Movie Recommendation System Using Machine Learning

BY

Saikat Rahman

ID: 221-25-115

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science and Engineering

Supervised By

Professor Md. Monzur Morshed

Professor

Department of CSE

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

This Project/Thesis titled “**Movie Recommendation System Using Machine Learning**”, submitted by Saikat Rahman (ID: 221-25-115) to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 17-01-2023.

BOARD OF EXAMINERS

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Professor Md. Monzur Morshed, Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

[illegible]

Professor Md. Monzur Morshed
Professor
Department of CSE
Daffodil International University

Submitted by:

[illegible]

Saikat Rahman
ID: 221-25-115
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year thesis successfully.

We really grateful and wish our profound our indebtedness to **Professor Md. Monzur Morshed**, **Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Machine Learning*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Professor Dr. Touhid Bhuiyan and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

In today's incredibly busy environment, recommendation systems are becoming more and more crucial. The most frequently used areas for recommender systems are, among other things, books, news, articles, music, videos, and movies. I have suggested a movie recommendation system in this paper. The importance of recommendation systems can be attributed to their ability to help people make the best decisions without having to use their cognitive faculties. Due to the numerous duties that need to be completed in the 24 hours available, people are constantly pressed for time. Like any other kind of recommendation system, a movie recommendation system relies on the similarities between users (collaborative filtering) or the user's desired activity (content-based filtering) to provide suggestions. By fusing collaborative and content-based filtering, we can get beyond the limitations of both approaches and build a more robust recommendation system. In this research, a movie recommendation system is presented that uses a hybrid model based on cosine similarity techniques and feature extraction. The list is ranked according to how highly members of the website rated each film. After that, it will be simple for the user to browse the suggestions and find their preferred movie. As a result, movie recommendations will be more pertinent to users' needs. The process culminates in a search of movie databases for all pertinent information, including popularity and beauty, needed for a recommendation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Introduction

The explosion in data storage has ushered in a new era of knowledge. When it comes to leveraging data to create better systems, recommendation systems play a vital role. It can be used by the user to select the best choice from the available options. Today, recommendations are what we buy online. When we wish to do things like purchase books, listen to music, view movies, etc., a recommendation system is usually active in the background, making suggestions to the user based on his or her past actions. Businesses like Netflix and Amazon utilize recommender systems to help consumers find the best movies and TV shows for them.

Users have come to anticipate consistently high-quality suggestions from recommender systems. When it comes to services that can't provide useful recommendations, they have a low tolerance. A user will cease using a music streaming app if it is unable to accurately forecast and play music that the user enjoys listening to. As a result, several IT firms are investing heavily in refining their own recommendation mechanisms. However, there is more to this issue than meets the eye.

Each individual has unique tastes and interests. Also, even a single user's preferences might shift based on a wide range of contextual circumstances, including their current state of mind, the time of year, and the task at hand. It's not the same music that one might listen to while working out as it is when making supper. Exploration vs. exploitation is another challenge faced by recommendation systems. They need to leverage what they currently know about the user effectively while also venturing into uncharted territory to learn more.

The steps for developing the movie recommendation system are as follows. Analyze the data first using exploratory data analysis (EDA). Create the recommendation system after that. Lastly, ask for references. There are three main kinds of recommendations used by recommendation systems, and they are: collaborative filtering, content-based filtering, and demographic filtering. In this study, I use both collaborative filtering and content-based filtering. All other recommender systems will use the small dataset, but I'm going to make a simple recommender that draws from the large dataset's movie selection.

It's common practice for recommender systems to take one of two major routes. Content-based filtering is one example; in which we provide recommendations based on a user's preferences as determined by their profile data. The other is collaborative filtering, in which we attempt to build groups of users who are similar to one another and then utilize the data collected from these groups to generate individual suggestions.

1.2 Motivation

The field of recommendation systems is very broad and is applied in all industries. Particularly with streaming services, recommendation algorithms are an integral component of item suggestions. For streaming movie services such as Netflix, recommendation algorithms are crucial for assisting consumers in discovering new films to watch. Recommendations are used because they save time, which is why they are important in many different contexts. It is utilized in a variety of real-world contexts, including entertainment, e-commerce, services, social media, and others. Filtering and predicting only the movies that a matching user is most likely to wish to see is the main objective of movie recommendation systems. Users constantly expect obtaining great recommendations. They have a poor tolerance for services that cannot provide appropriate advice. If a music streaming app cannot forecast and play the user's favorite music, the user will discontinue usage of the app. IT businesses have devoted significant resources to improving their recommendation systems as a result. However, the situation is more complex than it first looks. In addition, the preferences of a single user may vary dependent on a wide range of factors, such as the user's mood, the season, or the kind of activity they are engaged in. Since every user is unique, this strategy is judged insufficiently complex. In order to achieve its main aims of increasing demand and involving customers, a movie recommendation system must do both. A variety of screening techniques and algorithms are used by movie recommendation systems to guide consumers toward the most pertinent movies.

1.3 Purpose of the research

Recommender systems are information filtering technologies that attempt to anticipate the rating of people and objects, mostly based on big data, in order to propose their preferences. Movie recommendation systems facilitate the classification of people with like interests. In addition, many factors go into making unique compilations of useful and interesting content for each user or individual. The algorithms powering recommendation systems are built on artificial intelligence, and they utilize this information to compile a list of options that are most likely to be of interest to a certain user. The user's profile, search and browsing history, what other people with similar traits or demographics are watching, and the possibility that you will watch the same movies all go into determining these results. This is done by using a mixture of heuristics and predictive modeling on the already-collected data.

A recommendation system's main goal is to estimate the user's final product rating. The program helps the user choose the optimal solution from a set of alternatives. Multiple corporations employ recommendation systems to improve customer service and boost revenue. Some of these corporations are Netflix, YouTube, and others. It is a worthwhile research topic because, since human preferences change over time, it can be difficult to identify what a user wants from the resources that are accessible. The study's goals are to cope with the massive amount of data and filter important information, suggest related films based on user preferences, and do user rating of the selected movie.

1.4 Expected Outcome

This study focuses on a recommendation system that suggests various items to consumers. This system will suggest films to users. This technology will provide more accurate results than the current methods. Ratings from individual users form the basis of the current method. There's a chance this won't work for people whose preferences don't align with the system's, given that every person is different. This technique compares users, finds those who have similar tastes in movies, and then makes recommendations based on those users' reviews. This will provide the user with a very specific recommendation.

Machine learning is used by a movie recommendation system (also known as a movie recommender system) to make predictions about or filter recommendations for movies for individual users (ML). The kinds of reviews that a film receives from viewers determine how well-liked it is. The decisions made by other users are also influenced by these reviews. The use of recommendation systems in the entertainment industry is common when watching movies, listening to music, or watching any TV program. The study's goals are to cope with the massive amount of data and filter important information, suggest related films based on user preferences, and do sentiment analysis on reviews of the selected movie.

1.5 Layout of the Report

The rest of the paper is organized into the following sections: The research's purpose, the rationale for the investigation, and the anticipated results are all covered in the first chapter. The scope and challenges of the issue are covered in the second chapter, along with related studies and a summary of the research. The methodology of this study, the technique of data collecting, statistical analysis, and potential applications are all covered in Chapter 3. In addition to discussing the study's numerical and graphical findings, chapter four also includes experimental evaluation and other pertinent discussions. Chapter 5 discusses the importance of this study to society. An overview of this research is provided in the sixth chapter.

CHAPTER 2
ANALYSIS OF THE CONTEXT

2.1 Introduction

Over the last decade, several different types of recommendation systems have been created and put into use. Content-based, collaborative, knowledge-based, utility-based, hybrid, etc. are only some of the methods used by these recommendation systems [23]. The majority of online recommendation systems for a range of things rely on ratings from past users to produce suggestions for current users with comparable interests. Jung, Harris, Webster, and Herlocker (2004) devised one such approach for enhancing search results. The system solicits feedback from users about the adequacy of search results in meeting their information requirements. These ratings are then used to generate suggestions for subsequent users with similar requirements.

The kinds of evaluations a movie receives from the general public determine its level of popularity. Other users' choices are influenced by these reviews as well. Content-based filtering (CBF), collaborative filtering (CF), and hybrid filtering are the three methods used most frequently to create recommendation systems CBF is a method used to evaluate each item's content and suggest further goods with comparable qualities. It is also common knowledge that many recommendation systems employ the hybrid-filtering methodology, which combines the advantages of both CBF and CF procedures.

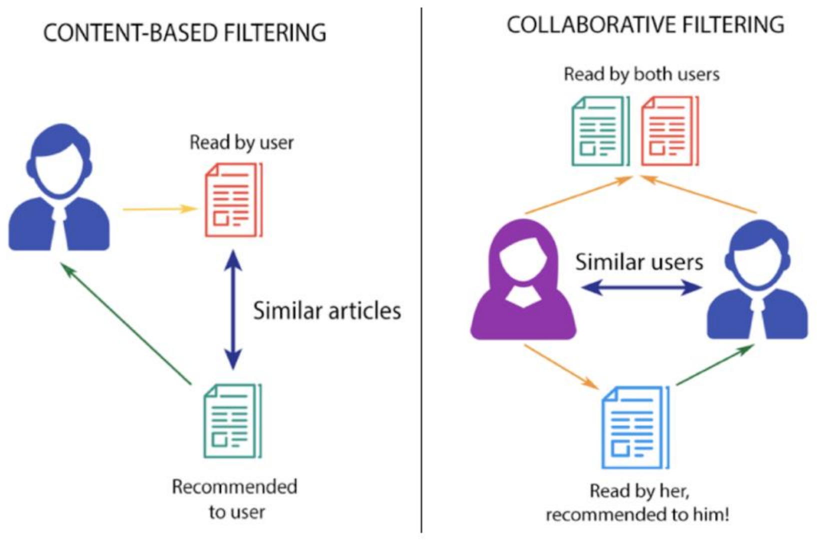


Fig 2.1: Both Filtering Content Base and Collaborative

2.2 Related Works

As far as we know, a number of methods have been put forth for recommendation systems. One of the most effective information management systems is recommendation systems (RS), which was first developed by the Tapestry project in 1992 [1]. Collaborative filtering (CF), which is based on the nearest-neighbor principle, is now the most effective method used by movie recommendation systems. Inputs from diverse users are used by collaborative systems, which then perform various comparisons on these inputs [2]. For instance, movie recommendation systems analyze user ratings for various movies [3] to discover additional users who share their opinions and suggest movies they have rated highly. For enhancing search results, Jung, Harris, Webster, and Herlocker (2004) developed one such approach. Longer and more detailed search queries are encouraged by the system, which also gathers user reviews on whether or not the results of the search satisfied their information needs. In collaborative filtering systems, memory-based and model-based strategies are both employed [2]. In order to provide recommendations, memory-based systems continuously examine user data [2]. The goal of model-based techniques is to create a model of a user's behavior, which is then used, together with a few other parameters, to forecast future behavior.

The factored user-genre matrix approach was used to identify latent genre influences in order to improve user profiles. In [11], content-based filtering using user category-based filtering was proposed as a solution to "item cold start," one of the most serious difficulties of recommender systems. Item cold starts refer to new items that have not received sufficient user feedback and, as a result, might impact the correctness of the recommendation. The authors of [12] propose the SEP framework for addressing recommender system concerns such as cold starts and sparsity. The authors of [13] created a strategy for a recommender system that uses genre information to address both the coverage and redundancy of the algorithms. According to our understanding, the majority of relevant efforts focus on developing a unique method for identifying user similarities, but the prediction of movie genres remains understudied. However, it may play an important role in suggesting creative products to consumers.

Balabanovic et al. developed a content-based recommendation system that could be used to many other fields, including but not limited to literature, film, video, and music. [5]. It makes use of a variety of elements, including author, genre, and terms that are used most frequently. These are often extracted using TF-IDF and Information Gain (IG) [4], [6].

In [9], Prafulla Bafna introduced a categorization strategy for text. In this method, stop words are eliminated from the text, and TF-IDF is used to normalize the data. To get the clusters, agglomerative clustering and fuzzy k-means are utilized. This method addresses the drawback of [5] by normalizing the data using TF-IDF. There have been several categorization systems for text. A hybrid strategy for a system of movie recommendation had been put forth by George et al. [7]. This is a Web-based recommendation tool that uses a graphical user interface to gather user movie ratings on a range of one to five.

Rasheed et al. [2], [3] used visual characteristics such as average shot duration, color variation, motion content, and the lighting key extracted from movie trailers to predict movie genres. Using hierarchical SVM, Yuan et al. [4] classified genres using visual information from films, including temporal and spatial characteristics. Zhou et al. [5] categorized film trailers using the bag-of-visual-words paradigm, using shot classes as vocabulary. Using a meta-heuristic optimization approach, Huang et al. [6] retrieved both visual and auditory data from movie trailers and conducted genre categorization.

2.3 Examination and Summary

To establish our final recommendation system, I will aim to generate a large number of different recommendation algorithms and then build an ensemble of these models. I'll create a simple recommender that draws from all of the movies in the dataset, but in practice, I'll only ever use the smaller one. I'm ready to begin laying the groundwork for my recommendation system now.

2.3.1 Simple Recommenders

Provide each user with suggestions based on the popularity and/or genre of movies. This method is based on the premise that more successful and highly acclaimed films have a greater chance of being appreciated by the general viewer. Examples include the IMDB Top 250. Users of the Simple Recommender get suggestions for movies based on popularity and genre. The assumption behind this suggestion is that the average moviegoer is more likely to like a film that has received

widespread praise and critical acclaim. To wit, this approach does not tailor its recommendations to each user.
There isn't much complexity involved in putting this idea into practice. Sort our movies by audience approval and uncover the best in our collection. To further narrow down the results, we can also choose a genre to get recommendations for films that fall within that category.

I will make my chart using IMDB's weighted rating method. Weighted Rating (WR) is

$$\left(\frac{v}{v+m} . R\right) + \left(\frac{m}{v+m} . C\right)$$

where,
1. The total number of votes for the movie is denoted by v.
2. To be included in the graph, m votes are required as a bare minimum.
3. An R is the typical recommendation for this movie.
4. C represents the typical response to the whole report.

2.3.2 Content-based recommenders

Based on a given item, recommend comparable things. This algorithm makes these suggestions using item information, such as genre, director, description, actors, etc., for movies. These recommender systems are predicated on the central idea that if a person like one thing, he or she will also love something similar. And to do so, it will use the user's previous item information. An excellent example would be YouTube, which offers content you may like to watch depending on your viewing history.
Our previously constructed recommendation suffers from severe limitations. To begin, it doesn't take into account individual tastes while giving recommendations. In our Top 15 list, most of the films probably wouldn't appeal to someone who prefers romantic flicks over action pictures. Even if he checked out the genre-specific charts, he wouldn't get the most helpful suggestions. If this person is like the rest of us, they must be obsessed with Shahrukh Khan and Karan Johar. These recommendations wouldn't even be at the top of the romance chart if the person had access to them.

I want to develop an engine that computes similarity between movies based on specific criteria and then recommends movies that are most similar to a user's previously loved movie in order to provide more tailored suggestions. Content-based filtering is another name for this method since we will be utilizing information about movies (metadata) to create this system.
Two content-based recommender systems will be created by me using:

- Movie Synopses and Slogans
- Producers, Directors, Actors, and Roles in the Film

2.3.3 Collaborative filtering

The goal of these systems is to anticipate a user's rating or preference for an item based on the ratings and preferences of other users. Unlike their content-based equivalents, collaborative filters do not need item metadata.
However, there are significant restrictions on what we can do with our content-based engine. However, it can only recommend films that are similar to the ones you've already seen. In other words, it can't learn preferences and make suggestions based on them across different musical styles. In addition, our engine is impersonal since it does not account for each individual user's preferences and prejudices. No matter who uses our recommendation engine to ask about a certain film, they will all get the same set of suggestions.
Since this is a guide for moviegoers, we'll be using a method called "collaborative filtering" to narrow down the options. Collaborative filtering operates on the premise that I can infer how much I will like using a service or product by looking at the reactions of others who have my demographic characteristics and who have already used but not yet tried that service or product.
Collaborative filtering into two categories:

- **User-based Filtering:** Filtering based on the preferences of individual users yields results in which goods are suggested to shoppers. Let's pretend that Alice and Bob have a passion for reading (that is, they largely like and dislike the same books). Let's pretend for a moment that a brand new book has just hit the shelves and that Alice has read it and really enjoyed it. Since it seems probable that Bob will like it as well, the system has selected this book as one to suggest to Bob.

- **Item-based Filtering:** Item-based for the most part, these filtering algorithms function similarly to the content recommendation engine you developed. These algorithms find things that are comparable to those that have been rated in the past. If Alice, Bob, and Eve all rated The Lord of the Rings and The Hobbit with the maximum five stars, the algorithm would classify them as being very comparable. This is why the algorithm suggests The Hobbit to customers who have purchased The Lord of the Rings.

2.3.4 Hybrid Recommender

I created a recommendation engine that suggested films to a user based on the system's predicted ratings for that user, drawing on ideas from content and collaborative filtering. The system is put into place with a plan that uses both collaborative filtering and context-based filtering. This method gets over the limitations of the component algorithms and boosts the overall system efficiency.

