DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

MOVIE RECOMMENDATION SYSTEM

By
Aishwarya D Shetty – ENG18CS0022
Anusha M – ENG18CS0042
Anusha R – ENG18CS0043
Bhavana– ENG18CS0059

Under the supervision of

Dr. Pramod Kumar Naik
Associate Professor, CSE Department

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering

Department of Computer Science & Engineering

Kudlu Gate, Bangalore – 560068

Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"MOVIE RECOMMENDATION SYSTEM"** is carried out by **Aishwarya D Shetty (ENG18CS0022), Anusha M (ENG18CS0042), Anusha R (ENG18CS0043), Bhavana (ENG18CS0059)** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Dr. Pramod Kumar Naik	**Dr. Girisha G S**	**Dr. A Srinivas**
Associate Professor Dept. of CSE, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Aishwarya D Shetty (ENG18CS0022), Anusha M (ENG18CS0042), Anusha R (ENG18CS0043), Bhavana (ENG18CS0059)** are students of the eighth semester B-Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **"Movie Recommendation System"** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student **Signature**

Name1: Aishwarya D Shetty
USN: ENG18CS0022

Name2: Anusha M
USN: ENG18CS0042

Name3: Anusha R
USN: ENG18CS0043

Name4: Bhavana
USN: ENG18CS0059

Place: Bangalore
Date:

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Dr. Pramod Kumar Naik**, **Associate Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinator Dr. Meenakshi Malhotra and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our parents, family and friends who provided us with every requirement throughout the course, We would like thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

Pages

LIST OF ABBREVIATIONS vi
LIST OF FIGURES vii
ABSTRACT viii
CHAPTER 1 INTRODUCTION 1
1.1. PURPOSE 2
1.1.1 INTENDED AUDIENCE 2
1.2. SCOPE 2
CHAPTER 2 PROBLEM DEFINITION 3-4
2.1 OBJECTIVES 4
CHAPTER 3 LITERATURE SURVEY 5-6
CHAPTER 4 PROJECT DESCRIPTION 7-9
4.1. PROPOSED DESIGN 8
4.2.WORK FLOW 8
4.3. ASSUMPTIONS AND DEPENDENCIES 9
CHAPTER 5 REQUIREMENTS 10-12
5.1. FUNCTIONAL REQUIREMENTS 11
5.2. NON-FUNCTIONAL REQUIREMENTS 11
5.2.1 SOFTWARE QUALITY ATTRIBUTES 11
5.3. SOFTWARE REQUIREMENTS 11
5.4 HARDWARE REQUIREMENTS 12
CHAPTER 6 METHODOLOGY 13-14
CHAPTER 7 EXPERIMENTATION 15-19
CHAPTER 8 TESTING AND RESULTS 20-23
CONCLUSION 24
REFERENCES 24-26

APPENDIX A 27
BOW 27
APPENDIX B 27
IMDB 27
APPENDIX C 27
KNN 27
APPENDIX D 27
SOURCE CODE 27-29

LIST OF ABBREVIATIONS

ML	Machine Learning
BOW	Bag of Words
IMDB	Internet Movie Database
HTML	Hyper Text Markup Language
KNN	K-Nearest Neighbor
SVM	Support Vector Machine

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
1	Proposed design	8
2	Work Flow	8
3	Confusion Matrix	16
4	K- nearest neighbor classifier	17
5	Implementation	17-18
6	Out Put Screenshots	21-23

ABSTRACT

Recommendation systems play an important role in our everyday life. Starting from movie/music streaming sites like Netflix or Spotify to the most basic search engines, the core component is recommendation systems. Even the technology giant Google is known for its search engine above everything else. To emphasize its importance we aim to build a simple recommendation system using the IMDB movie database. Our finished web application can suggest similar movies based on the input provided by the user, considering factors like the plot of the movie, actors present in it, as well as the director. Needless to say, some factors would be common between the input movie and the recommended ones. The web app uses the scikit-learn python library, to match with the input movie with corresponding movies in the dataset with the highest similarity score. Its frontend is designed using Flask and it has speed of 40 seconds. And it is deployed on Heroku. We aim to demystify the magic behind a recommendation system. When we applied different Machine learning algorithms we have obtained the accuracies of 98% using KNN, 99% using Decision tree classifier, 99% using SVM, 99% using Random forest classifier. For our project KNN is giving us better accuracy which is 98% with the ease of use. The precision scores are 0.98 and 0.97, recall is 0.97 and 0.99, fl-score of 0.98 and 0.98, support value is 638 and 746 are obtained.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

Every human being has his/her own nuances. Their taste or views of various subject matters may differ. Recommendation systems aim to exploit those nuances and predict what kind of content the user might prefer. Looking at the vast variety and volume of movies that appear when streaming sites like Netflix or Amazon Prime, it is only natural that humans get confused about what to watch. In this situation, recommendation systems come into play. A content based recommender works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). Based on that data, a user profile is generated, which is then used to make suggestions to the user .This recommendation system is known as content based recommendation system as we are suggesting movies based on the history of the users.

1.1 PURPOSE

To Provide relevant content out of relevant or irrelevant collection of items to the users which makes it hassle-free and easy for users to find contents based on their interest. recommendation system aims to provide users with accurate movie recommendations. Usually basic recommendation system to make recommendations consider one of the following factors; User preference known as content based Filtering or the preference of similar users known as collaborative filtering. To create a stable and accurate recommender system will use of content based filtering.

1.1.1 INTENDED AUDIENCE

The intended audience for this project is people who have interest in movies. Recommender System is a system used to predict or filter preferences according to the user's choices.

1.2 SCOPE

In today’s world searching for content/movies is a common problem. Every user wants an exact result for which he/she is looking for. No one wants to waste their time on searching. That’s when the movie recommendation system comes to picture.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

The total amount of time a person spends deciding which movie to watch due to the huge number of choices, that time could be utilized in watching at least half of a full-length movie. Hence an average customer would be saving about an hour of his time every week, doing something more productive. These systems use information filtering techniques to process information and provide the user with potentially more relevant items.

