The Ultimate Recommendation System: *Proposed Pranik System*

Vipin Kumar[1], Amit Kumar Gupta[1], Prakriti Yadav[2], Nikhil Kumar[1], Rajeev Kumar[1*]
[1]*KIET Group of Institutions, Delhi NCR Ghaziabad, India,*
[2]*Indian Institute of Management Indore, Indore, Madhya Pradesh, India*
rajeev.rakshit@gmail.com; rajeev.kumar@kiet.edu

Abstract*— In today's fast-paced world, recommendation systems have become indispensable tools, aiding users in making personalized decisions amidst an overwhelming array of choices. These systems leverage user data and preferences to generate tailor-made recommendations based on individual tastes and behaviors. This research paper introduces the development and implementation of Pranik Movies, an ultimate recommendation system for personalized movie suggestions. The system incorporates collaborative and content-based filtering techniques, utilizing machine learning algorithms to analyze user behaviors, ratings, and viewing histories. A comprehensive overview of the research framework is provided, encompassing system architecture, data pre-processing, feature engineering techniques, and model selection and design. Text processing methods such as stemming, bag-of-words (BoW), and TF-IDF (Term Frequency-Inverse Document Frequency) are employed for processing and analyzing textual movie data. The accuracy of recommendations is enhanced through the assessment of film similarities, utilizing algorithms like cosine similarity and Euclidean distance. The paper concludes by outlining future directions for advanced machine learning techniques, social media integration, expanded content support, and the refinement of the evaluation framework. Pranik Movies signifies a significant advancement in recommendation systems, enabling personalized and precise movie recommendations within a vast and diverse cinematic landscape.

Keywords: Recommendation engine, Machine learning, Deep learning, Natural language processing, Entertainment, Movies.

I. INTRODUCTION

In today's rapidly evolving world, the constant influx of decisions inundates us with a myriad of choices each day. This phenomenon, known as "decision fatigue," stems from the abundance of options available, ranging from selecting our evening meals to choosing books and films for leisure. This challenge is further compounded by the deluge of information characterizing our present era. Within this context, recommendation systems emerge as indispensable tools. Designed with meticulous precision, these systems alleviate decision fatigue by guiding users to pinpoint items, features, or products that align with their preferences and past interactions.

The process of selecting a movie has become a daunting task, entailing considerations of mood and interest. Amidst the vast cinematic landscape, identifying the right film that resonates with one's current mood and preferences poses a significant challenge. This is where the "Pranik System" recommendation system comes into play—a groundbreaking innovation engineered to redefine the movie selection process. Termed the "Pranik System," this advancement seamlessly integrates the capabilities of collaborative and content-based filtering techniques with cutting-edge strides such as deep learning and natural language processing. Our system excels in providing precise recommendations across a diverse spectrum of films, adeptly leveraging a nuanced blend of advanced methodologies. These personalized recommendations are meticulously tailored to resonate with the distinct cinematic inclinations of each individual user accessing our platform.

The Pranik System strategically positions itself to address the complexities inherent in the swiftly evolving cinematic landscape, characterized by an overabundance of streaming choices and an inundation of information. Purposefully designed to surmount these challenges, this ingenious system streamlines decision-making complexities and alleviates the cognitive load linked to such choices. By delivering individualized movie recommendations that seamlessly align with users' cinematic preferences and viewing histories, the Pranik System introduces a realm of simplicity into this intricate process.

The domain of recommendation systems, distinguished by its flexibility and robustness, continually expands. With each click, databases receive updates, leading to perpetually evolving suggestions that enrich user experiences while alleviating cognitive strain. These systems have etched remarkable footprints across various sectors, including e-commerce, entertainment, healthcare, and education, underscoring their multifaceted influence.

Platform	**Specifications**		
	Type	***Techniques used***	***Data used***
Netflix	Streaming platform	Collaborative, content-based filtering	Viewing habits, preferences, ratings
Amazon	E-commerce	Collaborative, content-based, item-based filtering	Browsing, purchase history, item similarities, customer reviews
Spotify	Music and Podcasts	Collaborative, content-based filtering, matrix factorization techniques	History, playlists
Youtube	Social media platform	Collaborative, content-based filtering, deep learning models	History, likes, dislikes, subscriptions

Table 1: Platform Specification Information

Table 1 can be used as a helpful reference point because it gives an overview of the platform requirements for well-known services. This table highlights the employed techniques and utilised data for recommendation systems, highlighting how different platforms utilise a variety of strategies to improve user experiences. This table becomes a

compass for comprehending the multifaceted approaches adopted by various platforms in order to recommend content to their users when it is viewed in the context of the proposed Pranik System.

II. RELATED WORK

In the field of movie recommendation systems, several notable research contributions have been made. One such contribution is the trust-based collaborative filtering algorithm proposed by Liaoliang Jiang, Yuting Cheng, Li Yang, Jing Li, Hongyang Yan and Xiaoqin Wang [1]. This algorithm incorporates trust relationships among users to improve the accuracy and reliability of recommendations in E-commerce systems. It estimates user trustworthiness using a trust propagation method based on historical behavior and interactions. By considering both item similarity and user trust, personalized recommendations are generated, outperforming other collaborative filtering approaches in terms of recommendation accuracy.

Important to the functioning of recommendation systems is the practise of collaborative filtering, which is utilised to a large extent by market leaders such as Netflix and Amazon. One such tried-and-true method utilised in the process of recommending films is called content-based filtering. It does this by analysing the item's qualities, such as the genre, cast, director, and storyline summaries, to recommend other movies that are comparable.

Within the realm of recommendation systems, content-based filtering is a tried-and-true method that centres its attention, in order to make suggestions, on the qualities of the objects under consideration. K. Iwahama, Y. Hijikata, and S. Nishida [2] developed a content-based filtering system for music data with the intention of creating a recommendation system that suggests music based on the content analysis of songs. This was accomplished by developing a content-based filtering system for music data. Their research is centred on the creation of a recommendation algorithm that is based on item profiles. This notion has been used as the basis for later content-based recommendation systems.

It has come to the notice of those working in the field of recommendation systems that hybrid recommendation systems are able to improve the accuracy of recommendations by integrating a number of different methods. A hybrid model that combines incremental collaborative filtering with content-based algorithms has been proposed by H. Wang, P. Zhang, T. Lu, H. Gu, and N. Gu [3]. This model is referred to as the hybrid model. Their research is focused on improving the efficiency and precision of recommendation systems, which is its primary objective. In order to generate recommendations that are both more accurate and more diversified, the proposed methodology incorporates both collaborative filtering and content-based filtering.

With the rise of deep learning, neural networks have gained popularity in recommendation systems. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua [4] proposed a neural collaborative filtering model that combines matrix factorization and neural networks to capture both user-item interactions and item-item similarities.

P. Sharma and L. Yadav [5] discuss a movie recommendation system based on item-based collaborative filtering. They explore the use of collaborative filtering techniques to generate personalized movie recommendations. The study focuses on improving recommendation accuracy by considering the similarities between movie items. The authors present a methodology for building the recommendation system and evaluate its performance.

H. Zhang, M. Gan, and X. [6] Sun discuss the challenges of movie recommendation in location-based social networks and propose an approach that incorporates memory-based preferences and point-of-interest stickiness. Their work aims to improve the accuracy and effectiveness of recommendations in such networks. By considering user preferences and the stickiness of locations, the proposed method offers personalized movie recommendations.