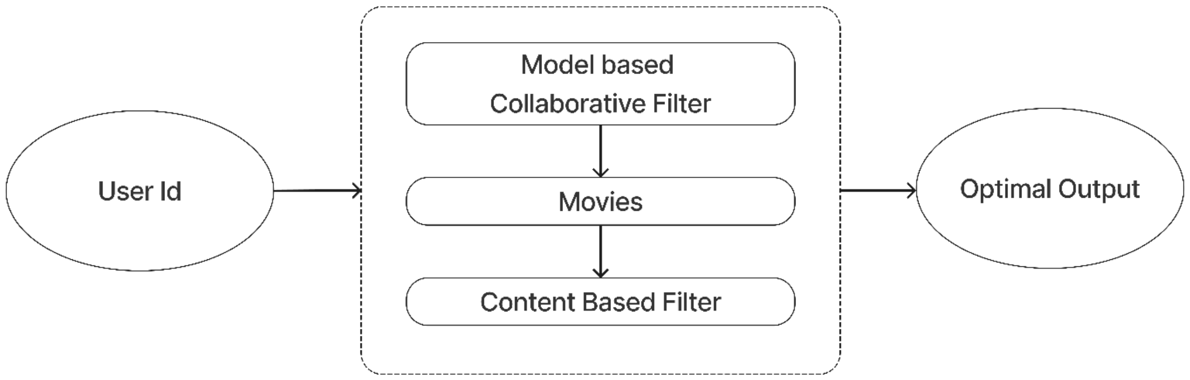


Fig 2.2: Architecture for hybrid approach

2.4 Scope of the problem

In order to combat the issue of information overload [14], recommender systems filter out the most relevant pieces of information from a sea of dynamically created data based on a user's stated or inferred preferences, interests, or past behavior with respect to a certain item. Based on a person's characteristics, a recommender system may determine whether or not that user would like a certain product. Analyze user preferences for genres and movies, and provide some basic recommendations based on those preferences. Rank actors, movies, and genres in order of popularity and profit.

Multiple film suggestions are provided to users by this method. The system's foundations are collaborative work, content-based methods, and a hybrid of the two. This study will provide more detailed results comparing systems that use a collaborative approach, a content-based approach, and a hybrid method that combines the two.

Content-based filtering aims to provide suggestions to the user based on their interests and previous actions. The algorithm uses the user's previous activities to extract patterns from the World Wide Web, and it then makes predictions about the user's interest in various news topics to suggest to them in the future. Since the model only needs information about the current user to make suggestions, it may ignore data about other users. Scaling to a big number of users is simplified in this way. The program is able to learn a user's tastes and then recommend things that are a good fit for that individual but would only interest a tiny subset of the population.

Sometimes called social filtering, collaborative filtering is a method of determining which information is most relevant to a given situation. In collaborative filtering, algorithms are used to sift through user ratings and comments to find those most likely to like a given service's suggestions. Individuals' social media feeds and advertisements are also curated through collaborative filtering. By far the most popular and well-proven approach to making suggestions is via the use of collaborative recommender systems, in which users make suggestions for one another. Collaborative filtering may either be done between users (user-to-user filtering) or between items (item-to-item filtering). When compared to user-collaborative filtering, content-based filtering performs better. The commonalities between items make more sense and are more similar than those between users.

The system is implemented using a method that combines context-based filtering with collaborative filtering. This method gets over the limitations of the component algorithms and boosts the overall system efficiency.

Users may browse through movie suggestions made by this system. Because it takes a collaborative rather than a content-based approach, this system will provide more transparent results than competing systems. Limiting themselves to human input, content-based recommendation algorithms don't make any out-of-the-box suggestions. These systems rely on the opinions of individual users, restricting your freedom of inquiry. In addition to evaluating movies, users may also get suggestions for similar films based on the ratings of others. The biggest issue in creating any system is making sure the target audience is happy. We had some difficulties as well while creating our system. Scalability and real-world feedback and verification of usage based on actual implementation One of the most popular techniques for predicting ratings inside an RS is collaborative filtering (CF). CF-based recommendation systems require greater processing power and storage space as users and things both increase in number. Scalability issues with CF algorithms might be a significant issue for a recommendation system. Using publicly accessible movie datasets, the majority of existing movie recommendation systems have worked to improve performance metrics like Root-Mean-Square Error (RMSE).

2.5 Difficulties

The recommendation system faces a number of difficulties. For the system to find a match, there must be sufficient users. For instance, we compare them with the pool of accessible individuals or products if we wish to find similar users or items. It sounds like Cold Start Problem. There are extremely few users or ratings in the matrix. Because the majority of users do not evaluate the things, it is quite difficult to discover users who have given the same items the same rating. Finding a group of users who rate the things thus becomes difficult. Collaborative filtering uses a significant amount of data to improve reliability, which calls for more resources. Due to the exponential growth of information, processing is becoming more and more expensive and inaccurate [10].

Machine learning-based movie recommendation systems might run into a number of difficulties, such as the following:

- **Data sparsity:** Due to the fact that many users may not have rated a substantial number of films, it may be challenging to properly anticipate the preferences that they have.
- **Cold start problem:** When making suggestions for a new user who has not yet reviewed any movies, it might be challenging to do so since there is no data on which to base the recommendations.
- **Scalability:** The computational cost of generating suggestions rises in proportion to the size of the user and movie databases since there are more potential combinations of user

and movie pairings. Scalability refers to the ability of a system to accommodate a growing number of users and movies.

- **Personalization:** Recommendation systems should be able to capture the unique preferences of each user, rather than making recommendations based on the average preferences of all users. This is preferable to the alternative, which is to make recommendations based on the average preferences of all users.
- **Diversity:** Recommendation systems should be able to deliver a varied selection of suggestions rather than only proposing the most popular things or limiting themselves to just recommending the most popular items.
- **Privacy:** When collecting and using data for recommendation systems, users' privacy must be kept safe as much as possible at all times.
- **Changing tastes:** Since people's interests tend to change over time, it might be hard to keep the suggestions up to date to reflect how the users' tastes are changing.

CHAPTER 3
The Research Methodology

3.1 Introduction

Research methodologies are studied in research methodology. It is a particular approach or technique applied to the discovery, collection, processing, and analysis of data pertaining to a topic. The technique part of a research paper gives the reader the chance to examine a study's overall validity and dependability. the methods or particular processes that aid in the selection, processing, and analysis of information. The research approach offers the study credibility and yields reliable scientific results. It also gives us a thorough plan that aids in keeping researchers on the course, facilitating a simple, efficient, and manageable approach. In order to help users and reduce the effects of information overload, recommender systems use knowledge discovery and statistical methods to provide product recommendations. Now, it is very important to figure out how to measure the effectiveness of recommender systems so that the best learning algorithms can be used on them. Popularity-based recommenders, which are a type of non-personalized recommendation system, usually tell people about the most popular things, like the newest movies, the best-selling books, and the most-bought goods. Collaborative filtering is the foundational form of the RS model, and it is predicated on the observation that consumers tend to prefer items that are similar to those they already like and to those that others with similar interests enjoy. In an effort to predict the user's preferences, content-based filtering relies on the user's past actions. The suggestions made by content-based filtering are based on the user's profile being compared to the keywords and characteristics associated with objects in the database. In this paper, I predict the movie genre based on the movie synopsis of the film by utilizing supervised machine learning. Here I used Content-based filtering, Collaborative-based filtering and hybrid approach for recommend movie based on user action, user rating, movie popularity, movie metadata, movie description. I used a Google Collaborate notebook for execution. Since the data we are gathering is in raw format, preprocessing is required to improve the accuracy of our classification model.

3.2 Data collection procedure

Data collection is gathered and analyzed through the process of data collection from diverse sources. In order to answer research questions, put forward hypotheses, and assess results, one must engage in data collecting, which is the systematic gathering and measurement of information on relevant variables. The gathered information is used as a foundation for dependability estimates. Therefore, a reliable dependability estimate relies heavily on a well-executed data collecting technique. Predictions are only as good as the information used to make them. As a result, maintaining high standards for data collecting is crucial. Data analysis relies on having high-quality input, thus it's crucial that the data obtained be of the best possible standard. When trying to learn more about a subject, a field of study, a research topic, or even a person, data is crucial. Because of this, it is seen as an important part of the many things that make up our modern world. In the current day, data may be used for many different purposes. If you want to acquire insights, make predictions, and run your operations in a manner that provides considerable value, data collection is a component you should never turn off, regardless of whether you are exploring digital transformation or not. I used the movies metadata, movies keywords, movies rating datasets for this paper that I downloaded from the internet.

Table 3.1: Collected Movies Languages and their count from movie dataset

Language	Count
en	32269
fr	2438
it	1529
ja	1350
de	1080
es	994
ru	826
hi	508
Ko	444
Zh	409

Table 3.2: Collected Movies Genres from movie dataset

Movie Genres	
1	[Animation, Comedy, Family]
2	[Adventure, Fantasy, Family]
3	[Romance, Comedy]
4	[Comedy, Drama, Romance]
5	[Comedy]
6	[Action, Crime, Drama, Thriller]
7	[Comedy, Romance]
8	[Action, Adventure, Drama, Family]
9	[Action, Adventure, Thriller]
10	[Adventure, Action, Thriller]

Table 3.3: Collected Movies title and popularity from movie dataset

Movie Title	Popularity
Toy Story	21.94694
Jumanji	17.01554
Grumpier Old Men	11.7129
Waiting to Exhale	3.859495
Father of the Bride Part II	8.387519
Heat	17.92493
Sabrina	6.677277
Tom and Huck	2.561161
Sudden Death	5.23158
GoldenEye	14.68604

3.3 Data evaluation

Evaluation of data is the process of determining the quality and usefulness of data. It is a crucial phase in the data analysis process that entails evaluating the data to find any faults or concerns that may influence its correctness or use. Evaluation of data may be undertaken at many stages of the data analysis process, including the data collection, cleaning, and transformation phases. It may use several strategies, including displaying the data, examining it for flaws or inconsistencies, and comparing it to other sources or standards.

The objective of data assessment is to guarantee that the data are trustworthy and appropriate for the intended use. It is a crucial phase in the data analysis process since it helps discover any issues with the data that must be resolved before the data can be utilized for analysis or decision-making.

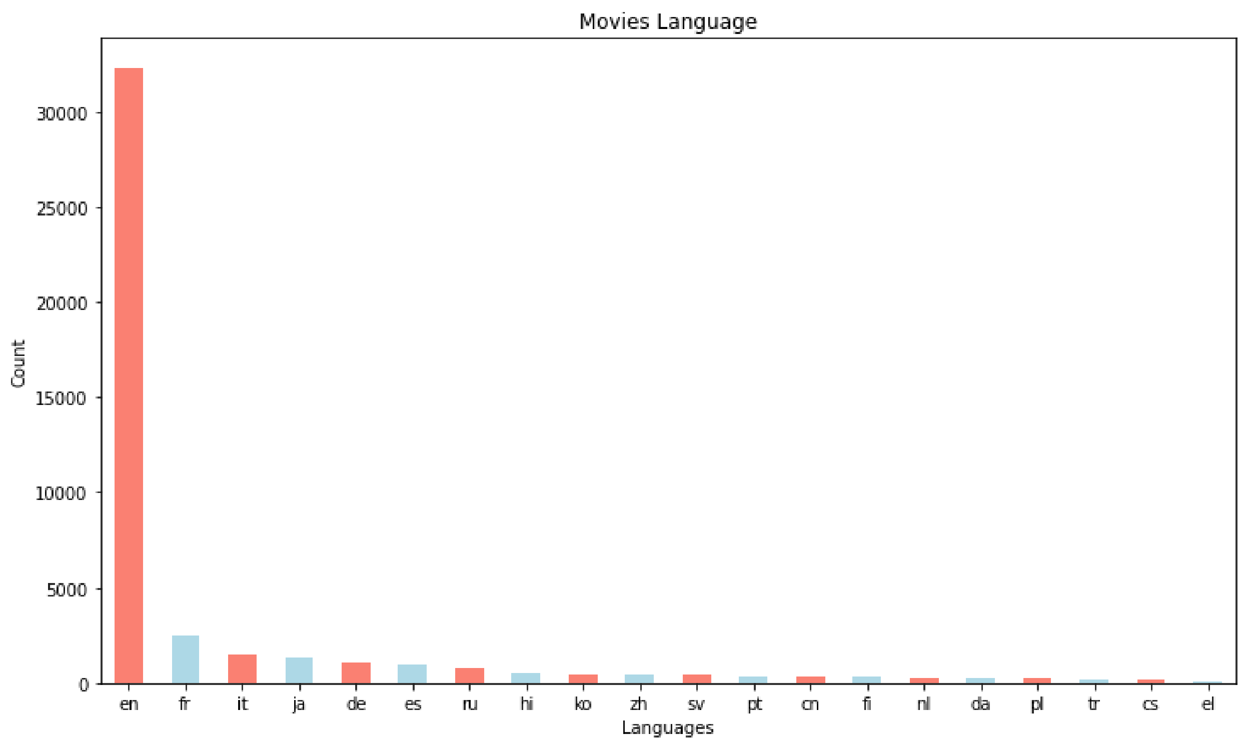


Fig 3.1: Movies top 20 languages count with languages name showing through Bar chart from movies dataset

Movie recommendation systems can suggest movies in any language. This depends on the language of the movies in the dataset and the language of the users for whom the suggestions are made. The language of the films in the dataset will depend on which films are included and which languages they are accessible in. It may be necessary to use machine translation or language identification to make sure that the suggestions are right for the user's preferred language.

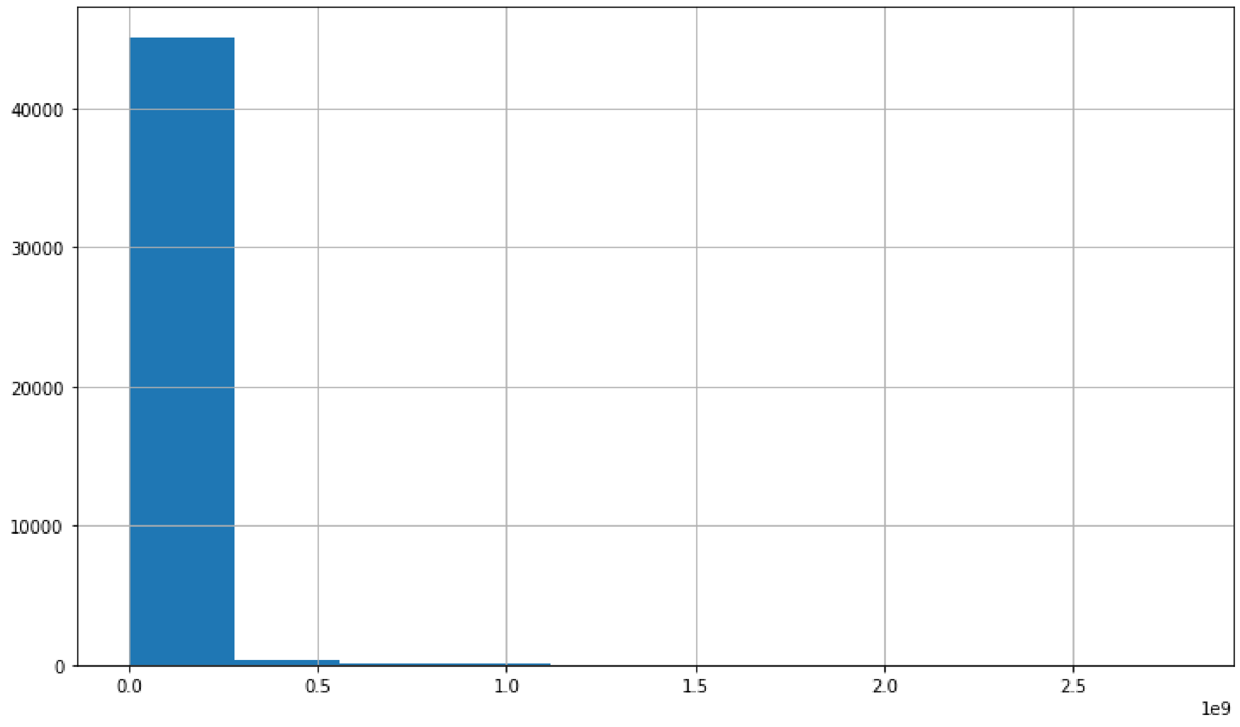


Fig 3.2: Collected Movies Revenue data showing through histogram chart from movie dataset

This histogram displays movie revenue from the movies dataset. Visualizing movie revenue data and analyzing the revenue distribution within a certain dataset This data may indicate that the bulk of films in the dataset have relatively modest revenues, while a smaller number of films have much greater revenues. This might suggest that the distribution of movie revenues has a long tail, with a small number of blockbuster movies accounting for a large amount of the overall income.

