# TV Shows Recommender

Niharika Singh
*Dept. of Engineering Physics*
*Indian Institute of Technology Guwahati*
Guwahati, India
niharika.singh@iitg.ac.in

Harsh Raj
*Mehta Family School of Data science and Artificial Integillence*
*Indian Institute of Technology Guwahati*
Guwahati, India
harsh.raj@iitg.ac.in

***Abstract*—The internet is often described as an "infinite pool of data" due to its vast and continuously expanding nature. Today's major challenge is getting the meaningful and desired information from such a humongous pool of data. For this reason, the users want a system that can provide them with suggestions of their own choice from this ample amount of data and the best technology about these is the recommendation system. This paper aims to describe the implementation of a movie and TV shows recommender system using spark. Recommender systems are meant to give suggestions to users based on their taste and liking. As, most of the recommendation system works only with collaborative filtering that analyze the user's behavior and preferences and predict what they would like based on similarity with other users. There are two kinds of collaborative filtering systems; user-based recommender and item-based recommender. Hence, to further improve accuracy in the recommendation system, we implemented more features such as the genre of the movie, the directors, the actors and so on to provide better suggestions.**

## I. Introduction

Modern technology has revolutionized the volume, variety, and velocity at which data are generated. The digitalization of our daily lives has led to the era of big data. Nevertheless, the abundance of data has also given rise to the challenge of information overload and that makes it difficult to process and make decisions. In today's digital age, recommendation systems have become an integral part of our online experiences. These systems play a pivotal role in helping users discover relevant products, services, content, and information in the vast sea of options available on the internet. Whether it's suggesting movies to watch, products to buy, music to listen to, or news articles to read, recommendation systems have the power to enhance user satisfaction, engagement, and decision-making.

At its core, a recommendation system is an information filtering system that leverages data, algorithms, and user behavior analysis to predict and provide personalized suggestions. The goal is to offer users content or items that align with their preferences, interests, and past interactions, ultimately improving their overall user experience. This benefits the providers to make the rise in revenue and customer satisfaction. This paper is structured as follows: Initially, we will discuss a concise overview of some recent and pertinent research carried out in the field of recommender systems, and an explanation of the collaborative filtering technique. Subsequently, we will delve into the implementation of the TV shows recommender using Spark. Finally, a qualitative evaluation of the techniques used will be presented and some future work.

## II. Background

The world of movie and TV shows entertainment has witnessed exponential growth in the volume of data generated, including user ratings, shows metadata, and user preferences. This explosion of data offers tremendous opportunities for enhancing the shows-watching experience through personalized recommendations. The importance of efficient recommendation systems in this context cannot be overstated. Traditional collaborative filtering approaches primarily rely on user ratings to generate recommendations. However, they often fail to capture the nuanced preferences of users, especially when it comes to diverse factors such as shows genres and actors.

### *A. The Challenge of Genre and Actor Preferences*

While user-based collaborative filtering has proven effective in many recommendation systems, it faces limitations when dealing with users who have diverse tastes within the same rating category. For instance, consider two users who both rated a particular movie highly. One user may appreciate the show because it belongs to a genre they adore, while the other user may have liked the show solely because of a specific actor's performance. In such cases, traditional collaborative filtering tends to recommend movies based solely on ratings, potentially leading to inaccurate recommendations. To address this challenge effectively, it is essential to consider additional dimensions of user preferences, such as genre and actor data.

### *B. The Significance of Genre and Actor Data*

Genre and actor information play pivotal roles in shaping a user's show preferences. Incorporating these dimensions into the recommendation system allows for a more nuanced understanding of user tastes. Users who favor a particular genre, such as romance or action, should be presented with TV show recommendations aligned with their genre preferences. Similarly, users who have a penchant for specific actors or actresses should receive recommendations that feature their favorite performers. By integrating genre and actor data into the recommendation model, we aim to enhance the accuracy and relevance of TV shows recommendations.

### C. The Role of Big Data and Distributed Computing

As the volume of shows-related data continues to grow exponentially, traditional computing resources and storage solutions become insufficient for processing and analyzing this vast dataset efficiently. To tackle this challenge, our project leverages the power of Hadoop, an open-source framework for distributed storage and processing of large datasets, and Hadoop Distributed File System (HDFS), a distributed file system designed to handle massive data storage. By adopting Hadoop and HDFS, we can store and manage vast amounts of movie data, including user ratings, genre information, and actor details. Moreover, Hadoop's MapReduce paradigm facilitates parallel processing of data, resulting in significantly reduced computation times for complex recommendation algorithms. This distributed computing approach not only ensures scalability but also enables real-time updates to the recommendation system as new data becomes available. In summary, our project on ”TV shows Recommendation System Using Collaborative Filtering” seeks to address the limitations of traditional user-based collaborative filtering by incorporating genre and actor preferences. We recognize the significance of these dimensions in shaping user tastes and aim to provide more accurate and personalized shows recommendations.