2.1 OBJECTIVES

Purpose of the movie recommendation system aims to provide users with accurate movie recommendations. Usually basic recommendation system to make recommendations consider one of the following factors; User preference known as content based Filtering or the preference of similar users known as collaborative filtering In today's world a lot of data is generated through various e-commerce sites and different platforms. Sometimes users may be confused or waste a lot of time in order to search for what he/she is actually looking for So our system helps in content based recommendation.

CHAPTER 3

LITERATURE REVIEW

Movie Recommender System Using K- Means Clustering AND K-Nearest Neighbor Published by Ahuja, R., Solanki, A., & Nayyar, A.in the year 2019 January Technology Used is K-Mean clustering and K-Nearest Neighbor Algorithms such as K-Means Clustering, KNN is used in recommending movies. Python implementation increased the accuracy of prediction.After studying different types of machine learning algorithms, there is a clear picture of where to apply which algorithm in different areas of industries such as recommender systems, e- commerce[1]. Using Collaborative Filtering Algorithms Combined with Doc2Vec for Movie Recommendation Published by Liu, G., & Wu, X in 2019 Technology Used Collaborative Filtering Algorithms,Doc2Vec Here, including the score data of the users' movie and the story introduction data of the movie is used to enrich the input of the algorithm model.Here movies were recommended based on their features, and word order of the sentences[2]. Research and Development of Movie Social System Published by Qui Bin Liu, Sheng Gao, Bi Feng Guo, Hui Liu in 2019 It uses Web development and deployed project using Heroku server.We came to Know about Heroku server which we used in our project[3]. In the year 2018,

Ifada, N., & Prasetyo, E. H proposed methods of handling the sparsity and scalability problems of the collaborative filtering approach. For handling sparsity, the approach estimates the rating entries by combining the similarities of the rating and their respective genres using relative weighting [8]

Yeole Madhavi B.1, Rokade Monika D.2, Khatal Sunil S.31 used Content-based Filtering, Text to vector, Vector similarity, Hybrid approach[9]

According to R. Lavanya, et. al. [10], in order to tackle the information explosion problem, recommendation systems are helpful. Authors mentioned about the problems of data sparsity, cold start problem, scalability, etc. Authors have done a literature review of nearly 15 research papers related to movie recommendation system. After reviewing all these papers, they observed that most of the authors have used collaborative filtering rather than content-based filtering. Also, the authors noticed that a lot of authors have used hybrid-based approach. Even though a lot of research has been done on recommendation systems, there is always a scope for doing more in order to solve the existing drawbacks. Ms. Neeharika Immaneni, et. al. [11] proposed a hybrid recommendation technique which takes into consideration both content-based filtering approach as well as collaborative filtering approach in a hierarchial manner in order to show a personalized movie recommendation to the users. The most unique thing about this research work is that the authors have made movie recommendations using a proper sequence of images which actually describe the movie story plot. This actually helps for better visuals. The author have also described the graph based recommendation system, content-based approaches, hybrid recommender systems, collaborative filtering systems, genere correlations based recommender system, etc. The proposed algorithm has 4 major phases. Initially, social networking website like Facebook is used to know the user interest. Later, the movie reviews needs to be analysed and the recommendations needs to be made. Finally, story plot needs to be generated for better visuals. Md. Akter Hossain, et. al. [12] proposed NERS which is an acronym for neural engine-based recommender system. The authors have done a successful interaction between 2 datasets carefully. Moreover, the authors stated that the results of their system are better than the existing systems because they have incorporated the usage of general dataset as well as the behaviour-based dataset in their system. The authors have used 3 different estimators in order to evaluate their system against the existing systems.

CHAPTER 4
PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1 PROPOSED DESIGN

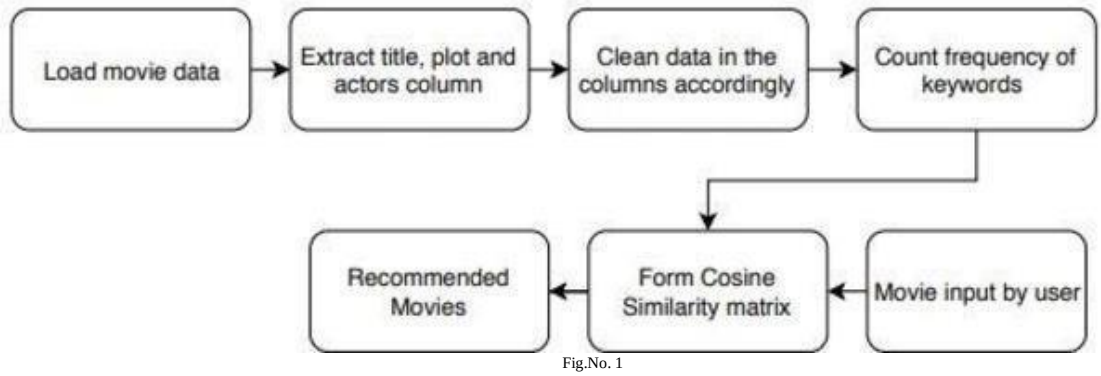

Fig.No. 1

- Fetching data:
  The IMDB dataset is used to generate movie recommendations based on the input movie by the user.
- Pre-processing the data:

  The selected columns are cleaned by removing punctuations and extra spaces. Then, keywords are extracted from the plot, which replaces the 'Plot' column.
- Building the model:

  Bag of words model is implemented by combining the actors, directors, and the keyword plots into a single column.

  After the table is generated, we count the frequency of each word in each row with the help of the count vectorizer. And form a cosine similarity matrix.

  A cosine similarity matrix looks at the frequency of common word between two movies and gives a rating from 0 to 1 accordingly
- Developing a website:

  The frontend of the application is built using Flask
- Deploying the website:Deployed on Heroku.