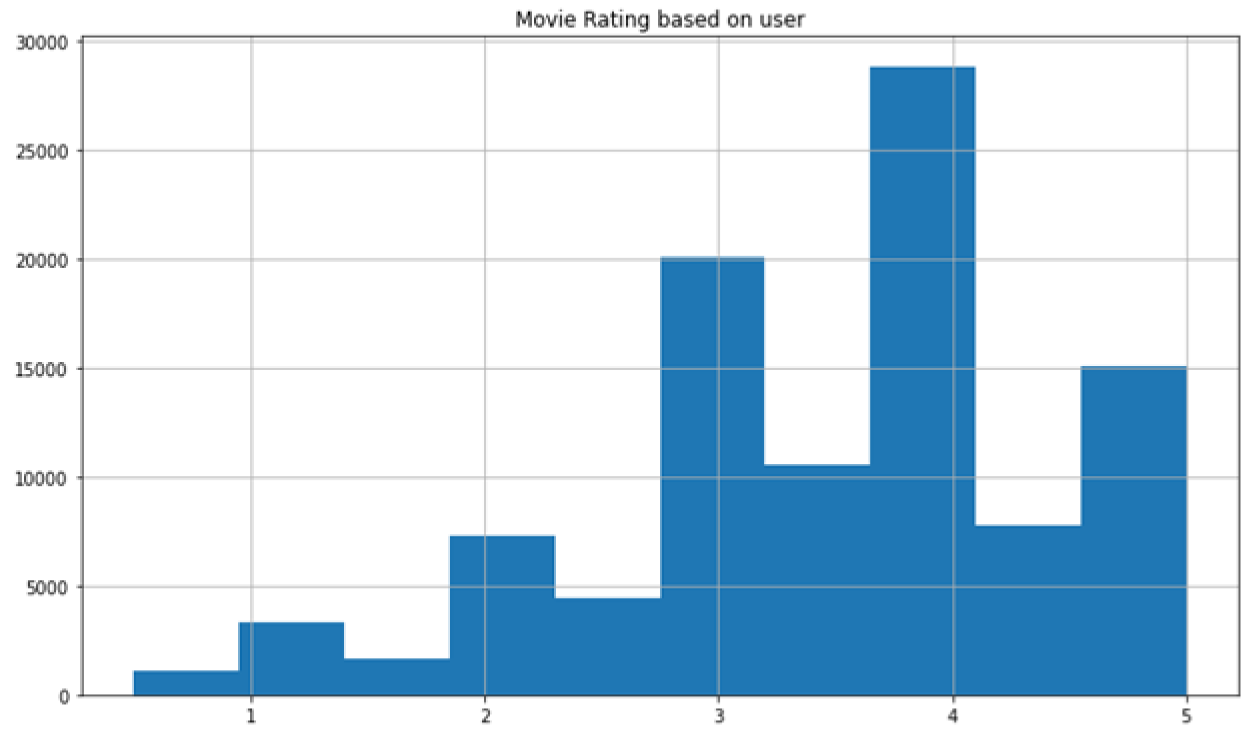


Fig 3.3: Collected all movies rating data showing through Histogram chart from movie rating dataset

In this histogram, I display movie ratings from the movies dataset. The bulk of movies within the dataset may have relatively high ratings, with a smaller number of movies receiving lower ratings. This may imply that the dataset contains a large number of well-liked, popular films.

In a system for recommending movies to users, movie ratings play an essential role in choosing which films are recommended. Ratings play an essential part in movie recommendation systems since they supply the input data necessary to make suggestions and represent the tastes of the

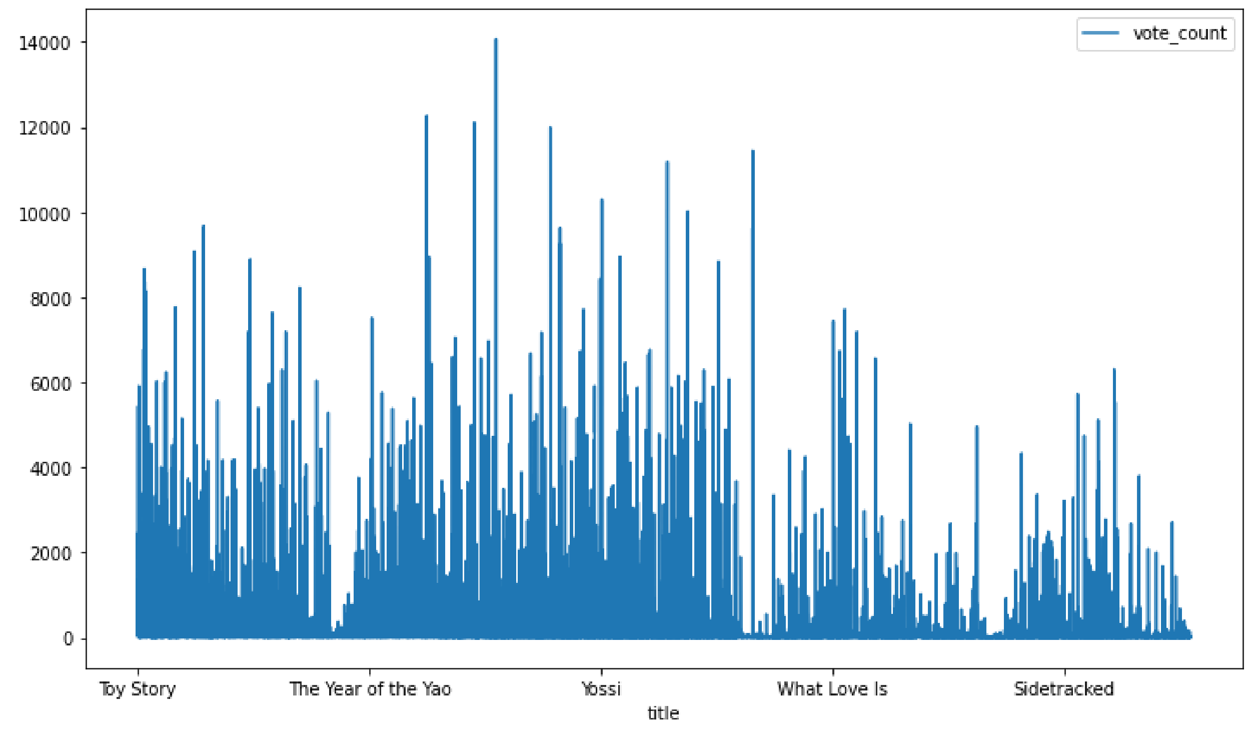


Fig 3.4: Collected Movies title and Movies user vote count data showing through Line chart from movie dataset

In this picture I show movies title and movies user vote count data through line chart from movie dataset. Information about movie user vote counts may be utilized to produce suggestions for users. The vote count data represents a movie's popularity among users and may be used to determine the most popular movies or to adapt suggestions to the tastes of specific users. By evaluating the vote count data for groups of users, the recommendation engine is able to find patterns in the data and provide suggestions based on the preferences of people with similar tastes. This method, also known as collaborative filtering, may facilitate the creation of more tailored suggestions for each user. This is a simple and successful method for recommending films that are likely to appeal to a large audience.

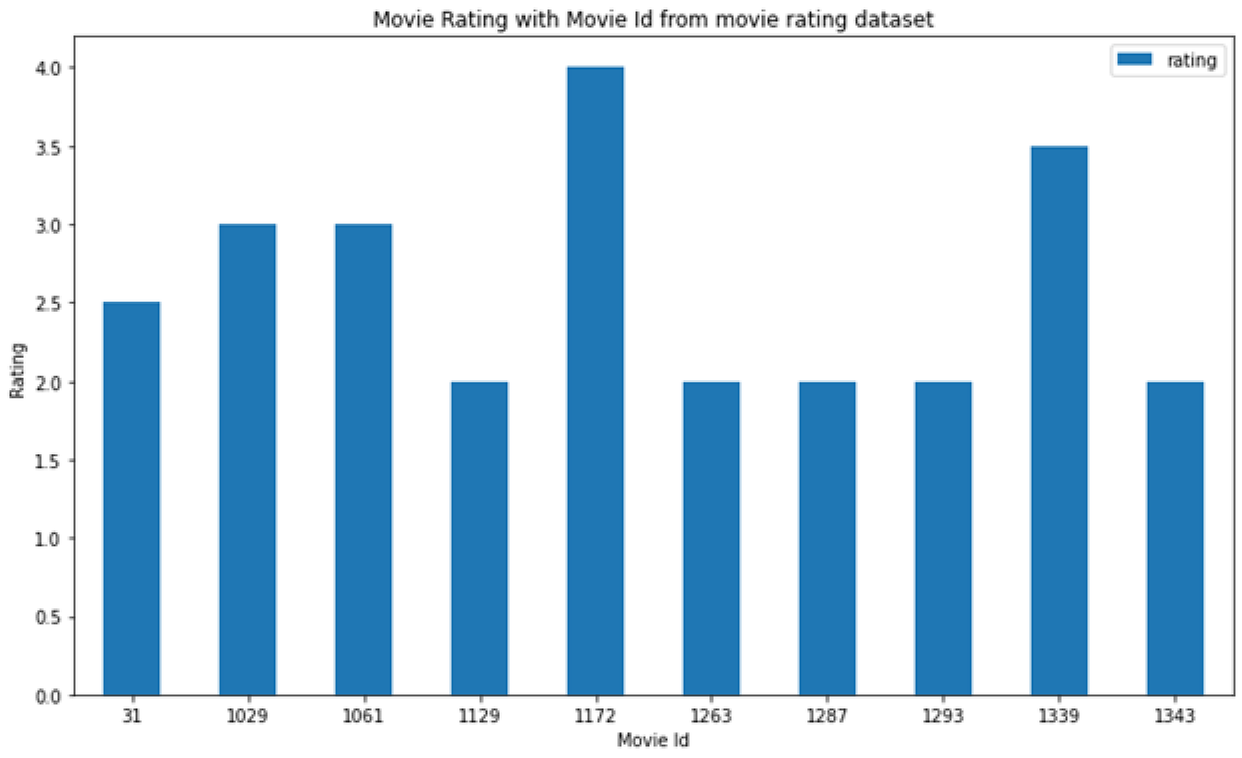


Fig 3.5: Collected first ten movies rating with movie id data showing through Bar chart from movie rating dataset

In this picture I show movies Id and movies user rating data through bar chart from movie dataset. In a movie recommendation system, rating data from users may be utilized to produce suggestions for consumers. The rating data shows the opinion of individual users on a film and may be used to identify the highest-rated films or to personalize recommendations to the tastes of specific users. By evaluating rating data for individual users, the recommendation system may determine the sorts of films that a user tends to like and propose comparable films to that user. By looking at rating data for groups of users, the recommendation system may be able to find patterns and make suggestions based on what people with similar tastes like.

A table displaying the correlation coefficients between several variables is called a correlation matrix. The correlation between two variables is shown in each column of the table. The correlation coefficient, which ranges from -1 to 1, is a metric for determining the direction and strength of a linear connection between two variables. The formula for Pearson's correlation coefficient is:

$$r = covariance(x, y) / (std(x) * std(y))$$

Where:

- x and y are the two variables for which the correlation coefficient is being calculated
- covariance(x, y) is the measure of the degree to which two variables change together
- std(x) and std(y) are the standard deviations of x and y, respectively.

Table 3.4: Correlation Matrix from movie rating dataset

	UserId	**MovieId**	**Rating**	**Timestamp**
UserId	1	0.007126	0.010467	-0.035072
MovieId	0.007126	1	-0.028894	0.514742
Rating	0.010467	-0.028894	1	-0.038996
Timestamp	-0.035072	0.514742	-0.038996	1

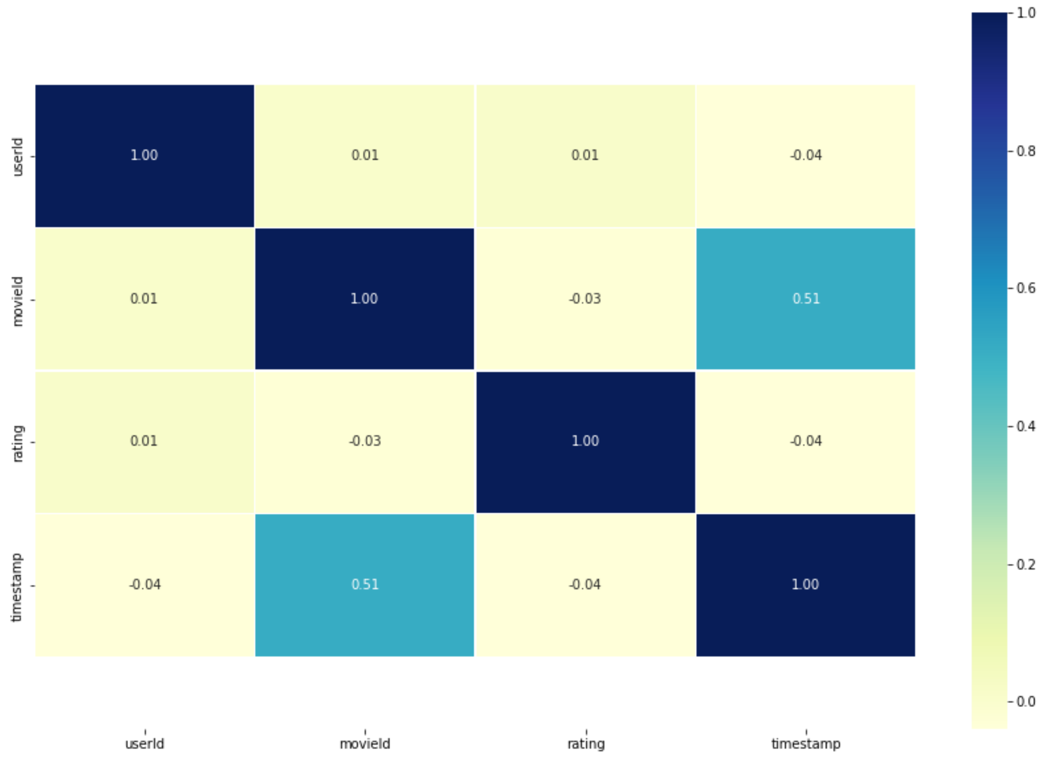


Fig 3.6: Correlation matrix from movie rating dataset

3.4Proposed Methodology

The suggested methodology is a strategy or framework that defines the technique that will be used to attain a certain objective or resolve a specific issue. Typically, the suggested approach comprises a description of the actions to be taken, the tools and methods to be used, and the needed resources. The suggested methodology is an essential component of any research effort since it specifies the strategy for collecting and analyzing data to answer the research question. In a business environment, the suggested methodology may define the technique that will be used to produce a new product or service or to resolve a particular business challenge.

To be successful, the suggested approach must be well organized and well-reasoned, and it must account for any possible constraints or obstacles that may develop. It should also be targeted at the particular issue or objective being addressed and intended to provide accurate and trustworthy findings.

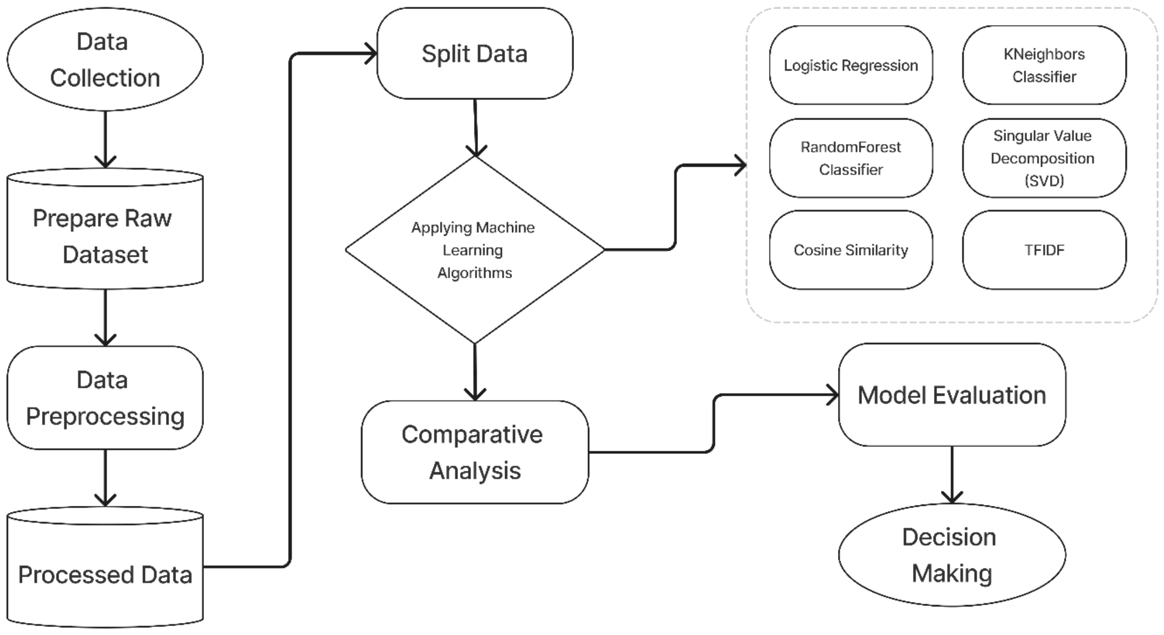


Fig 3.7: Ten movies with their rating and Id showing through Bar chart from movie rating dataset

A suggested approach for a movie recommendation system might contain the following steps:

1. **Define the problem:** Clearly defining the issue that the movie recommendation system is designed to tackle is the first stage in its development. This could mean figuring out who the system is meant for, what kinds of movies it will suggest, and what the exact goals of the system are.
2. **Gather data:** To create a recommendation engine, I must collect a dataset of movie and rating data. This information might be gathered from several sources, including online movie databases, box office records, and user ratings on social media sites. I gathered data from the Internet for this movie recommendation system.
3. **Preprocess the data:** Once I have acquired the data, I will need to preprocess it so that the recommendation system can use it. This could mean cleaning and organizing the data, getting rid of outliers or wrong data points, and pulling out important qualities or traits from the data.
4. **Select a recommendation algorithm:** Various techniques, such as collaborative filtering, content-based filtering, and matrix factorization, may be used to construct a recommendation system. I must choose an algorithm that is suitable for our particular issue and data collection. I use Logistic Regression, K Neighbors Classifier, Random Forest Classifier, Singular Value Decomposition (SVD), Cosine Similarity, and TFIDF algorithms for this recommendation system.
    - **Logistic Regression:** In logistic regression, a model is created to estimate the likelihood of an occurrence based on one or more independent variables (also known as predictor variables). A training dataset, including observations of the independent variables and their corresponding values for the dependent variable, is used to create the model. The model is then used to forecast the likelihood that the event will occur based on fresh observations of the independent variables.
    - **K-Nearest Neighbors:** The classification method K-Nearest Neighbors (KNN) is used for supervised learning. It is founded on the notion that an item is classified according to the qualities of its closest neighbors. KNN is a simple and effective technique that is extensively used in several applications, such as image and text categorization, anomaly detection, and recommendation systems. It is easy to use and can be used with

a number of different distance metrics, making it a flexible and reliable way to put things into groups.

