation using Python, assign weights to project attributes (such as genre or director) and calculate the weighted average of user ratings. Use pandas for data management. For example, create a DataFrame that contains video data (with properties such as text format) and other properties including user ratings. Assign weights to features, calculate the weighted index, and recommend items based on weighted averages. This simple example explains the idea; but real...world recommendation engines often require more complex algorithms and take userpreferences into account in a collaborative way. Special libraries such as Surprise or LightFM are very useful for these tasks.

### TF-IDF Method

TFIDF (time frequency converted data frequency) plays an important role in content-based filtering of recommendations. This method calculates the importance of words in a document by combining two important indices. Term frequency (TF) measures the frequency of term in a document, indicating its relevance to certain terms.

Inverse Document Frequency (IDF) measures the importance of a word in the entire text by referring to fewer terms in the document. The TFIDF score is calculated by multiplying the TF and IDF to create a vector representation based on the content of each element. In content filtering, vectors can evaluate similar items and help display items to users based on their interests and the content of the items they are associated with. This method is good at capturing key features of projects and improving the accuracy of content-driven recommendations.



### KNN Model

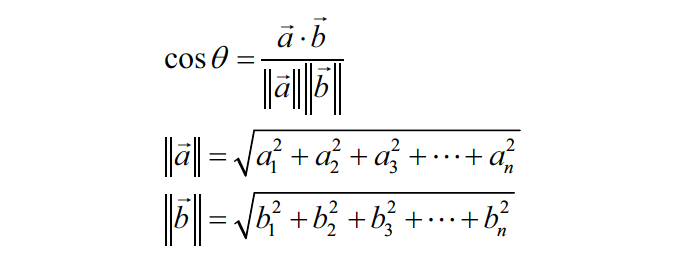
K-Nearest Neighbours (KNN) is a machine learning algorithm commonly used for collaborative filtering in recommendation systems. Collaborative filtering relies on user-item interactions and finds similarities between users or items to make predictions or recommendations.  
Collaborative filtering aims to predict a user's preference for an item based on the preferences of similar users. In the case of a KNN model, the similarity is determined by how closely the preferences of users align.  
Steps to Implement KNN for Collaborative Filtering:

* **User Similarity Calculation:** The system works on the similarity between different users using various metrics such as cosine similarity or Pearson correlation. This comprises comparing the historical choices or ratings of users for different movies.
* **Nearest Neighbours Selection:** The KNN algorithm selects the K most similar users to the target user. These users form the “community” of the target user.
* **Aggregating Ratings:** The system consolidates the ratings or preferences of the selected community to predict how the target user might rate the movies they have not yet watched.
* **Recommendation Generation:** According to the aggregated preferences, the system suggests items to the targeted user.

### Cosine Similarity

A cosine similarity matrix is a mathematical tool used in various fields, including information retrieval, natural language processing, and recommendation systems. It quantifies the similarity between different vectors, often representing documents, items, or features, based on the cosine of the angle between them.

We will be using the cosine similarity to calculate a numeric quantity that denotes the similarity between two movies.[2] When we say two vectors, they could be two product descriptions, two titles of articles or simply two arrays of words.

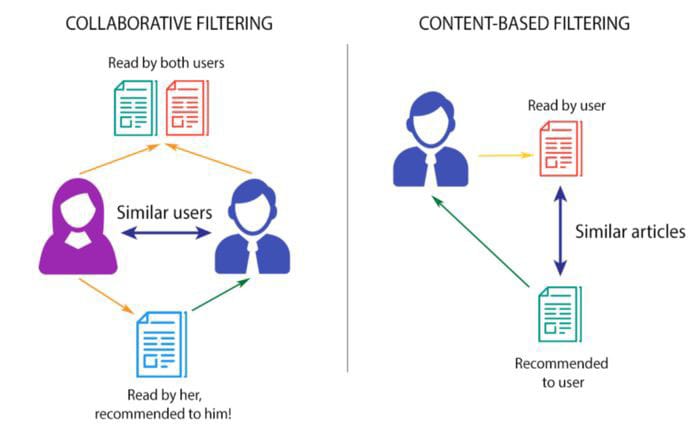


Mathematically, if ‘a’ and ‘b’ are two vectors, cosine equation gives the angle between the two.