4.2 Work Flow

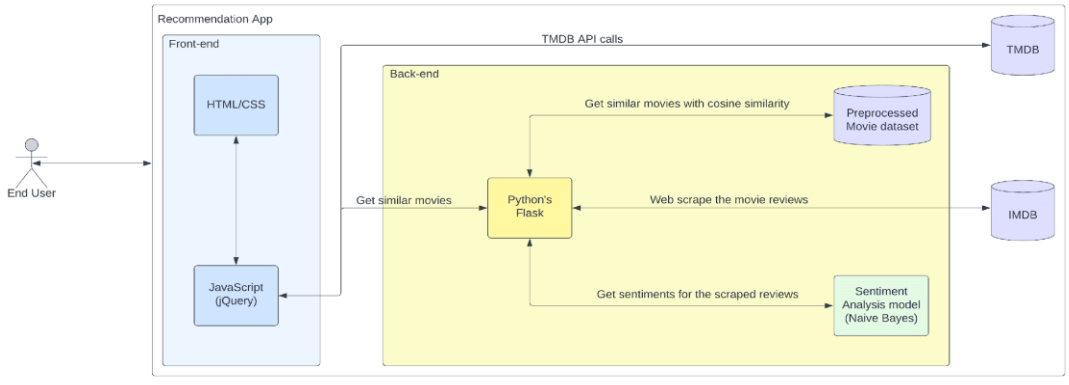

Fig.No.2

4.3 ASSUMPTIONS AND DEPENDENCY

Because of the hardware constraints, we would need very large storage to hold all of the data and we would need the cooperation of many entities to access their data, which is not practical, therefore we considered a tiny self-proctored sample of the data for our project. Due to these constraints, the training of the ML model will be restricted in the initial stages. Creating a synchronous application that works fluently on the website.

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. Here, the system has to perform the following tasks :

- Ask the user to enter the movie name and we have an auto complete function to suggest movies which are entered into the placeholder.
- On selecting the proper movie, the website displays the top 10 recommendations for the entered movie.
- And we also display the review of the movie based on the sentiment analysis of the comments.

5.2 NON-FUNCTIONAL REQUIREMENTS

5.2.1 Software Quality Attributes

The built project must provide availability, flexibility, usability, and a straightforward end product to the user with high efficiency, effectiveness, and satisfaction. When the user input the movie recommendation system should quickly display the results.

5.3 SOFTWARE REQUIREMENTS

- Jupyter notebook
- Python 2.7 or above versions
- Anaconda software/Google Colab
- Windows7 or higher OS

5.4 HARDWARE REQUIREMENTS

- Windows 8 or above
- Processor : Any processor above 500 MHz
- RAM : 512Mb
- Hard Disk : 10GB
- Input device : Standard Keyboard and Mouse
- Output device : High Resolution Monitor

CHAPTER 6
METHODOLOGY

CHAPTER 6 METHODOLOGY

- Loading movie data.
- Extract title plot and actors column.
- Cleaning data in the columns accordingly.
- Count frequency of keywords.
- Movie input by the user.
- Uses the concept of cosine similarity.
- The movie plots are transformed as vectors in a geometric space. Therefore the angle between two vectors represents the closeness of those two vectors.
- Cosine similarity calculates similarity by measuring the cosine of the angle between two vectors.
- A simple but elegant methodology to represent similarity between two entities in numerical value between 0 and 1.
- 0 meaning the entities are absolutely different, and 1 meaning both are identical.

PREDICTION ALGORITHM

- Our application uses the concept of cosine similarity, a simple but elegant methodology to represent similarity between two entities in a numerical value between 0 and 1, 0 meaning the entities are absolutely different, and 1 meaning both are identical. The formula for deriving cosine similarity is as follows

$$similarity(A,B) = \frac{\mathrm{A.B}}{||\mathrm{A}|.||\mathrm{B}||} = x = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- . First, the selected columns are cleaned by removing punctuations and extra spaces.
- Then, keywords are extracted from the plot, which replaces the ‘Plot’ column.
- Finally, the bag of words model is implemented by combining the actors, directors, and the keyword plots into a single column.
- After the above table is generated, we count the frequency of each word in each row with the help of the count Vectorizer. Find the index of the movie located in the similarity matrix.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

Accuracy Table

Algorithm	Accuracy
KNN	0.98
Decision tree classifier	0.99
SVM	0.99
Random forest classifier	0.99
Logistic Regression	0.99