- **Random Forest Classifier:** The classification algorithm Random Forest is used for supervised learning. It is an ensemble approach, which aggregates the predictions of numerous decision trees to produce a single forecast. Random Forest is a potent and widely-used classification algorithm that is renowned for its precision and resilience. It is resistant to overfitting and works well on several datasets, particularly those with high dimensionality or intricate feature interactions. It is also easy to use and can be used to solve a wide range of categorization problems, which makes it a popular choice for people who work with machine learning.
- **Singular Value Decomposition (SVD):** SVD is a mathematical method for dividing a matrix into its component elements. It is a potent instrument utilized in several domains, such as machine learning, natural language processing, and image processing. SVD is used to decrease the dimensionality of a matrix, approximate a matrix with a matrix of lower rank, and identify a matrix's major components. In collaborative filtering methods for recommendation systems and natural language processing, it is used to determine the underlying meaning of words in a text.
- **Cosine similarity:** By calculating the cosine of the angle between two vectors in an inner product space, we may assess how well they align with one another. It is calculated by dividing the dot product of the vectors by the product of their magnitudes. Cosine similarity is often used in information retrieval and natural language processing to figure out how similar two documents or passages of text are.
- **TFIDF:** The TF-IDF formula is used to calculate the inverse document frequency of a document or collection of documents to determine the relative importance of individual phrases. It is often used in information retrieval and natural language processing applications to identify the significance of a word in a document relative to a corpus of documents. TF-IDF is used to provide more weight to essential words and less weight to common terms, so that the value of a word in a text may be portrayed more accurately.

5. **Train and evaluate the model:** After selecting a method and preprocessing the data, I must train the model using the data and assess its performance. This could mean separating the data

into sets for training and testing, fitting the model to the training data, and then using the model to predict the test data.

6. **Deploy the system:** I may implement the recommendation system and make it accessible to users at the final phase. This may require integrating the system into an existing application or developing a recommendation system on its own.

CHAPTER 4
Experimental Findings and Review

4.1 Introduction

In a research endeavor, the outcomes of experiments and reviews are the conclusions and findings obtained from the data gathered and processed as part of the study. These data are often provided as tables, graphs, and text and serve to answer the research question and support the study's major conclusions. The outcomes of experiments and reviews should be presented clearly and simply, backed by the data gathered for the investigation. This could mean looking at the statistical significance of the data, taking into account the limitations of the study, and figuring out if there are any possible implications for future research or real-world uses.

The recommendation mechanism is a system used to filter information in the system that forecasts an item's rating. The recommended system creates suggestions for individual users based on their prior purchases and searches, as well as the behavior of other users. Recommended systems are software applications and approaches that recommend useful objects to a user. The system matches users with goods in which they are interested. It will facilitate and enhance the quality of choices made by internet shoppers.

When a user hasn't rated many things previously, collaborative filtering works better than the content-based method. Cosine similarity, which calculates distance between individuals based on their profiles, is used to determine how similar the users are to one another. Following the computation of user similarity, it is inevitable that certain films that the user hasn't seen but that comparable users have seen might end up being among those films. This similarity is attained by comparing the user's movie preferences and user ratings with those of all other users, and then sorting comparable individuals based on distances between 0 and 1.

As a hybrid recommendation component, it adds weights to the outcomes from both the content-based and collaborative filtering processes. The majority of the characteristics that vary depending on the user rating profile that are accessible are used to apply weights to the suggestion. The content-based technique will be given more weight if the user has less rating history. Conversely, if the user has enough rating history to identify comparable users, the collaborative filtering approach will be given more weight. The users will be given recommendations based on the expected top results when the findings are pooled and computed.

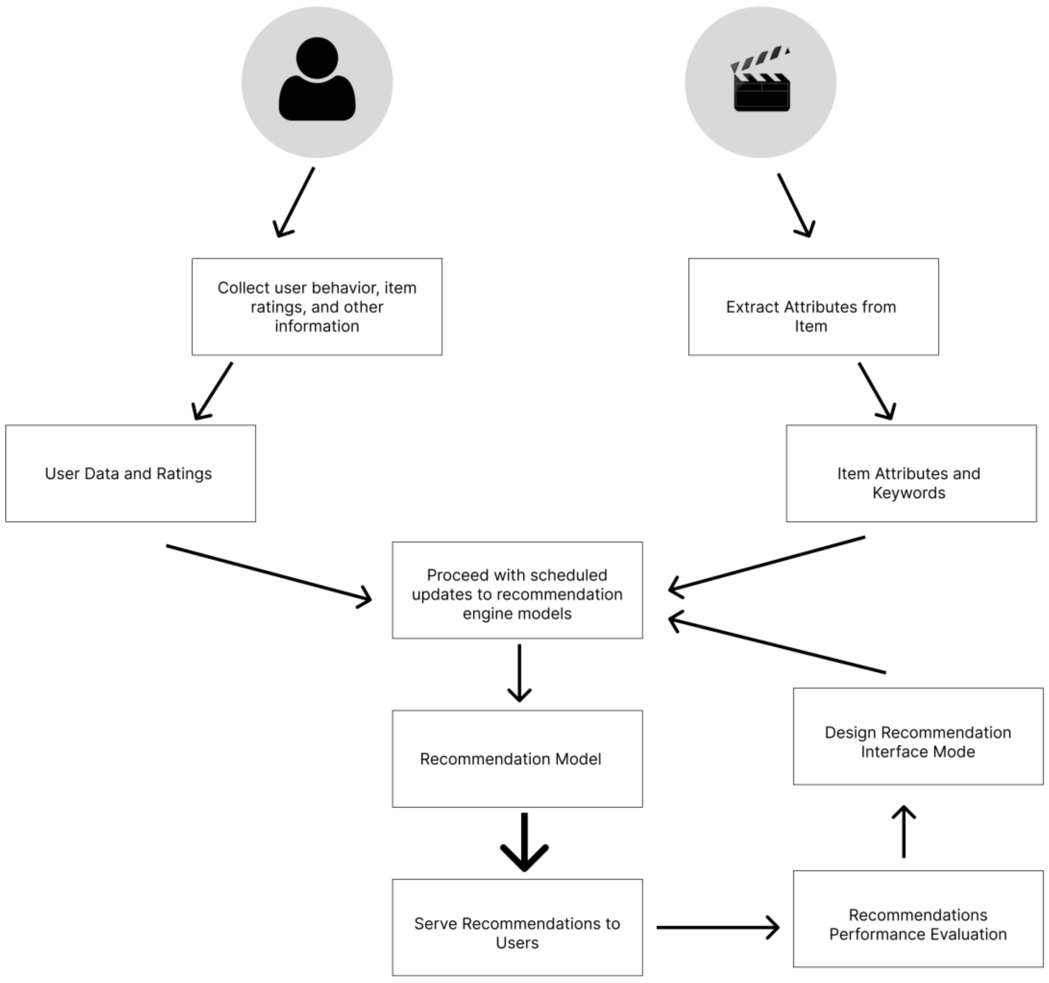

Collect user behavior, item ratings, and other information
Extract Attributes from Item
User Data and Ratings
Item Attributes and Keywords
Proceed with scheduled updates to recommendation engine models
Recommendation Model
Design Recommendation Interface Mode
Serve Recommendations to Users
Recommendations Performance Evaluation

Fig 4.1: Process of movie recommendation system

4.2 Analyzing the Results of Experiments

4.2.1 Simple Recommender

There are primarily two categories of recommender systems: customized and non-personalized. Both have their advantages and disadvantages. Users are given recommendations for the most popular things via non-personalized recommendation systems such as popularity-based recommenders [20]. These recommenders provide users with information about the top 10 movies, the best-selling books, and the most frequently bought products.

Every user of The Simple Recommender receives recommendations that are generic and based on the popularity of movies and (sometimes) the genre of such movies. The fundamental reasoning behind this recommender is based on the hypothesis that movies that have garnered widespread appreciation from both audiences and critics would have a greater chance of being enjoyed by the typical audience member. This approach does not provide suggestions that are specifically tailored for the end user.

The implementation of this approach is ridiculously easy to understand. It is enough for us to filter our movies according to their ratings and how popular they are, and then we can provide the best movies from our list. We may take it one step further by specifying a genre to get the top results for films in that category.

In accordance with the quantity of votes from frequent visitors, IMDb additionally modifies the weighted rating used for the IMDb Top 250. IMDb may also make further statistical corrections to lessen the impact of efforts to improve or harm a film's rating. IMDb doesn't share any information about the algorithms that are used to make vote-rigging less likely to happen. I also use a similar weighted rating for recommended movie systems in my study report.

$$\mathrm{WR} = \frac{v}{v+m} \cdot R + \frac{m}{v+m} \cdot C$$

- v stands for the quantity of movie votes.
- m stands for the number of votes a film must receive to make the Top Movies list.
- The R represents the film's mean rating.
- The overall report grade is a C.

The next step is to settle on a respectable figure for m, the bare minimum of votes required to make the graph. To set a limit, we'll choose the 95th percentile. Therefore, for a movie to be ranked, it must get more votes than 95% of the other movies in the top 100.

Table 4.1: Qualified movies dataset after calculating movies weighted rating

Title	**Year**	**Vote Count**	**Vote Average**	**Popularity**	**Genres**	**WR**
Inception	2010	14075	8	29.108149	Action, Thriller, Science Fiction, Mystery, Adventure	7.917588058
The Dark Knight	2008	12269	8	123.167259	Drama, Action, Crime, Thriller	7.905871458
Interstellar	2014	11187	8	32.213481	Adventure, Drama, Science Fiction	7.897107403
Fight Club	1999	9678	8	63.869599	Drama	7.881752881
The Lord of the Rings: The Fellowship of the Ring	2001	8892	8	32.070725	Adventure, Fantasy, Action	7.871786954
Pulp Fiction	1994	8670	8	140.950236	Thriller, Crime	7.868660493
The Shawshank Redemption	1994	8358	8	51.645403	Drama, Crime	7.863999674
The Lord of the Rings: The Return of the King	2003	8226	8	29.324358	Adventure, Fantasy, Action	7.861926689
Forrest Gump	1994	8147	8	48.307194	Comedy, Drama, Romance	7.860655533
The Lord of the Rings: The Two Towers	2002	7641	8	29.423537	Adventure, Fantasy, Action	7.851923855

In this table 4.1, after calculating weighted rating, I show 10 movies list with movie title, movie release year, movie vote count, movie average vote, movie popularity, movie genres and movie weighted rating count from our qualified movies data list.

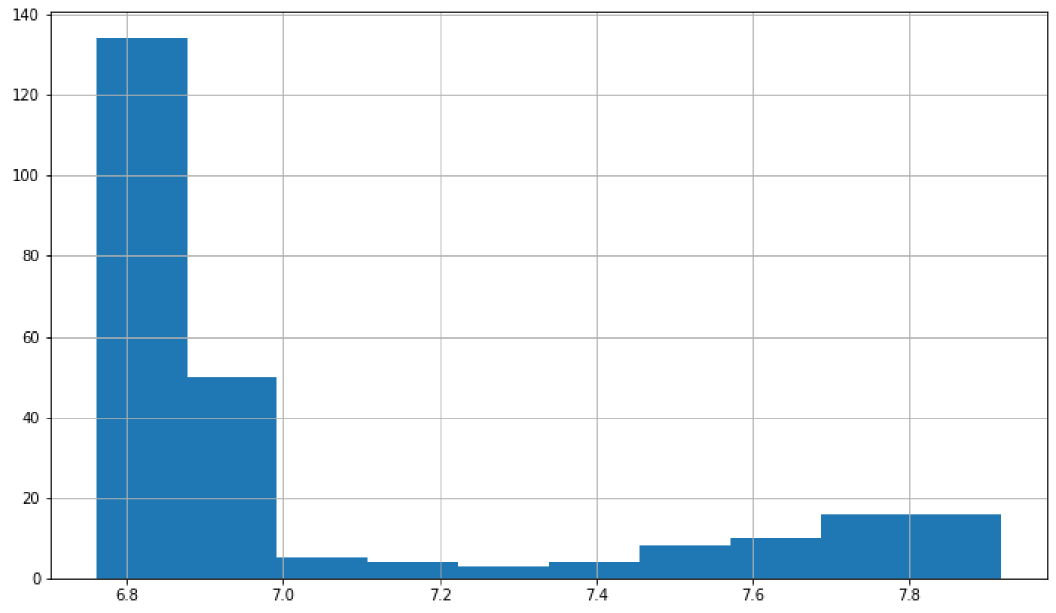


Fig 4.2: Calculating weighted rating matrix for movie data set showing through histogram chart

As a result, a movie needs at least 434 votes on TMDB to be eligible for the chart. The average movie rating on TMDB is 5.244 out of 10, which means that 2274 films are eligible for inclusion in our ranking. After calculating weighted rating, I show the qualified movie dataset using histogram chart in Fig 4.2.

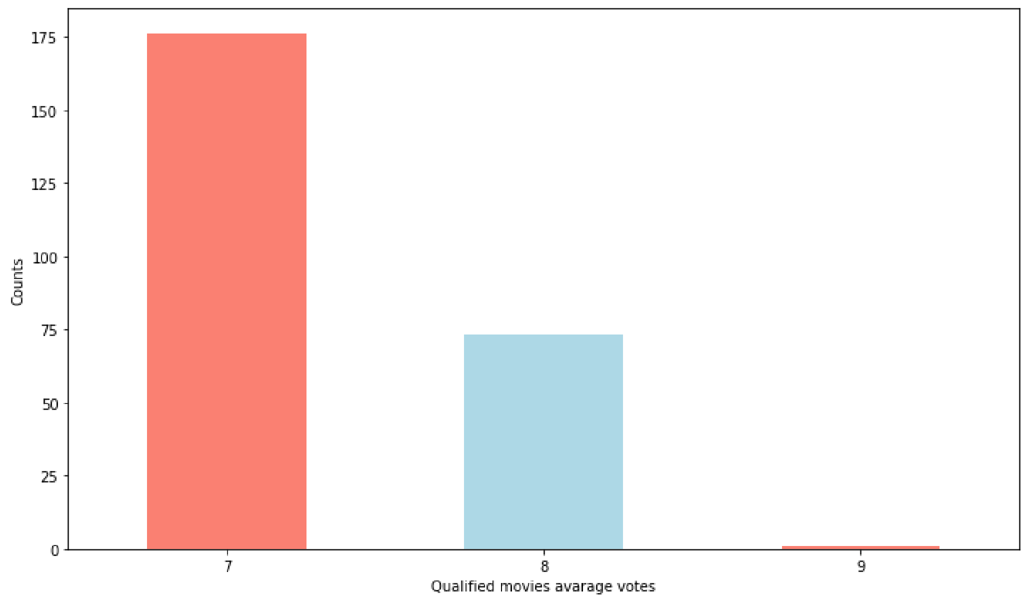


Fig 4.3: Qualified movies average votes showing through Bar chart from movie dataset

The average number of votes that movies have gotten from viewers is shown using an average votes bar chart once weighted rating calculations have been completed. This graph (Fig 4.3) is helpful for comparing the popularity of various movies and seeing trends and patterns in the data.

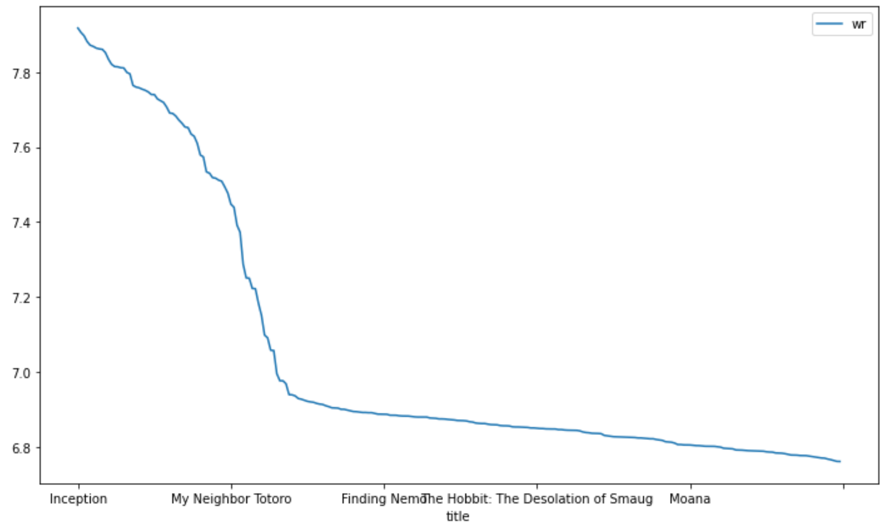


Fig 4.4: After calculating weighted rating matrix movie title and wr showing through Line chart

Table 4.2: Top 10 Animation movies based on the most popular movie genres

Title	**Year**	**Vote Count**	**Vote average**	**Popularity**	**WR**
The Lion King	1994	5520	8	21.60576	7.909339
Spirited Away	2001	3968	8	41.04887	7.875933
Howl's Moving Castle	2004	2049	8	16.13605	7.772103
Princess Mononoke	1997	2041	8	17.16673	7.771305
My Neighbor Totoro	1988	1730	8	13.5073	7.735274
Your Name.	2016	1030	8	34.46125	7.58982
Grave of the Fireflies	1988	974	8	0.010902	7.570962
Paperman	2012	734	8	7.198633	7.465676
Piper	2016	487	8	11.24316	7.285132
Wolf Children	2012	483	8	10.2495	7.281198

I include the top 10 animated films from our certified movies dataset in table 4.2. I'll do this by lowering our default conditions from 95 to the 85th percentile. A common film genre, animation movies contain animated characters and backdrops and often use speech, music, and sound effects to create tales. Animation films may have a broad range of subjects and plots, and they can appeal to a wide range of people.