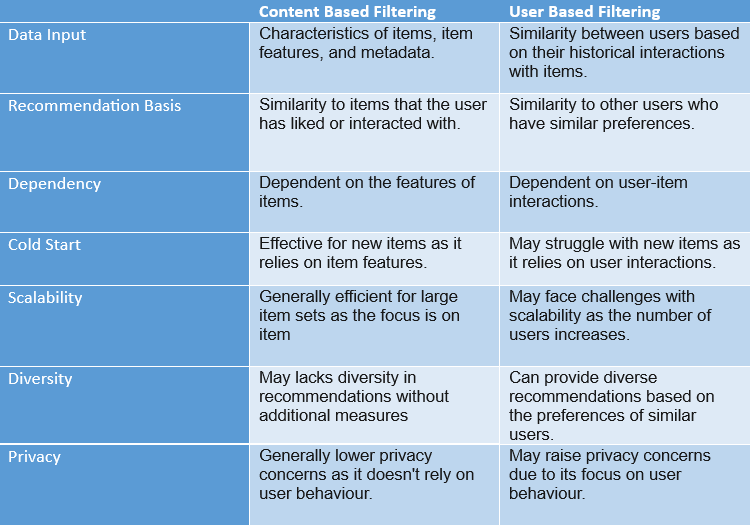
In the context of recommendation systems, the cosine similarity matrix is frequently employed in content-based filtering. Each row of the matrix corresponds to an item (e.g., a movie), and the columns represent the features or attributes associated with those items. The values in the matrix denote the cosine similarity between pairs of items.

Once the cosine similarity matrix is constructed, recommendation systems can utilize it to identify items that are most similar to a user's preferences. For instance, if a user has liked certain movies with specific features, the system can recommend items with high cosine similarity to those liked items, as they share similar characteristics.

**Fig 1.** Difference between Collaborative Filtering and Content-based Filtering



**Table 1.** Difference between Content Based Filtering and User Based Filtering

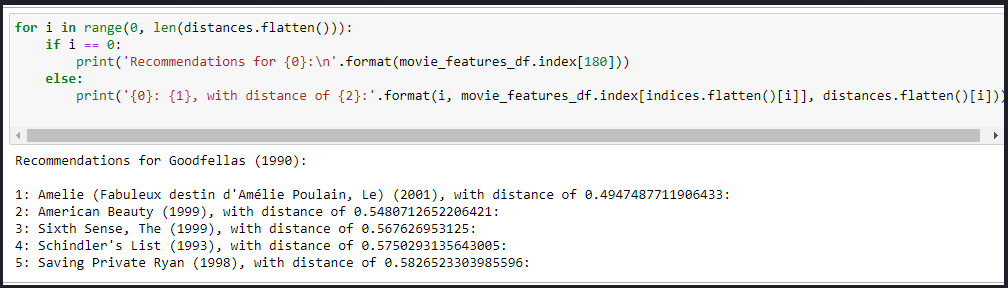


# Discussions and Experimental Analysis

Model-based collaborative filtering is a recommendation system based on the user’s behavior and preference information. It makes recommendations through similar models.[4]

Under collaborative filtering, a user-item matrix is created which represents user interactions with movies. Cosine similarity has been used as a measuring technique for calculating similarity. Further bifurcations are there such as for user-based filtering, where users having the same past preferences are identified, on the other hand, for item-based collaborative filtering, similarity is calculated between movies based on ratings given by the user. K- nearest neighbors for a given item or user are identified having the most similar entities. User ratings from these neighbors are gathered to form personalized recommendations based on the early preferences of the movies by the user. Eventually, the top N recommendations are generated and presented to the user, utilizing collaborative filtering to enhance the performance of the recommendation system and the accuracy and relevance of movie suggestions.

Compared with other recommendation algorithms, it is very convenient to extract item features from objects in a collaborative filtering algorithm.[4] Collaborative filtering algorithms can use other users' responses to similar information to automatically improve the recommended content. Thereby, it can achieve an excellent degree of automation with its high level of intellectualization and personalization.[4]

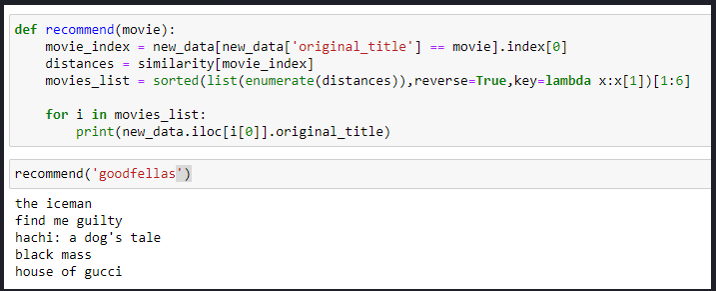


Therefore, the results of the recommendation increase being influenced by other’s behavior.