	precision	recall	f1-score	support
0	0.98	0.97	0.98	638
1	0.97	0.99	0.98	746

Confusion Matrix

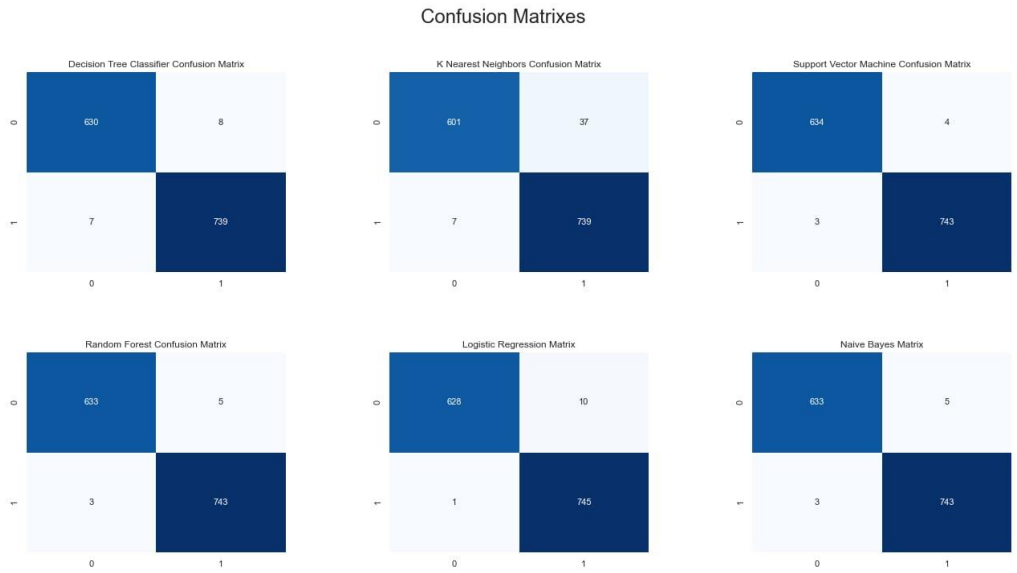

Fig. No.3

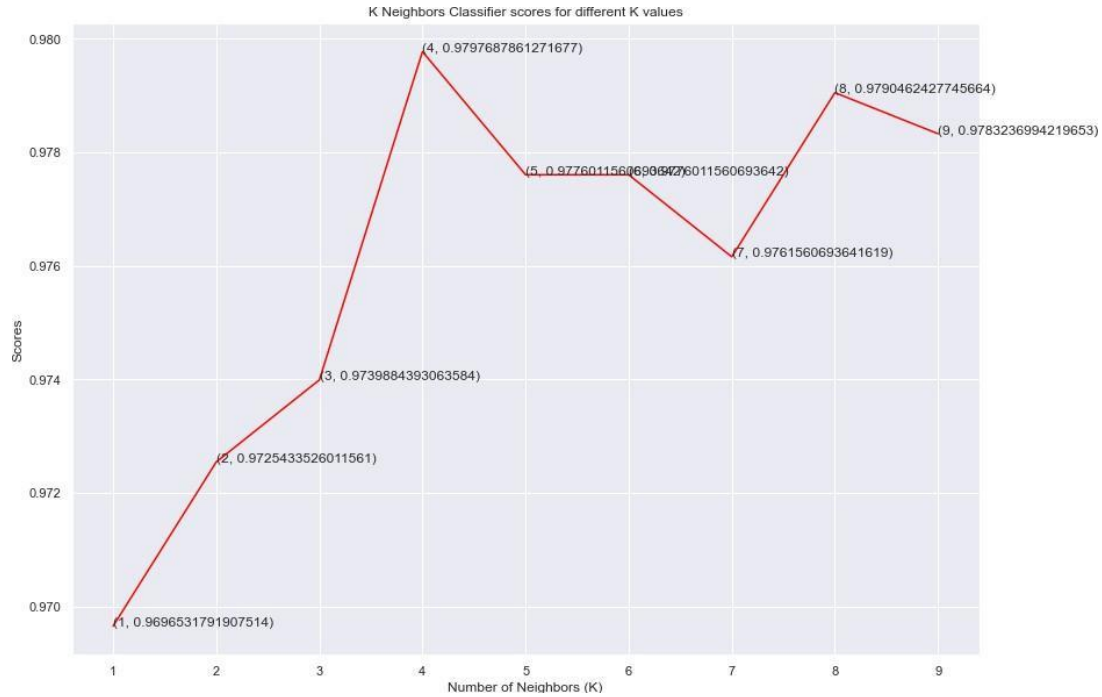

Fig. No.4

IMPLEMENTATION

```
In [45]: import numpy as np
         import pandas as pd

In [46]: movies = pd.read_csv('tmdb_5000_movies.csv')
         credits = pd.read_csv('tmdb_5000_credits.csv')

In [47]: movies.head(2)
```

Out[47]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_compan
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ingeni Film Partners", " 28
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Walt Dis Pictures", "id": 2},

```
In [48]: movies.shape
```

Out[48]: (4803, 20)

```
In [49]: credits.head()
```

Out[49]:

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
In [50]: movies = movies.merge(credits,on='title')
```

Fig. No.5

```
In [52]: movies = movies[['movie_id','title','overview','genres','keywords','cast','crew']]

In [53]: movies.head()
```

Out[53]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
In [54]: import ast

In [55]: def convert(text):
             L = []
             for i in ast.literal_eval(text):
                 L.append(i['name'])
             return L

In [56]: movies.dropna(inplace=True)

In [57]: movies['genres'] = movies['genres'].apply(convert)
         movies.head()
```

Out[57]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

Fig.No.6

APPLICATION

e-Commerce

Is an industry where recommendation systems were first widely used. With millions of customers and data on their online behavior, e-commerce companies are best suited to generate accurate recommendations.

Retail

Target scared shoppers back in the 2000s when Target systems were able to predict pregnancies even before mothers realized their own pregnancies. Shopping data is the most valuable data as it is the most direct data point on a customer's intent. Retailers with troves of shopping data are at the forefront of companies making accurate recommendations.

Media

Similar to e-commerce, media businesses are one of the first to jump into recommendations. It is difficult to see a news site without a recommendation system.

Banking

A mass-market product that is consumed digitally by millions. Banking for masses and SMEs are prime for recommendations. Knowing a customer's detailed financial situation, along with their past preferences, coupled with data of thousands of similar users is quite powerful.

Telecom

It Shares similar dynamics with banking. Telcos have access to millions of customers whose every interaction is recorded. Their product range is also rather limited compared to other industries, making recommendations in telecom an easier problem.

Utilities

Similar dynamics with telecom but utilities have an even narrower range of products, making recommendations rather simple.