## III. Previous Investigations

In this section, we briefly discuss the previous work that has been done on recommendation systems. There are three main types of recommendation systems, namely collaborative filtering, content-based filtering, and hybrid systems.

*a) Collaborative Filtering:* Collaborative filtering systems analyze the user's behavior and preferences and predict what they would like based on similarity with other users. There are two kinds of collaborative filtering systems; user-based recommender and item-based recommender.

*b) Content Based Filtering:* Content-based systems consider the description and features of an item along with the user's preferences to provide suggestions.

*c) Hybrid System:* Hybrid recommendation systems are a combination of both collaborative and content-based filtering methods. In these types of systems, collaborative and content-based predictions are performed separately and then the results of both techniques are combined to provide recommendations.

However, the most recommendation system is using collaborative filtering methods to predict the needs of the user due to this method gives the most accurate prediction.

### A. Collaborative Filtering Model

The implementation of a collaborative movie recommendation system centered on user ratings as a primary source for generating recommendations. This system is constructed using Mahout, a framework that operates on the Hadoop platform. This approach relies on measuring the similarity or correlation between items, and it encompasses both user-based and item-based collaborative filtering methods. In this section, we will delve into the details of these two collaborative filtering techniques.

*a) User-based filtering:* User-based preferences are commonly employed in the realm of personalized systems. This approach operates under the assumption that a user's preferences are not random and can be discerned through historical analysis. The process commences with users assigning ratings, typically on a scale of 1 to 5, to various items in a catalog. These ratings can be explicit or implicit in nature.

Explicit ratings occur when a user directly rates an item on a scale or provides a thumbs-up/thumbs-down assessment. However, gathering explicit ratings can be challenging, as not all users are inclined to offer feedback. In such cases, implicit ratings are derived from user behavior. For example, if a user repeatedly purchases a product, it implies a positive preference. In the context of movie systems, if a user watches an entire movie, it suggests a level of liking, although the determination of implicit ratings lacks clear-cut rules.

Subsequently, for each user, we identify a predefined number of nearest neighbors. We calculate the correlation between users' ratings using the Pearson Correlation algorithm. The underlying assumption is that when two users' ratings exhibit a high degree of correlation, these users are likely to have similar preferences for items and products. This correlation is then utilized to make item recommendations to users.

*b) Item based filtering:* In contrast to the user-based filtering approach, item-based filtering centers its attention on the resemblance between the items that users have shown interest in, rather than the users themselves. This method precomputes the most similar items in advance. When making recommendations, it suggests items that bear the highest similarity to the item of interest, which is then presented to the user.

## IV. Problem Statement

The TV Shows Recommendation System using Spark is designed to provide personalized and accurate TV show recommendations to users based on their viewing history and preferences. Leveraging Apache Spark, a powerful and distributed computing framework, the system processes large datasets efficiently and delivers real-time recommendations. This project aims to enhance user experience and engagement on a streaming platform by offering content tailored to individual tastes.

### A. Project Objectives

- Develop a scalable recommendation system using Apache Spark.
- Utilize collaborative filtering and content-based filtering techniques for accurate TV show suggestions.
- Implement real-time processing of user interactions to continuously update recommendations.
- Enhance system performance by leveraging Spark's distributed computing capabilities.
- Evaluate and optimize the recommendation algorithms to improve accuracy and relevance

## V. Architecture

Our research approach involves several key steps:

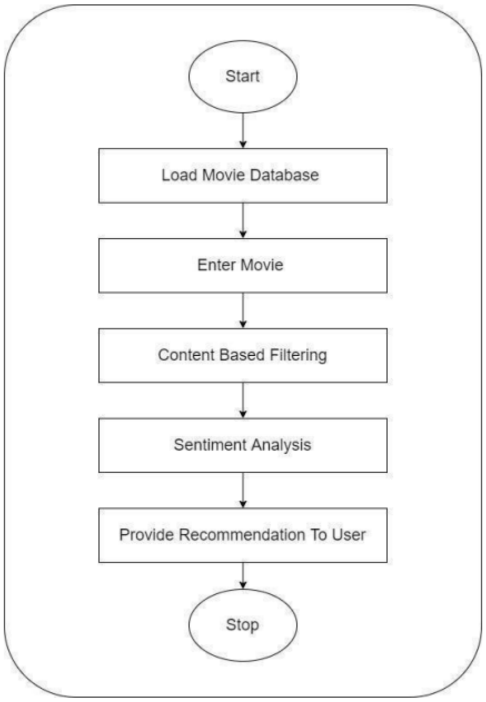


Fig. 1. Flowchart of the TV shows recommendation system.

