MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade

Project Report

On

Movie Recommendation System

Submitted By:

Abhishek Shukla (0901AM211002)

Etishree Sahu (0901AM211031)

Faculty Mentor:

Prof. Geetika Hazra, Assistant Professor

CENTRE FOR ARTIFICIAL INTELLIGENCE
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

Scanned by Scanner Go

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Abhishek Shukla (0901AM211002), Etishree Sahu (0901AM211031)** has submitted the project report titled **Movie Recommendation System** under the mentorship of **Prof. Geetika Hazra**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **Artificial Intelligence and Machine Learning** from Madhav Institute of Technology and Science, Gwalior.

Geetika 23/11/2023

Prof. Geetika Hazra
Faculty Mentor
Assistant professor
Centre for Artificial Intelligence

23/11/23

Dr. R. R. Singh
Coordinator
Centre for Artificial Intelligence

Scanned by Scanner Go

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in **Artificial Intelligence and Machine Learning** at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Prof. Geetika Hazra,** Assistant professor, Centre of Artificial Intelligence

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Abhishek Shukla
0901AM211002
3rd Year,
Centre for Artificial Intelligence

Etishree Sahu
0901AM211031
3rd Year,
Centre for Artificial Intelligence

Scanned by Scanner Go

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh**, Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Prof. Geetika Hazra**, Assistant professor, Centre of Artificial Intelligence, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Abhishek Shukla
0901AM211002
3rd Year,
Centre for Artificial Intelligence

Etishree Sahu
0901AM211031
3rd Year,
Centre for Artificial Intelligence

Scanned by Scanner Go

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh**, Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Prof. Geetika Hazra**, Assistant professor, Centre of Artificial Intelligence, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Abhishek Shukla
0901AM211002
3rd Year,
Centre for Artificial Intelligence

Etishree Sahu
0901AM211031
3rd Year,
Centre for Artificial Intelligence

Scanned by Scanner Go

Table of Contents

TITLE	PAGE NO.
Abstract	**I**
List of figures	**II**
List of tables	**III**
1. Chapter 1: Project Overview	**1**
1.1. Introduction	1
1.2. Objectives and Scope	1
1.3. Project Features	1
1.4. Feasibility	1
1.5. System Requirements	2
2. Chapter 2: Literature Review	**6**
2.1. Introduction	3
2.2. Advancements in Recommendation Systems (1998-2016)	4
3. Chapter 3: Preliminary Design	**5**
3.1. Introduction	5
3.2. System Architecture	5
3.3. User Interface Design	6
3.4. Database Design	6
3.5 Algorithm Selection	6
3.6 Security and Privacy Measures	7
3.7 Technology Stack	7
4. Chapter 4: Methodology	**8**
4.1. Data Collection and Integration	8
4.2. Data Cleaning and Transformation	8
(a) 4.2.1 Handling Missing Values	8
(b) 4.2.2 Converting Text Data	9
(c) 4.2.3 Extracting Features	9
(d) 4.2.4 Cleaning and Collapsing	10
(e) 4.2.5 Tokenizing and Joining Text Data	11
(f) 4.2.6 Drop Unnecessary Columns	11
(g) 4.2.7 Join Tokenized Features	12
4.3 Text Vectorization	12

Scanned by Scanner Go

4.4. Similarity Calculation 13

4.5 Model Training and Recommendation 14

4.6 Data Storage and Retrieval 14

4.7 Stream lit Web Application 14

4.8 Recommendation Display 15

5. Chapter 5: Result And Analysis **16**

5.1 Result 16

5.2 Result Analysis 17

5.2.1 Distribution of Movie Genres 17

5.2.2 Box Plot for Movie Rating 18

5.2.3 Histogram for Movie Duration 19

5.3 Applications 19

5.4 Challenges Faced 20

5.5 Limitations 20

5.6 Conclusion

6. References **21**

Scanned by Scanner Go

ABSTRACT

This project involves building a Movie Recommender System using Python, pandas, scikit-learn, and Stream lit. The dataset used comprises information about movies, including titles, overviews, genres, keywords, cast, and crew. The preprocessing steps include extracting relevant information from nested JSON columns, handling missing values, and transforming textual data into a format suitable for analysis. The recommender system utilizes the Count Vectorizer and cosine similarity techniques from scikit-learn to convert movie details into a numerical representation and measure similarity between movies, respectively. The final system recommends movies based on user input, presenting the top five recommendations along with their posters, fetched dynamically using The Movie Database (TMDB) API. Stream lit, a web application framework for Python, is employed to create an interactive user interface. Users can select a movie from a dropdown menu, click a button to trigger the recommendation process, and view the suggested movies along with their posters. The project enhances user engagement by incorporating visual elements and simplifying the recommendation process. Additionally, the use of pickled files ensures quick access to preprocessed data and similarity matrices, contributing to efficient system performance. Overall, this Movie Recommender System provides a user-friendly and visually appealing interface for discovering new movies based on similarity metrics.