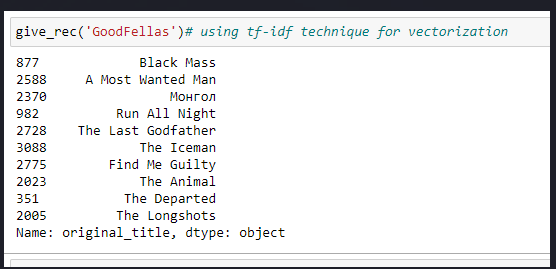
Content-based filtering algorithm takes the personal behavior of individual users as the main measurement standard.[4] Content-based filtering algorithm compares the similarity degree by extracting the text information of things. It can extract the text information of users or items in the background of the software platform, such as the basic personal information of the user and the marketing description text of the product. These are the text feature attributes required by the content-based filtering algorithm. The algorithm would organize them into feature sets based on the individuals.[4] In content-based filtering, items are recommended based on comparisons between the item and user profile is content that is found to be relevant to the user in the form of keywords (or features).[2] A set of assigned keywords (terms, features) might be seen by the user profile which, the algorithm collects from the user interest. A set of keywords (or features) of an item is the Item profile.[2]

TF-IDF and Bag of Words are both methods that are part of Natural Language Processing for extracting information and their application in the Movie recommendation system holds great significance.

Bag of words considers the frequency of words and ignores the structure of terms. In the movie recommendation system, it is applied to movie descriptions and user reviews, by which vectors are created of the words and their frequencies in each document.

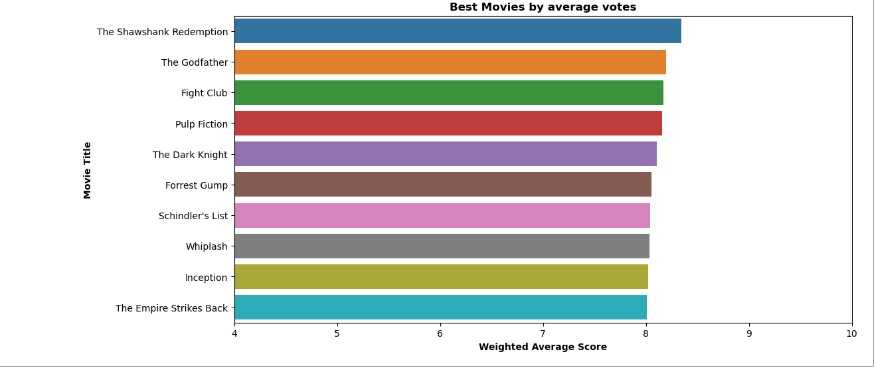


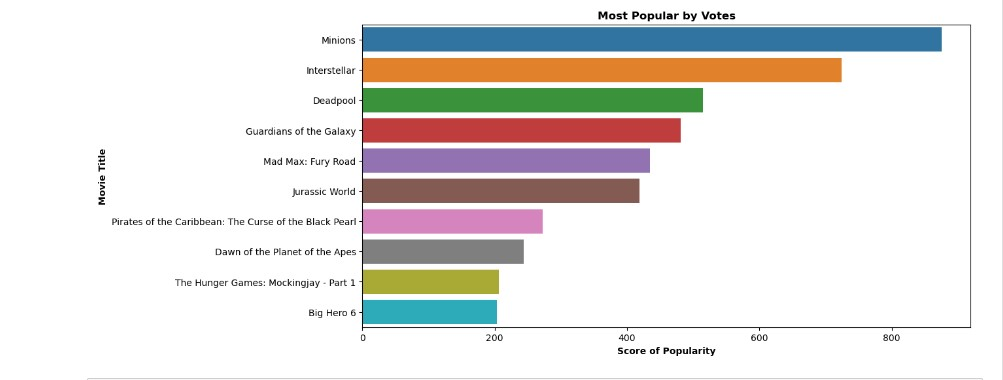
TF-IDF is a statistical measure that evaluates the significance of a word in a document relative to a collection of documents. It identifies the important features or keywords distinguishing movies.