*a) Data Collection:* We will gather a diverse dataset of user ratings, movie metadata (including genres and actors), and historical user interactions with movies. This data collection will be conducted through a combination of methods, including:

- Web Scraping: We will explore the possibility of scraping movie-related websites or databases to acquire up-to-date movie metadata, including genres and actor information.
- Existing Data Sources: We will also investigate the availability of publicly accessible datasets related to movie ratings, genres, and actor details. Utilizing existing data sources can save time and resources.

*b) Data Preprocessing:* The collected data will undergo preprocessing to clean and structure it for analysis. This includes handling missing data, encoding genres, and actors, and normalizing user ratings.

*c) Collaborative Filtering Enhancement:* We will adapt and extend traditional collaborative filtering algorithms to incorporate genre and actor information. This may involve matrix factorization techniques and hybrid recommendation methods.

*d) Evaluation:* We will evaluate the performance of the enhanced recommendation system using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Precision-Recall.

*e) User Testing:* To validate the system's effectiveness in capturing nuanced preferences, we will conduct user testing and gather feedback from users on the recommendations provided.

Including web scraping as a data collection method and considering the utilization of existing data sources will enhance the research approach's flexibility and data acquisition strategies.

### A. Expected Difficulties

Several challenges are expected during the research process:

*a) Data Quality:* Ensuring the quality and reliability of the dataset, including accurate genre and actor information, can be challenging.

*b) Algorithm Complexity:* Incorporating genre and actor preferences into collaborative filtering algorithms while maintaining computational efficiency can be complex.

*c) User Engagement:* Encouraging user participation in testing and gathering meaningful feedback may pose challenges.

### B. Significance

The proposed research is significant for several reasons:

*a) Enhanced Recommendations:* The research aims to provide more accurate and personalized movie recommendations, improving the movie-watching experience for users.

*b) Practical Application:* The findings can be applied to various recommendation systems beyond movies, such as e-commerce and content streaming platforms.

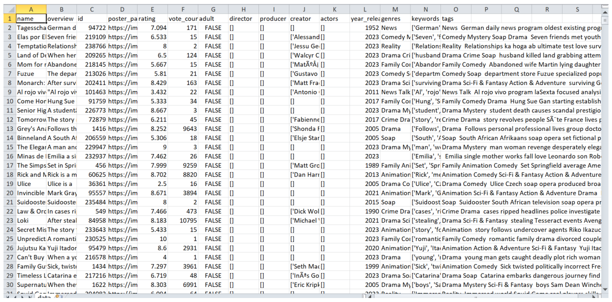
*c) Understanding User Preferences:* The research contributes to a deeper understanding of how users' nuanced preferences for genres and actors can be captured and utilized in recommendation systems.

## VI. Methodology

In this section, we will discuss various steps involved in the designing of a proposed TV shows recommender system in detail. First of all, we need to get datasets representing different TV shows. Then we have to preprocess and clean our raw datasets. After that, we will use the appropriate text vectorizer method to find out the relevance of each word in the corpus. And finally, we need to use a similarity measuring algorithm to find out the similarity between the TV shows, and then the system will recommend videos to the user based on the similarity score.

### A. Data Gathering and Preprocessing

The efficiency of the recommendation system is highly dependent on the amount and quality of data available in the system. So, we took the data from the TMDB website which consists of 8926 TV shows datasets. Each TV show consists of several features such as genre, original language, release date, cast, overview, production company, and so on. Out of this, we took four major features: Genre, Cast, rating and Overview for each show. Then we create a separate table including the mentioned three features as given below : After creating sample datasets, several data cleaning and preprocessing steps were carried out such as removing stop words, null values, and special characters from each document.