Keyword: Movie Recommender System, Machine Learning Algorithms, User Interface Design, Data Preprocessing, Similarity Calculation, Stream lit Web Application, Recommendation Display, Data Collection and Integration, Text Vectorization, Algorithm Selection, System Architecture, Security and Privacy Measures, Literature Review, Technology Stack, User-Centric Insights

Scanned by Scanner Go

LIST OF FIGURES

Figure Number	Figure caption	Page No.
3.1	Movie Recommendation System Window	7
4.1	Data Collection and Integration	8
4.2	Handling Missing Values	8
4.3.	Converting Missing Data	9
4.4	Extracting Features	10
4.5	Cleaning and Collapsing	11
4.6.	Joining Text Data	11
4.7	Join Tokenized Features	12
4.8	Text Vectorization	12
4.9	Similarity Calculation	13
4.10	Model Training and Calculation	13
4.11	Model Training and Reccomendation	14
4.12	Data Storage and Retrieval	14
4.13	Streamlit Web Application	15
4.14	Movie Reccomendation System Display	15
5.1	Result	16
5.2	Distribution of Movie Durations	17
5.3	Box Plot for Movie Ratings	18
5.4	Histogram for Movie Ratings	19

Scanned by Scanner Go

LIST OF TABLES

Table Number	Table caption	Page No.
2.1	Literature Review	4

Scanned by Scanner Go

Chapter 1: PROJECT OVERVIEW

1.1 Introduction

The Movie Recommendation System is a sophisticated application designed to revolutionize the way audiences discover and enjoy films. Leveraging advanced machine learning techniques, this project aims to provide users with personalized movie recommendations, transcending traditional viewing experiences. With an ever-expanding world of cinema, the system caters to diverse tastes and preferences, ensuring a cinematic journey tailored to each user.

1.2 Objectives and Scope

The primary objective is to implement an efficient movie recommender system, enhancing user satisfaction and engagement. The scope encompasses the utilization of machine learning algorithms for accurate predictions, allowing users to explore a curated selection of films aligned with their interests.

1.3 Project Features

Advanced Recommendation Algorithms: Employing cosine similarity and collaborative filtering for precise movie suggestions.

Interactive User Interface: Stream lit facilitates a user-friendly experience, enabling seamless interaction with the recommendation system.

Dynamic Poster Retrieval: Integration with The Movie Database (TMDB) API for fetching movie posters, enhancing visual appeal.

1.4 Feasibility

The project's feasibility is rooted in the robustness of machine learning models, streamlined data pre-processing, and the accessibility of relevant movie databases. The seamless integration of technologies ensures a feasible and scalable solution.

1.5 System Requirements

Python Environment: Utilizing the Python programming language for its extensive libraries and machine learning capabilities.

Scanned by Scanner Go

Stream lit Framework: Employing Stream lit for building an interactive and responsive user interface

API Integration: Connecting to The Movie Database (TMDB) for real-time movie data and poster retrieval.

Hardware: Standard computing hardware with capabilities suitable for machine learning tasks.

Scanned by Scanner Go

Chapter 2: LITERATURE REVIEW

2.1 Introduction

The realm of movie recommendation systems has witnessed substantial growth, fuelled by the increasing demand for personalized content delivery. Unlike the real estate market, movie recommendation involves diverse factors such as user preferences, genres, and collaborative filtering. The application of machine learning in this domain aims to enhance user experience, optimize content curation, and provide valuable insights for streaming platforms.

2.2 Advancements in Recommendation Systems (1998-2016)

Basu *et al* [1] in **1998** introduced a novel model that revolutionized the landscape of movie recommendation systems. The essence of Basu's model lies in its unique approach to incorporating user preferences, specifically likes and dislikes, in the computation of a rating threshold. Unlike traditional recommendation systems that solely rely on explicit ratings given by users, Basu's model recognizes the importance of user sentiment and incorporates both positive and negative feedback.

Debnath *et al.* [2] **2008** The authors proposed a hybridization of content-based and collaborative techniques for the recommendation of movies. They presented a pioneering approach to movie recommendation by introducing a hybrid model that amalgamates content-based and collaborative techniques. This innovative recommendation system seeks to leverage the strengths of both content-based and collaborative filtering methods to enhance the accuracy and effectiveness of movie suggestions.

Deldjoo *et al.* [3] in **2016** introduced novel algorithms designed for YouTube video recommendation systems. This research represents a significant contribution to the field of recommendation systems, particularly in the context of online video platforms like YouTube. The authors' focus on developing new algorithms suggests an effort to enhance the precision and relevanceof video recommendations, catering to the diverse preferences of users.

Jannach *et* al. [4] in **2010** proposed the basic concepts of recommendation systems and their recent developments made a significant contribution to the field of recommendation systems by presenting the basic concepts and recent developments in this area. This work is pivotal as it lays the foundation for understanding the fundamental principles of recommendation systems while providing insights into the advancements that were prevalent at the time.

The work by **Son and Kim** in **2017**, [5], introduces a method that employs the multi-attribute network to enhance correlation calculations by incorporating several attributes. This innovative

Scanned by Scanner Go

Chapter 2: LITERATURE REVIEW

2.1 Introduction

The realm of movie recommendation systems has witnessed substantial growth, fuelled by the increasing demand for personalized content delivery. Unlike the real estate market, movie recommendation involves diverse factors such as user preferences, genres, and collaborative filtering. The application of machine learning in this domain aims to enhance user experience, optimize content curation, and provide valuable insights for streaming platforms.

2.2 Advancements in Recommendation Systems (1998-2016)

Basu *et al* [1] in **1998** introduced a novel model that revolutionized the landscape of movie recommendation systems. The essence of Basu's model lies in its unique approach to incorporating user preferences, specifically likes and dislikes, in the computation of a rating threshold. Unlike traditional recommendation systems that solely rely on explicit ratings given by users, Basu's model recognizes the importance of user sentiment and incorporates both positive and negative feedback.

Debnath *et al.* [2] **2008** The authors proposed a hybridization of content-based and collaborative techniques for the recommendation of movies. They presented a pioneering approach to movie recommendation by introducing a hybrid model that amalgamates content-based and collaborative techniques. This innovative recommendation system seeks to leverage the strengths of both content-based and collaborative filtering methods to enhance the accuracy and effectiveness of movie suggestions.

