THE ULTIMATE RECOMMENDATION SYSTEM: PRANIK MOVIES

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted by

NIKHIL KUMAR
(2200290140098)
MAYANK SHARMA
(2200290140089)

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Amit Kumar
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(FEBRUARY 2024)

CERTIFICATE

Certified that **Nikhil Kumar 2200290140098, Mayank Sharma 2200290140089** have carried out the project work having "**The ultimate recommendation system: Pranik Movies**" (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date: 24/02/2024

Nikhil Kumar (2200290140098)
Mayank Sharma (2200290140089)
……
……

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 24/02/2024

Dr. Amit Kumar
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

This project report details the creation and execution of Pranik Movies, a sophisticated recommendation system tailored to offer personalised movie recommendations in the age of excessive information. The system uses collaborative and content-based filtering methods, incorporating machine learning algorithms to examine user behaviours, ratings, and viewing histories. The paper offers a thorough examination of the research framework, encompassing system architecture, data pre-processing, feature engineering techniques, and model selection and design.

Text processing techniques like stemming, bag-of-words (BoW), and TF-IDF are used to analyse textual movie data. The system improves recommendation accuracy by evaluating film similarities through algorithms such as cosine similarity and Euclidean distance. The research concludes by proposing future directions that focus on exploring advanced machine learning techniques, integrating social media, enhancing content support, and refining the evaluation framework.

Pranik Movies is an innovative recommendation system that provides personalised and accurate movie suggestions in the extensive and varied world of cinema. The study contributes to the continuous development of recommendation systems, focusing on the difficulties presented by the increasing content options and the requirement for personalised recommendations in the current rapidly changing digital landscape.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Dr. Amit Kumar** for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Nikhil Kumar

Mayank Sharma

…….....

…….....

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the fast-paced and ever-changing landscape of today's world, we face a constant influx of decisions that inundate us with a multitude of choices daily. This phenomenon, referred to as "decision fatigue," arises from the overwhelming abundance of options available, spanning from selecting our evening meals to choosing books and films for leisure. This challenge is further intensified by the deluge of information characterizing our present era. Within this context, recommendation systems become essential tools. Crafted with careful precision, these systems mitigate decision fatigue by assisting users in identifying items, features, or products that resonate with their preferences and past interactions.

1.1 OVERVIEW

The process of choosing a movie has become a challenging task, involving 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 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.

1.2 THE DYNAMIC LANDSCAPE OF RECOMMENDATION SYSTEMS

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. 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 utilized data for recommendation systems, showcasing how different platforms utilize a variety of strategies to improve user experiences. This table becomes a compass for comprehending the multifaceted approaches adopted by various platforms to recommend content to their users when viewed in the context of the proposed Pranik System.

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

Table. 1.2.1 Platform Specification Information

CHAPTER 2

COMPREHENSIVE WORK

In the field of movie recommendation systems, several noteworthy 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. This algorithm incorporates trust relationships among users to enhance the accuracy and reliability of recommendations in E-commerce systems, estimating user trustworthiness through a trust propagation method based on historical behavior and interactions. By considering both item similarity and user trust, personalized recommendations outperform other collaborative filtering approaches in terms of recommendation accuracy.

Collaborative filtering, a widely utilized practice by market leaders like Netflix and Amazon, plays a crucial role in recommendation systems. Content-based filtering, another tried-and-true method in recommending films, analyzes the qualities of items such as genre, cast, director, and storyline summaries to suggest comparable movies.

Within the realm of recommendation systems, content-based filtering centers its attention on the qualities of the objects under consideration. K. Iwahama, Y. Hijikata, and S. Nishida developed a content-based filtering system for music data, creating a recommendation system suggesting music based on the content analysis of songs. Their research focuses on the creation of a recommendation algorithm based on item profiles, serving as a foundation for later content-based recommendation systems.

Recognizing the potential for improved accuracy, hybrid recommendation systems integrate various methods. H. Wang, P. Zhang, T. Lu, H. Gu, and N. Gu proposed a hybrid model combining incremental collaborative filtering with content-based algorithms to enhance the efficiency and precision of recommendation systems. This methodology incorporates both collaborative filtering and content-based filtering to generate more accurate and diversified recommendations.

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 proposed a neural collaborative filtering model combining matrix factorization and neural networks to capture both user-item interactions and item-item similarities.

P. Sharma and L. Yadav discuss a movie recommendation system based on item-based collaborative filtering, focusing on improving recommendation accuracy by considering the similarities between movie items.

H. Zhang, M. Gan, and X. Sun discuss the challenges of movie recommendation in location-based social networks and propose an approach incorporating memory-based preferences and point-of-interest stickiness to improve the accuracy and effectiveness of recommendations.

Emphasizing explainable recommendations, N. Tintarev and J. Masthof 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 Stefen Rendle. This work introduces the concept of Factorization Machines (FM) as a powerful tool for recommendation tasks, excelling in capturing interactions between categorical variables, making them suitable for recommendation scenarios where user-item interactions are prevalent.

This study introduces the Pranik Movies recommendation system, incorporating innovative components. Collaborative and content-based filtering provides user-specific movie suggestions, with machine learning algorithms analyzing user activity, ratings, and watching history. Advanced text processing methods like stemming enhance movie text analysis, and similarity algorithms improve movie suggestions. The study framework covers system architecture, data pre-processing, feature engineering, model selection, and design, along with implementation, deployment, user interface functionality, and assessment metrics for a comprehensive understanding of the system.

CHAPTER 3

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.

3.1 ARCHITECTURE

The architecture of our movie recommendation system is depicted in Fig. 3.1.1, illustrating the fow of methods and processes that contribute to delivering accurate and personalized movie recommendations.