| | name | overview | id | poster_pa | rating | vote_cour | adult | director | producer | creator | actors | year_relea | genres | keywords | tags |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Tagesscha | German d | 94722 | https://im | 7.094 | 171 | FALSE | [] | [] | [] | [] | 1952 | News | ['German' | News  German daily news program oldest existing progr |
| 3 | Elas por El | Seven frie | 219109 | https://im | 6.533 | 15 | FALSE | [] | [] | ['Alessand | [] | 2023 | Comedy M | ['Seven', 'f | Comedy Mystery Soap Drama  Seven friends met youth |
| 4 | Temptatio | Relationsh | 238766 | https://im | 8 | 2 | FALSE | [] | [] | ['Jessu Ge | [] | 2023 | Reality | ['Relations | Reality  Relationships ka hoga ab ultimate test love surv |
| 5 | Land of De | When her | 209265 | https://im | 6.5 | 124 | FALSE | [] | [] | ['Walcyr C | [] | 2023 | Drama Cri | ['husband | Drama Crime Soap  husband killed land grabbing attem |
| 6 | Mom for r | Abandone | 218145 | https://im | 5.667 | 15 | FALSE | [] | [] | ['MatÃºÅ¡ | [] | 2023 | Family Co | ['Abandon | Family Comedy  Abandoned wife Martin lying daughter |
| 7 | Fuzue | The depar | 213026 | https://im | 5.81 | 21 | FALSE | [] | [] | ['Gustavo | [] | 2023 | Comedy S | ['departm | Comedy Soap  department store Fuzue specialized popu |
| 8 | Monarch: | After survi | 202411 | https://im | 8.429 | 163 | FALSE | [] | [] | ['Matt Fra | [] | 2023 | Drama Sci | ['surviving | Drama Sci-Fi & Fantasy Action & Adventure  surviving G |
| 9 | Al rojo viv | "Al rojo viv | 101463 | https://im | 3.432 | 22 | FALSE | [] | [] | ['Antonio | [] | 2011 | News Talk | ['Al', 'rojo' | News Talk  Al rojo vivo program laSexta focused analysi |
| 10 | Come Hor | Hung Sue | 91759 | https://im | 5.333 | 34 | FALSE | [] | [] | [] | [] | 2017 | Family Co | ['Hung', 'S | Family Comedy Drama  Hung Sue Gan starting establish |
| 11 | Senior Hig | A studenta | 226773 | https://im | 8.667 | 3 | FALSE | [] | [] | [] | [] | 2023 | Drama My | ['student', | Drama Mystery  student death causes scandal prestigio |
| 12 | Tomorrow | The story | 72879 | https://im | 6.211 | 45 | FALSE | [] | [] | ['Fabienne | [] | 2017 | Crime Dra | ['story', 're | Crime Drama  story revolves people SÃ¨te France lives p |
| 13 | Grey's Ana | Follows th | 1416 | https://im | 8.252 | 9643 | FALSE | [] | [] | ['Shonda F | [] | 2005 | Drama | ['Follows', | Drama  Follows personal professional lives group docto |
| 14 | Binneland | A South Af | 206559 | https://im | 5.306 | 18 | FALSE | [] | [] | ['Elsje Star | [] | 2005 | Soap | ['South', 'A | Soap  South African Afrikaans soap opera set fictional p |
| 15 | The Elegar | A man and | 229947 | https://im | 9 | 3 | FALSE | [] | [] | [] | [] | 2023 | Drama My | ['man', 'wo | Drama Mystery  man woman revenge desperately elega |