The importance of explainable recommendations has also been emphasized. N. Tintarev and J. Masthoff [7] discussed the significance of providing users with understandable explanations for movie recommendations and proposed different techniques for generating explanations.

In the realm of movie recommendation systems, a notable contribution is the "Factorization Machines for Movie Recommendations" approach proposed by Steffen Rendle [8]. This work introduces the concept of Factorization Machines (FM) as a powerful tool for recommendation tasks. FM are a type of machine learning model that excels in capturing interactions between categorical variables, making them particularly suitable for recommendation scenarios where user-item interactions are prevalent. The paper presents how FM can be applied to movie recommendation, leveraging user and item features to predict user preferences. The model's effectiveness is demonstrated through experimental results showcasing improved recommendation accuracy compared to traditional collaborative filtering techniques.

This study introduces the Pranik Movies recommendation system, which has innovative components. Collaborative and content-based filtering provide user-specific movie suggestions. Machine learning algorithms analyse user activity, ratings, and watching history to make suggestions. Advanced text processing methods like stemming increase movie text analysis. The technology improves movie suggestions with similarity algorithms. The study framework includes system architecture, data pre-processing, feature engineering, model selection, and design. It also covers implementation, deployment, user interface functionality, and assessment metrics to better understand the system.

III. RESEARCH FRAMEWORK

A comprehensive movie recommendation system is an intricate interplay of several components meticulously designed to offer personalized recommendations. Here, we outline the architecture of our proposed system and delve into the data pre-processing, feature engineering, exploratory data analysis, text processing techniques, and similarity algorithms that constitute its core.

A. Architecture

The architecture of our movie recommendation system is depicted in Figure 1, illustrating the flow of methods and processes that contribute to delivering accurate and personalized movie recommendations.

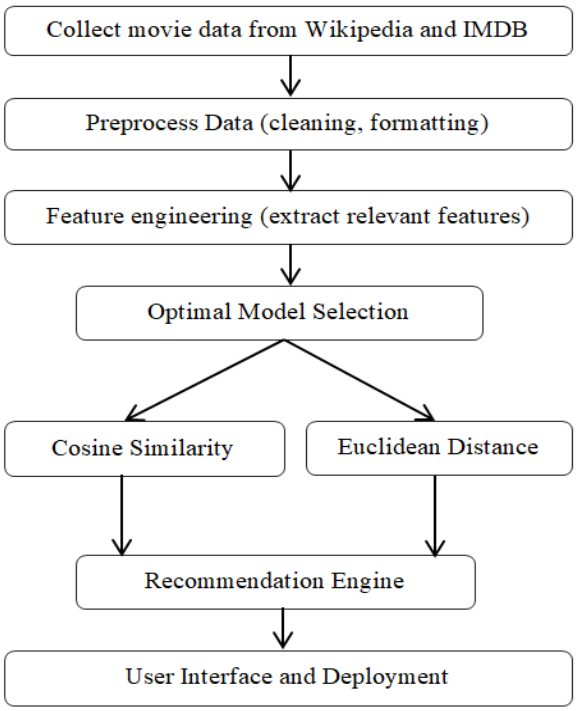

Figure 1: Flowchart of Proposed Methods

B. *Data Pre-processing and Feature Engineering*

The foundation of any recommendation system lies in the quality and relevance of its data. In the data-gathering phase, we aimed to create a comprehensive database by amalgamating information from various sources, including Wikipedia and the TMDB API. While Wikipedia provided an overview, the TMDB API furnished us with granular details such as title, genre, cast, release date, duration, box office earnings, and reviews for movies released from 2000 to 2023.