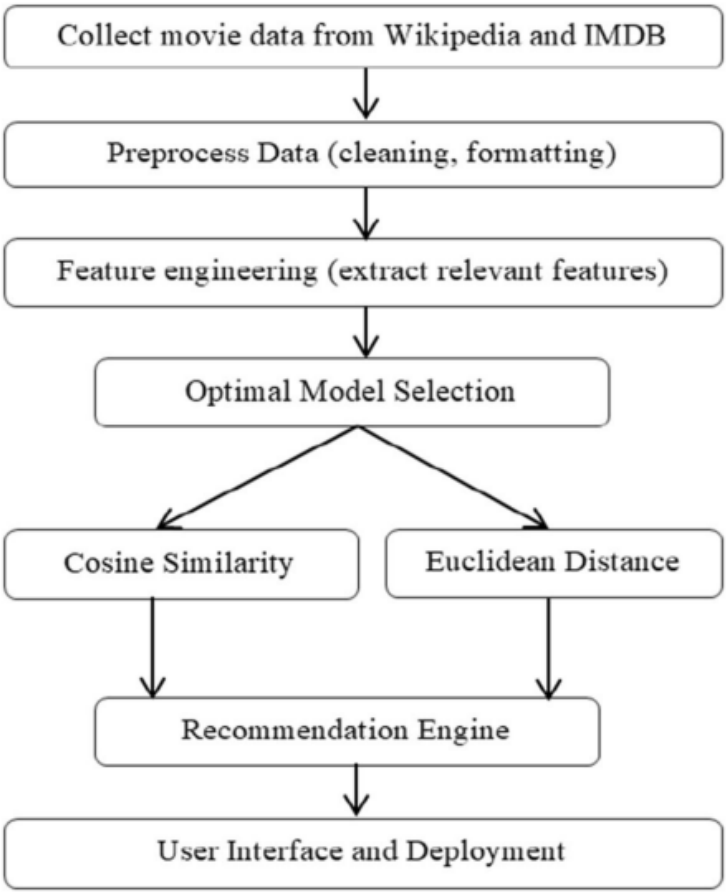

Fig. 3.1.1 Flowchart of Proposed Methods

3.2 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 (Fig. 3.2.1).

```
<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
```

Fig. 3.2.1 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 (Table 3.2.1).

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 identifers for movie genres
Genre	A specifc genre classification
Budget	The fnancial 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 ofcial 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. 3.2.1 Essential variables encompassed by the dataset

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

3.3 EXPLORATORY DATA ANALYSIS (EDA):

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

Variable Distributions Exploration: First, we examine the variable distributions, looking for patterns using statistical methods and graphical representations like histograms, box plots, scatter plots, 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 (Fig. 3.3.1).

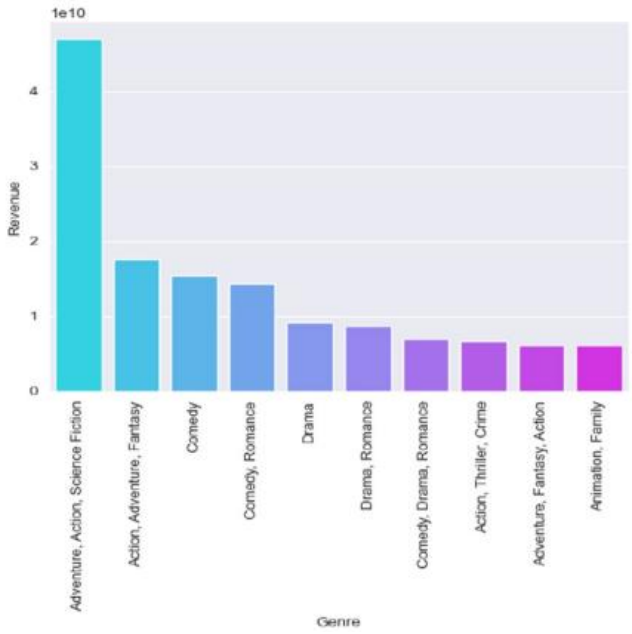

Fig. 3.3.1 Top 10 Genres by Revenue in the Dataset

Insights into Quantitative Attributes: Further insights into movies' quantitative attributes were gained by examining statistical labels. These visualizations unveil distributions, central tendencies, and variability (Fig. 3.3.2).

	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

Fig. 3.3.2 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. 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 (Fig. 3.3.3)

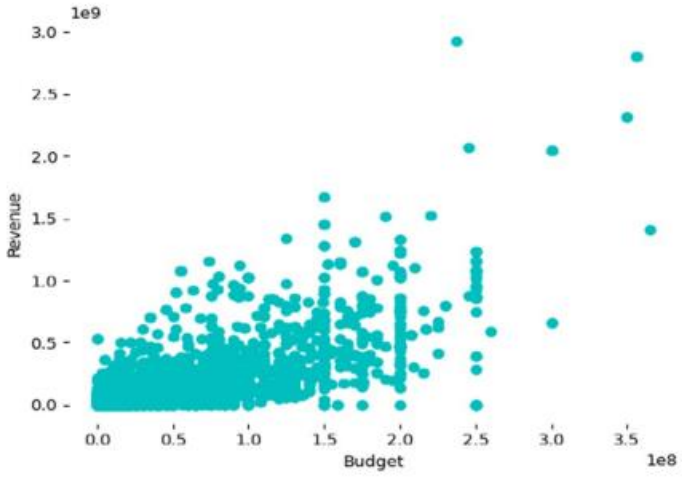

Fig. 3.3.3 Comparison of Budget and Revenue (using Gathered Dataset)

Exploring the 'Year' distribution provides a snapshot of movie release trends over the years (Fig. 3.3.4).