| 16 | Minas de I | Emilia a si | 232937 | https://im | 7.462 | 26 | FALSE | [] | [] | [] | [] | 2023 | | ['Emilia', 's | Emilia single mother works fall love Leonardo son Rob |
| 17 | The Simps | Set in Spri | 456 | https://im | 7.999 | 9259 | FALSE | [] | [] | ['Matt Gro | [] | 1989 | Family Ani | ['Set', 'Spr | Family Animation Comedy  Set Springfield average Ame |
| 18 | Rick and M | Rick is a m | 60625 | https://im | 8.702 | 8820 | FALSE | [] | [] | ['Dan Harr | [] | 2013 | Animation | ['Rick', 'me | Animation Comedy Sci-Fi & Fantasy Action & Adventure |
| 19 | Ulice | Ulice is a | 36361 | https://im | 2.5 | 16 | FALSE | [] | [] | [] | [] | 2005 | Drama Co | ['Ulice', 'Cz | Drama Comedy  Ulice Czech soap opera produced broa |
| 20 | Invincible | Mark Gray | 95557 | https://im | 8.671 | 3894 | FALSE | [] | [] | [] | [] | 2021 | Animation | ['Mark', 'G | Animation Sci-Fi & Fantasy Action & Adventure Drama |
| 21 | Suidooste | Suidooste | 235484 | https://im | 8 | 2 | FALSE | [] | [] | [] | [] | 2015 | Soap | ['Suidoost | Soap  Suidooster South African television soap opera pr |
| 22 | Law & Ord | In cases ri | 549 | https://im | 7.466 | 473 | FALSE | [] | [] | ['Dick Wol | [] | 1990 | Crime Dra | ['cases', 'ri | Crime Drama  cases ripped headlines police investigate |
| 23 | Loki | After steal | 84958 | https://im | 8.183 | 10795 | FALSE | [] | [] | ['Michael ' | [] | 2021 | Drama Sci | ['stealing', | Drama Sci-Fi & Fantasy  stealing Tesseract events Aveng |
| 24 | Secret Mis | The story | 233643 | https://im | 5.433 | 15 | FALSE | [] | [] | [] | [] | 2023 | Animation | ['story', 'fo | Animation  story follows undercover agents Riko Ikazuc |
| 25 | Unpredict | A romanti | 230525 | https://im | 10 | 1 | FALSE | [] | [] | [] | [] | 2023 | Family Co | ['romantic | Family Comedy  romantic family drama divorced couple |
| 26 | Jujutsu Ka | Yuji Itador | 95479 | https://im | 8.6 | 2931 | FALSE | [] | [] | [] | [] | 2020 | Animation | ['Yuji', 'Ita | Animation Action & Adventure Sci-Fi & Fantasy  Yuji Itac |
| 27 | Can't Buy | When a yo | 216578 | https://im | 4 | 1 | FALSE | [] | [] | [] | [] | 2023 | Drama | ['young', 'r | Drama  young man gets caught deadly plot rich woman |
| 28 | Family Gu | Sick, twiste | 1434 | https://im | 7.297 | 3961 | FALSE | [] | [] | ['Seth Mac | [] | 1999 | Animation | ['Sick', 'twi | Animation Comedy  Sick twisted politically incorrect Fre |
| 29 | Timeless L | Catarina e | 217216 | https://im | 6.719 | 48 | FALSE | [] | [] | ['InÃªs Go | [] | 2023 | Drama Soa | ['Catarina' | Drama Soap  Catarina embarks dangerous journey find |
| 30 | Supernatu | When the | 1622 | https://im | 8.303 | 6991 | FALSE | [] | [] | ['Eric Kripk | [] | 2005 | Drama My | ['boys', 'Sa | Drama Mystery Sci-Fi & Fantasy  boys Sam Dean Winche |