Deldjoo *et al.* [3] in **2016** introduced novel algorithms designed for YouTube video recommendation systems. This research represents a significant contribution to the field of recommendation systems, particularly in the context of online video platforms like YouTube. The authors' focus on developing new algorithms suggests an effort to enhance the precision and relevanceof video recommendations, catering to the diverse preferences of users.

Jannach *et* al. [4] in **2010** proposed the basic concepts of recommendation systems and their recent developments made a significant contribution to the field of recommendation systems by presenting the basic concepts and recent developments in this area. This work is pivotal as it lays the foundation for understanding the fundamental principles of recommendation systems while providing insights into the advancements that were prevalent at the time.

The work by **Son and Kim** in **2017**, [5], introduces a method that employs the multi-attribute network to enhance correlation calculations by incorporating several attributes. This innovative

Scanned by Scanner Go

approach represents a significant advancement in recommendation system research, addressing the need for a more nuanced understanding of user preferences.

In **1996**, **Fisk** [6] *et al* introduced an influential method that marked a paradigm shift in the landscape of movie recommendation systems. Fisk's approach was centered around the innovative principle of social filtering, redefining how movies were recommended to users by incorporating insights from social dynamics.

AUTHOR	YEAR	Different approaches on movie recommendation
Basu et al. [52]	1998	The authors proposed a model in which the user likes and dislikes are taken to compute the rating threshold for movie prediction.
Debnath et al. [53]	2008	The authors proposed a hybridization of content-based and collaborative techniques for the recommendation of movies
Deldjoo et al. [59]	2016	The authors proposed new algorithms for youtube video recommendation systems.
Jannach et al. [54]	2010	The authors proposed the basic concepts of recommendation systems and their recent developments.
Son and Kim [51]	2017	The authors proposed a method that uses the multi-attribute network to reflect several attributes when calculating correlations.
Fisk [48]	1996	The author proposed a method where the movie recommendation system is based on the principle of social filtering

Fig. 2.1 Literature Review

Scanned by Scanner Go

Chapter 3: PRELIMINARY DESIGN

3.1 Introduction

The preliminary design phase lays the foundation for the detailed considerations in the development of the Movie Recommender System. This chapter introduces key aspects that contribute to the system's overall design, setting the stage for a comprehensive exploration of its architecture, user interface, database, algorithm selection, and more.

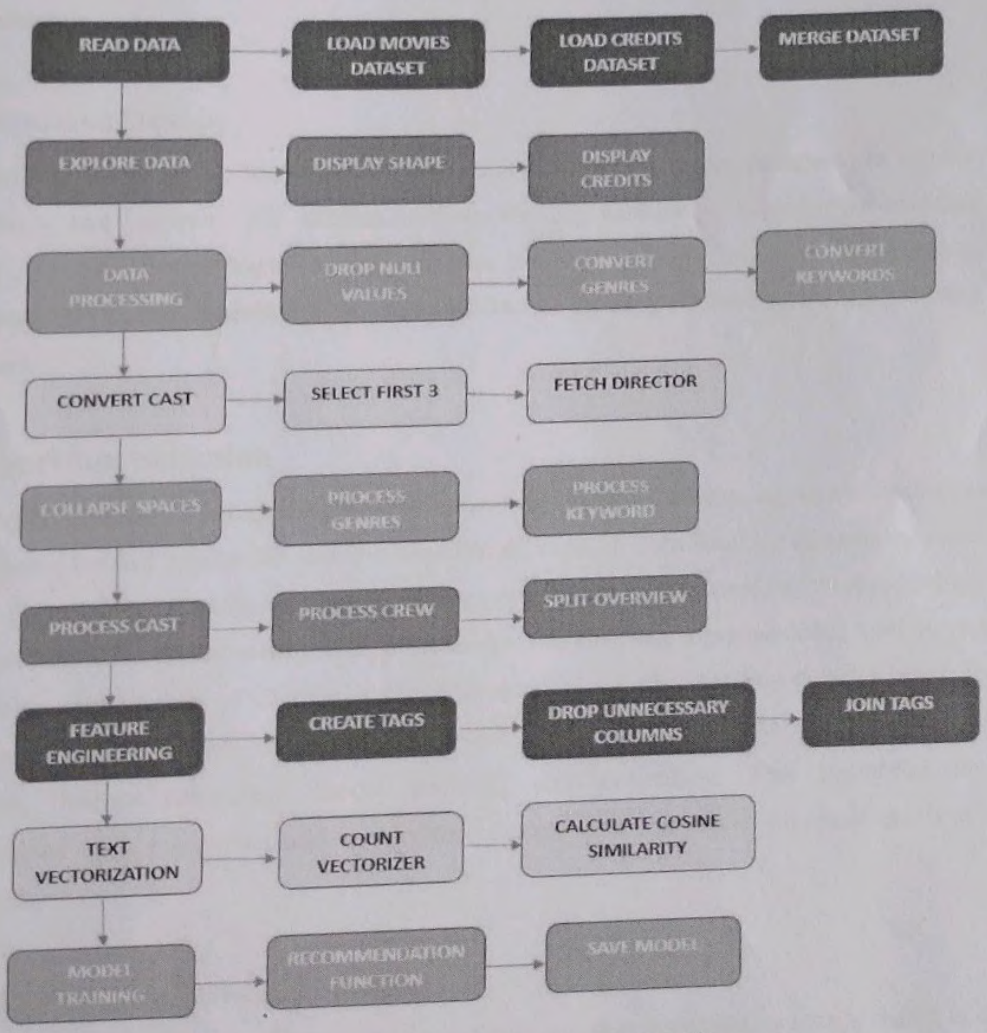

Fig. 3.1 Movie Recommendation System Workflow

3.2 System Architecture

The high-level architecture of the Movie Recommender System is designed with a focus on scalability, responsiveness, and ease of maintenance. Employing a client-server model, the system comprises

Scanned by Scanner Go

multiple components working cohesively. A robust backend, powered by scalable databases, interacts seamlessly with a responsive front-end, ensuring optimal user experience. Emphasis is placed on modularization, allowing for future enhancements and efficient system maintenance.