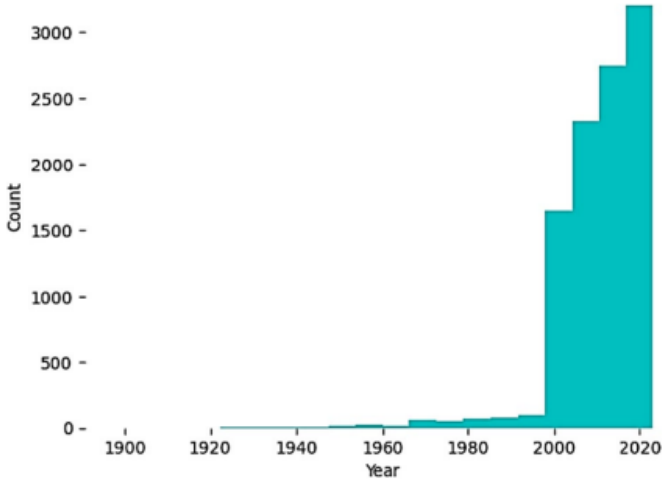

Fig. 3.3.4 Year Distribution (using the Gathered Dataset)

Runtime Distribution Analysis: 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 centered on a particular period, and the distribution is roughly normal (Fig. 3.3.5).

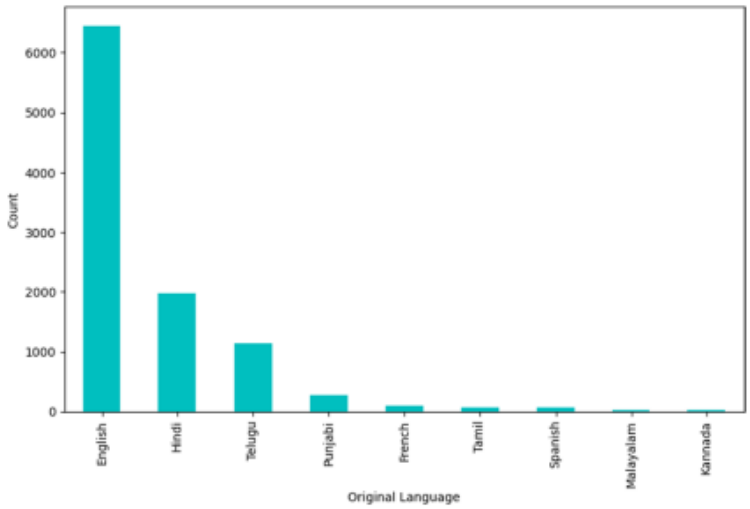

Fig. 3.3.5 Distribution of Movies by Original Language (using the Gathered Dataset)

Language Diversity and Global Impact: 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 utilized 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 (Fig. 3.3.6).

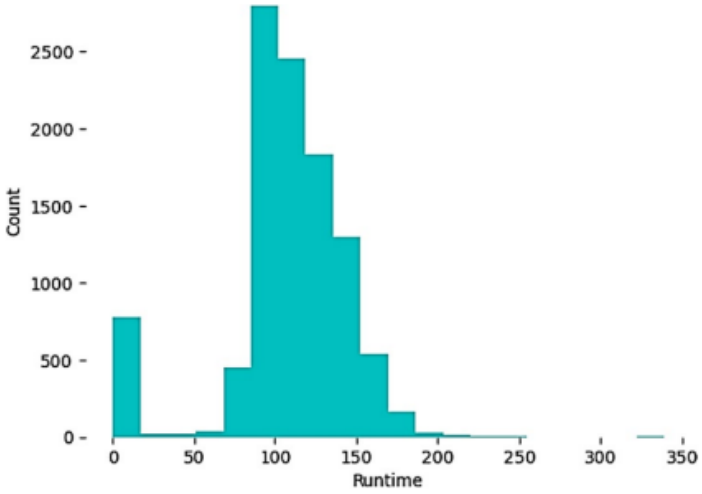

Fig. 3.3.6 Runtime Distribution (using the Gathered Dataset)

Correlation Matrix Exploration: 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, vote average, and count (Fig. 3.3.7).

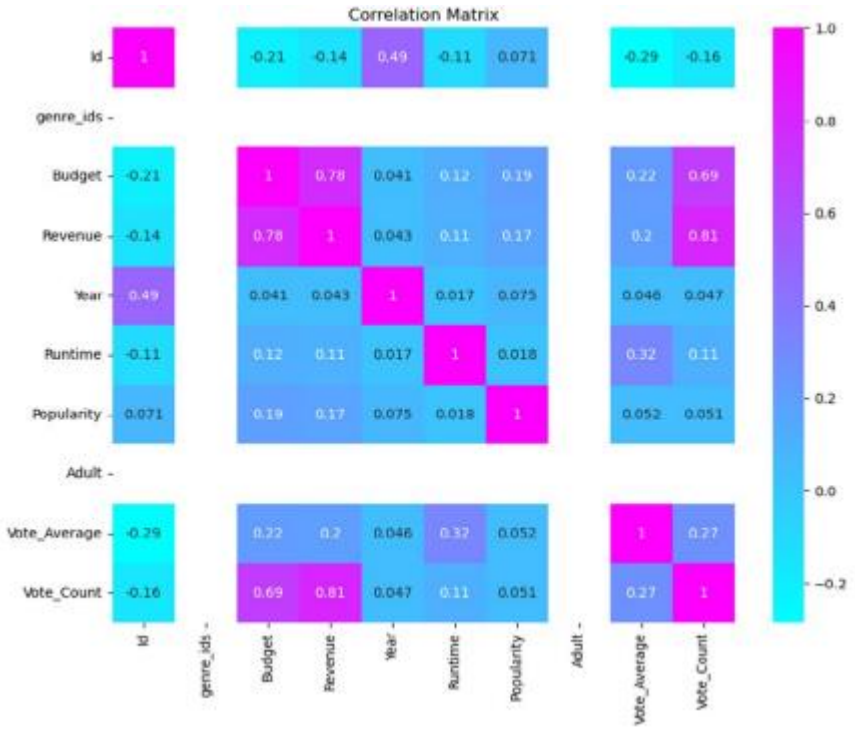

Fig. 3.3.7 Correlation Matrix of Quantitative Columns

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 (Fig. 3.3.8)