The dataset was meticulously pre-processed, including integrity review, anomaly identification, and data cleaning and manipulation. This crucial step ensured that our analyses were based on cohesive and reliable data.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10455 entries, 0 to 10454
Data columns (total 22 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Title              10455 non-null  object
 1   Id                 10455 non-null  int64
 2   Trailer Link       7185 non-null   object
 3   Director           9993 non-null   object
 4   Cast               10455 non-null  object
 5   genre_ids          0 non-null      float64
 6   Genre              9736 non-null   object
 7   Budget             10455 non-null  int64
 8   Revenue            10455 non-null  int64
 9   Overview           10344 non-null  object
 10  Homepage           3609 non-null   object
 11  Year               10368 non-null  float64
 12  Runtime            10455 non-null  int64
 13  Popularity         10455 non-null  float64
 14  Adult              10455 non-null  bool
 15  Release_Date       10368 non-null  object
 16  Original_Title     10455 non-null  object
 17  Original_Language  10455 non-null  object
 18  Tagline            6089 non-null   object
 19  Vote_Average       10455 non-null  float64
 20  Vote_Count         10455 non-null  int64
 21  Reviews            3420 non-null   object
dtypes: bool(1), float64(4), int64(5), object(12)
memory usage: 1.8+ MB
```

Figure 2: Information on Collected Data

From the gathered dataset, we meticulously selected crucial attributes that encompassed the most relevant aspects of movie data. By adeptly addressing missing values and ensuring consistency through a comprehensive data cleansing process, we laid a solid foundation for conducting exploratory data analysis and developing predictive models.

Variable	***Description***
Title	The title of the movie.
Trailer Link	A link to the movie's trailer.
Director	The director(s) responsible for the movie.
Cast	The cast members featured in the movie.
Genre IDs	Numerical identifiers for movie genres.
Genre	A specific genre classification.
Budget	The financial allocation for creating the movie.
Revenue	The earnings generated by the movie.
Overview	A brief synopsis or overview of the movie's plot.
Homepage	A link to the official homepage of the movie.
Year	The year in which the movie was released.
Runtime	The duration of the movie.
Popularity	A measure of the movie's popularity.
Adult	An indication of whether the movie is intended for adult audiences.
Release Date	The date of the movie's release.
Original Title	The original title of the movie (if applicable).
Original Language	The language in which the movie was originally produced.
Tagline	A memorable tagline associated with the movie.
Vote Average	The average user rating for the movie.
Vote Count	The total count of user votes for the movie.
Reviews	Reviews or critical commentary related to the movie.

Table 2: Essential variables encompassed by the dataset

Overall, the data collection preprocessing phase [9] laid the groundwork for the project's later phases of exploratory data analysis and model building.

C. *Exploratory Data Analysis (EDA)*

Our system for recommending films is built on a foundation of exploratory data analysis, also known as EDA[10]. As a result of going through this process, we are able to unearth insights, recognise patterns, and locate relationships within the dataset. We are able to understand the distribution of variables, discover correlations, and comprehend the subtleties of the movie landscape thanks to EDA.

First, we examine the variable distributions, looking for patterns using statistical methods and graphical representations like histograms, box plots, scatter plots [11], and frequency tables. By conducting these explorations, we are able to address any biases in our data, find previously unknown trends, and unearth the characteristics of our data.

For instance, our analysis revealed the top revenue-generating genres, spotlighting their impact on the film industry's financial success.

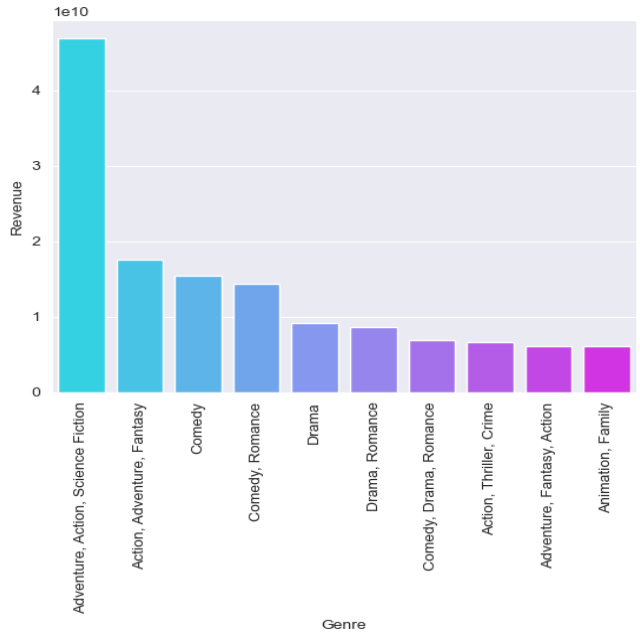

Figure 3: Top 10 Genres by Revenue in the Dataset

Further insights into movies' quantitative attributes were gained by examining statistical labels. These visualizations unveil distributions, central tendencies, and variability.

	Budget	Revenue	Year	Runtime	Popularity	Vote_Average	Vote_Count
count	1.045500e+04	1.045500e+04	10368.000000	10455.000000	10455.000000	10455.000000	10455.000000
mean	1.789673e+07	5.385357e+07	2010.547454	106.692874	40.084622	5.533678	1250.579818
std	3.829793e+07	1.591697e+08	10.009770	38.530287	415.762614	2.221473	2866.090257
min	0.000000e+00	0.000000e+00	1897.000000	0.000000	0.600000	0.000000	0.000000
25%	0.000000e+00	0.000000e+00	2005.000000	93.000000	1.823500	5.253000	4.000000
50%	0.000000e+00	0.000000e+00	2012.000000	108.000000	8.214000	6.155000	98.000000
75%	1.917500e+07	3.119954e+07	2017.000000	130.000000	18.786500	6.840000	1087.500000
max	3.650000e+08	2.923706e+09	2023.000000	339.000000	10773.574000	10.000000	33609.000000

Figure 4: Describe the quantitative columns

By examining these charts, we can learn more about the distribution, central tendency, and variability of these statistical features. These data are useful in understanding the movie's qualitative and statistical properties [12]. By analyzing the 'budget' and 'revenue' columns you can better understand the relationship between investment and financial returns in the film industry and identify patterns of returns and indications of potential success factors.

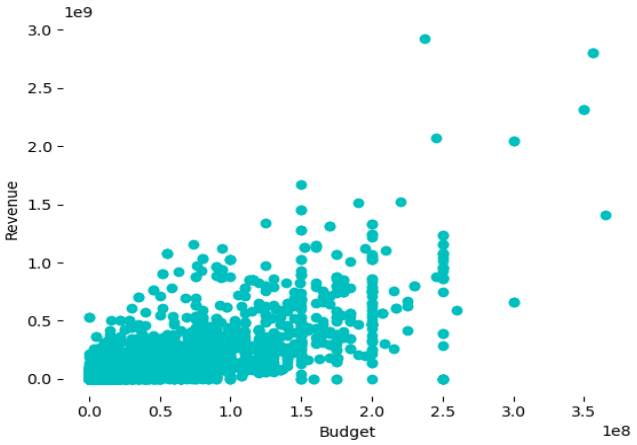

Figure 5: Comparison of Budget and Revenue (using Gathered Dataset)

Exploring the 'Year' distribution provides a snapshot of movie release trends over the years.

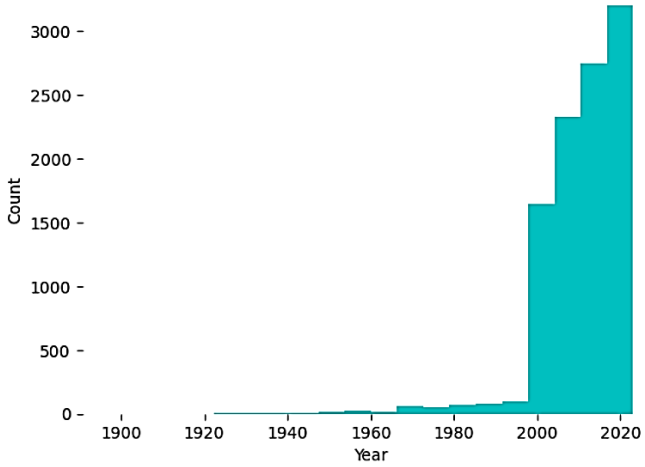

Figure 6: Year Distribution (using the Gathered Dataset)

The 'Runtime' column in our dataset represents the duration of movies. Analysis of the runtime distribution reveals a broad range of values, from a few minutes to many hours. The bulk of the films have runtimes that are centred on a particular period, and the distribution is roughly normal.

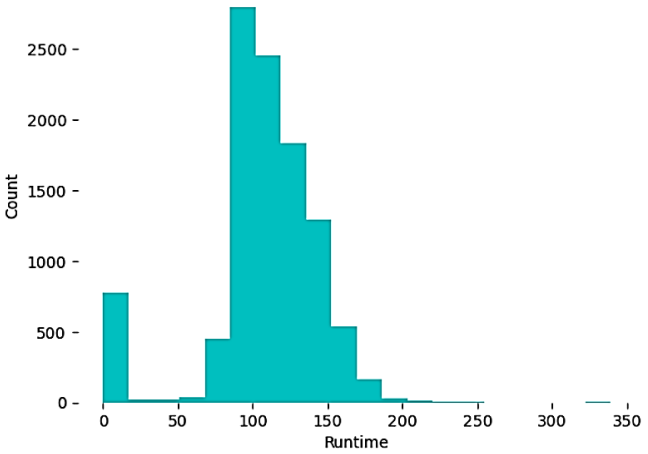