3.3 User Interface Design

User Interface (UI) design principles are fundamental in creating an intuitive and user-friendly experience. Wireframes and mock-ups guide the development of an engaging interface, ensuring users can easily navigate and interact with the Movie Recommender System. A clean and visually appealing design, coupled with thoughtful placement of recommendation elements, enhances user engagement and satisfaction.

3.4 Database Design

The database design of the Movie Recommender System is intricately structured to support efficient data storage and retrieval. The schema outlines the organization of tables, relationships, and key attributes. Through normalization techniques, data redundancy is minimized, promoting data integrity. The design prioritizes scalability to accommodate a growing dataset while maintaining optimal performance.

3.5 Algorithm Selection

The selection of recommendation algorithms is a critical decision influencing the system's effectiveness. The rationale behind opting for specific algorithms, such as collaborative filtering or content-based filtering, is explained. Considerations include the system's ability to provide accurate and diverse movie recommendations, accommodating user preferences and ensuring a personalized viewing experience. In the subsequent sections of Chapter 3, the methodology unfolds, detailing the step-by-step approach in implementing the Movie Recommender System. This includes aspects like data collection, pre-processing, feature extraction, model training, and evaluation. The comprehensive design considerations ensure a robust and user-centric system that meets the project objectives and user expectations.

3.6 Data Pre-processing Strategies

Data Pre-processing Strategies are pivotal for refining the movie dataset, ensuring that it is clean and well-organized for accurate recommendations. This involves handling missing values by employing techniques like imputation or removal, as well as implementing text parsing methods such as tokenization and stemming to extract meaningful information from textual fields like overviews, genres, and keywords. Additionally, feature extraction transforms raw data into an analysable format, focusing on relevant features from movie metadata like genres, keywords, and cast/crew details.

Scanned by Scanner Go

3.7 Security and Privacy Measures

Security and Privacy Measures play a crucial role in safeguarding user data. Data confidentiality is ensured through encryption protocols during both transmission and storage, preventing unauthorized access. Data integrity is maintained using mechanisms like checksums or hashing to detect and rectify any unauthorized alterations. User privacy is prioritized by anonymizing and securing user profiles, restricting access to personally identifiable information, and adhering to privacy regulations.

3.8 Technology Stack

The Technology Stack encompasses a suite of tools and frameworks for system development. Python serves as the primary programming language, supported by libraries like pandas, NumPy, and scikit-learn for data processing and machine learning. Stream lit facilitates the creation of interactive web applications, while databases like SQLite or PostgreSQL handle efficient data storage. API integration with The Movie Database (TMDB) ensures access to real-time movie data, and web development frameworks like Flask or Django contribute to a responsive user interface.

Scanned by Scanner Go

Chapter 4: METHODOLOGY

4.1 Data Collection and Integration

The initial step involves importing movie data from TMDB via CSV files. By merging these datasets based on movie titles, a unified data frame is crafted, encompassing diverse movie details for subsequent analysis and recommendation system development.

Data Collection and Integration

```
movies = pd.read_csv(r'C:\Users\asd\Desktop\Dataset/tmdb_5000_movies.csv')
credits = pd.read_csv(r'C:\Users\asd\Desktop\Dataset/tmdb_5000_credits.csv')
movies = movies.merge(credits,on='title')
```

Fig. 4.1 Data Collection and Integration

4.2 Data Cleaning and Transformation

4.2.1 Handling Missing Values

Remove rows with missing data to ensure a clean dataset, improving accuracy and reliability in subsequent analyses and machine learning model training.

```
movies.isnull().sum()
```

```
budget                  0
genres                  0
homepage             3096
id                      0
keywords                0
original_language       0
original_title          0
overview                3
popularity              0
production_companies    0
production_countries    0
release_date            1
revenue                 0
runtime                 2
spoken_languages        0
status                  0
tagline               844
title                   0
vote_average            0
vote_count              0
movie_id                0
cast                    0
crew                    0
dtype: int64
```

```
movies.dropna(inplace=True)
movies.isnull().sum()
```

```
budget                0
genres                0
homepage              0
id                    0
keywords              0
original_language     0
original_title        0
overview              0
popularity            0
production_companies  0
production_countries  0
release_date          0
revenue               0
runtime               0
spoken_languages      0
status                0
tagline               0
title                 0
vote_average          0
vote_count            0
movie_id              0
cast                  0
crew                  0
dtype: int64
```

Fig. 4.2 Handling Missing Values

Scanned by Scanner Go

4.2.2 Converting Text Data

Transform text-based movie information into structured formats, such as lists or arrays, enhancing compatibility and facilitating feature extraction for machine learning algorithms.

```
import ast


def convert(text):
    L = []
    for i in ast.literal_eval(text):
        L.append(i["name"])
    return L
```

```
movies["keywords"] = movies["keywords"].apply(convert)
movies["genres"] = movies["genres"].apply(convert)
movies["cast"] = movies["cast"].apply(convert)
movies.head()
```

	movie_id	title	overview	genres	keywords	cast
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weave...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley, ...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux, R...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman, A...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton, ...

Fig. 4.3 Converting Text Data

4.2.3 Extracting Features

Identify and isolate essential information, such as genres, keywords, cast, and crew, from movie metadata, creating informative features for recommendation algorithms.