	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...
...	...	...	...	...	...	...
5553	560204	Arkansas	Clark Duke	['Liam Hemsworth', 'Clark Duke', 'Vince Vaughn...	Crime, Thriller	Kyle and Swin live by the orders of an Arkansa...
5554	19053	Valley Girl	Martha Coolidge	['Nicolas Cage', 'Deborah Foreman', 'Elizabeth...	Comedy, Romance	Julie, a girl from the valley, meets Randy, a ...
5555	429422	Capone	Josh Trank	['Tom Hardy', 'Linda Cardellini', 'Matt Dillon...	Crime, Drama	The 47-year old Al Capone, after 10 years in p...
5556	582596	The Wrong Missy	Tyler Spindel	['David Spade', 'Lauren Lapkus', 'Candace Smit...	Comedy, Romance	A guy meets the woman of his dreams and invite...
5557	385103	Scoob!	Tony Cervone	['Amanda Seyfried', 'Christina Hendricks', 'Fr...	Animation, Comedy, Family, Mystery	In Scooby-Doo's greatest adventure yet, see th...

5558 rows × 8 columns

Fig. 3.3.8 Useful Data for Further Processing

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

These steps are as follows:

- 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.
- 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.
- 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 (Fig. 3.3.9).

	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,...

5554 rows × 3 columns

Fig. 3.3.9 Combined Multiple Features Columns into a Single Tags Column

- To ensure consistency and eliminate case-related discrepancies, text data is converted to lowercase (Fig. 3.3.10).

	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,...

5554 rows × 3 columns

Fig. 3.3.10 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 for information pre-processing. The system can give consumers reliable and relevant movie suggestions by extracting logical properties from the text.

2.2.2. 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.

Porter Stemming Algorithm: To normalize text data, the Porter stemming algorithm 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 (Fig 3.3.11).

Input Word	*Rule Applied*	*Stemmed Word*
recommendation	Remove "-action"	recommend

Fig. 3.3.11 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 recognize 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 can recognize 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 visualization 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 (Fig. 3.3.12).

Fig. 3.3.12 Combined Multiple Features Columns into a Single Tags Column

Bag-of-Words Approach: We applied the Bag-of-Words (BoW) approach to process and analyze textual data 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 sci-kit-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 (Fig. 3.3.13).

```
[[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]]
```

Fig. 3.3.13 Matrix of Tags Column After Applying the Bad-ofWords Approach

This BoW matrix served as input for further analysis and modeling. 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 personalized recommendations.

Term Frequency-Inverse Document Frequency Approach: The TF-IDF (Term Frequency-Inverse Document Frequency) methodology is a popular method for analyzing and processing textual data in movie recommendation systems. The TF-IDF technique is used in the movie recommendation system to extract relevant information and identify keywords or phrases that indicate the substance of each movie from the 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 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.

3.5 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. We can discover which films are most similar to one another and provide recommendations based on those similarities by comparing them.

```
(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
:     :
```

Fig. 3.3.14 Feature Matrix After Apply TF-IDF Approach

In our movie recommendation system, we considered multiple similarity algorithms, including cosine similarity, Euclidean distance, Jaccard similarity, Tversky index, and Pearson correlation coefficient. However, for our specific implementation, we focused on utilizing cosine similarity and Euclidean distance. 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.

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 Bag-of-Words (BoW) and TF-IDF matrices.

The formula for cosine similarity can be expressed as:

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

Where:

- A and B are vectors representing two items or documents.

- ||A|| and ||B|| are the magnitudes (also known as the Euclidean norms) of vectors A and B, respectively.

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, 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: The films are listed in descending order of similarity score. The film 'Get Out' has the highest similarity score of 0.5654, followed by 'Hereditary,' 'A Quiet Place,' 'You're Next,' and 'The Conjuring 2' (Fig. 3.3.15).

```
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)
```

Fig. 3.3.15 Prediction of Movies using Cosine Similarity Algorithm with BoW

Cosine Similarity 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' scored 0.2586, 0.2560, and 0.2549, respectively (Fig. 3.3.16).

```
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)
```

Fig. 3.3.16 Prediction of Movies using Cosine Similarity Algorithm with TF-IDF

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.

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.

The Euclidean distance equation calculates the distance between two points in an 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:

$$d = sqrt\big((x2 - x1)\text{^}2 + (y2 - y1)\text{^}2\big)$$

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

$$d = sqrt\big((x1 - y1)\text{^}2 \ + \ (x2 - y2)\text{^}2 \ + \ \ldots \ + \ (xn - yn)\text{^}2\big)$$

The Euclidean distance equation measures the straight-line distance between two points in space, considering all dimensions. It is a commonly used distance metric in various fields, including data science, machine learning, and image processing, 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: 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 (Fig. 3.3.17).

```
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)
```

Fig. 3.3.17 Prediction of Movies using Euclidean Distance Algorithm with BoW

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 (Fig. 3.3.18).

```
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)
```

Fig. 3.3.18 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, those who liked "The Witch" might find these suggested films interesting to watch.