CHAPTER 8
TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

```
In [81]: recommend('Gandhi')

Gandhi, My Father
The Wind That Shakes the Barley
A Passage to India
Guiana 1838
Bloody Sunday

In [84]: recommend('Iron Man')

Iron Man 2
Iron Man 3
Avengers: Age of Ultron
Captain America: Civil War
The Avengers

In [87]: recommend('Inception')

Duplex
The Helix... Loaded
Star Trek II: The Wrath of Khan
Timecop
Chicago Overcoat

In [93]: recommend('The Silence of the Lambs')

Red Dragon
Suspect Zero
Eddie: The Sleepwalking Cannibal
Blood Work
Hannibal Rising
```

Fig.No.7

Result Screenshots

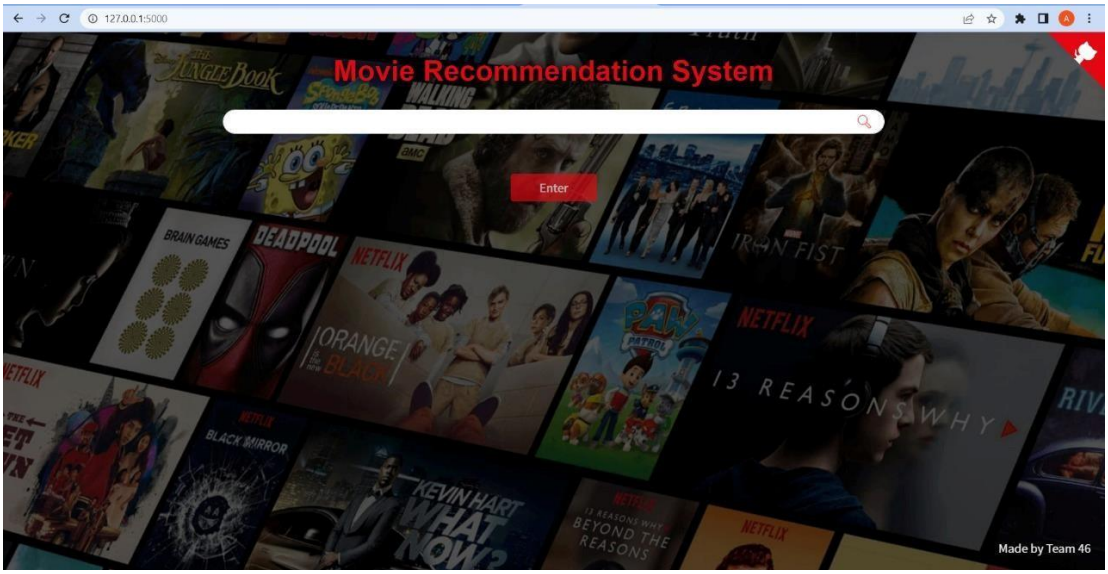

Fig.No.8

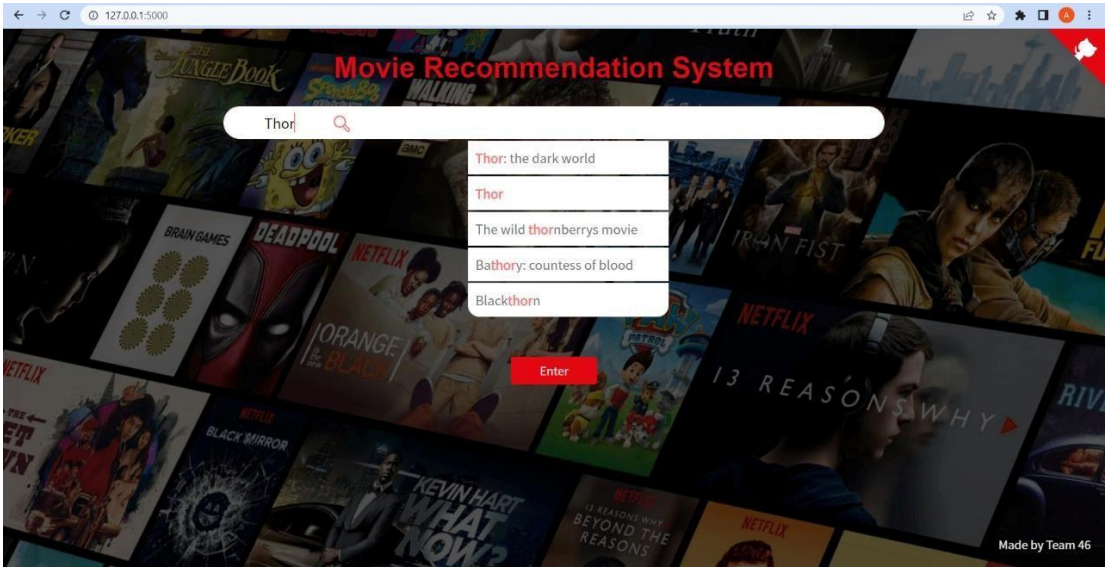

Fig.No.9

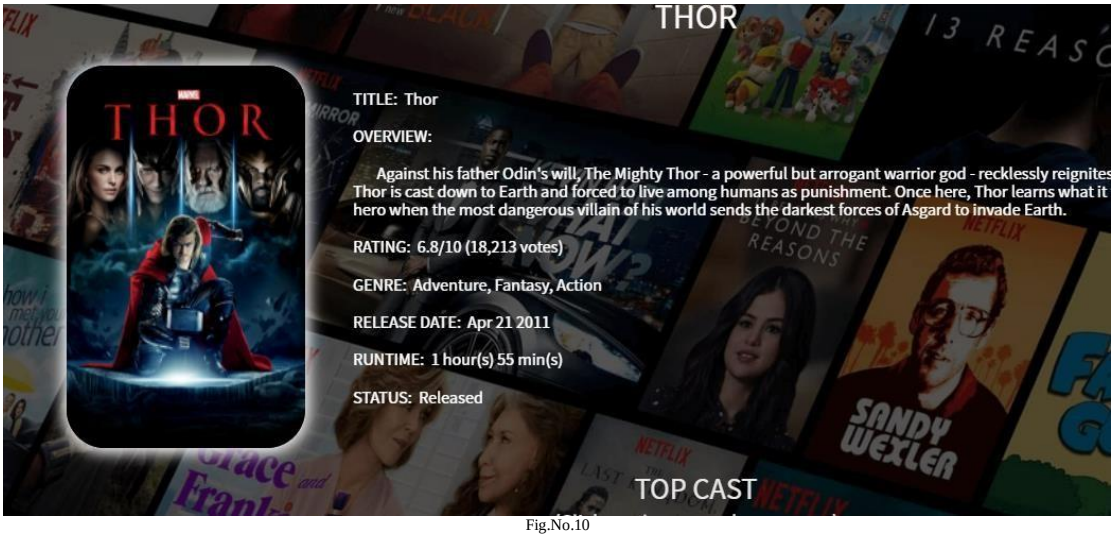

Fig.No.10

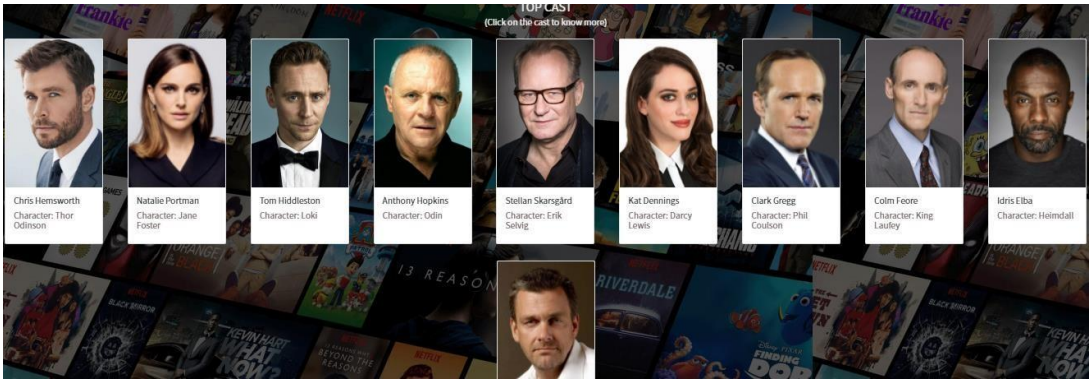

Fig.No.11

Comments	Sentiments
Thor is a film that takes risks - and they pay off wonderfully. The movie grounds itself in reality while keeping a mythological and fantasy-esque tone, and it never loses sight of the humanity within the characters, with elevates this above standard superhero fare.	Good :
Who knew Thor was the perfect drinking buddy and is domesticated lol? Perfect writing. Perfect directing. Perfect casting. Perfect cinematography. Perfect amount of S/WFF. Perfect score. Perfect comedy/drama/thrills. Etc... 10/10	Good :
Great movie! The storyline is very well done! I love how they show Asgard and explain how everything fits into place. The humor is very tasteful, mixed with a good amount of action. The only thing that could have used some improving was the way they filmed the film. The shakey camera and weird angles are a little annoying, but the rest of the film makes up for this. Very good choice in the cast, could not have picked a better actor for Thor or Loki. Natalie Portman's part isn't the best. She has cheesy lines and her acting doesnt flow right, even though she is a talented actress. She does better with serious parts instead of light and comical. This movie is very clean and makes a great fun movie for the entire family.	Good :
I'm always a fan of the first films with each character because of the originality. I'm watching them again from the start and Thor is not as good as Iron Man but is still a very good film. The relationship between Thor and Loki is a massive part of the MCU story and is at the forefront of this film, still really good action and continuation of the MCU.	Good :
The Norse god Thor is admittedly one of Marvel's second-league superheroes, but just like Iron Man, both are inevitable steps in Marvel's continuing march towards arguably the most highly anticipated comic book adaptation in movie history next year - "The Avengers". Yet, if there's anything to learn from the "Iron Man" experience, it's that it's entirely possible to turn a B-league superhero into an A-grade feature film with just the right star and the right director. And kudos to the Marvel team for doing it once again with "Thor", a thoroughly entertaining blockbuster of a film that succeeds thanks to the unlikely choices of Kenneth Branagh as director and Chris Hemsworth as the star of the titular character. Kenneth is the English director best known as cinema's foremost exponent of the Bard, while Chris is the Australian actor probably only known as the father to James T. Kirk in last year's "Star Trek". Both are equally unlikely choices to headline this US$150mil tentpole, though Marvel's gamble has paid off handsomely. Unlike Iron Man, Incredible Hulk, or for that matter the upcoming Captain America who were products of science, Thor was the hammer-wielding alien-god who was always going to be a tricky proposition adapting for