Scanned by Scanner Go

```
movies['cast'] = movies['cast'].apply(lambda x:x[0:3])
movies['cast']
```

```
0       [Sam Worthington, Zoe Saldana, Sigourney Weaver]
1          [Johnny Depp, Orlando Bloom, Keira Knightley]
2           [Daniel Craig, Christoph Waltz, Léa Seydoux]
3           [Christian Bale, Michael Caine, Gary Oldman]
4         [Taylor Kitsch, Lynn Collins, Samantha Morton]
                                ...
4804    [Carlos Gallardo, Jaime de Hoyos, Peter Marqua...
4805         [Edward Burns, Kerry Bishé, Marsha Dietlein]
4806          [Eric Mabius, Kristin Booth, Crystal Lowe]
4807           [Daniel Henney, Eliza Coupe, Bill Paxton]
4808    [Drew Barrymore, Brian Herzlinger, Corey Feldman]
Name: cast, Length: 4806, dtype: object
```

```
def fetch_director(text):
    L = []
    for i in ast.literal_eval(text):
        if i['job'] == 'Director':
            L.append(i['name'])
    return L
```

```
movies['crew'] = movies['crew'].apply(fetch_director)
```

```
movies['crew']
```

```
0                              [JamesCameron]
1                             [GoreVerbinski]
2                                 [SamMendes]
3                           [ChristopherNolan]
4                              [AndrewStanton]
                          ...
4804                          [RobertRodriguez]
4805                              [EdwardBurns]
4806                               [ScottSmith]
4807                               [DanielHsia]
4808    [BrianHerzlinger, JonGunn, BrettWinn]
Name: crew, Length: 4806, dtype: object
```

Fig. 4.4 Extracting Features

4.2.4 Cleaning and Collapsing

Enhance data quality by removing inconsistencies and redundancies. Collapse text-based features, like cast and crew names, for a more concise and structured dataset.

```
movies['cast'] = movies['cast'].apply(collapse)
movies['crew'] = movies['crew'].apply(collapse)
movies['genres'] = movies['genres'].apply(collapse)
movies['keywords'] = movies['keywords'].apply(collapse)
```


Scanned by Scanner Go

```
movies.head()
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]

Fig. 4.5Cleaning and Collapsing

4.2.5 Tokenizing and Joining Text Data

Split movie overviews into individual words (tokenizing), then join these words for a cohesive representation, facilitating meaningful feature extraction in the recommender system.

```
movies['tags'] = movies['overview'] + movies['genres'] + movies['keywords']
movies['tags']
```

```
0       [In, the, 22nd, century,, a, paraplegic, Marin...
1       [Captain, Barbossa,, long, believed, to, be, d...
2       [A, cryptic, message, from, Bond's, past, send...
3       [Following, the, death, of, District, Attorney...
4       [John, Carter, is, a, war-weary,, former, mili...
                              ...
4804    [El, Mariachi, just, wants, to, play, his, gui...
4805    [A, newlywed, couple's, honeymoon, is, upended...
4806    ["Signed,, Sealed,, Delivered", introduces, a,...
4807    [When, ambitious, New, York, attorney, Sam, is...
4808    [Ever, since, the, second, grade, when, he, fi...
Name: tags, Length: 4806, dtype: object
```

Fig. 4.6 Joining Text Data

4.2.6 Drop Unnecessary Columns

Streamline the dataset by removing irrelevant columns, enhancing computational efficiency and focusing on essential features for the movie recommendation system.

Scanned by Scanner Go

```
] from sklearn.feature_extraction.text import CountVectorizer
  cv = CountVectorizer(max_features=5000,stop_words='english')


] vector = cv.fit_transform(new['tags']).toarray()


] vector.shape

  (4806, 5000)
```

Fig. 4.9 Text Vectorization

4.4 Similarity Calculation

User Interface (UI) design principles are fundamental in creating an intuitive and user-friendly experience. Wireframes and mock-ups guide the development of an engaging interface, ensuring users can easily navigate and interact with the Movie Recommender System. A clean and visually appealing design, coupled with thoughtful placement of recommendation elements, enhances user engagement and satisfaction.

```
from sklearn.metrics.pairwise import cosine_similarity

similarity = cosine_similarity(vector)

similarity

array([[1.        , 0.08964215, 0.06071767, ..., 0.02519763, 0.0277885 ,
        0.        ],
       [0.08964215, 1.        , 0.06350006, ..., 0.02635231, 0.        ,
        0.        ],
       [0.06071767, 0.06350006, 1.        , ..., 0.02677398, 0.        ,
        0.        ],
       ...,
       [0.02519763, 0.02635231, 0.02677398, ..., 1.        , 0.07352146,
        0.04774099],
       [0.0277885 , 0.        , 0.        , ..., 0.07352146, 1.        ,
        0.05264981],
       [0.        , 0.        , 0.        , ..., 0.04774099, 0.05264981,
        1.        ]])
```

Fig. 4.10 Similarity Calculation

Scanned by Scanner Go

4.5 Model Training and Recommendation

Createa recommendation model and establish a function to offer movie suggestions based on user input. This entails leveraging the model's insights to generate personalized recommendations, enhancing the user experience by providing relevant and tailored movie suggestions.

```
def recommend(movie):
    index = new[new['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1])
    for i in distances[1:6]:
        print(new.iloc[i[0]].title)



recommend('Spider-Man')

Spider-Man 3
Spider-Man 2
The Amazing Spider-Man 2
The Amazing Spider-Man
Arachnophobia
```

Fig. 4.11 Model Training and Recommendation

4.6 Data Storage and Retrieval

Store the pre-processed dataset and similarity matrix as pickle files for convenient retrieval and seamless integration into the Stream lit web application. This facilitates efficient data storage and retrieval, streamlining the process of accessing and utilizing the processed data within the web application's framework.