CHAPTER 4

IMPLEMENTATION AND DEPLOYMENT

The Pranik Movies Recommendation System underwent the practical implementation and deployment phase, where it was deployed on a production server or cloud infrastructure to ensure user accessibility. Considerations for scalability, reliability, and security were paramount during the deployment process. Optimizations were implemented for performance, load balancing, and resource allocation to guarantee a seamless user experience, even during peak usage periods. The successful deployment now allows users to access personalized movie recommendations based on their preferences, enriching their overall movie-watching experience.

The user-friendly interface of the Pranik Movies Recommendation System ensures easy navigation and delivers personalized movie recommendations aligned with 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 detailed results page. Information provided 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 furnishes a list of suggested films that share similar genres, themes, or user preferences, accompanied by comprehensive information similar to the searched movie. The integration of YouTube allows the system to display movie trailers, both for the searched movies and recommended movies, further enriching the user experience and engagement. With the inclusion of trailer videos, users can preview the visual and audio elements of the films, aiding them in gauging their interest and making informed decisions about their movie choices.

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

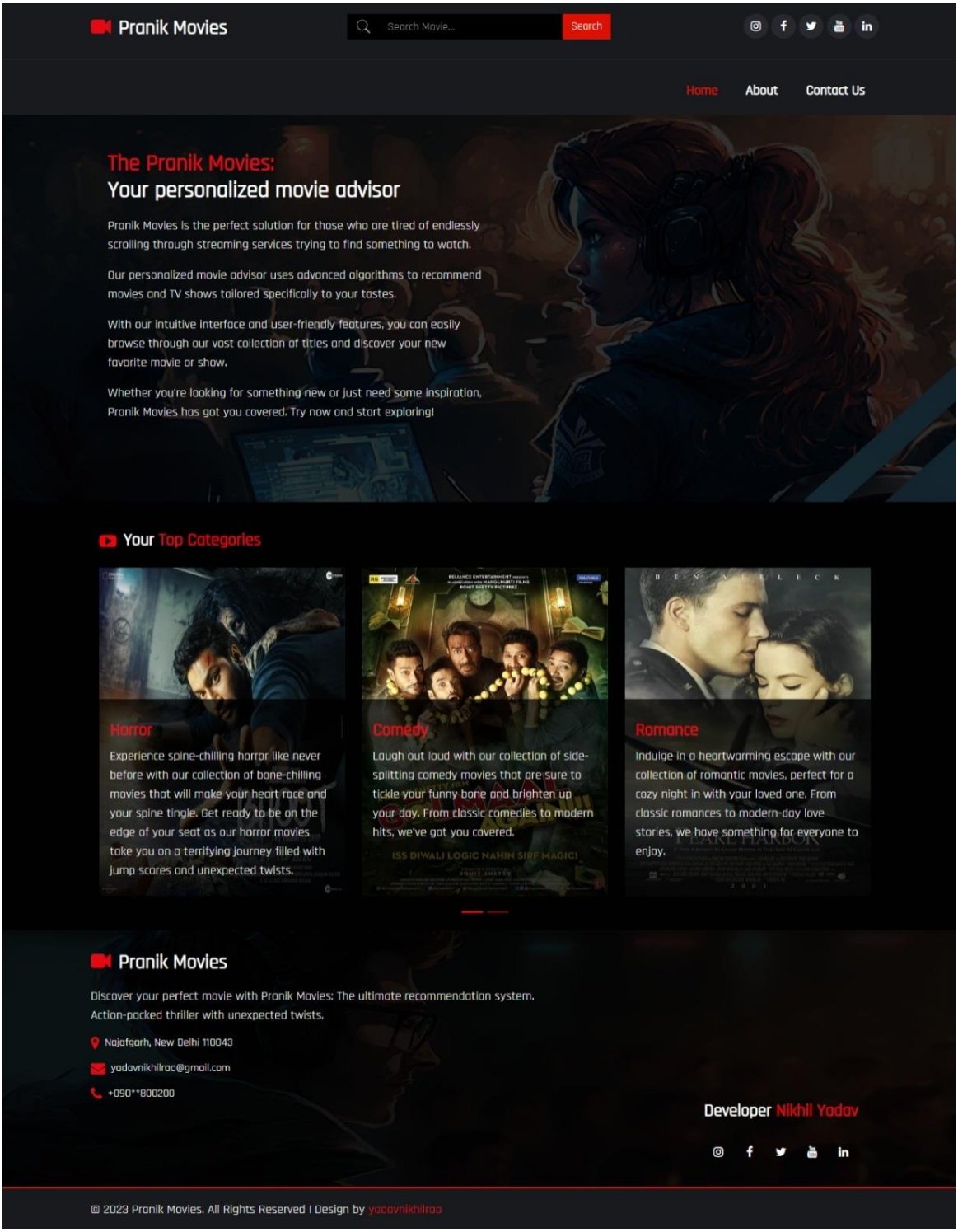

Fig. 4.1 Home Page

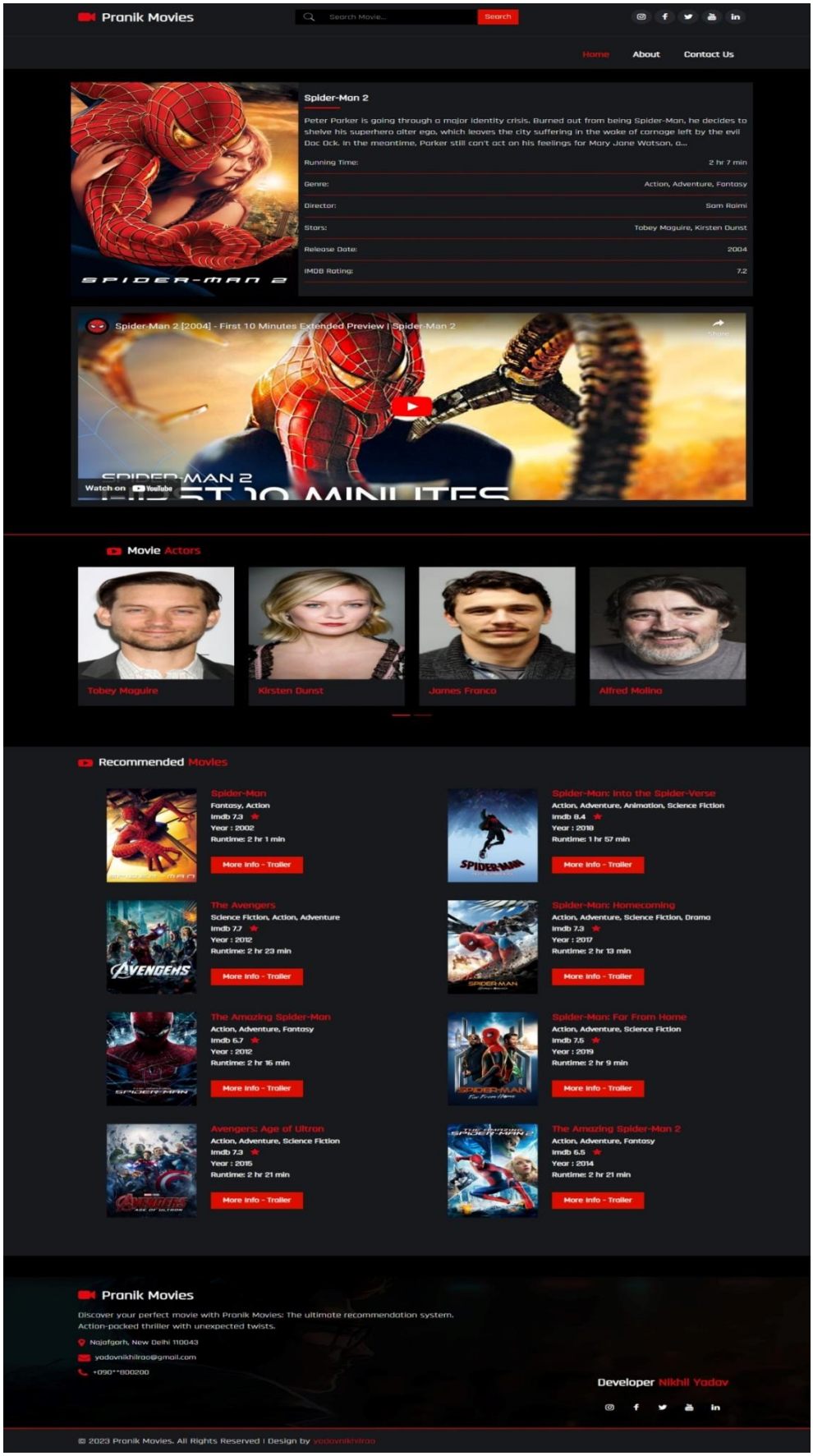

Fig. 4.2 Recommended Movies

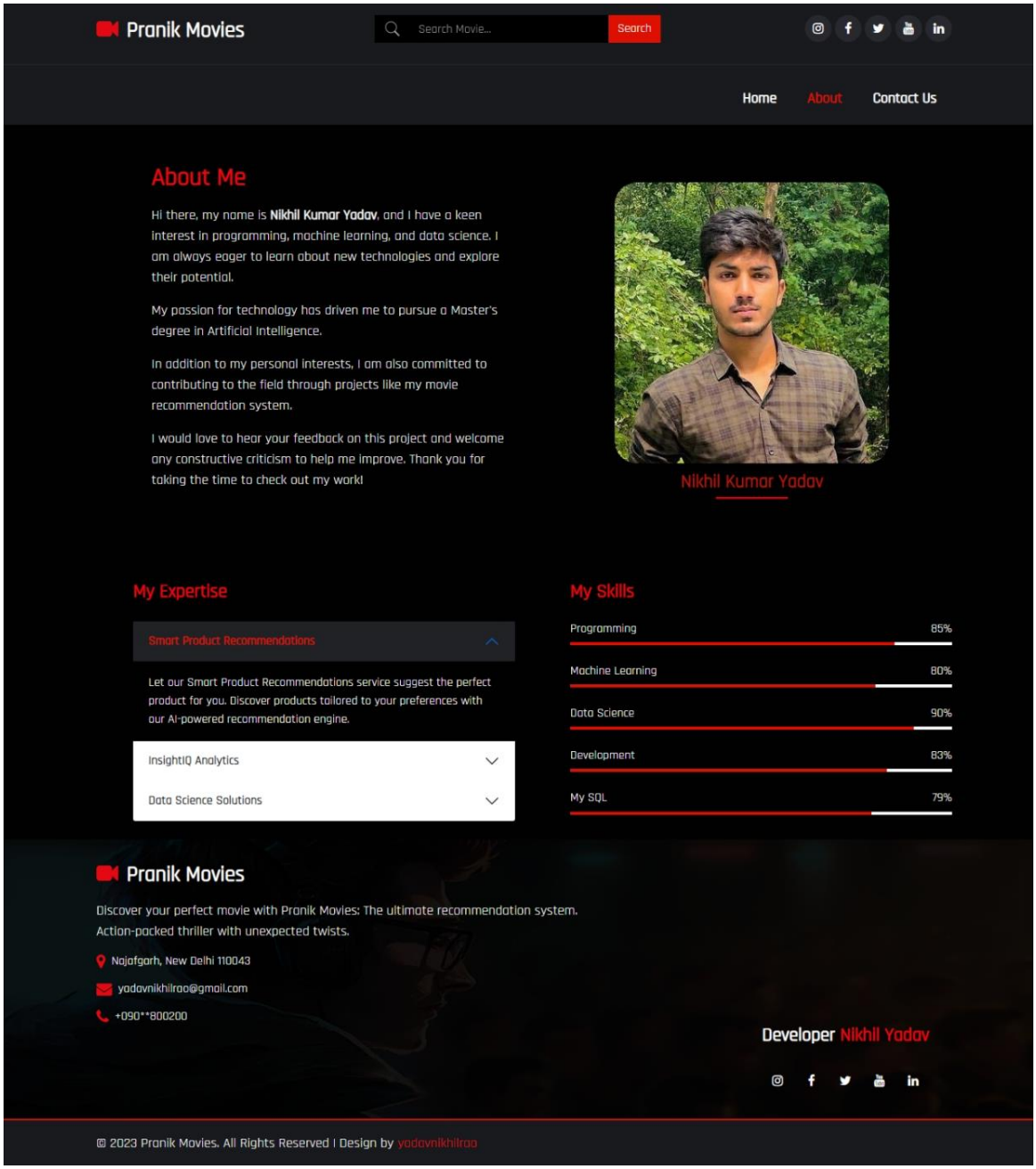

Fig. 4.3 About Page

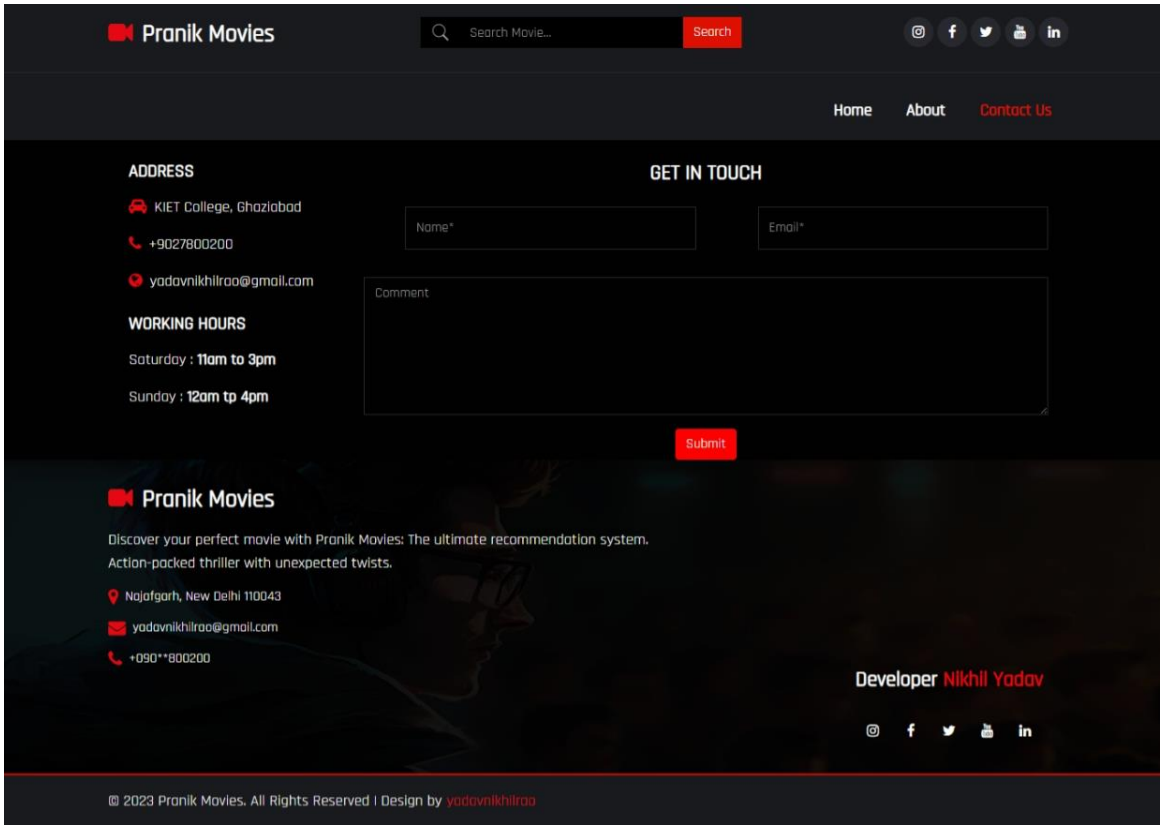

Fig. 4.4 Contact Us Page

CHAPTER 5

DISCUSSION AND FUTURE WORK

The evaluation and implementation of the Pranik Movies Recommendation System have yielded insightful findings regarding its functionality and performance. By integrating the TMDB API, Bag-of-Words (BoW), and TF-IDF approaches, the system has effectively enhanced recommendation accuracy and improved user experiences. Evaluation methods, such as surveys, interviews, and user interaction analyses, have contributed to assessing user satisfaction and identifying areas for potential improvements. Overall, the system has demonstrated optimistic outcomes and garnered positive user feedback.

Numerous promising avenues exist for future enhancements of the Pranik Movies Recommendation System. Exploring advanced machine learning techniques, such as deep learning algorithms or neural networks, holds potential for uncovering intricate patterns within the movie dataset, resulting in more personalized and accurate recommendations.