Figure 8: Runtime Distribution (using the Gathered Dataset)

Distribution analysis of movies' original language showcases the global linguistic diversity of the dataset, informing us about language preferences and their influence on the film industry. English is the most commonly utilised language in our database due to the large quantity of Hollywood films. Hindi is becoming increasingly widely represented, demonstrating the significance of Indian cinema. Furthermore, the dataset includes films in well-known languages such as Spanish, French, and Mandarin, demonstrating the breadth of available films. These discoveries shed light on the significance of specific languages and their impact on the global film landscape. Such insights are priceless when it comes to developing movie recommendation systems that can accommodate users' diverse language preferences.

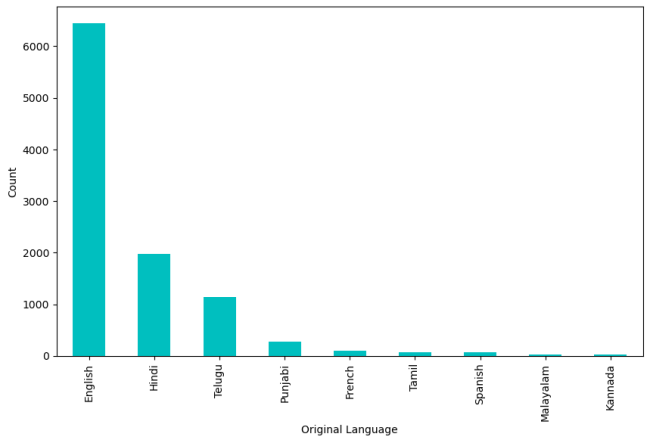

Figure 7: Distribution of Movies by Original Language (using the Gathered Dataset)

Furthermore, we explore correlations between attributes through correlation matrices, shedding light on relationships between elements like budget and income, runtime and popularity, or vote average and count. The correlation matrix, which illuminates numerical attribute correlations, is crucial to the movie recommendation system. The correlation matrix can show whether movie recommendation attributes are positively or negatively correlated. This data shows the relationships between budget, income, runtime, popularity, and vote average and count.

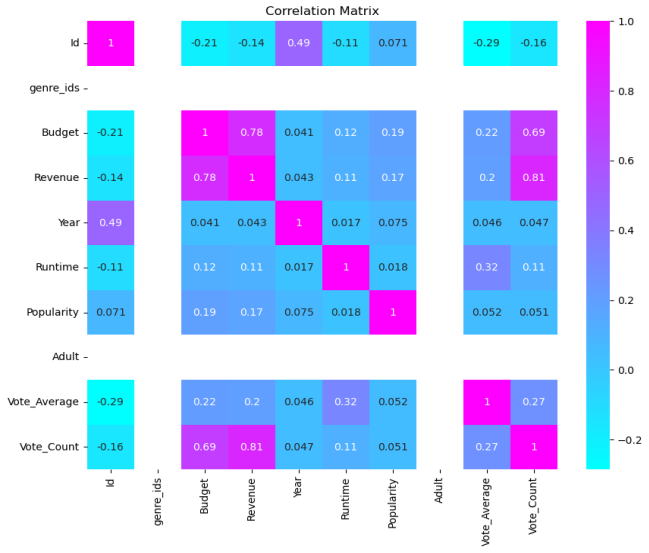

Figure 9: Correlation Matrix of Quantitative Columns

We carefully examined our dataset during the exploratory data analysis (EDA) phase and discovered several issues, such as missing or irrelevant records. We decided to remove certain variables that were deemed unnecessary for our recommendation system or had significant missing values to maintain the integrity and reliability of our dataset. We specifically removed genre_ids, Homepage, Adult, and Runtime variables.

We hoped to eliminate any potential biases or inconsistencies in our analysis by removing these variables. This was a critical step in ensuring the quality and reliability of our dataset for subsequent stages of analysis and modeling in the movie recommendation system. We were able to concentrate on the most significant elements of our dataset by removing these variables. By removing these variables, we were able to focus on the key attributes that are more relevant and informative for our recommendation.

Based on our study, we opted to focus on the labels most relevant to our movie recommendation processes, such as Title, Director, Artist, Genre, Overview, and Reviews.

	Id	Title	Director	Cast	Genre	Overview
0	391629	Baaghi	Sabbir Khan	['Tiger Shroff', 'Shraddha Kapoor', 'Sunil Gro...	Action, Thriller, Romance	Ronny is a rebellious man, who falls in love w...
1	25918	Champion	Mark Robson	['Kirk Douglas', 'Marilyn Maxwell', 'Arthur Ke...	Drama	An unscrupulous boxer fights his way to the to...
2	1104040	Gangs of Lagos	Jadesola Osiberu	['Demi Banwo', 'Adesua Etomi-Wellington', 'Tob...	Crime	A group of friends who each have to navigate t...
3	157800	Har Dil Jo Pyar Karega	Raj Kanwar	['Salman Khan', 'Rani Mukerji', 'Preity Zinta'...	Comedy, Drama	Raj is a struggling singer chasing his dreams ...
4	60579	Hey Ram	Kamal Haasan	['Kamal Haasan', 'Shah Rukh Khan', 'Hema Malin...	History, Drama, Crime	Saketh Ram's wife is raped and killed during d...
...	...	...	...	...	...	...
5586	560204	Arkansas	Clark Duke	['Liam Hemsworth', 'Clark Duke', 'Vince Vaughn...	Crime, Thriller	Kyle and Swin live by the orders of an Arkansa...
5587	19053	Valley Girl	Martha Coolidge	['Nicolas Cage', 'Deborah Foreman', 'Elizabeth...	Comedy, Romance	Julie, a girl from the valley, meets Randy, a ...
5588	429422	Capone	Josh Trank	['Tom Hardy', 'Linda Cardellini', 'Matt Dillon...	Crime, Drama	The 47-year old Al Capone, after 10 years in p...
5589	582596	The Wrong Missy	Tyler Spindel	['David Spade', 'Lauren Lapkus', 'Candace Smit...	Comedy, Romance	A guy meets the woman of his dreams and invite...
5590	385103	Scoob!	Tony Cervone	['Amanda Seyfried', 'Christina Hendricks', 'Fr...	Animation, Comedy, Family, Mystery	In Scooby-Doo's greatest adventure yet, see th...

Figure 10: Useful Data for Further Processing

Feature engineering techniques can be applied to a variety of attributes chosen for movie recommendation. Following that, we used feature engineering techniques [13] to improve the performance of movie recommendation systems.

These steps are as follows:

1) Editing line breaks: This step entails modifying the spaces in columns to ensure consistency in formatting. Standardizing the format makes data processing and analysis easier.

2) Cleaning the columns: The ‘reviews' column has been cleaned up by removing unnecessary lines and icons. This procedure aids in the removal of noise and irrelevant information, resulting in a more accurate representation of the movie reviews.

3) Combining multiple sources of information: To create tags, various sources of information such as 'cast, ‘reviews,' 'genre,' and 'director' are combined. These tags act as additional metadata and provide useful information about the movies, facilitating the recommendation process.

	movie_id	movie_title	Tags
0	391629	Baaghi	TigerShroff,ShraddhaKapoor,SunilGrover,Sudheer...
1	25918	Champion	KirkDouglas,MarilynMaxwell,ArthurKennedy,PaulS...
2	1104040	Gangs of Lagos	DemiBanwo,AdesuaEtomi-Wellington,TobiBakre,Ade...
3	157800	Har Dil Jo Pyar Karega	SalmanKhan,RaniMukerji,PreityZinta,NeerajVora,...
4	60579	Hey Ram	KamalHaasan,ShahRukhKhan,HemaMalini,RaniMukerj...
...	...	...	...
5549	560204	Arkansas	LiamHemsworth,ClarkDuke,VinceVaughn,JohnMalkov...
5550	19053	Valley Girl	NicolasCage,DeborahForeman,ElizabethDaily,Mich...
5551	429422	Capone	TomHardy,LindaCardellini,MattDillon,KyleMacLac...
5552	582596	The Wrong Missy	DavidSpade,LaurenLapkus,CandaceSmith,SarahChal...
5553	385103	Scoob!	AmandaSeyfried,ChristinaHendricks,FrankWelker,...

Figure 11: Combined Multiple Features Columns into a Single Tags Column

To ensure consistency and eliminate case-related discrepancies, text data is converted to lowercase.