Fig. 2. TV shows data with multiple features.

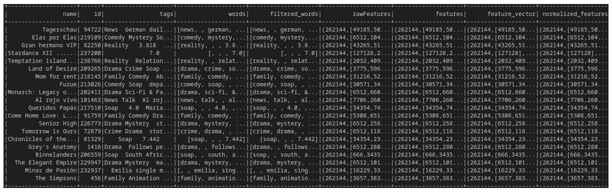
## B. Text Vectorization

After getting the filtered dataset, we have to convert this corpus into numerical representation i.e. in the form of vectors. This vector plays an important role in building a recommendation model in the later stage. There are several text vectorization techniques in natural language processing such as Bag of Words, TF-IDF (Term Frequency and Inverse Document Frequency), Word2Vec, and so on. In this paper, we used the TF-IDF text vectorization method to convert our filtered corpus into vectors. TF-IDF shows how important a word is to a document. It is obtained by computing two metrics: TF and IDF.

*1) Term frequency (TF):* Term frequency finds out the frequency of appearance of a particular term in the document. Suppose we have a document d, then term frequency of a word ‘w’ will be

$$TF = \frac{\texttt{No. of word 'w' in document 'd'}}{\texttt{Total no. of document 'd'}}$$

Using this formula, we calculate term frequency of each word in a corpus and obtained result is shown in the following table



```
+--------------------+------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------
|                name|    id|                tags|               words|      filtered_words|         rawFeatures|            features|      feature_vector| normalized_features
+--------------------+------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------
|          Tagesschau| 94722|News  German dail...|[news, , german, ...|[news, , german, ...|(262144,[49185,50...|(262144,[49185,50...|(262144,[49185,50...|(262144,[49185,50...
|       Elas por Elas|219109|Comedy Mystery So...|[comedy, mystery,...|[comedy, mystery,...|(262144,[6512,104...|(262144,[6512,104...|(262144,[6512,104...|(262144,[6512,104...
|    Gran hermano VIP| 82250|Reality   3.818  ...|[reality, , , 3.8...|[reality, , , 3.8...|(262144,[43265,51...|(262144,[43265,51...|(262144,[43265,51...|(262144,[43265,51...
|Stardance XII ......|237200|             7.0    |          [, , , 7.0]|          [, , , 7.0]|(262144,[127128,2...|(262144,[127128,2...|(262144,[127128],...|(262144,[127128],...
|Temptation Island...|238766|Reality  Relation...|[reality, , relat...|[reality, , relat...|(262144,[2032,409...|(262144,[2032,409...|(262144,[2032,409...|(262144,[2032,409...
|      Land of Desire|209265|Drama Crime Soap ...|[drama, crime, so...|[drama, crime, so...|(262144,[3775,596...|(262144,[3775,596...|(262144,[3775,596...|(262144,[3775,596...
|        Mom for rent|218145|Family Comedy  Ab...|[family, comedy, ...|[family, comedy, ...|(262144,[31216,52...|(262144,[31216,52...|(262144,[31216,52...|(262144,[31216,52...
|               Fuzue|213026|Comedy Soap  depa...|[comedy, soap, , ...|[comedy, soap, , ...|(262144,[30571,34...|(262144,[30571,34...|(262144,[30571,34...|(262144,[30571,34...
|Monarch: Legacy o...|202411|Drama Sci-Fi & Fa...|[drama, sci-fi, &...|[drama, sci-fi, &...|(262144,[6512,660...|(262144,[6512,660...|(262144,[6512,660...|(262144,[6512,660...
|        Al rojo vivo|101463|News Talk  Al roj...|[news, talk, , al...|[news, talk, , al...|(262144,[7706,260...|(262144,[7706,260...|(262144,[7706,260...|(262144,[7706,260...
|       Queridos Papás|217510|Soap   4.0  Maria...|[soap, , , 4.0, ,...|[soap, , , 4.0, ,...|(262144,[34354,74...|(262144,[34354,74...|(262144,[34354,74...|(262144,[34354,74...
|Come Home Love: L...| 91759|Family Comedy Dra...|[family, comedy, ...|[family, comedy, ...|(262144,[5308,651...|(262144,[5308,651...|(262144,[5308,651...|(262144,[5308,651...
|         Senior High|226773|Drama Mystery  st...|[drama, mystery, ...|[drama, mystery, ...|(262144,[6512,258...|(262144,[6512,258...|(262144,[6512,258...|(262144,[6512,258...
|     Tomorrow is Ours| 72879|Crime Drama  stor...|[crime, drama, , ...|[crime, drama, , ...|(262144,[6512,118...|(262144,[6512,118...|(262144,[6512,118...|(262144,[6512,118...
|Chronicles of the...| 81329|    Soap   7.442    |   [soap, , , 7.442]|   [soap, , , 7.442]|(262144,[34354,23...|(262144,[34354,23...|(262144,[34354,23...|(262144,[34354,23...
|      Grey's Anatomy|  1416|Drama  Follows pe...|[drama, , follows...|[drama, , follows...|(262144,[6512,280...|(262144,[6512,280...|(262144,[6512,280...|(262144,[6512,280...
|        Binnelanders|206559|Soap  South Afric...|[soap, , south, a...|[soap, , south, a...|(262144,[666,3435...|(262144,[666,3435...|(262144,[666,3435...|(262144,[666,3435...
|  The Elegant Empire|229947|Drama Mystery  ma...|[drama, mystery, ...|[drama, mystery, ...|(262144,[6512,101...|(262144,[6512,101...|(262144,[6512,101...|(262144,[6512,101...
|      Minas de Pasión|232937|  Emilia single m...|[, , emilia, sing...|[, , emilia, sing...|(262144,[16229,33...|(262144,[16229,33...|(262144,[16229,33...|(262144,[16229,33...
|        The Simpsons|   456|Family Animation ...|[family, animatio...|[family, animatio...|(262144,[3657,383...|(262144,[3657,383...|(262144,[3657,383...|(262144,[3657,383...
+--------------------+------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------
```

Fig. 3. Normalised vector data.

*2) Inverse document frequency (IDF):* Higher the term frequency (TF), we can say that the word has a significant meaning in the recommendation of the particular TV show. But this is not applicable in some cases such as stop words, or less meaningful words. For example, suppose a document has the word ‘m’ four times. In that case, it doesn’t mean the word ‘movie’ is significant while recommending because this word is less significant and doesn’t play a vital role while recommending. So to overcome this issue, we have to calculate inverse document frequency.

$$TF = \log \frac{\texttt{Total no. of shows}}{\texttt{No.of shows of doc 'd' which contains word 'w'}}$$

We take logarithm while calculating IDF because sometimes the value of IDF without log is larger and more dominant over TF. So, to negate its dominance and make it have the same effect as TF, we use a log while calculating IDF.