"The Lion King" is a children's animated film. Typically geared at younger viewers, this kind of animation includes kid-friendly characters and plots. This movie release year 1994, 5520 vote, 8 average vote and popularity 21.6 and weighted rating 7.9.

In 2012, the romantic comedy short "Paperman" came out in black and white on computers. It was made by Walt Disney Animation Studios. A guy uses a squadron of paper airplanes to try to get a girl's attention after being rejected by her on the train ride to work. This movie 734 vote, 8 average vote and popularity 7.19 and weighted rating 7.46.

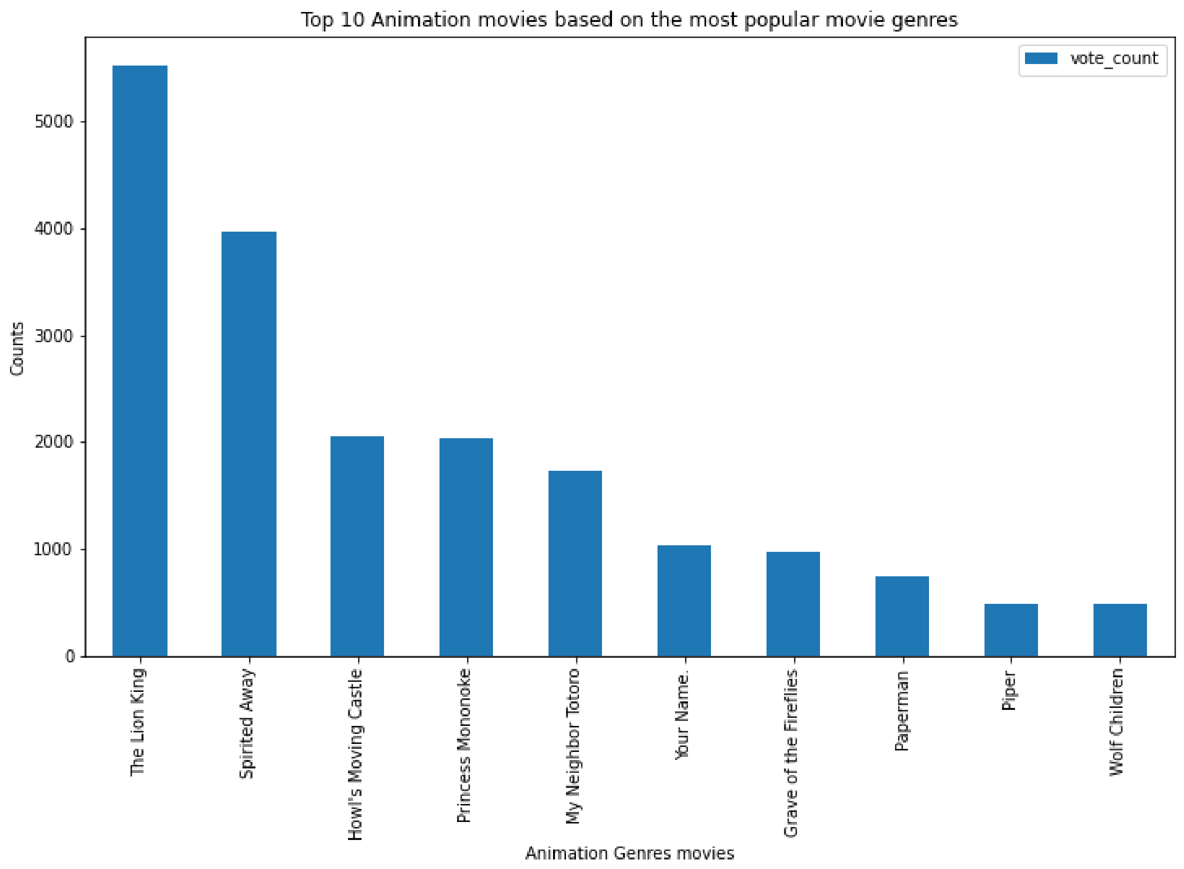


Fig 4.5: Top 10 Animation movies based on the most popular movie genres showing through Bar chart

In this bar chart (Fig 4.5) I show top 10 animation movies based on the most popular movie genres using movie title and movie vote count. Between this 10 movies "The Lion King" movie have 5520 vote, 7.9 weighted rating and "Wolf Children" movie have 483 vote, 7.28 weighted rating.
After calculating this weighted rating as simple recommendation system, we can easily recommend movie based their weighted rating. The highest weighted rating movies are top listed movie in this simple recommendation system and lowest weighted rating movies are bellow listed movie.

4.2.2 Content-based Recommendation

Content-based filtering will recommend more goods with similar features depending on the user's actions or input. To further explain how content-based filtering works, let's create our own version of the Google Play store. As an example of a feature matrix, the accompanying graphic shows a table with rows representing apps and columns representing features [21]. It is possible to include a wide variety of features, including categorization (into broad groups like "Educational," "Casual," and "Health"), the app's publisher, and many more. Assume for the sake of argument that this feature matrix is binary, with non-zero values indicating the existence of the feature in the app.
Content-based advice is a strategy often used by recommender systems. In general, the objective is to suggest products that the consumer will like just as much as the ones they have already shown some preference for. The basic purpose of a content-based recommender system is to identify how similar two objects are. The vector space model is the most well-known of the several approaches to modeling items. The TF-IDF model uses item-specific keywords to assign a value. Set k_i to be the ith keyword of item d_j and w_{ij} to be the weight of k_i for d_j, then you may specify the contents of d_j as:

$$Content(d_j) = \{w_{1j}, w_{2j}, ...\}$$

Using the user's past preferences as a guide, a content-based recommender system will make suggestions. A user's preferences may be predicted based on their past interactions with the system.

So, if we think of *ContentBasedProfile(u)* as u's personal preference vector, we may define it as follows:

$$ContentBasedProfile(u) = \frac{1}{|N(u)|} sum_{d \in N(u)} Content(d)$$

What was previously popular with user u is denoted by *N(u).* Given a user u and an item d, the degree to which u enjoys d is defined as the degree to which *ContentBasedProfile(u)* is comparable to *Content(d)*, where *Content(.)* and *ContentBasedProfile(.)* have already been computed for all users:

$$p(u, d) = sim(ContentBasedProfile(u), Content(d))$$

The recommender I created in the preceding section has several significant drawbacks. For starters, it makes the same suggestion to everyone, regardless of their own preferences. A person who enjoys romantic comedies or animated films (and despises action) would probably not like the majority of the films in our Top 10 Chart. If s/he went one step further and looked at our genre charts, the suggestions would no longer be the best.

Think of someone who enjoys "The Lion King," "Paperman," or "Toy Story," for instance. One conclusion we may draw is that the individual adores both John Lasseter and Tom Hanks. Even if s/he had access to the romance chart, they wouldn't be the top suggestions.

I'm going to create an engine that determines the similarity between movies based on certain criteria and proposes films that are most similar to a specific movie that a user loved in order to further personalize our suggestions. This method is sometimes called "content-based filtering" because the engine will be made from information about the movie.

I will make two content-based recommendations in the following:

- Synopsis and taglines for movies
- Cast, crew, keywords, and genre for movies

Movie Description Based Recommender

I start by attempting to create a recommender system utilizing taglines and movie descriptions. We must assess our machine's performance qualitatively since we lack a quantitative measure for doing so. For read the movie description I used TF-IDF, Cosine Similarity.

TF-IDF

The TF-IDF is calculated by multiplying the term frequency (TF) of a word by its inverse document frequency (IDF). The number of times a word occurs in a document is known as its term frequency, and the collection's inverse document frequency calculates the frequency of a word over all documents. Words that are frequent or unique to a single text will have a higher IDF value than words that are common to all publications. A document's or a group of documents' most important words may be found using TF-IDF, which is also often used to express a document's content in a vector space model. It is a popular method for finding the important words in a text and is used in both information retrieval and natural language processing jobs.

The relative frequency of the word *t* in document *d* is known as term frequency, or *tf(t, d)*.

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

where *ft,d* is the total number of times a word appears in a document, i.e., how many times term *t* appears in document *d*. Note that the document's total number of phrases serves as the denominator (counting each occurrence of the same term separately). Other definitions of the word "frequency" include []:

- the actual raw count: *tf(t,d) = ft,d*
- *tf(t,d)* = 1 if *t* occurs in *d* and 0 otherwise in terms of boolean "frequencies"
- Frequency scaled logarithmically: *tf(t,d) = log (1 + ft,d)*

The inverse document frequency shows how often a word appears in all texts, and hence how much information it carries. To get the inverse logarithmic scale, divide the total number of documents by the percentage of documents that include the phrase, and then take the logarithm of that quotient.

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

- *N* is the number of documents in the corpus as a whole. *N* = |*D*|

- The number of documents that include the phrase t $\mathrm{tf}(t,d) \neq 0)$ is given by the formula $|\{d \in D : t \in d\}|$ If the word is absent from the corpus, a division-by-zero will result. Therefore, changing the denominator to $1 + |\{d \in D : t \in d\}|$ is usual.

Cosine Similarity:

The purpose of content-based recommendation engines is to determine which films are most like one another. Text, video, and still images are all acceptable formats for the material. In this study, we use a feature vector to characterize each film. That being said, I will now explain the cosine similarity method.

The most common method for determining how similar two documents are is called "Cosine Similarity." The cosine of the angle between the feature vectors may be used to determine the degree of similarity between the characteristics.

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

The similarity scale goes from -1 to 1. With a value of -1, the two vectors point in opposing directions, whereas a value of 1 indicates that they point in the same direction. If the two vectors are completely unrelated, their cosine similarity will be 0. Weights must be positive for text matching to work, so our range must be between 0 and 1. An example of cosine similarity is shown below. [14]

Given two sentences:

- A: I like watching movie, but I don’t like watching drama.
- B: I don’t like watching movie and drama.

How do we quantify how closely the two statements correspond to one another? The underlying principle is that two sentences are more similar if they include more terms in common.

First step: Word segmentation.

- A: I / like / watching / movie, but / I / don’t / like / watching / drama.
- B: I / don’t / like / watching /movie / and / drama

Second step: List all the words.

- I, like, watching, TV, but, don’t, films, and

Third step: Word frequency calculation.

- A: I 2, like 2, watching 2, movie 1, but 1, don’t 1, drama 1, and 0.
- B: I 1, like 1, watching 1, movie 1, but 0, don’t 1, drama 1, and 1.

Forth step: Get word frequency vector.

- A: [2, 2, 2, 1, 1, 1, 1, 0]
- B: [1, 1, 1, 1, 0, 1, 1, 1]

The cosine of the two vectors may then be determined.

$$\cos(\theta) = \frac{2\times1+2\times1+2\times1+1\times1+1\times0+1\times1+1\times1+0\times1}{\sqrt{2^2+2^2+2^2+1^2+1^2+1^2+1^2+0^2}\times\sqrt{1^2+1^2+1^2+1^2+0^2+1^2+1^2+1^2}}$$

There is a high degree of similarity between the two statements (0.85).

Table 4.3: 10 most similar movies like "The Godfather" using pairwise cosine similarity matrix from movies dataset

Movie Title	
1	The Godfather: Part II
2	The Family
3	Made
4	Johnny Dangerously
5	Shanghai Triad
6	Fury
7	American Movie
8	The Godfather: Part III
9	8 Women
10	Summer of Sam

Table 4.4: 10 most similar movies like "Toy Story" using pairwise cosine similarity matrix from movies dataset

Movie Title	
1	Toy Story 2
2	Toy Story 3
3	The 40 Year Old Virgin
4	Man on the Moon
5	Factory Girl
6	What's Up, Tiger Lily?
7	Rebel Without a Cause
8	For Your Consideration
9	Rivers and Tides
10	Condorman

In the same way that we did with the Description Recommender, I first extract movie genres, directors, and keywords from our dataset before using a count vectorizer to generate our count matrix. The rest of the process is quite similar to the first: we return the films that are the most similar after calculating their cosine similarity.

Here is how I compile information on genres and credits:

- Remove any commas or spacing and change the case of every option to lowercase. As a result, our system will be able to tell the difference between Johnny Depp and Johnny Galecki.
- Mention the Director three times to make their role stand out more among the rest of the ensemble.

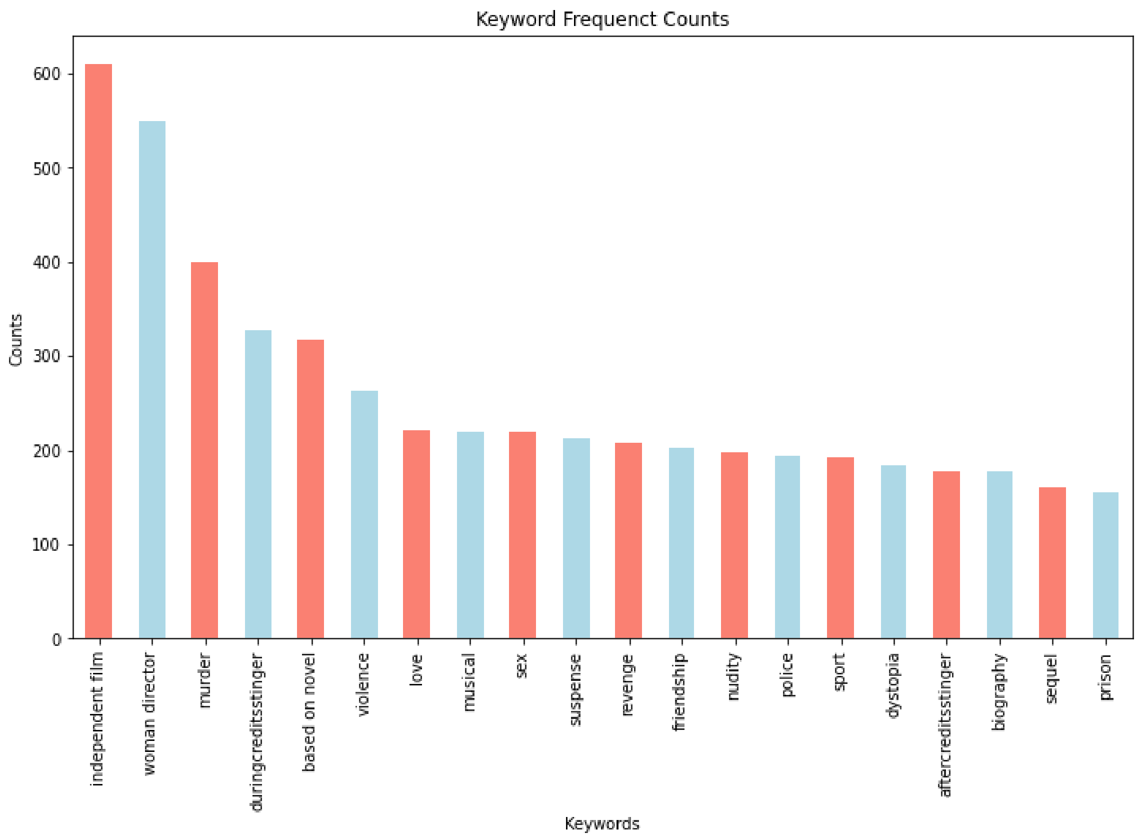


Fig 4.6: Calculate the frequency counts of every keyword that appears in the dataset showing through Bar chart

Before I use our keywords, I will process them. Each keyword's occurrences in the dataset are counted to determine its frequency of use.

The frequency with which a given phrase or term occurs in a dataset is referred to as the "frequent counts of every keyword." Text categorization, topic modeling, and information retrieval are just a few of the natural language processing and information retrieval activities that might benefit from this data.

The most crucial words or phrases in the movie's narrative synopsis, title, or other fields that characterize the movie's qualities may be found using frequent counts of every term in a movie recommendation system. The algorithm could count how often different types of movies, actors, directors, or keywords appear in each film.

The system would preprocess the movie data by tokenizing the text fields, eliminating stop words, and stemming the words before calculating the frequent counts of each term. The machine would then count the number of times each word or phrase occurs in the dataset. NLTK, SpaCy, and other libraries may be used for this.

Table 4.5: 10 most similar movies like "Toy Story" after changing cosine similarity scores

Movie Title	
1	Luxo Jr.
2	Toy Story 2
3	Cars 2
4	Cars
5	A Bug's Life
6	Toy Story of Terror!
7	Toy Story 3
8	Creature Comforts
9	Monsters, Inc.
10	Meet the Deedles

Table 4.6: 10 most similar movies like "The Godfather" after changing cosine similarity scores

Movie Title	
1	Tucker: The Man and His Dream
2	The Godfather: Part II
3	The Rainmaker
4	The Cotton Club
5	One from the Heart
6	Gardens of Stone
7	The Godfather: Part III
8	The Conversation
9	Rumble Fish
10	Peggy Sue Got Married

After changing the cosine similarity scores I found recommended movie list like “The Godfather” or “Toy Story” based on movies metadata. The movie metadata is used to describe information about a movie, such as its title, release date, genre, cast and crew, narrative synopsis, and other elements. This data may be used to determine a film's defining features and assign it a proper place in a larger collection of films.