```
0       tigershroff,shraddhakapoor,sunilgrover,sudheer...
1       kirkdouglas,marilynmaxwell,arthurkennedy,pauls...
2       demibanwo,adesuaetomi-wellington,tobibakre,ade...
3       salmankhan,ranimukerji,preityzinta,neerajvora,...
4       kamalhaasan,shahrukhkhan,hemamalini,ranimukerj...
                              ...
5549    liamhemsworth,clarkduke,vincevaughn,johnmalkov...
5550    nicolascage,deborahforeman,elizabethdaily,mich...
5551    tomhardy,lindacardellini,mattdillon,kylemaclac...
5552    davidspade,laurenlapkus,candacesmith,sarahchal...
5553    amandaseyfried,christinahendricks,frankwelker,...
Name: Tags, Length: 5554, dtype: object
```

Figure 12: Text Data of Tags Columns Converted into Small Letters

Overall, the proposed feature engineering methodology highlights the usefulness of NLP methods in movie recommendation systems [14] for information pre-processing. The system can give consumers reliable and relevant movie suggestions by extracting logical properties from the text.

D. Text Processing Techniques

Text processing is a fundamental facet of movie recommendation, enabling the extraction of meaningful information from movie descriptions, reviews, and other textual data. Here, we outline key techniques utilized in this phase.

1) Porter Stemming Algorithm: To normalize text data, the Porter stemming algorithm [15] is used. By converting words to their base forms, this algorithm reduces the dimensionality of the data. Normalization via stemming improves recommendation system accuracy by capturing the underlying meaning of words. The Porter stemming algorithm is a popular natural language processing (NLP) technique for normalizing words by reducing them to their base or root form. It contributes to the accuracy and effectiveness of text analysis tasks, such as movie recommendation systems. By removing common suffixes from words using a set of rules, the algorithm permits different spellings of the same word to be treated equally.

Input Word	***Rule Applied***	***Stemmed Word***
recommendation	Remove “-action”	recommend

Table 3: Porter Stemming Algorithm

Here's an example of how the Porter stemming algorithm works step by step: "recommendation" is the input word. Use stemming rules: "recommend" without the common suffix "-action", "recommend" is the resulting stemmed word. We use the Porter stemming algorithm to reduce the word "recommendation" to its simplest form, "recommend."

This process of normalization aids in capturing the essence of the word and enables the recommendation system to recognise and group similar words together. The Porter stemming algorithm was applied to the movie tags, which were combined into a single string.

By breaking down these words into their most basic forms, the system is able to recognise movies, actors, or directors with names that are similar and provide precise recommendations based on their textual characteristics.

Following that, we created a word cloud visualisation [16] to highlight the tags' most frequently occurring stemmed words. In the word cloud, which is a visual representation of word frequency, the size of each word correlates to its frequency. This allows us to identify the most frequently occurring themes or topics associated with the films.

The matplotlib library was used to create the word cloud, and the resulting graph was titled "Word Cloud of Stemmed Tags." The graph depicts the prominent stemmed words in a concise and visually appealing manner, allowing for a quick understanding of the key themes in the movie dataset.

Figure 13: Word Cloud of Stemmed Tag (using the Gathered Dataset)

2) Bag-of-Words Approach: We applied the Bag-of-Words (BoW) approach to process and analyse textual data [17] related to movie reviews. The BoW method is a widely used technique in natural language processing and text mining tasks.

To begin, we used the scikit-learn library's CountVectorizer module to convert the text data into a numerical representation. To do so, the text was tokenized into individual words or tokens, and a vocabulary of all unique words in the dataset was created.

Next, we constructed a matrix where each row represented a movie and each column represented a unique word from the vocabulary. The values in the matrix indicated the frequency of each word in the corresponding movie's review.

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

Figure 14: Matrix of Tags Column After Applying the Bad-of-Words Approach

This BoW matrix served as input for further analysis and modelling. It allowed us to capture the textual information of the movie reviews in a structured and quantitative manner, enabling us to apply machine learning algorithms for recommendation purposes. Using the BoW approach, we were able to extract important features from movie reviews and uncover patterns or relationships between words and movie preferences. This data was useful in understanding

user preferences, identifying similar movies, and making personalised recommendations.

3) Term Frequency-Inverse Document Frequency Approach: The TF-IDF (Term Frequency-Inverse Document Frequency) methodology is a popular method for analysing and processing textual data [18] in movie recommendation systems.

The TF-IDF technique is used in the movie recommendation system to extract relevant information and identify key words or phrases that indicate the substance of each movie from tag column.

The technique includes the following steps:

- The technique of dividing text data into individual words or tokens is known as tokenization.. This stage aids in the division of textual material into smaller parts for subsequent investigation.
- Calculation of phrase Frequency (TF): The frequency of each phrase in a movie's synopsis or review is computed. This metric indicates how frequently a phrase appears in a certain film.
- Inverse Document Frequency (IDF) calculation: The IDF score is calculated for each term, which measures the rarity or importance of a term across all movies in the dataset. Terms that occur frequently across all movies will have a lower IDF score, while terms that are unique to specific movies will have a higher IDF score.
- TF-IDF calculation: The TF-IDF score is obtained by multiplying the term frequency (TF) of a term in a movie by its inverse document frequency (IDF) across all movies. This score reflects the significance of the term within the specific movie as well as its distinctiveness in the entire dataset.
- Feature vector representation: The TF-IDF scores are used to create a feature vector representation [19] for each movie. This vector captures the importance of different terms in the movie's description or review, allowing for meaningful comparisons and similarity calculations between movies.

By applying the TF-IDF approach, the movie recommendation system can capture the unique characteristics and content of each movie, enabling more accurate matching and recommendation of movies based on their textual features. This technique enhances the system's ability to understand the context and relevance of movies, providing users with more personalized and relevant recommendations.

We created a TF-IDF feature matrix using the scikit-learn library's TfidfVectorizer. This matrix represents the importance of each term in the tags based on its frequency in a specific movie and its inverse frequency across all movies. We limited the maximum number of features to 10,000 and applied pre-processing steps such as stripping accents and removing English stop words. This TF-IDF feature matrix serves as a numerical representation of the movie tags, capturing the relevance and significance of each term for further analysis and modelling in the movie recommendation system.