Another area of investigation involves the integration of user-generated content or data from social media platforms into the recommendation algorithm. By incorporating user ratings, evaluations, and social connections data, the system could generate recommendations based on the community's collective experiences, thereby enhancing the algorithm's ability to provide personalized suggestions. Expanding the recommendation system to include a broader variety of content types, such as television shows, documentaries, and streaming platforms, is also a worthwhile endeavor.

The evaluation of the Pranik Movies Recommendation System has produced encouraging results, validating its design and implementation success. Notable outcomes include commendable recommendation accuracy, high levels of user satisfaction, and diverse recommendations spanning various genres, languages, and release years.

The Pranik System has demonstrated adaptability by improving recommendation accuracy based on user feedback, contributing to its practical utility. The system's architecture exhibits scalability, effectively handling a growing user base and expanding the movie database while maintaining optimal performance.

Implications for Interdisciplinarity: Beyond its primary function of recommending films, the Pranik System shows potential for interdisciplinary applications. The methodologies employed in its design could influence various disciplines, including e-commerce, content delivery, and personalized marketing.

CHAPTER 6

CONCLUSION

The combination of the TMDB API and Bag-of-Words (BoW) and TF-IDF approaches has proven successful, demonstrating commendable results in recommendation precision and user satisfaction. The system's capability to navigate extensive movie-related data, process textual information effectively, and recognize relevant patterns showcases its practical utility. Positive user feedback reinforces the system's value and contribution to the field of movie recommendations.

Furthermore, the evaluation results shed light on potential avenues for future enhancements, such as incorporating advanced machine learning techniques and leveraging user-generated content and social media data. Expanding the system's scope to encompass diverse content categories and refining the evaluation framework are identified as opportunities for continuous improvement.

In conclusion, the Pranik Movies Recommendation System exemplifies the successful integration of innovative methodologies and cutting-edge technologies in the field of movie recommendations. The tangible outcomes and user-centric design underscore the system's importance in reducing decision fatigue and enhancing the cinematic experience. The ongoing evolution of the system holds the potential to make movie recommendation systems indispensable companions for all cinephiles.

CHAPTER 7

BIBLIOGRAPHY

1. Iwahama K, Hijikata Y, Nishida S (2004) "Content-based filtering system for Music Data," 2004 International Symposium on Applications and the Internet Workshops. 2004 Workshops. doi:https://doi. org/10.1109/saintw.2004.1268677

2. Tintarev N, Masthof J (2007) A Survey of Explanations in Recommender Systems. 2007 IEEE 23rd International Conference on Data Engineering Workshop, Istanbul, Turkey, pp. 801–810, doi: https://doi.org/10.1109/ICDEW.2007.4401070

3. Rendle S (2010) Factorization machines. In: 2010 IEEE International Conference on Data Mining. IEEE Sydney, NSW, Australia, pp 995–1000. https://doi.org/10.1109/ICDM.2010.127

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

5. Wang H, Zhang P, Lu T, Gu H, Gu N (2017) "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). doi:https://doi.org/10.1109/cscwd.2017.8066717

6. Zhang H, Gan M, Sun X (2021) Incorporating memory-based preferences and point-of-interest stickiness into recommendations in location-based social networks. ISPRS Int J Geo Inf 10(1):36. https://doi.org/10.3390/ijgi10010036

7. Sharma P, Yadav L (2020) Movie recommendation system using item based collaborative filtering. Int J Innov Res Comput Sci Technol 8(4). doi:https://doi.org/10.21276/ijircst.2020.8.4.2

8. Chen B (2022) Data Collection and preprocessing. SpringerBriefs in Computer Science, pp. 5–16. doi:https://doi.org/10.1007/978-981-19-7369-7_2

9. Camizuli E, Carranza EJ (2018) Exploratory Data Analysis (EDA). The Encyclopedia of Archaeological Sciences, pp. 1–7. doi:https://doi.org/10.1002/9781119188230.saseas0271

10. Gallavotti G, Bonetto F, Gentile G (2004) General qualitative properties. Aspects of Ergodic, Qualitative and Statistical Theory of Motion, pp. 1–26. doi:https://doi.org/10.1007/978-3-662-05853-4_1

11. Dr PN (2020) Leukemia drug prediction using machine learning techniques with feature engineering. J Adv Res Dynamic Control Syst 12(SP4):141–146. https://doi.org/10.5373/jardcs/v12sp4/20201475

12. Kapoor N, Vishal S, KKS (2020) Movie recommendation system using NLP Tools. 2020 5th International Conference on Communication and Electronics Systems (ICCES). doi:https://doi.org/10. 1109/icces48766.2020.9137993

13. Arifsiswandi A, Permana Y, Emarilis A (2021) Stemming analysis Indonesian language news text with Porter algorithm. J Phys: Confer Series 1845(1):012019. https://doi.org/10.1088/1742-6596/1845/1/012019

14. Lohmann S, Heimerl F, Bopp F, Burch M, Ertl T (2015) Concentri Cloud: Word cloud visualization for multiple text documents. 2015 19th International Conference on Information Visualisation. doi:https://doi.org/10.1109/iv.2015.30

15. Passalis N, Tefas A (2018) Learning bag-of-embedded-words representations for textual information retrieval. Pattern Recogn 81:254–267. https://doi.org/10.1016/j.patcog.2018.04.008

16. Christian H, Agus MP, Suhartono D (2016) “Single Document Automatic text summarization using term frequency-inverse document frequency (TF-IDF),” ComTech: Computer. Math Eng Appl 7(4):285. https://doi.org/10.21512/comtech.v7i4.3746

17. Pan X, Cheng J, Xia Y, Zhang X, Wang H (2012) “Which feature is better? TF*IDF feature or topic feature in text clustering,” 2012 Fourth International Conference on Multimedia Information Networking and Security. doi:https://doi.org/10.1109/mines.2012.249

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