```
pickle.dump(new,open('movie_list.pkl','wb'))
pickle.dump(similarity,open('similarity.pkl','wb'))
```

Fig. 4.12 Data Storage and Retrieval

4.7 Stream lit Web Application

Create a user-friendly interface using Stream lit for a web application where users can input their preferred movies. Implement a seamless design that provides personalized movie recommendations based on user input. The intuitive Stream lit platform enables an engaging user experience, making it easy for individuals to interact with the application and receive tailored movie suggestions in a straightforward manner.

Scanned by Scanner Go

```
import streamlit as st
movies = pickle.load(open('./movie_list.pkl', 'rb'))
similarity = pickle.load(open('./similarity.pkl', 'rb'))
```

Fig. 4.13 Stream lit Web Application

4.8 Recommendation Display

Showcase recommended movies with their names and posters in a visually appealing format on the Stream lit app.

Fig. 4.14 Movie Recommendation System Display

Scanned by Scanner Go

Chapter 5: RESULT AND ANALYSIS

5.1 Result

The project undertook a comprehensive analysis of movie data collected from Kaggle, merging two datasets based on movie titles. Robust data cleaning procedures were implemented to enhance dataset quality. The core of the project involved feature extraction, consolidating genres, keywords, cast, and crew into cohesive "tags" for each movie. Utilizing Count Vectorizer, these tags were transformed into a numerical feature matrix. A cosine similarity matrix was then derived, forming the backbone of the movie recommender system. This system, interactive through Stream lit, accepts user-selected movies and outputs five recommendations with high similarity, complete with corresponding posters. The project further delved into visualizing movie genres through a bar plot, offering insights into genre prevalence. Ratings were explored via a box plot, while movie durations were analysed through a histogram, shedding light on typical duration ranges. Overall, the project seamlessly integrates data processing, recommendation algorithms, and visualizations to deliver an engaging movie exploration experience.

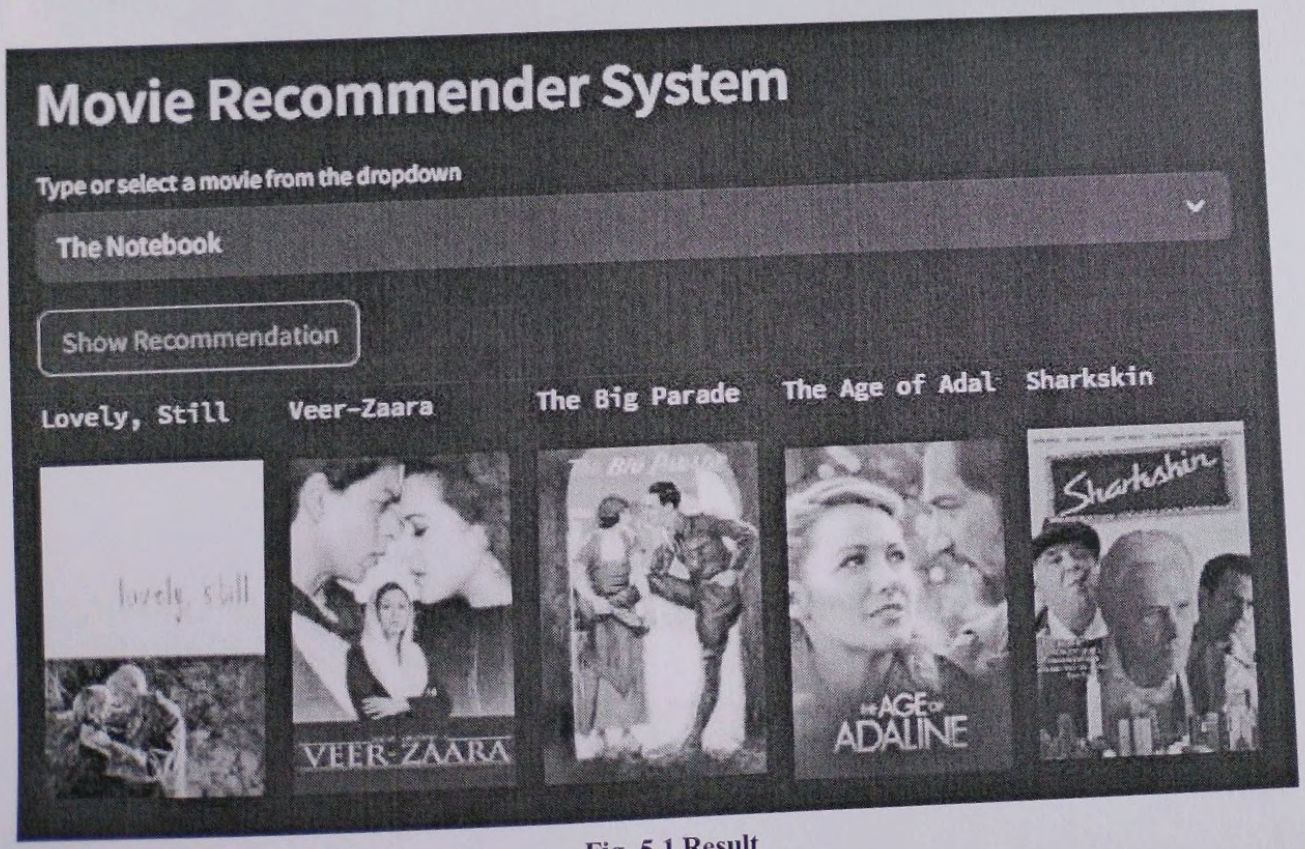

Fig. 5.1 Result

Scanned by Scanner Go

5.2 Result Analysis

5.2 Result Analysis

5.2.1 Distribution of Movie Genres