the big screen. After all, the combination of Norse mythology and science fiction could easily have degenerated into kitschy camp, so it's a good thing that Kenneth knows when to get serious and when to have fun. He films the scenes of Thor in his homeland Asgard with the utmost gravitas, while leaving the humour for his brief fish-out-of-water stint on Earth. Indeed, Kenneth handles the power feuds and sibling rivalry with the same intensity as he would a Shakespearian production, and brings a deeply human perspective to his godly characters. In particular, the tussle between father and sons is emotionally charged - whether Odin's heartbreak at banishing his son Thor, whom he was about to coronate as king, out of the kingdom for arrogantly and impetuously reigniting a war with an ancient enemy; or Loki's confrontation with Odin about his true origins - and therefore electrifying to watch. To add weight to the proceedings at Asgard, Kenneth brings in two excellent character actors alongside Chris- Anthony Hopkins and Tom Hiddleston. Anthony's patriarchal role isn't new for the actor, but he delivers a restrained yet powerful and dignified performance as Thor's father Odin. Tom, on the other hand, is deliciously snarky playing the sibling jealous of Thor's impending accession to the throne, and the British actor gives a surprisingly intimate and affecting villainous turn that brings unexpected depth to his character. In contrast to the godly solemnity, Kenneth brings some welcome Earthly humour to Thor's subsequent banishment. The trio of scriptwriters Ashley Edward Miller, Zack Stentz and Don Payne (working from an effective origin story by J. Michael Straczynski and Mark Protosevich) reserve the movie's wittiest lines for Thor's interplay with three mortal beings- astrophysicist Jane Foster (Natalie Portman), her gal-pal/assistant Darcy Lewis (Kat Dennings) and Jane's mentor, Dr. Erik Selvig (Stellan Skarsgard), while Kenneth directs Thor's awkward adjustment to life on Earth (an absolutely hilarious scene has Thor storming into a pet store and asking for a horse to go about his mission to find his mysterious magical hammer Mjolnir) with just the right amount of infectious fun. There's also an interplanetal romance thrown in for good measure, as Thor and Jane develop a growing attraction for each other. But more than just obligatory, the love between god and human becomes a lesson in humility and self-sacrifice for the fallen Thor to pave his restoration back to glory. Though thinly written, Portman makes the best out of her role, and it is to her credit that Jane becomes an engaging and believable character that the audience roots for. Portman also shares a warm chemistry with Chris, which brings poignancy to their relationship as well as Thor's subsequent transformation. Thor's time on Earth is also used to forge the necessary narrative links with "The Avengers", in the form of Clark Gregg's S.H.I.E.L.D. agent Coulson, an unbilled cameo by Jeremy Renner as the archer Hawkeye, and Samuel L. Jackson's Nick Fury appearance right at the end of the credits. Fans of the Thor universe from the comic books will also delight in the awe-inspiring presence of Heimdall (Idris Elba), as well as Thor's entourage of Asgard warrior pals Sif (Jaimie Alexander), Volstagg (Ray Stevenson), Fandral (Josh Dallas), and Hogun (Tadanobu Asano). These pals also add colour to the action sequences, which Kenneth choreographs with surprising aplomb. He stages a thunderous battle very early on as Thor leads an attack against the Frost Giants on the planet of Jotunheim with every bit of weight and muscle one would expect from a clash of godly