*3) TF-IDF:* Word in the document is most important when it occurs frequently in its own document but rarely in other documents. Since TF gives how frequent or common a word is in its own document, and IDF gives how rare a word is in other documents so we need to multiply both terms to find the actual relevance of the word in the document. Table I and Table II show TF and IDF respectively, thus we calculate the value from both tables to get TF-IDF.
TF-IDF = TF*IDF

## C. Cosine Similarity

The similarity score is a numeric value that ranges from zero to one and is used to determine how similar two items are to each other on a scale from zero to one. This score is obtained by comparing the texts of the two documents and measuring their similarity. This can be calculated using the cosine similarity measure, which determines the similarity of texts regardless of their size. Cosine similarity is a measure used to calculate the cosine of the angle between two vectors projected in a multi-dimensional space.

$$\cos(\theta) = \frac{\mathbf{A}.\mathbf{B}}{||\mathbf{A}||\ ||\mathbf{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Here the range of cosine similarity lies between 0 and 1.

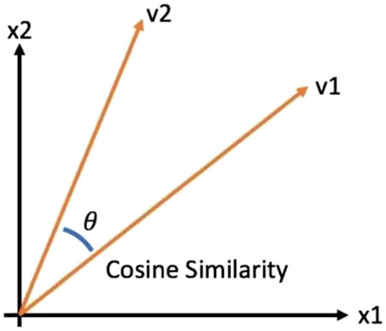


Fig. 4. Cosine similarity/distance.

Value 1 means the document is exactly similar to each other whereas 0 means two documents are completely dissimilar. In the diagonal, the value is 1 because here both shows are the same shows. Now we have a similarity score between each shows. Finally, our recommendation engine can suggest videos based on this similarity score.

## VII. Results

Click here to visit the website.

## A. User Registration and Login

The user registration and login functionalities have been successfully implemented, enhancing the overall user experience by providing a personalized space for each user. Upon registration, users create accounts with unique credentials, allowing them to log in securely. This feature ensures the security of user data and offers a seamless transition between sessions.

### *B. Favorite TV shows Section*

The Favorite TV shows section empowers users to curate a list of TV shows they particularly enjoy. This functionality contributes to user engagement and loyalty, enabling users to easily access and manage their preferred content. The implementation of this feature involves creating a dynamic and interactive interface where users can add and remove TV shows from their favorites.

### *C. Genre Selection*

The Genre Selection feature enables users to explore TV shows based on their preferred genres. This is achieved by categorizing TV shows into different genres and allowing users to filter content based on their interests. The underlying mechanism involves tagging each TV show with relevant genres and optimizing the search functionality to deliver genre-specific recommendations

### *D. Detailed Movie Information*

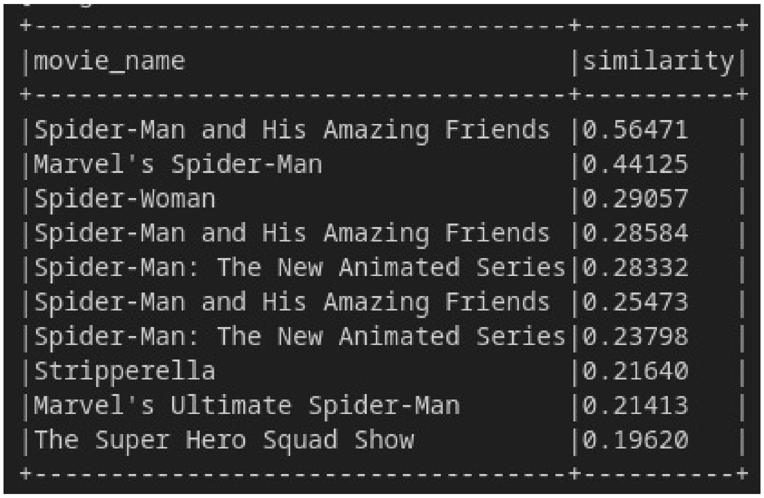
Clicking on a TV show reveals detailed information, providing users with insights into the selected content. This feature enriches the user experience by offering comprehensive details such as the plot summary, cast and crew information, release dates, and viewer ratings. The system retrieves and displays this information from a structured and well-maintained database.