Table 4.7: 10 most similar movies like "The Dark Knight" after calculate similarity scores and calculate the vote

Title	**Vote Count**	**Vote Average**	**Year**	**WR**
Inception	14075	8	2010	7.917588
Interstellar	11187	8	2014	7.897107
The Prestige	4510	8	2006	7.758148
Memento	4168	8	2000	7.740175
The Dark Knight Rises	9263	7	2012	6.921448
Batman Begins	7511	7	2005	6.904127
Batman Returns	1706	6	1992	5.846862
Batman Forever	1529	5	1995	5.054144
Batman v Superman: Dawn of Justice	7189	5	2016	5.013943
Batman & Robin	1447	4	1997	4.287233

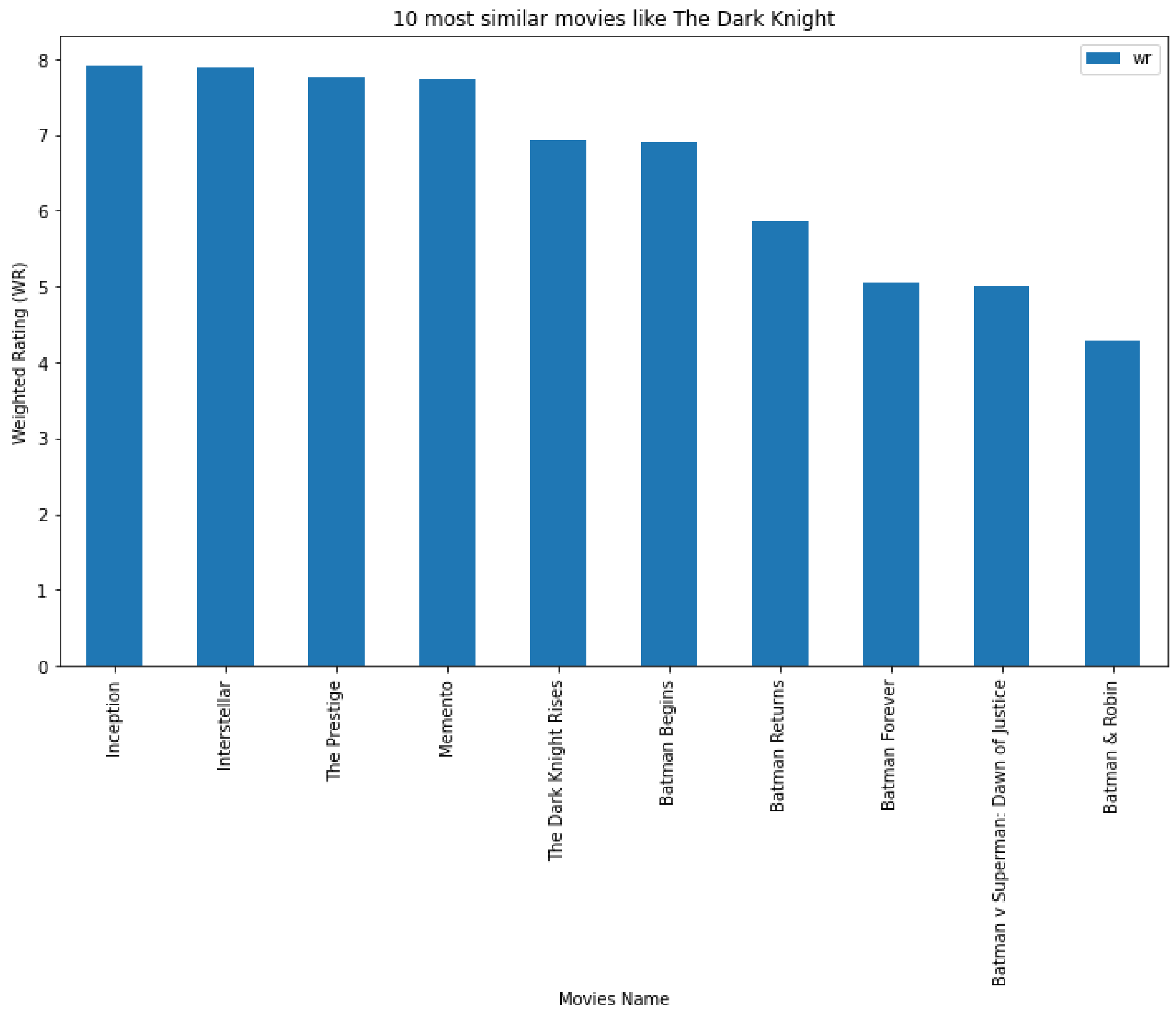


Fig 4.7: After calculate similarity scores 10 movies like "The Dark Knight" movies name and their weighted Rating showing through Bar chart

4.2.3 Collaborative-filtering Recommendation

The most popular application recommendation engine, collaborative filtering, uses data to make educated predictions about what users want next [22]. This engine relies on the premise that previous users of a product are good predictors of how they will react to similar products in the future. Collaborative filtering makes suggestions based on shared characteristics between people and material in order to overcome some of the shortcomings of content-based filtering [18]. This paves the way for serendipitous suggestions, in which a product is suggested to user A based on the preferences of user B. As an added bonus, the embedding's may be learned automatically

without requiring any human intervention or feature engineering [19]. For collaborative filtering to work, it requires a user's previous selections to be used as a basis for a set of criteria. This setup doesn't need a lot of fancy hardware to function. An object and a person are both described by an embedding or feature vector, which places them in the same embedding position. Enclosures for things and people are automatically generated.
The collaborative filtering procedure might be either of two types:

- Memory-based collaborative filtering
- Model-based collaborative filtering

Memory-based collaborative filtering:

One kind of CF, known as memory-based CF, uses the user's prior data in the form of rankings to determine how similar two people or products are. Primarily, this method seeks to describe the degree of similarity between users or items and to unearth consistent evaluations that may be used to promote the concealed features.
The two techniques that make up memory-based CF are as follows:

- User-based Collaborative Filtering
- Item-based Collaborative Filtering

User-based Collaborative Filtering:

User-based collaborative filtering is a technique for generating suggestions based on user similarities. Finding people who are similar to a certain user and then recommending things those users have enjoyed is the core notion.
The similarity between two users is the key component of user-based collaborative filtering. The Pearson correlation coefficient, which ranges from -1 to 1, is a popular metric for comparing similarities. When the value is -1, there is no correlation at all between the users' ratings; when it's 1, there is a perfect positive correlation between the users' ratings; and when it's 0, there is no connection between the users' ratings.

The Pearson correlation coefficient between two users, u and v, can be calculated as follows:

$$r(u, v) = \Sigma(i)\ (r(u,i) - \mu u)\ (r(v,i) - \mu v) \ /\ (\ \sqrt{\Sigma(i)(r(u,i) - \mu u)^2} * \sqrt{\Sigma(i)(r(v,i) - \mu v)^2})$$

where:

- The value of r(u,i) is the rating that user u has given to item i.
- The value of r(v,i) is the rating that user v has given to item i.
- u represents the average rating given by users.
- v represents the average rating of all users.
- The value denoted by I is the aggregate rating for all products for which both users have provided ratings

Once the degree of similarity between two users has been determined, it may be used to identify the other users who are most like a particular user and then propose products that those other users have enjoyed.

Another Method is Cosine Similarity

$$\cos(u, v) = \Sigma(i)\ (r(u,i) * r(v,i))\ /\ (\ \sqrt{\Sigma(i)(r(u,i)^2)} * \sqrt{\Sigma(i)(r(v,i)^2)}\ )$$

In CF algorithms, Pearson and Cosine similarity are both often used. Cosine Similarity, on the other hand, is more resilient when dealing with sparse data, which often occurs in recommendation systems.

Item-based Collaborative Filtering:

A technique for creating suggestions based on the similarity of things is item-based collaborative filtering. The fundamental concept is to identify products that are comparable to those that a user has previously enjoyed and then suggest those comparable products to the user.

When it comes to item-based collaborative filtering, similarity is king. The cosine similarity is a popular measure of similarity that ranges from -1 to 1. A value of 1 denotes entire identity between the two items, a value of -1 denotes complete dissimilarity, and a value of 0 denotes no resemblance at all.

The cosine similarity between two items, i and j, can be calculated as follows:

$$cos(i, j) = \Sigma(u)\ (r(u,i) * r(u,j)) / (\ \sqrt{\Sigma(u)(r(u,i)^2)} * \sqrt{\Sigma(u)(r(u,j)^2)}\ )$$

where:

- The value of r(u,i) is the rating that user u has given to item i.
- The value of r(u,j) is the rating that user u has given to item j.
- The symbol u signifies the aggregate rating of all users who have provided feedback on both items I and j.

Once the similarity between two goods has been determined, it may be used to identify the products that are most similar to a certain product and then suggest those products to the user.
Another method which also works well is Pearson Correlation.

$$r(i, j) = \Sigma(u)\ (r(u,i) - \mu i)\ (r(u,j) - \mu j) / (\ \sqrt{\Sigma(u)(r(u,i) - \mu i)^2} * \sqrt{\Sigma(u)(r(u,j) - \mu j)^2})$$

Cosine Similarity is better capable of managing sparse data in Item-Based CF, similar to User-Based CF.
Depending on use case, data, and computing capabilities, you can utilize any of these similarity metrics. It's also crucial to keep in mind that item-based CF necessitates precalculating the similarity between each item pair, which might be costly computationally if your data is extremely vast.

Model-based collaborative filtering

Model-based collaborative filtering is a strategy for formulating advice that makes predictions by understanding the underlying patterns in the data. This strategy is distinct from user-based collaborative filtering and item-based collaborative filtering, which depend on locating comparable persons or products to provide suggestions.
Matrix factorization, a popular kind of model-based collaborative filtering, aims to extract the latent representations of users and objects from the rating data. Matrix factorization may be used to break the rating matrix R down into two low-rank matrices, U and V, where U represents latent

representations of the users and V represents latent representations of the items. R(u,i) is the rating user u provided to item i.

The purpose is to bring the gap between actual and predicted results closer together. The most common optimization method is alternating least squares (ALS) or stochastic gradient descent (SGD) (ALS).
These approaches aim at approximating the original high-dimensional data matrix with a lower-dimensional one while retaining as much information as feasible. Then, predictions regarding missing ratings may be made using this lower-dimensional form.

Using neural networks, also known as deep learning-based collaborative filtering (CF), is an additional kind of model-based filtering. Autoencoders, Restricted Boltzmann machines (RBM), and other models are used in this situation. With the additional flexibility of neural networks, they are trained to learn the underlying representations of the data in a manner similar to matrix factorization.

Represent-based CF provides a number of benefits over memory-based CF, including increased scalability, managing massive amounts of data, the ability to deal with sparse data, and the capacity to more accurately model user preferences. It may be challenging to comprehend, tougher to execute, and harder to adjust. It also requires more computer resources and data pretreatment, among other disadvantages.

Logistic Regression

A supervised machine learning approach for resolving categorization issues is logistic regression. In order to determine whether a statement will be true or false, we utilize a set of independent variables. Instead of evaluating the value of the result, the method is used to estimate the likelihood of the outcome.
Finding the optimum parameters (weights) for the input characteristics that maximize the probability of the observed result is the fundamental tenet of the logistic regression process. In

order to discover the set of parameters that best fits the data, the method begins with an initial set of parameter estimations and iteratively updates those estimates.

The logistic regression formula is typically written as:

$$p(y=1|x) = 1 / (1 + e^{(-w^T x)})$$

Where:

- p(y=1|x) is the probability that the output y is 1, given the input x
- x is the feature vector used as input
- Weight vector w
- Natural logarithm base is e.
- -w^T x is the dot product of the weight vector and the input feature vector, which is also known as the input to the logistic function.

This function is called the sigmoid function, which is a special case of the logistic function. It is used to map the input to the probability of a binary outcome.

The logistic regression model may be used to forecast fresh data after the process has converged and the ideal set of parameters has been identified. For a given collection of input characteristics, the model is used to estimate the likelihood of the result, and a threshold is used to turn the probability into a binary result (1 or 0).

Random Forest

Random Forest may be used for both classification and regression, since it is a supervised machine learning technique. It is an ensemble method, which means it combines predictions from several models into a single prediction. The core idea behind Random Forest is to take an average of the predictions made by each decision tree after training it on separate subsets of data.

The precise mathematical formulation for the Random Forest technique relies on the classification or regression issue you're attempting to solve as well as how the approach is being used. However, there are a few overarching ideas that apply to the majority of Random Forest implementations.
For classification:

- Each tree in the forest "votes" for the category it believes the input point falls into.
- The class that obtains the most votes is the final forecast.

For Regression:

- The target variable is predicted by each tree in the forest.
- The final prediction is the average of all the trees' predictions.

In both situations, the following are used in the training of the decision trees within the random forest:

- To train each decision tree, choose a random subset of the data (with replacement).
- At each split, for each tree, choose a random subset of the features (without replacement).
- Utilizing the chosen subset of the characteristics and data, train a decision tree.
- Repeat the procedure until there are a certain number of trees.

In conclusion, the random forest technique may be compared to an ensemble of decision trees, where each tree is trained using a different random subset of the data and the final prediction is created by averaging the predictions of all the trees.

K-Nearest Neighbors (KNN)

The supervised machine learning method K-Nearest Neighbors (KNN) is utilized for both classification and regression applications. An input point is identified using the KNN method depending on how similar it is to other input points in the training dataset [25].
KNN determines the distance between a new observation and each observation in the training dataset before classifying it. Any legitimate distance metric, including the Minkowski distance, Manhattan distance, and Euclidean distance, may be used as the distance metric.

For instance, the following formula represents the Euclidean separation between two points x and y in a d-dimensional feature space:

$$Distance(x,y) = sqrt((x1-y1)^2 + (x2-y2)^2 + ... + (xd-yd)^2)$$

Where,
x and ***y*** are the two observations, and ***xi*** and ***yi*** are the values of the ith feature.

The method determines the distances and then chooses the K closest observations based on the distance measure. The chosen observations are referred to as the new observation's "neighbors." The procedure chooses the most prevalent class among the K chosen points to be the input point's class in classification problems. The method predicts the input point for regression problems using the mean of the target variable among the K chosen points.

In conclusion, KNN is a straightforward technique that is simple to grasp, but it may be computationally costly when working with big datasets since it has to calculate the distance between every point in the training dataset and the input point. Additionally, the curse of dimensionality, which is the decrease in algorithm performance as data complexity rises, may have an impact.

Singular Value Decomposition (SVD)

In linear algebra and data analysis, the singular value decomposition (SVD) method is a potent mathematical tool. It is the factorization of a real or complex matrix, which reduces the matrix to smaller, easier-to-understand components by decomposing it into a canonical form.

A matrix A is divided by the SVD into the following three matrices:
U: a unitary matrix representing the left singular vectors of A
S: a diagonal matrix with non-negative singular values of A on the diagonal
V*: the conjugate transpose of a unitary matrix representing the right singular vectors of A

The general equation for SVD is:

$$A = U * S * V^T$$

Where,

- A is the original matrix,
- U, S, and V^T are the three matrices obtained by SVD.

The decomposition is distinct and enables straightforward and effective low-rank approximations, such as truncation. Data compression, data visualization, and identifying a data set's key elements may all benefit from this.

Additionally, it aids in picture compression, noise reduction, and dimensionality reduction.

One of the most potent and adaptable mathematical techniques for data analysis, SVD is extensively utilized in a variety of applications, including computer vision, natural language processing, data mining, and information retrieval.

A data set's dimensionality may be decreased with SVD by locating and eliminating duplicate or pointless features. This may reduce the complexity of the data, making it simpler to read and evaluate. By deleting elements from the data that have minimal bearing on its general structure, SVD may be utilized to reduce its size. Working with huge data sets may be more efficient in terms of compute and storage.

After evaluating RMSE, MAE value for SVD algorithm given bellow:

Table 4.8: Evaluating RMSE, MAE of algorithm SVD on 5 split(s)

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Mean**	**Std**
RMSE (testset)	0.8933	0.9074	0.8904	0.8952	0.8964	0.8965	0.0058
MAE (testset)	0.6902	0.6971	0.6882	0.6868	0.6894	0.6903	0.0036
Fit time	1.3	1.36	1.31	1.32	1.32	1.32	0.02
Test time	0.13	0.15	0.14	0.3	0.19	0.18	0.06

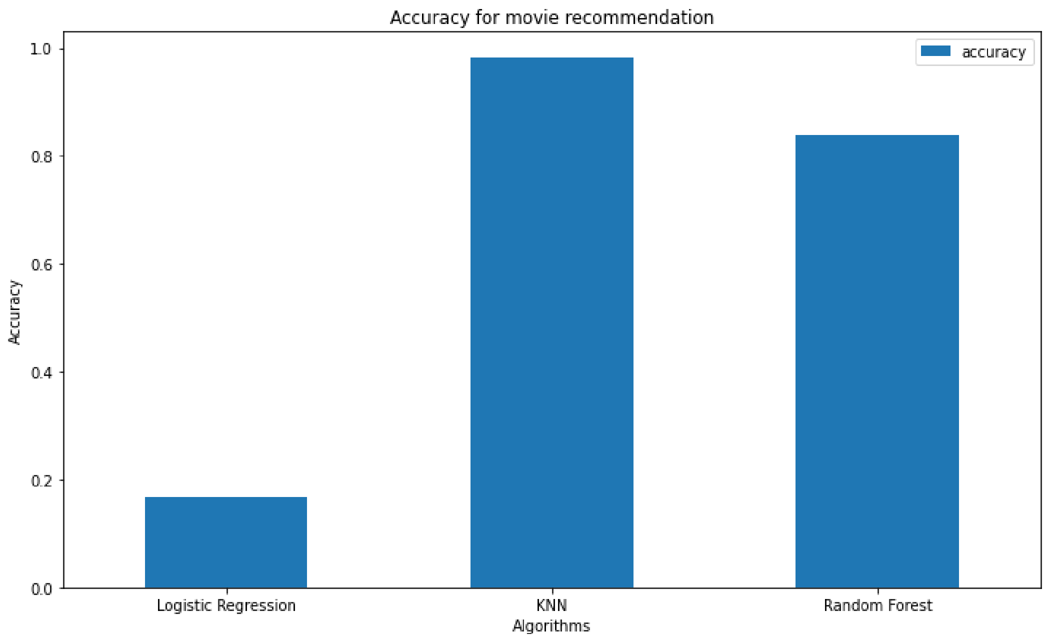


Fig 4.8: Accuracy score for movie recommendation system showing through Bar chart

A machine learning model's performance is measured using an accuracy score. It gauges how many of the model's predictions were accurate and is often used to classification difficulties. The number of accurate forecasts divided by the total number of predictions is the accuracy score. The outcome is a number between 0 and 1, where 1 denotes a perfect score and 0 denotes a score that is entirely off.