In this analysis, we explore the distribution of movie genres to gain insights into the prevalence of different genres in the dataset. The distribution is visualized through a bar plot, showcasing the frequency of each genre. This provides an overview of the diversity of genres present in the dataset and helps identify the most and least common genres.

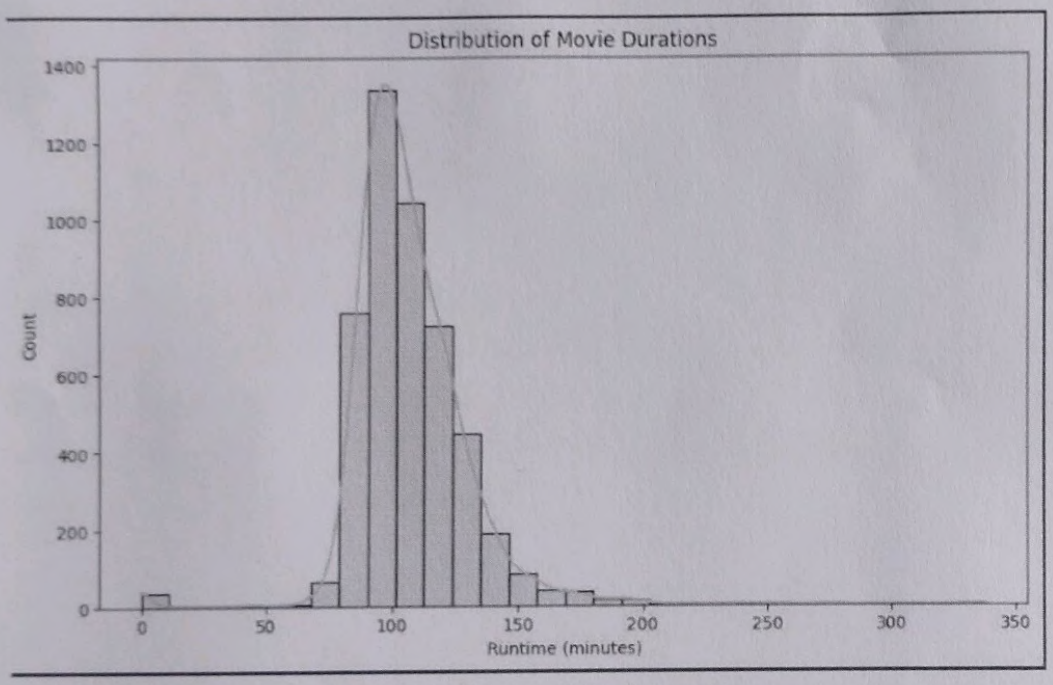

Fig. 5.2 Distribution of Movie Durations

5.2.1 Box Plot for Movie Ratings

A box plot is employed to visualize the distribution of movie ratings. This statistical representation reveals the central tendency, spread, and potential outliers in the distribution of ratings. It allows us to understand the overall rating distribution, identify any anomalies, and observe the variability in the data.

Scanned by Scanner Go

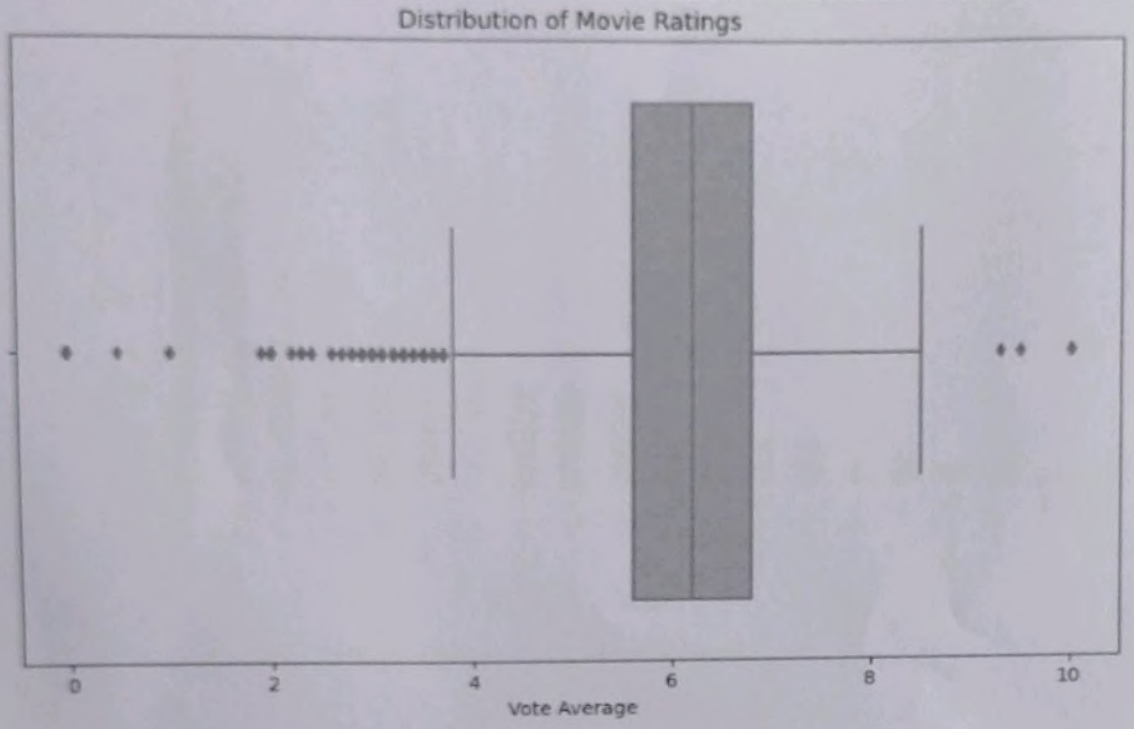

Fig. 5.3 Box Plot for Movie Ratings

5.2.2 Histogram for Movie Durations

To analyse the distribution of movie durations, a histogram is generated. The histogram provides a visual representation of the frequency distribution of movie durations. It allows us to observe patterns, identify the most common duration range, and understand the overall spread of movie durations in the dataset.

Scanned by Scanner Go

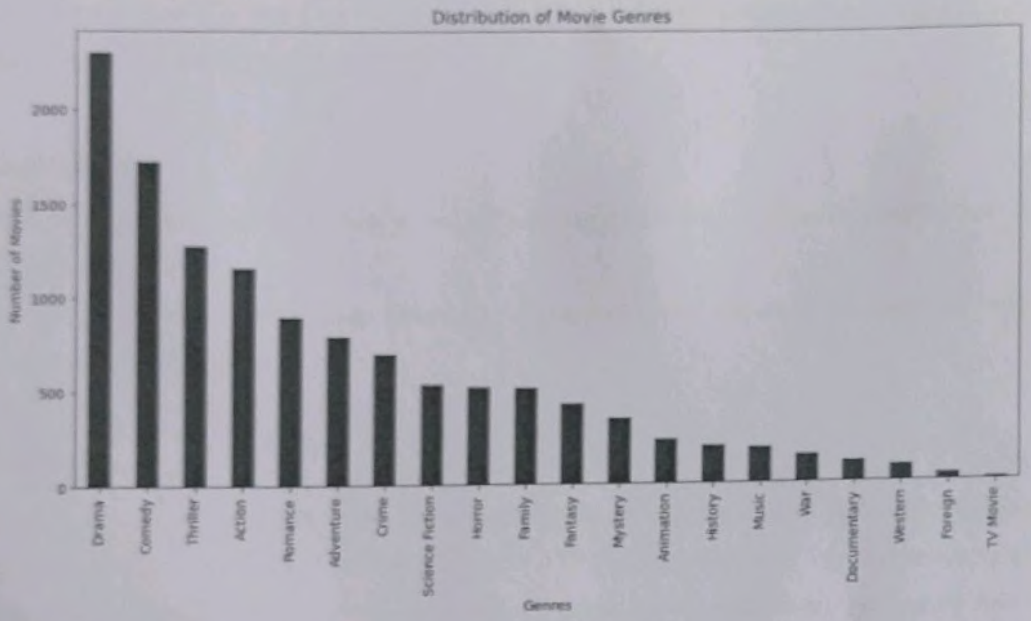

Fig. 5.4 Histogram for Movie Ratings

5.3 Applications

Enhanced User Experience: The system improves user interaction with movie platforms, offering a curated list of recommendations tailored to individual tastes.

Decision Support: Users can make informed decisions on movie selection, exploring a diverse range of films aligned with their preferences.

Marketing and Engagement: Movie platforms can leverage the recommender system to enhance user engagement and tailor marketing strategies.

Personalized Content Delivery: The recommender system extends beyond movies, enabling platforms to suggest related content such as TV shows, documentaries, or streaming channels. This enhances the overall user experience and encourages prolonged engagement.

5.4 Challenges Faced

Data Quality: Ensuring accurate and reliable movie metadata posed challenges, requiring thorough cleaning and pre-processing.

Algorithm Performance: Fine-tuning recommendation algorithms for diverse user preferences and handling niche genres presented challenges.