### *E. Content-Based Recommendation System*

The Content-Based Recommendation System utilizes cosine similarity to suggest TV shows that are most similar to the one selected by the user. This system leverages natural language processing techniques to analyze textual features such as show descriptions, genres, and cast. By calculating cosine similarities between the feature vectors of different shows, the system identifies content with high relevance to the user's preferences. Recommendations for movie Spider-Man are:

### *F. User-Based Collaborative Filtering*

The User-Based Collaborative Filtering algorithm employs the Alternating Least Squares (ALS) model with 24 latent factors and a regularization parameter of 0.6. Collaborative filtering works by predicting user preferences based on the preferences of similar users. The ALS model factors in latent features representing user and item preferences, and regularization helps prevent overfitting. In the context of TV show recommendations, the model processes the historical watch data of users, identifying patterns and correlations between their preferences. By capturing latent factors that influence user choices, the ALS model produces accurate and personalized recommendations. The regularization parameter fine-tunes the model to balance accuracy and generalization.



```
+-----------------------------------+----------+
|movie_name                         |similarity|
+-----------------------------------+----------+
|Spider-Man and His Amazing Friends |0.56471   |
|Marvel's Spider-Man                |0.44125   |
|Spider-Woman                       |0.29057   |
|Spider-Man and His Amazing Friends |0.28584   |
|Spider-Man: The New Animated Series|0.28332   |
|Spider-Man and His Amazing Friends |0.25473   |
|Spider-Man: The New Animated Series|0.23798   |
|Stripperella                       |0.21640   |
|Marvel's Ultimate Spider-Man       |0.21413   |
|The Super Hero Squad Show          |0.19620   |
+-----------------------------------+----------+
```

Fig. 5. website image.

### *G. User Watch Data Storage*

The effective storage of user watch data is crucial for the success of collaborative filtering. User watch histories are stored in a structured database, allowing the system to capture and analyze user preferences over time. This historical data serves as the foundation for training and refining the collaborative filtering model, ensuring that recommendations align with individual viewing habits.

### *H. Model Performance*

The ALS model's performance is assessed through cross-validation and evaluation metrics such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE). These metrics quantify the accuracy of the model's predictions against the actual user preferences. The choice of 24 latent factors and a regularization parameter of 0.6 is based on a balance between model complexity and performance. The collaborative filtering approach exhibits robust performance, effectively predicting user preferences and providing valuable insights into content relevance. Continuous monitoring and evaluation of the model's performance contribute to ongoing refinement and optimization.

### *I. Overall System Performance*

The integration of content-based and collaborative filtering recommendations results in a well-rounded and personalized viewing experience for users. The system's performance is characterized by its smooth navigation, quick response times, and the delivery of relevant and engaging content. The combination of algorithmic approaches ensures a versatile and adaptive recommendation system that caters to diverse user preferences.

## VIII. Conclusion

The recommender system has become more and more important because of the information overload. For the content-based recommender system specifically, we attempt to find a new way to improve the accuracy of the representative of the shows. For the problems we mentioned at the beginning, firstly, we use a content-based recommender algorithm which means there is no cold start problem. Our proposed algorithm utilizes textual metadata, such as the plot, cast, genre, release year, and other production details of movies, to analyze and recommend the most similar shows. The user only needs to input a movie of interest, and the system will generate appropriate recommendations. We tested our algorithm on a subset of the shows available on IMDb and found that the cosine similarity measure was effective for forecasting recommendations in recommendation systems. Website Page :

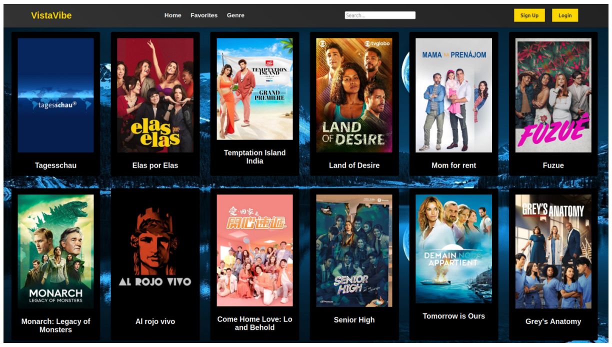


Fig. 6. Cosine similarity/distance.

## References

[1] C. -S. M. Wu, D. Garg and U. Bhandary, ”Movie Recommendation System Using Collaborative Filtering,” 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018, pp. 11-15, doi: 10.1109/ICSESS.2018.8663822.

[2] M. Wasid and K. Anwar, ”An Augmented Similarity Approach for Improving Collaborative Filtering based Recommender System,” 2022 International Conference on Data Analytics for Business and Industry (ICDABI), Sakhir, Bahrain, 2022, pp. 751-755, doi: 10.1109/ICDABI56818.2022.10041638.