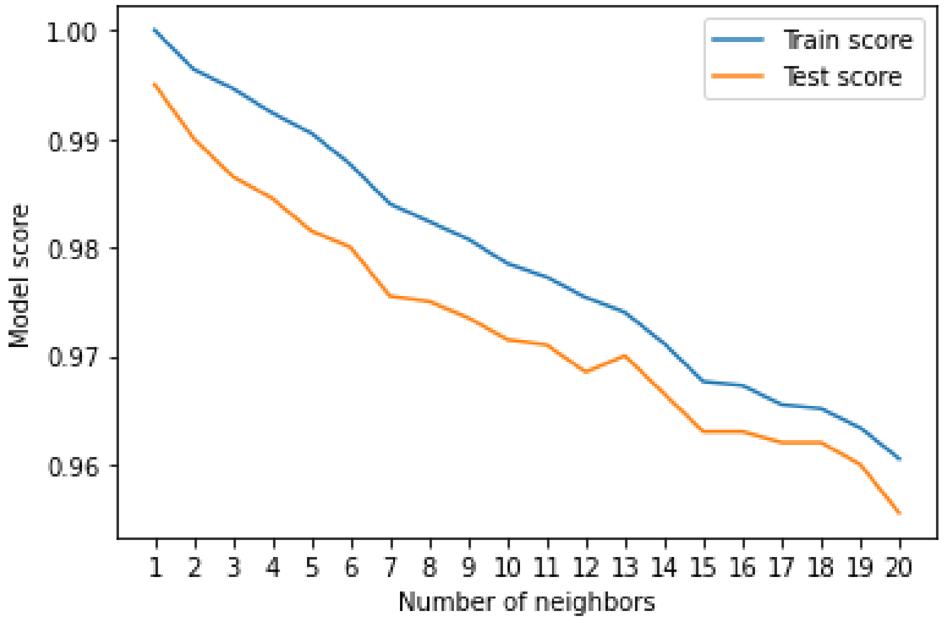


Fig 4.9: Accuracy score for KNN showing through Line chart

After calculation train score and test score K-Nearest Neighbors (KNN) model with an accuracy score of 0.9 means that the model correctly predicted the class of 90% of the instances in the test set. This is generally considered to be a very high accuracy score and indicates that the model is performing well on the test data. It's crucial to keep in mind that accuracy is not always the greatest statistic to assess a model's performance, particularly when the data is unbalanced, meaning that one class is much more prevalent than the others.

4.2.4 Hybrid Recommender Systems

Hybrid recommender systems are gaining popularity. Recent studies have shown that improving the effectiveness of both collaborative filtering and content-based filtering by combining them is possible. Combining the results of collaborative filtering (CF) with collaborative filtering (CB) suggestions is one technique to construct a hybrid recommender system [24]. Another is to add CF capabilities to a CB approach.

Few hybridization strategies exist, including:

- **Weighed:** the sum of ratings from several recommender features.
- **Swapping:** Select strategies by rearranging recommender subsystems.
- **Mixed:** Exhibit the outcome of several systems' recommendations.
- **Features Combination:** Gather characteristics from many inputs and use them all together.
- **Feature Augmentation:** Features are calculated by one recommender and sent into the next.
- **Cascade:** Use a recommender approach to provide a rough result, and then base further recommendations on that.
- **Meta-level:** Put the model created by one recommendation method into another as its input.

There may be many possible theoretical combinations, but that doesn't mean they'll all work well when applied to a given issue. In order to compensate for or circumvent the shortcomings of each recommender method, hybrid suggestion operates under a fundamental premise.

Table 4.9: For User Id 1 Avatar Movie Recommendation Movies Chart

Title	**Vote Count**	**Vote Average**	**Year**	**Movie id**	**Est Value**
The Terminator	4208	7.4	1984	218	3.2316
Terminator 2: Judgment Day	4274	7.7	1991	280	3.1012
Aliens	3282	7.7	1986	679	3.0385
Star Trek Into Darkness	4479	7.4	2013	54138	2.9618
Fantastic Planet	140	7.6	1973	16306	2.8328
Sinbad and the Eye of the Tiger	39	6.3	1977	11940	2.8305
Darby O'Gill and the Little People	35	6.7	1959	18887	2.8185
X-Men: Days of Future Past	6155	7.5	2014	127585	2.7532
The Abyss	822	7.1	1989	2756	2.7136
True Lies	1138	6.8	1994	36955	2.6779

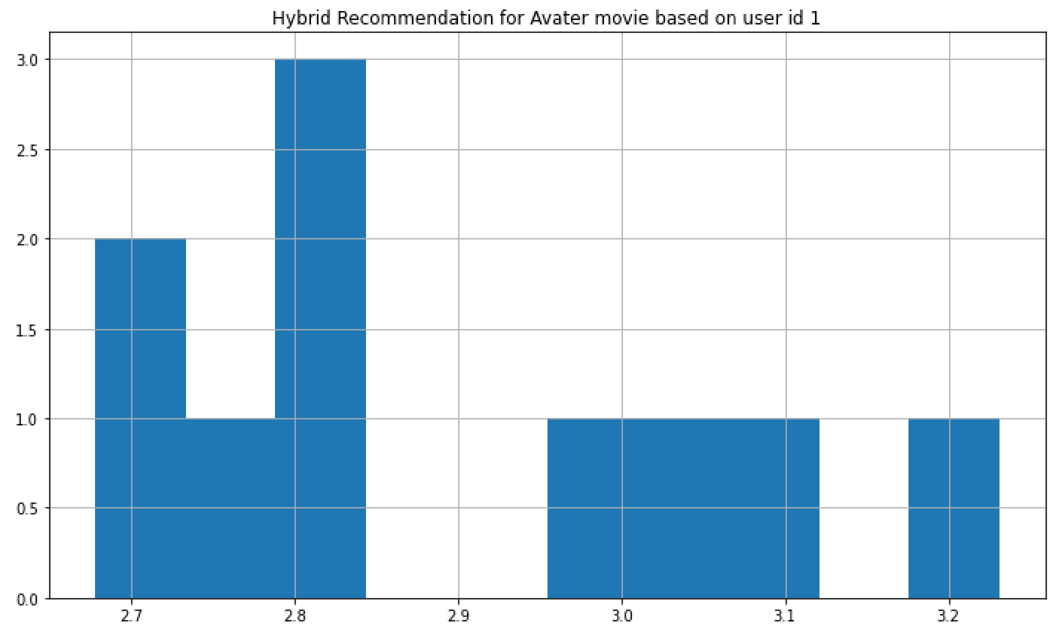


Fig 4.10: Hybrid Recommendation system for "Avater" movie estimation value based on user Id 1

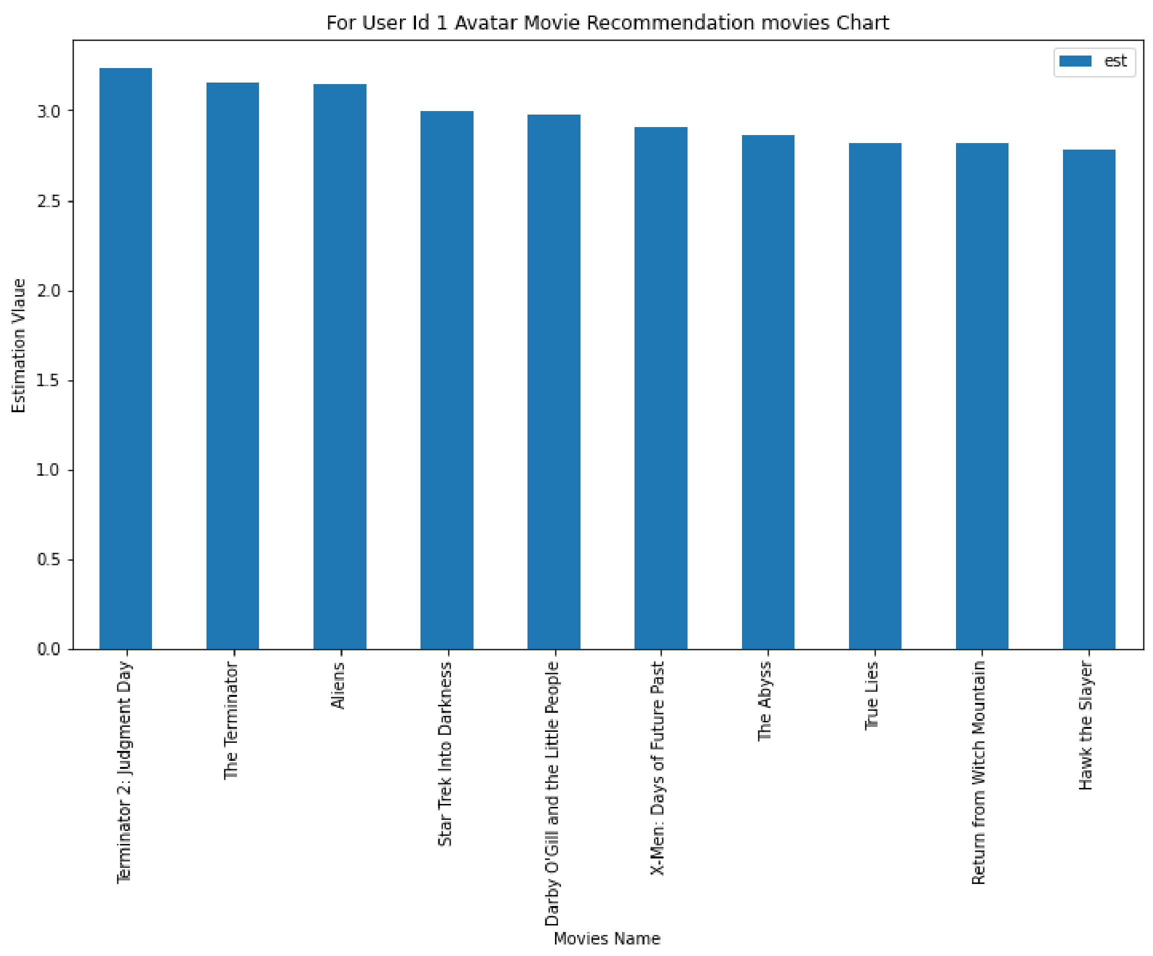


Fig 4.11: Recommended movie list for "Avater" movie based on user Id 1 using hybrid recommendation system

Table 4.10: For User Id 500 Avatar Movie Recommendation Movies Chart

Title	**Vote Count**	**Vote Average**	**Year**	**Movie id**	**Est Value**
Aliens	3282	7.7	1986	679	3.4702
Return from Witch Mountain	38	5.6	1978	14822	3.3685
Man of Steel	6462	6.5	2013	49521	3.3026
Fantastic Planet	140	7.6	1973	16306	3.2761
Star Trek Into Darkness	4479	7.4	2013	54138	3.2304
Sinbad and the Eye of the Tiger	39	6.3	1977	11940	3.1987

X-Men: Days of Future Past	6155	7.5	2014	127585	3.1586
Terminator 2: Judgment Day	4274	7.7	1991	280	3.1557
The Terminator	4208	7.4	1984	218	3.0484
Piranha Part Two: The Spawning	41	3.9	1981	31646	2.9634

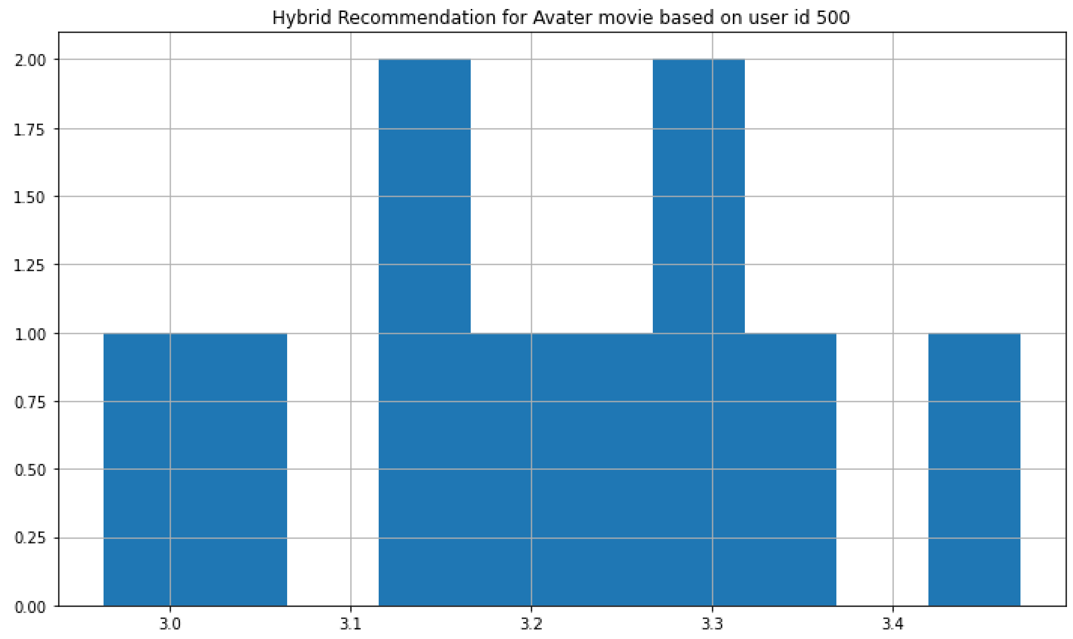


Fig 4.12: Hybrid Recommendation system for “Avater” movie estimation value based on user Id 500

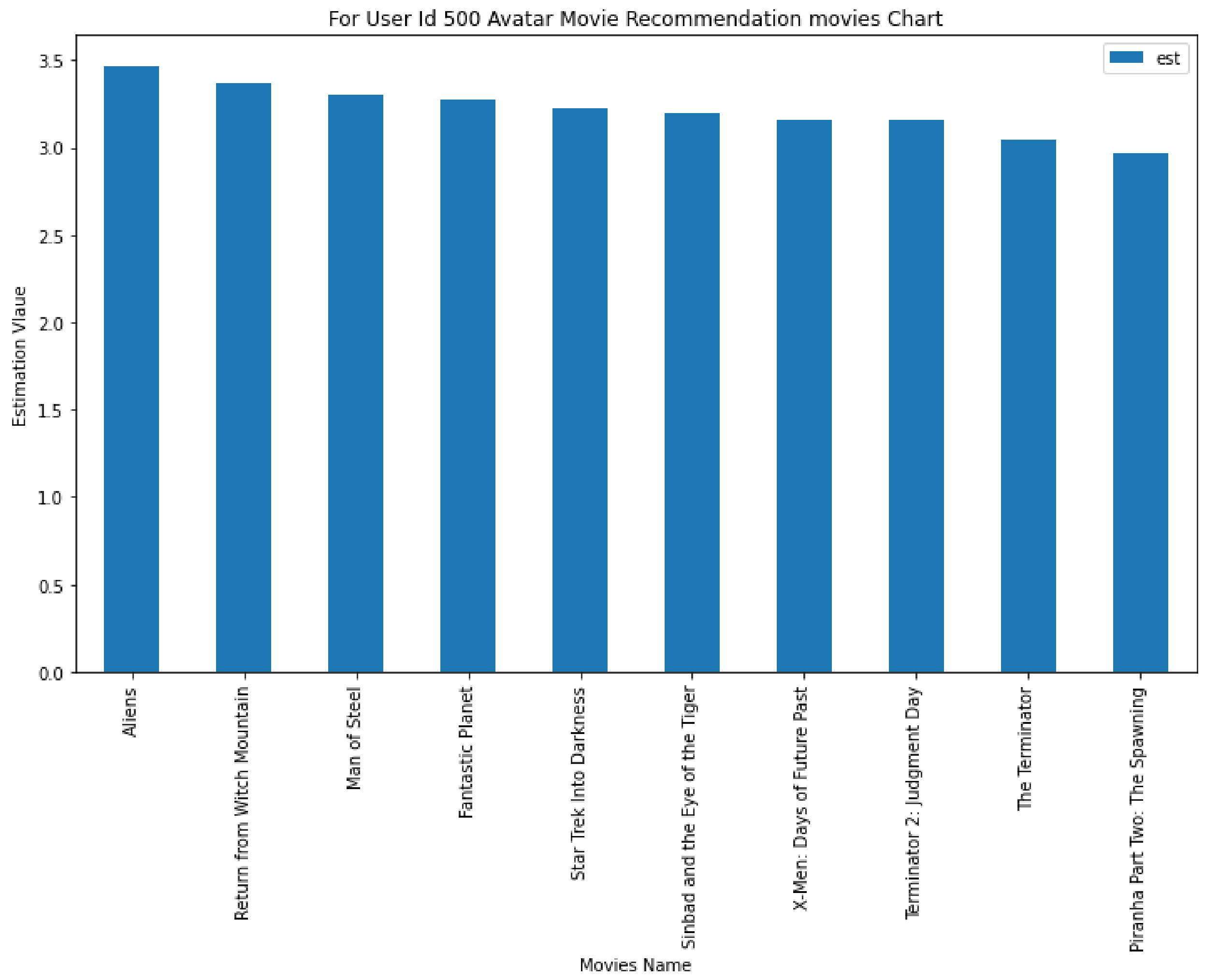


Fig 4.13: Recommended movie list for "Avater" movie based on user Id 500 using hybrid recommendation system

4.3 Final Decision

After calculating accuracy of Logistic Regression, Random Forest and KNN, KNN perform best for recommendation system. And after evaluation RMSE, MAE using SVD algorithm, it also best for recommendation system. Singular Value Decomposition (SVD) and K-Nearest Neighbors (KNN) are both powerful techniques that can be used in a movie recommendation system, but they have different strengths and weaknesses.