```
(0, 9009)     0.1595812504194538
(0, 103)      0.12485551122180803
(0, 1385)     0.29969792201718726
(0, 550)      0.27192552583521773
(0, 5575)     0.28973911424829474
(0, 34)       0.3139748967967174
(0, 5191)     0.22324931049468436
(0, 5147)     0.18170894516209032
(0, 9947)     0.14180667436100566
(0, 7996)     0.2659927852709681
(0, 1541)     0.2613263528748404
(0, 5391)     0.12870257305114263
(0, 3218)     0.19050197357555532
(0, 5500)     0.13557492778525135
(0, 7252)     0.2920477916166908
(0, 5063)     0.34314534638028776
(0, 7815)     0.31042855893197113
(1, 4073)     0.40797957427004883
(1, 6595)     0.23103537230299978
(1, 3041)     0.38964160749800403
(1, 9703)     0.4230225834977054
(1, 3344)     0.40439553279861895
(1, 1030)     0.5346689520937824
(2, 6109)     0.4373596482532358
(2, 8579)     0.38981628203021773
:     :
```

Figure 15: Feature Matrix After Apply TF-IDF Approach

After creating the TF-IDF feature matrix, it can be used as input for machine learning algorithms to train models that can make movie recommendations based on user preferences. Based on the similarities between a user's tastes and the characteristics of various films, the trained models may then be used to forecast which films a user is most likely to appreciate.

E. Similarity Algorithms

After converting the text data into numerical representations, we can proceed with applying similarity algorithms. These algorithms measure the similarity between movies based on their numerical features, such as TF-IDF values or other derived representations [20]. We can discover which films are most similar to one another and provide recommendations based on those similarities by comparing them.

In our movie recommendation system, we considered multiple similarity algorithms, including cosine similarity, Euclidean distance, Jaccard similarity, [21] Tversky index, and Pearson correlation coefficient. However, for our specific implementation, we focused on utilizing cosine similarity and Euclidean distance. [22] These algorithms allow us to measure the similarity between movies based on their numerical features and help us identify the most similar movies for recommendation purposes.

a) Cosine Similarity: Cosine similarity is an important factor in measuring the similarity of films in movie recommendation systems. We computed the similarity scores between pairs of movies by applying the cosine similarity algorithm to the BoW and TF-IDF matrices.

The formula for cosine similarity can be expressed as:

$$cosine\_similarity\ (A, B) = (A \cdot B) / (||A|| * ||B||) \qquad (1)$$

Where:

- A and B are vectors representing two items or documents
- A and B are vectors representing two items or documents.
- (A · B) is the dot product of vectors A and B.

  ||A|| and ||B|| are the magnitudes (also known as the Euclidean norms) of vectors A and B, respectively.
- The dot product (A · B) is calculated by summing the element-wise multiplication of the corresponding elements in vectors A and B.
- The magnitude of a vector (||A|| or ||B||) can be computed as the square root of the sum of the squares of its elements.

Note that cosine similarity values range from -1 to 1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates perfect dissimilarity.

This improved the accuracy and relevance of movie recommendations by making it easier to identify movies with similar textual features. The combination of the BoW and TF-IDF approaches, [23] as well as cosine similarity, provided a strong framework for investigating textual similarity in movie recommendation systems.

Movies for 'The Witch' based on Cosine Similarity with BoW and TF-IDF:
We used the cosine similarity algorithm in conjunction with the Bag-of-Words (BoW) and TF-IDF approaches to recommend films similar to 'The Witch.' Based on the calculated similarity scores, we identified the top recommended films for 'The Witch.'

Cosine similarity with BoW:

```
Recommended movies for 'The Witch':
-----------------------------------
Get Out (Similarity: 0.5654563715919486)
Hereditary (Similarity: 0.4947602158954139)
A Quiet Place (Similarity: 0.4928602229664295)
You're Next (Similarity: 0.4896491402837018)
The Conjuring 2 (Similarity: 0.48679227477939746)
```

Figure 16: Prediction of Movies using Cosine Similarity Algorithm with BoW

The films are listed in descending order of similarity score. The film 'Get Out' has the highest similarity score of 0.5654563715919486, followed by 'Hereditary', 'A Quiet Place', 'You're Next', and 'The Conjuring 2'.

Cosine similarity with TF-IDF:

```
Recommended movies for 'The Witch':
-----------------------------------
Get Out (Similarity: 0.3009230723052134)
The Lighthouse (Similarity: 0.28781352919927045)
Doctor Strange (Similarity: 0.2585652739476842)
A Quiet Place (Similarity: 0.2560381540983614)
Hereditary (Similarity: 0.2549029384648677)
```

Figure 17: Prediction of Movies using Cosine Similarity Algorithm with TF-IDF

'Get Out' had the highest similarity score of 0.3009 among the recommended movies, followed by 'The Lighthouse' with a similarity score of 0.2878. 'Doctor Strange,' 'A Quiet Place,' and 'Hereditary' scores of 0.2586, 0.2560, and 0.2549, respectively.

The textual features of films were compared using the BoW and TF-IDF representations to generate these recommendations. The higher the similarity score, the closer the film is to 'The Witch' in terms of plot.

b) Euclidean distance: Using the Bag-of-Words (BoW) and TF-IDF Approaches with Euclidean Distance to Recommend Movies. We used Bag-of-Words (BoW) and TF-IDF (Term Frequency-Inverse Document Frequency) approaches with the Euclidean distance metric in our movie recommendation system. These techniques enabled us to convert textual data into numerical representations and compare the similarity of films.

Using the Bag-of-Words (BoW) and TF-IDF Approaches with Euclidean Distance to Recommend Movies. We used Bag-of-Words (BoW) and TF-IDF (Term Frequency-Inverse Document Frequency) approaches with the Euclidean distance metric in our movie recommendation system. These techniques enabled us to convert textual data into numerical representations and compare the similarity of films.

The formula for cosine similarity Approaches with Euclidean Distance to Recommend Movies. We used Bag-of-Words (BoW) and TF-IDF (Term Frequency-Inverse Document Frequency) approaches with the Euclidean distance metric in our movie recommendation system. These techniques enabled us to convert textual data into numerical representations and compare the similarity of films. Using the BoW method, we created a matrix of word frequencies in movie descriptions, with each movie represented as a vector. Similarly, we assigned weights to each term in the TF-IDF approach based on its importance in the corpus. The Euclidean distance between two films was then calculated to determine their similarity.

The Euclidean distance [24] equation calculates the distance between two points in a Euclidean space, which is a space with a fixed number of dimensions. In a two-dimensional space, the Euclidean distance between two points (x1, y1) and (x2, y2) can be calculated using the following formula:

$$d = sqrt((x2 - x1)^2 + (y2 - y1)^2) \quad (2)$$

In general, for a n-dimensional space, the Euclidean distance between two points (x1, x2, ..., xn) and (y1, y2, ..., yn) can be calculated as:

$$d = sqrt((x1 - y1)^2 + (x2 - y2)^2 + ... + (xn - yn)^2) \quad (3)$$

where, "d" represents the Euclidean distance between two points in a multi-dimensional space.
"x1, x2, ..., xn" are the coordinates of the first point in the n-dimensional space.

"y1, y2, ..., yn" are the coordinates of the second point in the same n-dimensional space.
The expression "(x1 - y1)^2" calculates the squared difference between the corresponding coordinates of the first and second points along the first dimension.
The expression "(x2 - y2)^2" calculates the squared difference between the corresponding coordinates of the first and second points along the second dimension, and so on for each dimension.
The sum of these squared differences across all dimensions is calculated, and then the square root is taken to obtain the Euclidean distance "d" between the two points.
The Euclidean distance equation measures the straight-line distance between two points in the space, considering all dimensions. It is a commonly used distance metric in various fields, including data science, machine learning, and image processing [25], to quantify the similarity or dissimilarity between data points.

By integrating these approaches with the Euclidean distance metric, we effectively captured the semantic similarity between movies and improved the accuracy of our movie recommendation system.

Euclidean distance with BoW:

```
Recommended movies using Euclidean Distance for 'The Witch':
-------------------------------------
Incarnate (Similarity: 26.888659319497503)
My Big Fat Greek Wedding 2 (Similarity: 27.640549922170507)
Fright Night (Similarity: 27.76688675382964)
102 Dalmatians (Similarity: 27.820855486487112)
Instant Family (Similarity: 27.85677655436824)
```

Figure 18: Prediction of Movies using Euclidean Distance Algorithm with BoW

A closer match to "The Witch" using Euclidean distance with BoW can be determined by a movie receiving a higher similarity score, which is used to choose the recommended films [26].

Euclidean distance with TF-IDF:

To recommend films similar to 'The Witch,' we used the TF-IDF approach combined with Euclidean Distance. The output displays the top recommended films as well as their similarity scores.