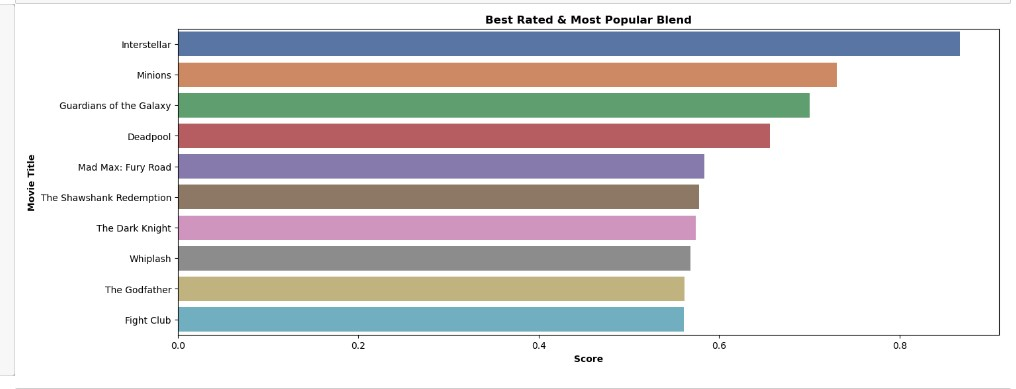


Comparison of similarity measurements in recommendation systems is important as it determines the effectiveness and accuracy of the recommendations generated by the system.[5]

The results have been displayed using the graphs representing the visualization on the movies that are in the data sets. The graphs formed represent the best popular movies by votes, user ratings, etc. which helped us draw insights from the data.







The difference is that collaborative filtering algorithms usually improve their recommendation system through users’ responses to the projects provided by the platform, while content-based filtering algorithm requires the system to accurately extract items’ precise information and users’ operation behaviors. This is to construct the user’s preference portraits of items, to find the target objects that match the traits.[4]

# Conclusion

If we talk about Content-Based algorithms, the simplicity of building a recommendation system and the convenience of its easy operation prove to be its advantage, however, how to improve the accuracy of its extraction for important information by this algorithm is also worth thinking problem.[4]

Collaborative thinking is an advanced technique for generating tailored recommendations in recommendation systems. In collaborative filtering Similarity measurement is a crucial component. The choice of Similarity depends on the specific requirements and goals of the recommendation system, as well as the type of data used.[5] Cosine Similarity is mostly used as a Similarity measure and implemented on real data sets. Moreover, in collaborative filtering models, embedding can be learned automatically. In future, we can work on hybrid recommender systems using clustering and similarity for better performance. Our approach can be further extended to recommend songs, video, venues, news, books, tourism and e-commerce sites, etc.[2]

# Acknowledgement

We would like to express our heartfelt gratitude to our mentor Dr. Mala Saraswat who has guided us throughout our study for this research paper. His guidance has helped us draw insights and inferences from our study. We also want to give special thanks to all the participants who participated in our research and contributed their precious time without whom it would not have been possible for us to complete our research within the required time frame.

We would also like to express our special thanks to our fellow members who have put in their efforts to guide us through their suggestions for improvement in our study. Last but not least, we want to extend our gratitude and appreciation to our faculty for their patience and encouragement during the research paper which inclined the development of this research paper.

# References

[1] Y. Madhavi, M. D. Rokade, and K. Sunil, “Movie Recommendation System using Content-based Filtering.” [Online]. Available: www.ijariie.com633

[2] P. Sharma and L. Yadav, “MOVIE RECOMMENDATION SYSTEM USING ITEM BASED COLLABORATIVE FILTERING,” *International Journal of Innovative Research in Computer Science & Technology*, vol. 8, no. 4, Jul. 2020, doi: 10.21276/ijircst.2020.8.4.2.

[3] A. Aziz and M. Fayyaz, “Comparison of Content Based and Collaborative Filtering in Recommendation Systems.” [Online]. Available: https://www.researchgate.net/publication/348659288

[4] X. Wu, “Comparison Between Collaborative Filtering and Content-Based Filtering,” 2022.

[5] P. Kumar, M. K. Gupta, C. R. S. Rao, M. Bhavsingh, and M. Srilakshmi, “A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, pp. 184–192, Feb. 2023, doi: 10.17762/ijritcc.v11i3s.6180.