SVD is a linear algebra technique that is widely used for dimensionality reduction, data compression, and data visualization. It can be used to find the underlying structure of a movie dataset and extract the most important features. SVD can be used to identify patterns and relationships in a dataset and find latent features, which are useful in movie recommendation systems.

KNN, on the other hand, is a supervised machine learning method often used in classification and regression settings. It does this by determining which other films are most like the one you're looking for. By analyzing a user's watching habits, KNN may provide insightful suggestions for new movies to watch.

SVD could be useful in a movie recommendation system by reducing the dimensionality of the data and by finding the underlying structure of a movie dataset. This can make it easier to identify patterns and relationships in the data and provide a better understanding of the movie preferences. For this reason I use SVD algorithm for this paper.

Finding the most comparable movies to a given one might be a helpful feature of a movie recommendation system, and KNN could be used for this purpose. This may be used to provide users with specific movie suggestions depending on their watching habits.

It's also worth considering that you can use both SVD and KNN together in a recommendation system. SVD could be used to preprocess the data and reduce the dimensionality, followed by KNN for finding the nearest neighbors, combining the benefits of both methods for better recommendation system.

In summary, SVD and KNN are both powerful techniques that can be used in a movie recommendation system, but they have different strengths and weaknesses. The choice of which technique to use depends on the characteristics of the data, the problem and the specific requirements of the recommendation system.

CHAPTER 5
Planning for social, environmental, and sustainable factors

5.1 Effect on society

The goal of movie recommendation systems is to provide consumers with suggestions based on the elements they enjoy the most. A high performance rating on a movie suggestion will propose movies that most closely resemble the similarities. In this paper, a comprehensive review of the literature on movie recommendation systems is undertaken. It discusses the filtering standards used by movie recommendation systems, the algorithms employed there, the performance evaluation standards, the implementation challenges, and recommendations for additional research. Some of the most well-known machine learning techniques used in movie recommendation systems include principal component analysis, self-organizing maps employing principal component analysis. The application of recommendation systems based on metaheuristics in research initiatives is given special consideration. The study aims to highlight the advancements made in developing movie recommender systems and what has to be done to minimize the current challenges in implementing practical solutions. The report will help both researchers who study recommender systems in general and data scientists who work on putting these systems into place.

5.2 Feasibility Study

After determining whether or not to proceed with the project, a business proposal summarizing the project's high points and offering some ballpark cost estimates is provided to the client. During the system analysis, the feasibility of the proposed system will be taken into account. This step is used to verify that the suggested system won't end up costing the business more money than it's worth. A thorough grasp of the essential needs of the system is necessary for conducting a feasibility study. The feasibility analysis involves three main factors, which are:

- Feasibility in terms of the Economy
- Feasibility in terms of Technology
- Feasibility in terms of society

5.2.1 Feasibility in terms of the economy

The study will be used to calculate how much money the system will end up costing the business. There is a limit on how much resources the company may put on developing the system. There has to be a thorough breakdown of all fees. That's why it's not only efficient, but also cheap: the system. This is so because the vast majority of used technologies are freely available to the public. Specifically, only the customized products were mandated purchases.

5.2.2 Feasibility in terms of technology

Examining the system's technical viability, or whether or not it meets the system's technical criteria, is the purpose of this research. Any system created shouldn't place an excessive load on the existing technological infrastructure. This will place a significant strain on the available technical resources. That's going to put a lot of pressure on the customer. It's important that the built system has minimal needs so that implementing it requires as little effort as possible.

5.2.3 Feasibility in terms of society

The goal of this research is to determine how well the system is received by its end users. Making the most of technological resources relies on proper training of end users. The user shouldn't be afraid to utilize the system. User adoption is directly proportional to the effort put into familiarizing and training new users on the system. His self-assurance has to be bolstered so that, as the system's end user, he can provide the constructive feedback that is always appreciated.

5.3 Sustainable planning

Entertainment is essential if any of us are to renew our spirits and energy in this hectic environment. Entertainment helps us regain our sense of purpose at work, which enables us to work more fervently. To re-energize ourselves, we may watch movies or listen to the music of our choosing. Given that picking a movie to watch is becoming more time-consuming, and time is a resource we can't afford to squander, we may turn to movie recommendation algorithms to help us choose worthwhile content to watch online. These technologies are more reliable. Recommended in this research is a hybrid strategy that combines content-based filtering with collaborative filtering using a support vector machine as a classifier and a genetic algorithm. The outcomes of comparative analyses have been presented, revealing that the proposed method is superior to the pure techniques investigated in this study in terms of accuracy, quality, and scalability. By incorporating the finest aspects of both approaches, the hybrid approach seeks to strike a compromise between their benefits and drawbacks.

CHAPTER 6
Summary, Findings, Recommendations, and Implications for Future Research

6.1 Summary of research

A machine learning-based movie recommendation system is one that uses machine learning algorithms to suggest movies to consumers based on their watching habits and interests. Although there are many different approaches to create these systems, collaborative filtering or content-based filtering are usually used. In this paper I use content-based filtering techniques for the qualities of the movies itself, such as genre, director, and actors, and also use collaborative filtering techniques depend on the watching habits and histories of other users to produce suggestions.

The system would examine a user's watching history and preferences and compare them to the viewing histories and preferences of other users via collaborative filtering. The technology will suggest films that the other user has seen and liked if they have a similar watching history and tastes. In content-based filtering, the system would examine the film's attributes, such as its genre, director, and actors, and suggest films that are comparable to those that a user has already seen and found enjoyable.

Machine learning methods used in recommender systems are designed to tailor their suggestions to each individual user. Users may get individualized movie suggestions using a machine learning-based recommendation engine, which can also assist them in finding new films they would like.

6.2 Limitations and conclusions

In this thesis paper, I am doing 4 different types of recommendation systems. There are simple recommendation system, a content-based recommendation system, a collaborative filtering recommendation system, and a hybrid recommendation system.

In Simple recommendation system, I create Top Movies Charts for both a broad category and a specialized genre, this technique employed Vote Count and Vote Averages. The ratings on which the final sorting was conducted were determined using the IMDB Weighted Rating System. In Content-based recommendation system, I created two content-based engines: one that generated predictions using information like cast, crew, genre, and keywords, and the other that used movie summaries and taglines as input. We have created a straightforward filter to favor movies with more votes and better ratings. In Collaborative filtering recommendation system, I created a collaborative filter based on single value decomposition using the potent Surprise Library. The engine provided projected ratings for a certain user and movie based on the RMSE, which was less than 1. In Hybrid recommendation system, I used concepts from content and collaborative filtering to create an engine that suggested movies to a specific user based on estimated ratings that the engine had internally generated for that user. I need a way to sort movies off of a user's disliked list when they actively want to avoid them. I can't do that right now, but doing so would greatly enhance the quality of my recommendations.

6.3 Future research implications

Since its inception many years ago, the recommender system has never experienced a low moment. The advancement of machine learning, large-scale networks, and high speed computers in recent years has encouraged fresh research in this area. In our next work, we'll take the following factors into account.

Few topics will be the focus of future study. The first step is investigating more specialized/personalized recommendations while taking into account the unique settings. This strategy may be used in different contexts, including service clustering. In addition to the stated interests ratings, users' implicit interests may be discovered via mining use data or user evaluations. This method enables the generation of suggestions even when there are minimal ratings. This will somewhat address the sparsity issue.

Introduce the movie's more accurate and appropriate elements. Common collaborative filtering suggestions substitute the rating for object characteristics. In the future, we should extract attributes from movies like color and subtitles that may provide a more accurate description of the film.

Create a database of movies that users actively detest. In recommender systems, user input is always valuable. A future goal is to compile a list of movies that users have expressed distaste towards. We'll also feed in movies we don't want to see into the recommender system so it can add those ratings to the ones it already has. A better recommender system outcome may be achieved in this manner.

I need a way to sort movies off of a user's disliked list when they actively want to avoid them. I can't do that right now, but doing so would greatly enhance the quality of my recommendations.

APPENDICES

Abbreviation

CF = Collaborative Filtering	
CBF = Content-based filtering	
TF = Term Frequency	
IDF = Inverse Document Frequency	

REFERENCES

[1] Goldberg, D. et al. (1992) “Using collaborative filtering to weave an information tapestry,” Communications of the ACM, 35(12), pp. 61–70. Available at: https://doi.org/10.1145/138859.138867.

[2] Bhatt, B.: A review paper on machine learning based recommendation system. Int. J. Eng. Dev. Res. (2014)

[3] Zhao, L., et al.: Matrix factorization for movie recommendation. In: IJCAI (2016)

[4] M. Balabanovic. An adaptive web page recommendation ´ service. In Proceedings of the first international conference on Autonomous agents, pages 378–385. ACM, 1997.

[5] M. Balabanovic and Y. Shoham. Fab: content-based, col- ´ laborative recommendation. Communications of the ACM, 40(3):66–72, 1997.

[6] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. Machine learning, 27(3):313–331, 1997.

[7] G. Lekakos and P. Caravelas. A hybrid approach for movie recommendation. Multimedia tools and applications, 36(1):55–70, 2008.

[8] J. A. Konstan and J. Riedl, ”Recommender systems: From algorithms to user experience”, User Modeling and User-Adapted Interaction 22(1-2):101-123, 2012. DOI: http://dx.doi.org/10.1007/s11257-011-9112-x

[9] R. Katarya and O. P. Verma, ”An effective collaborative movie recommender system with cuckoo search”, Egyptian Informatics Journal 18(2):105-112, 2017. DOI: http://dx.doi.org/10.1016/j.eij.2016.10.002

[10] F. Mansur, V. Patel, and M. Patel, ”A review on recommender systems”, IEEE International Conference on Innovations in Information, Embedded and Communication Systems, 1-6, 2017. DOI: http://dx.doi.org/10.1109/ICIIECS.2017.8276182

[11] M, D. (2022) What is cosine similarity and how is it used in machine learning?, Analytics India Magazine. Available at: https://analyticsindiamag.com/cosine-similarity-in-machine-learning/#:~:text=Cosine%20similarity%20is%20used%20as,of%20texts%20in%20the%20document. (Accessed: January 13, 2023).

[12] Das, S. (2022) Build a movie recommendation system on your own, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/ (Accessed: January 13, 2023).

[13] Jayaswal, V. (2020) Text vectorization: Term frequency‐inverse document frequency (TFIDF), Medium. Towards Data Science. Available at: https://towardsdatascience.com/text-vectorization-term-frequency-inverse-document-frequency-tfidf-5a3f9604da6d (Accessed: January 13, 2023).

[14] J.A. Konstan, J. Riedl Recommender systems: from algorithms to user experienceUser Model User-Adapt Interact, 22 (2012), pp. 101-123

[15] B. Pathak, R. Garfinkel, R. Gopal, R. Venkatesan, F. Yin Empirical analysis of the impact of recommender systems on sales J Manage Inform Syst, 27 (2) (2010), pp. 159-188

[16] TF–IDF (2023) Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Tf%E2%80%93idf (Accessed: January 13, 2023).

[17] Cosine similarity (2020) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/cosine-similarity/ (Accessed: January 13, 2023).

[18] Su, X. and Khoshgoftaar, T.M. (2009) A survey of collaborative filtering techniques, Advances in Artificial Intelligence. Hindawi. Available at: https://www.hindawi.com/journals/aai/2009/421425/ (Accessed: January 13, 2023).

[19] Collaborative filtering recommender systems - cs229.stanford.edu (no date). Available at: http://cs229.stanford.edu/proj2014/Rahul%20Makhijani,%20Saleh%20Samaneh,%20Megh%20Mehta,%20Collaborative%20Filtering%20Recommender%20Systems.pdf (Accessed: January 13, 2023).

[20] Goyani, M., & Chaurasiya, N. (2020). A review of movie recommendation system: Limitations, Survey and Challenges. ELCVIA: electronic letters on computer vision and image analysis, 19(3), 0018-37.

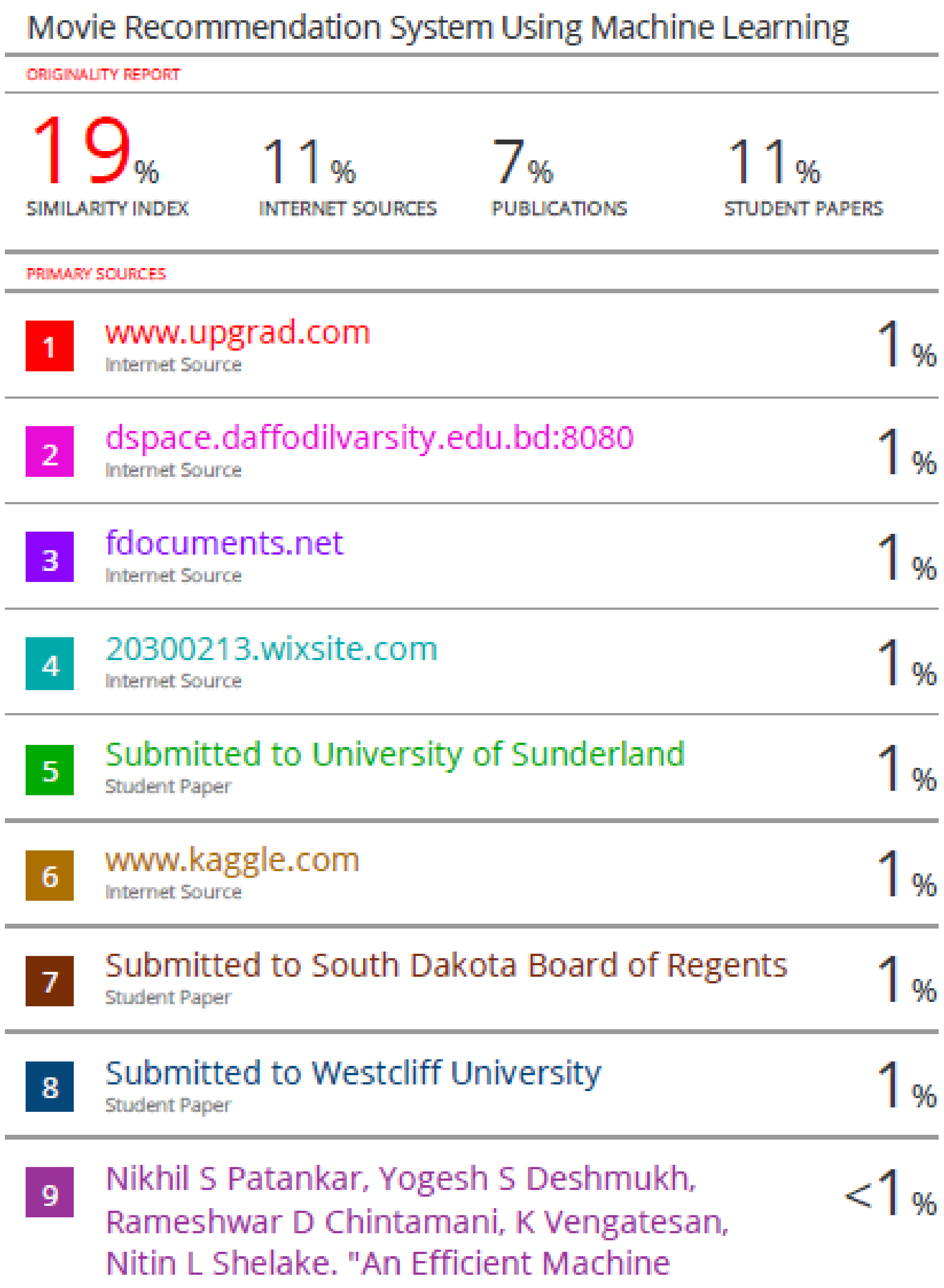
[21] Reddy, S. R. S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2019). Content-based movie recommendation system using genre correlation. In Smart Intelligent Computing and Applications (pp. 391-397). Springer, Singapore.

[22] Wu, C. S. M., Garg, D., & Bhandary, U. (2018, November). Movie recommendation system using collaborative filtering. In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) (pp. 11-15). IEEE.

[23] Furtado, F., & Singh, A. (2020). Movie recommendation system using machine learning. International journal of research in industrial engineering, 9(1), 84-98.

[24] Christakou, C., Vrettos, S., & Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. International Journal on Artificial Intelligence Tools, 16(05), 771-792.

[25] Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie recommender system using K-Means clustering and K-Nearest Neighbor. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 263-268). IEEE.



Movie Recommendation System Using Machine Learning

ORIGINALITY REPORT

19%	11%	7%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.upgrad.com Internet Source	1%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	fdocuments.net Internet Source	1%
4	20300213.wixsite.com Internet Source	1%
5	Submitted to University of Sunderland Student Paper	1%
6	www.kaggle.com Internet Source	1%
7	Submitted to South Dakota Board of Regents Student Paper	1%
8	Submitted to Westcliff University Student Paper	1%
9	Nikhil S Patankar, Yogesh S Deshmukh, Rameshwar D Chintamani, K Vengatesan, Nitin L Shelake. "An Efficient Machine	<1%