beings. There are at least two other money sequences - Thor's duel with a hulking piece of metal known as the Destroyer (which Coulson first tongue-in-cheek mistakes to be one of Stark's equipment) in a small New Mexico town; and the thrilling climactic battle between Thor and Loki back on Asgard. But the star that holds it all together is really Chris Hemsworth, his performance a well-calibrated blend of brawn and heart that will put sceptics to rest. His physicality is apparent right from the start, but Chris gradually reveals a winning charm and a heartfelt sincerity that enables his audience to relate with his character's inner struggles. It is a breakout star-making turn, which shows Chris adept at action, comedy and even romance. And certainly, it is to Kenneth Branagh's credit that these disparate elements can belong so comfortably in a movie that also combines mythology and science fiction. True to its origins, "Thor" is nothing less than superhero entertainment, and a definitive A-list one - despite the initial regard of its source character that kicks off this summer with a bang, boom and rumble.	Good :
Good movie, my time has well spend on this, so I recommend this to others too :)	Good :
Though I'm old and did read some comics in my time, Thor was never one of them. I do teach some Norse mythology, though, so that aspect interested me. There was an odd steampunk approach to the mythology, which was OK, I suppose. I really didn't expect this movie to be War and Peace, and I was certainly right on the money with that. It went for the snarky Iron Man approach more than I thought it would. It's hard to pull off without veteran snarkmeister Robert Downey, Jr., though. Chris Hemsworth is impossibly charming and good-looking in the title role, and does pretty well, considering some of the lines he had to say. Anthony Hopkins is a safe choice for his patented Older Mentor/Authority Figure performance (enjoy that check, Anthony), and Tom Hiddleston is somewhat intriguing as Loki. Stellan Skarsgard is completely wasted in a role that could have been phoned in, but you know who really stinks up the joint? Natalie Portman. It was DAMN hard to believe she was an astrophysicist, but it was even harder to believe that THOR (god of thunder, possessor of unspeakable power) would find her even remotely interesting. All she does is whine. The alleged falling in love is quite unbelievable, BTW. Yeesh. I hope the inevitable sequel loses her character.	Good :
Having just seen the global premiere of Thor - God of Thunder, in Sydney Australia, all I can say is.....wow! Don't worry readers I am not about to give away one single spoiler, however after some time and anticipation, Thor is a winner. Australian Chris Hemsworth graced the red carpet in all his native glory, talking up the film and his future entity in The Avengers. He was joined by Shakespeare aficionado Director Kenneth Branagh and co stars Jamie Alexander & Tom Hiddleston who plays crowd favourite Loki. Thor is a good movie, plenty of action, a lovely heroine (Natalie Portman) and some interesting sidekicks. Costumes rock, not to mention the obligatory Marvel characters making significant cameos, not just the extra scene after the credits, but within the film itself. I was a little over	Bad :

Fig.No.12

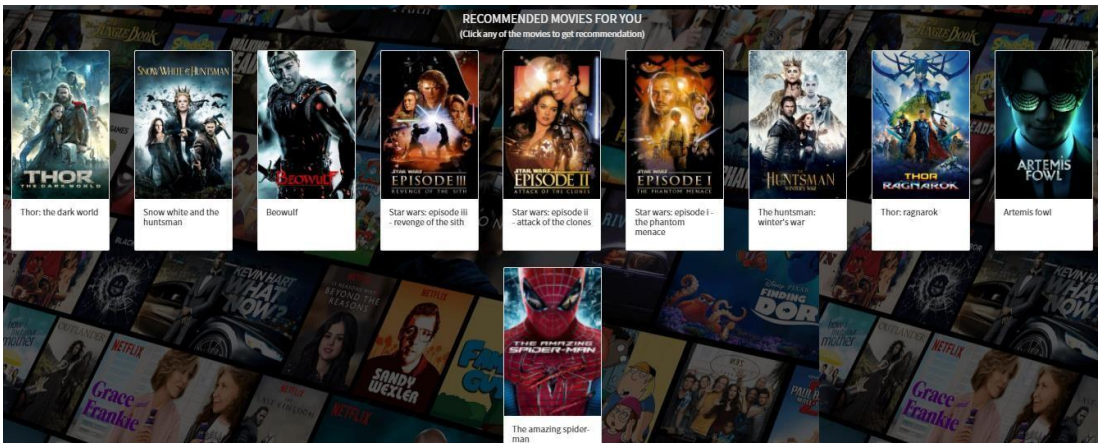

Fig.No.13

Ask the user to enter the movie name and we have an auto complete function to suggest movies which are entered into the placeholder.
On selecting the proper movie, the website displays the top 5 recommendations for the entered movie. And we also display the review of the movie based on the sentiment analysis of the comments.