```
Recommended movies using Euclidean Distance for 'The Witch':
-------------------------------------
Keep Safe Distance (Similarity: 0.9999999999999998)
Zeher (Similarity: 0.9999999999999998)
Radhe (Similarity: 0.9999999999999998)
Charminar (Similarity: 0.9999999999999998)
Get Out (Similarity: 1.1824355607768113)
```

Figure 19: Prediction of Movies using Euclidean Distance Algorithm with TF-IDF

Based on how closely these films resemble "The Witch," they have been recommended, with greater similarity scores indicating a stronger match in terms of their textual elements. Due to their textual similarities [27], those who liked "The Witch" might find these suggested films interesting to watch.

IV. Implementation and Deployment

The implementation and deployment phase of the Pranik Movies Recommendation System involved its practical deployment on a production server or cloud infrastructure, ensuring accessibility for users. Scalability, reliability, and security were key considerations during the deployment process, with optimizations made for performance, load balancing, and resource allocation to ensure a seamless user experience, even during peak usage periods. This successful deployment enables users to access personalized movie recommendations based on their preferences, enhancing their overall movie-watching experience.

The Pranik Movies Recommendation System boasts a user-friendly interface that enables easy navigation and personalized movie recommendations based on user preferences. For instance, when a user searches for a specific movie, such as "Blade Runner 2049," the system retrieves relevant information and presents it in a movie card or a page of detailed results. The information includes the title, genre, release year, director, cast, rating, and other pertinent details. Additionally, the system may enhance the visual representation by incorporating images or movie posters. Alongside the searched movie, the system provides a list of suggested films that share similar genres, themes, or user preferences, accompanied by comprehensive information similar to the searched movie. Moreover, the integration of YouTube allows the system to display movie trailers, both for the searched movie and recommended movies, further enhancing the user experience and engagement. By including trailer videos, users can preview the visual and audio elements of the films, helping them gauge their interest and make informed decisions about their movie choices.

By combining detailed information about the searched movie, recommendations for similar movies, and the inclusion of trailer videos, the Pranik Movies Recommendation System [28] offers users a comprehensive and engaging experience. Users can easily access detailed information, explore recommended movies, and preview trailers, facilitating a more enjoyable and informed movie selection process.

V. Discussion and Future Work

The implementation and evaluation of the Pranik Movies Recommendation System have produced illuminating findings regarding its functionality and performance. By incorporating the TMDB API, Bag-of-Words (BoW), and TF-IDF approaches, the system has effectively improved the accuracy of recommendations and enhanced user experiences. The incorporation of surveys, interviews, and user interaction analyses has facilitated the evaluation of user satisfaction, thereby facilitating the identification of potential areas for improvement and advancements. Overall, the system has produced optimistic outcomes and received positive user feedback.

Several promising avenues exist for enhancing the Pranik Movies Recommendation System in the future. Advanced machine learning techniques, such as deep learning algorithms or neural networks [29], have the potential to reveal intricate patterns and interdependencies within the movie dataset. This innovation could result in more personalised and accurate recommendations.

Integration of user-generated content or data from social media platforms into the recommendation algorithm is another area of investigation. By leveraging user ratings, evaluations, and social connections data, the system could generate recommendations based on the community's collective experiences. This social dimension could improve the algorithm's ability to provide personalised recommendations [30].

Investigating the expansion of the recommendation system to include a broader variety of content types, such as television shows, documentaries, and streaming platforms, is also worthwhile. Adapting existing algorithms and data processing pipelines to accommodate diverse content formats, along with the incorporation of relevant metadata, would enable the system to serve a larger audience and provide exhaustive recommendations.

The evaluation of the Pranik Movies Recommendation System has produced encouraging results, demonstrating its design and implementation success. Extensive testing and analysis have validated the system's capacity to provide users with relevant and customised movie recommendations. Among the notable outcomes of the evaluation are:

Recommendation Accuracy: Using metrics such as precision, recall, and F1-score, the Pranik System has consistently demonstrated commendable recommendation accuracy. These metrics attest to the system's efficacy in delivering user-preferred suggestions.

User Satisfaction: User satisfaction surveys and feedback have been instrumental in determining the efficacy of the system. A significant proportion of users expressed high levels of contentment with the Pranik System's recommendations. This positive feedback validates the practical utility and user-centric design of the system.

Diverse Recommendations: The system's recommendations have demonstrated a remarkable level of diversity, providing users with a well-balanced selection of films encompassing various genres, languages, and release years. This variety contributes to an enhanced user experience that caters to various cinematic preferences.

The Pranik System has demonstrated its adaptability by accommodating changing user preferences. As users interact with the system and provide feedback, the recommendation engine improves the accuracy and relevance of its suggestions.

Scalability: The system's architecture demonstrates scalability by effectively administering a growing user base and expanding movie database while preserving optimal performance. This scalability ensures that the system is able to provide seamless and effective recommendations despite increased usage.

Implications for Interdisciplinarity Beyond its primary function of recommending films, the Pranik System has demonstrated potential for interdisciplinary applications. The methodologies utilised in the design of the system have the potential to influence a variety of disciplines, including e-commerce, content delivery, and personalised marketing.

VI. CONCLUSION

The Pranik Movies Recommendation System has demonstrated its prowess in generating precise and customised movie recommendations that correspond to the specific preferences of individual users. Our study's findings demonstrate the efficacy of the proposed methodology, which incorporates data preprocessing, feature engineering, text processing techniques, similarity algorithms, and rigorous evaluation. These accomplishments demonstrate the system's significance in expediting the movie selection process and enhancing user experiences.

The combination of the TMDB API and the Bag-of-Words (BoW) and TF-IDF approaches has produced commendable results in terms of recommendation precision and user satisfaction. The system's capacity to navigate through vast amounts of movie-related data, effectively process textual data, and recognise pertinent patterns demonstrates its practical utility. Positive user feedback reinforces the system's usefulness and contribution to the field of movie recommendations.

In addition, our evaluation results cast light on potential avenues for future enhancements. The incorporation of sophisticated machine learning techniques, such as deep learning algorithms and neural networks, has the potential to reveal complex relationships and improve the accuracy of recommendations. The incorporation of user-generated content and data from social media platforms has the potential to further personalise recommendations, aligning them with the community's collective intelligence.

Extending the scope of the system to include diverse content categories, such as television programmes and documentaries, provides an opportunity to reach a larger audience. The addition of additional metadata and the modification of existing algorithms would allow the system to provide suggestions that are comprehensive and context-aware. In addition, refining the evaluation framework will allow for more nuanced assessments, ensuring that the performance and impact of the system are continuously measured and enhanced.

In conclusion, the Pranik Movies Recommendation System exemplifies the successful integration of innovative methodologies and cutting-edge technologies in the field of movie recommendation. The tangible outcomes and user-centric design highlight the significance of the system in reducing decision fatigue and enhancing the cinematic experience for users. As we progress, the evolution of the system will be characterised by the incorporation of cutting-edge techniques, the leveraging of the power of social interactions, and the provision of diverse content preferences. This voyage of continuous improvement and innovation has the potential to make movie recommendation systems indispensable companions for all cinephiles.

REFERENCES

[1] Liaoliang Jiang, Yuting Cheng, Li Yang, Jing Li, Hongyang Yan and Xiaoqin Wang, “A trust-based collaborative filtering algorithm for e-commerce recommendation system,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 8, pp. 3023–3034, 2018. doi:10.1007/s12652-018-0928-7.

[2] K. Iwahama, Y. Hijikata, and S. Nishida, “Content-based filtering system for Music Data,” 2004 International Symposium on Applications and the Internet Workshops. 2004 Workshops., 2004. doi:10.1109/saintw.2004.1268677.

[3] H. Wang, P. Zhang, T. Lu, H. Gu, and N. Gu, “Hybrid recommendation model based on incremental collaborative filtering and content-based algorithms,” 2017 IEEE 21st International

Conference on Computer Supported Cooperative Work in Design (CSCWD), 2017. doi:10.1109/cscwd.2017.8066717.

[4] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, "Neural collaborative filtering," Proceedings of the 26th International Conference on World Wide Web, 2017. doi:10.1145/3038912.3052569

[5] P. Sharma and L. Yadav, "Movie recommendation system using item based collaborative filtering," International Journal of Innovative Research in Computer Science & Technology, vol. 8, no. 4, 2020. doi:10.21276/ijircst.2020.8.4.2.

[6] H. Zhang, M. Gan, and X. Sun, "Incorporating memory-based preferences and point-of-interest stickiness into recommendations in location-based social networks," ISPRS International Journal of Geo-Information, vol. 10, no. 1, p. 36, 2021. doi:10.3390/ijgi10010036.

[7] N. Tintarev and J. Masthoff, "A Survey of Explanations in Recommender Systems," 2007 IEEE 23rd International Conference on Data Engineering Workshop, Istanbul, Turkey, 2007, pp. 801-810, doi: 10.1109/ICDEW.2007.4401070.

[8] Rendle, S. (Year). Factorization Machines for Movie Recommendations. In Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM), pp. 713-722.

[9] B. Chen, "Data Collection and preprocessing," SpringerBriefs in Computer Science, pp. 5–16, 2022. doi:10.1007/978-981-19-7369-7_2.

[10] Camizuli, E. and Carranza, E.J. (2018) 'Exploratory Data Analysis (EDA)', The Encyclopedia of Archaeological Sciences, pp. 1–7. doi:10.1002/9781119188230.saseas0271.

[11] Camizuli, E. and Carranza, E. (2018) 'Exploratory Data Analysis (EDA)', The Encyclopedia of Archaeological Sciences, pp. 1–7. doi:10.1002/9781119188230.saseas0271.

[12] G. Gallavotti, F. Bonetto, and G. Gentile, "General qualitative properties," Aspects of Ergodic, Qualitative and Statistical Theory of Motion, pp. 1–26, 2004. doi:10.1007/978-3-662-05853-4_1.

[13] Dr. P. N., "Leukemia drug prediction using machine learning techniques with feature engineering," Journal of Advanced Research in Dynamical and Control Systems, vol. 12, no. SP4, pp. 141–146, 2020. doi:10.5373/jardcs/v12sp4/20201475.

[14] N. Kapoor, S. Vishal, and K. K. S., "Movie recommendation system using NLP Tools," 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020. doi:10.1109/icces48766.2020.9137993.

[15] A. Arif siswandi, Y. Permana, and A. Emarilis, "Stemming analysis indonesian language news text with Porter algorithm," Journal of Physics: Conference Series, vol. 1845, no. 1, p. 012019, 2021. doi:10.1088/1742-6596/1845/1/012019

[16] S. Lohmann, F. Heimerl, F. Bopp, M. Burch, and T. Ertl, "Concentri Cloud: Word cloud visualization for multiple text documents," 2015 19th International Conference on Information Visualisation, 2015. doi:10.1109/iv.2015.30.

[17] N. Passalis and A. Tefas, "Learning bag-of-embedded-words representations for textual information retrieval," Pattern Recognition, vol. 81, pp. 254–267, 2018. doi:10.1016/j.patcog.2018.04.008.

[18] H. Christian, M. P. Agus, and D. Suhartono, "Single Document Automatic text summarization using term frequency-inverse document frequency (TF-IDF)," ComTech: Computer, Mathematics and Engineering Applications, vol. 7, no. 4, p. 285, 2016. doi:10.21512/comtech.v7i4.3746.

[19] X. Pan, J. Cheng, Y. Xia, X. Zhang, and H. Wang, "Which feature is better? TF*IDF feature or topic feature in text clustering," 2012 Fourth International Conference on Multimedia Information Networking and Security, 2012. doi:10.1109/mines.2012.249.

[20] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "Netflix recommendation system based on TF-IDF and cosine similarity algorithms," Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning, 2021. doi:10.5220/0010727500003101.

[21] J. M. Hancock, "Jaccard distance (Jaccard Index, Jaccard similarity coefficient)," Dictionary of Bioinformatics and Computational Biology, 2004. doi:10.1002/9780471650126.dob0956.

[22] M. Kryszkiewicz, "The cosine similarity in terms of the Euclidean distance," Encyclopedia of Business Analytics and Optimization, pp. 2498–2508, 2014. doi:10.4018/978-1-4666-5202-6.ch223.

[23] M. Alodadi and V. P. Janeja, "Similarity in patient support forums using TF-IDF and cosine similarity metrics," 2015 International Conference on Healthcare Informatics, 2015. doi:10.1109/ichi.2015.99.

[24] A. Y. Alfakih, "Euclidean distance matrices (EDMS)," Euclidean Distance Matrices and Their Applications in Rigidity Theory, pp. 51–87, 2018. doi:10.1007/978-3-319-97846-8_3.

[25] Adate, A. and Tripathy, B.K. (2018) '3. Deep Learning techniques for Image Processing', *Machine Learning for Big Data Analysis*, pp. 69–90. doi:10.1515/9783110551433-003.

[26] Adate, A. and Tripathy, B.K. (2018) '3. Deep Learning techniques for Image Processing', Machine Learning for Big Data Analysis, pp. 69–90. doi:10.1515/9783110551433-003.

[27] Besancon, R., Rajman, M. and Chappelier, J.-C. (1999) 'Textual similarities based on a distributional approach', Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99 [Preprint]. doi:10.1109/dexa.1999.795163.

[28] Prajna, K.B. *et al.* (2022) 'Implementation of a hybrid recommendation system for Movies', *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)* [Preprint]. doi:10.1109/mysurucon55714.2022.9972580.

[29] Aggarwal, K. et al. (2022) 'Has the future started? the current growth of artificial intelligence, Machine Learning, and Deep Learning', Iraqi Journal for Computer Science and Mathematics, pp. 115–123. doi:10.52866/ijcsm.2022.01.01.013.

[30] Wenzel, T. et al. (2022) 'Providing personalised recommendations of critical incident narratives in a cross-platform mobile application', Proceedings of the 6th International Conference on Computer-Human Interaction Research and Applications [Preprint]. doi:10.5220/0011528400003323.

'Declarations'

Availability of data and materials- Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Competing interests- The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding- No funding has been received for this work.

Acknowledgements- NA