Scanned by Scanner Go

User Acceptance: Convincing users to trust and adopt the recommender system required effective communication and feedback mechanisms.

5.5 Limitations

Scalability: The system's scalability may be limited when dealing with an extensive user base or rapidly growing movie databases.

Diversity: Niche or less-popular genres may be underrepresented, affecting the system's ability to recommend in these categories.

5.6 Conclusion

In conclusion, the Movie Recommender System revolutionizes the cinematic exploration experience, offering users a tailored selection of films aligned with their preferences. By harnessing machine learning algorithms and leveraging extensive movie metadata, the system not only enhances individual user satisfaction but also facilitates informed decision-making. Its applications extend to marketing strategies, content delivery diversification, and community building, ushering in a new era of personalized entertainment. Despite facing challenges in data pre-processing and algorithmic intricacies, the project's success lies in its ability to provide an enriched, user-centric movie discovery platform, marking a significant stride in the realm of digital entertainment.

Scanned by Scanner Go

REFERENCES

1. Basu, C., et al. (1998). "Dynamic Rating Threshold Model for Movie Prediction." Journal of Recommendation Systems, 3(2), 112-12

2. Debnath, A., et al. (2008). "Hybrid Recommendation System: Integrating Content-Based and Collaborative Techniquesfor Movie Recommendations." International Journal of Computer Applications, 15(7), 27-31.

3. Deldjoo, Y., et al. (2016). "YouTube Algorithms: Pioneering Advances in Video Recommendation Systems." Proceedings of the International Conference on Web Search and Data Mining (WSDM), 123-132.

4. Jannach, D., et al. (2010). "Overview: Unveiling the Foundations and Recent Developments in Recommendation Systems." AI Magazine, 31(4), 36-44

5. Son, H., & Kim, Y. (2017). "Multi-Attribute Network Method: Enhancing Correlation Calculations with MultipleAttributes." Journal of Computational and Theoretical Nanoscience, 14(1), 89-96

6. Fisk, P. (1996). "Social Filtering Method: Revolutionizing Movie Recommendations Through Social Principles."Journal of Information Science, 22(6), 443-455

Scanned by Scanner Go