Conclusion

A study has shown that the total amount of time a person spends deciding which movie to watch due to the huge number of choices, that time could be utilized in watching at least half of a full-length movie. Hence an average customer would be saving about an hour of his time every week, doing something more productive. To encounter this we have come up with our content-Based movie Recommender System which recommends movies similar to the movie user likes and analyses the sentiments on the reviews given by the user for that movie.We have used k-NN and various metrics to evaluate the improvement of the new approach.

The user interface has been designed to promote user interactivity with the application. Due to the simple and compact UI, future changes can be made in the frontend with ease future work could focus on eliminating the limitation of the IMDB dataset.
We have developed the website using Flask which is a python framework and we have deployed on heroku and website loading time is about 40 seconds .

REFERENCES

[1] Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 263-268). IEEE.

[2] Liu, G., & Wu, X. (2019, March). Using Collaborative Filtering Algorithms Combined with Doc2Vec for Movie Recommendation. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 1461-1464). IEEE.

[3] Nakhli, R. E., Moradi, H., & Sadeghi, M. A.(2019) Proposed a simple way which used the correlation between the likes and percentage of views for the movies and filtering algorithms

[4] Hossain, M. A., & Uddin, M. N. (2018, September). A Neural Engine for Movie Recommendation System. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT) (pp. 443-448). IEEE.

[5] Zhang, R., & Mao, Y. (2019). Movie Recommendation via Markovian Factorization of Matrix Processes. IEEE Access, 7, 13189-13199.

[6] Chen, Xiaojie, Pengpeng Zhao, Yanchi Liu, Lei Zhao, Junhua Fang, Victor S. Sheng, and Zhiming Cui. "Exploiting Aesthetic Features in Visual Contents for movie recommendations." IEEE Access 7(2019):49813- 49821

[7] Kim, M., Jeon, S., Shin, H., Choi, W., Chung, H., & Nah, Y. (2019, June). Movie Recommendation based on User Similarity of Consumption Pattern Change. In 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (pp. 317-319).

[8] IEEE Ifada, N., & Prasetyo, E. H. (2018, November). Employing Sparsity Removal Approach and Fuzzy C-Means Clustering Technique on a Movie Recommendation System. In 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM) (pp. 329-334). IEEE

[9] Yeole Madhavi B.1 , Rokade Monika D.2 , Khatal Sunil S.3,Movie Recommendation System using Content based Filtering using Content-based Filtering, Text to vector, Vector similarity, Hybrid approach

[10] R. Lavanya, U. Singh and V. Tyagi, "A Comprehensive Survey on Movie Recommendation Systems," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 532-536, doi: 10.1109/ICAIS50930.2021.9395759.

[11] N. Immaneni, I. Padmanaban, B. Ramasubramanian and R. Sridhar, "A meta-level

hybridization approach to personalized movie recommendation," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 2193-2200, doi: 10.1109/ICACCI.2017.8126171.

[12] M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), 2018, pp. 443-448, doi: 10.1109/CEEICT.2018.8628128.

[13] Monika D.Rokade ,Dr.Yogesh kumar Sharma,"Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic." IOSR Journal of Engineering (IOSR JEN), ISSN (e): 2250- 3021, ISSN (p): 2278-8719

GITHUB LINK: https://github.com/durgaanusha/majorproject

APPENDIX A

BAG OF WORDS

Bag of Words (BOW) is a method to extract features from text documents. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in all the documents in the training set.

APPENDIX B

IMDB

IMDb is an online database of information related to films, television series, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews.

APPENDIX C

KNN

K-Nearest Neighbor (KNN) is a supervised algorithm in machine learning that is used for classification and regression analysis. This algorithm assigns the new data based on how close or how similar the data is to the points in training data. Here, 'K' represents the number of neighbors that are considered to classify the new data point.

APPENDIX D

Source code:

KNN ALGORITHM

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
knn=KNeighborsClassifier()
knn.fit(X_train,y_train)
KNeighborsClassifier()
prediction_knn=knn.predict(X_test)
```

```
accuracy_knn= accuracy_score(y_test,prediction_knn)*100
scores_dict['KNeighborsClassifier'] = accuracy_knn
accuracy_knn
97.76011560693641
print(classification_report(y_test,prediction_knn))
              precision    recall  f1-score  support

           0       0.98      0.97      0.98       638
           1       0.97      0.99      0.98       746

    accuracy                           0.98      1384
   macro avg       0.98      0.98      0.98      1384
weighted avg       0.98      0.98      0.98      1384
k_range=range(1,26)
scores={}
h_score = 0       # to find the best score
best_k=0          # to find the best k
scores_list=[]

for k in k_range:
    knn=KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train,y_train)
    prediction_knn=knn.predict(X_test)
    scores[k]=accuracy_score(y_test,prediction_knn)
    if scores[k]>h_score:
        h_score = scores[k]
```

```
best_k = k
    scores_list.append(accuracy_score(y_test,prediction_knn))
print('The best value of k is {} with score : {}'.format(best_k,h_score*100))
The best value of k is 4 with score : 97.97687861271676
knn=KNeighborsClassifier(n_neighbors=best_k)
knn.fit(X_train,y_train)
KNeighborsClassifier(n_neighbors=4)
import matplotlib.pyplot as plt
plt.figure(figsize=(14,10))
knn_scores = []
for k in range(1,10):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    knn_classifier.fit(X_train, y_train)
    knn_scores.append(knn_classifier.score(X_test, y_test))

plt.plot([k for k in range(1, 10)], knn_scores, color = 'red')
for i in range(1,10):
    plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
plt.xticks([i for i in range(1, 10)])

plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Scores')
plt.title('K Neighbors Classifier scores for different K values')
Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')
```