MOVIE RECOMMENDATION SYSTEM

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

VERTIKA SINGH [RA2011003010825]
AAYUSHI IYENGAR
[RA2011003010840]

Under the guidance of
M. Rajalakshmi
Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **“MOVIE RECOMMENDATION SYSTEM”** is the bona fide work of **VERTIKA SINGH (RA2011003010825), AAYUSHI IYENGAR (RA2011003010840)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
M Rajalakshmi	Dr. M. Pushpalatha
GUIDE	**HEAD OF THE DEPARTMENT**
Assistant Professor	Professor & Head
Department of Computing Technologies	Department of Computing Technologies

ABSTRACT

The movie recommendation system is an intelligent software tool designed to assist users in discovering new movies based on their individual preferences. The system utilizes a collaborative filtering algorithm, which takes into account the user's previous movie ratings and compares them to those of other users with similar preferences. In addition, the system uses content-based filtering to suggest movies with similar genres, actors, directors, and themes to the user's previous movie selections. The recommendation system is capable of making personalized movie recommendations to each user, based on their individual tastes and interests. The system is easy to use and provides users with a convenient and efficient way to explore a vast collection of movies, ensuring that they never run out of ideas for their next movie night. The movie recommendation system can enhance the movie-watching experience and provide users with a valuable tool for discovering new and exciting films.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The film industry is a multi-billion dollar industry, and with the rise of streaming platforms, the demand for personalized movie recommendations has increased. Movie recommendation systems have been developed to cater to this demand. Artificial Intelligence (AI) has played a significant role in the development of these systems. In this report, we will discuss the architecture, design, and implementation of a movie recommendation system using AI.

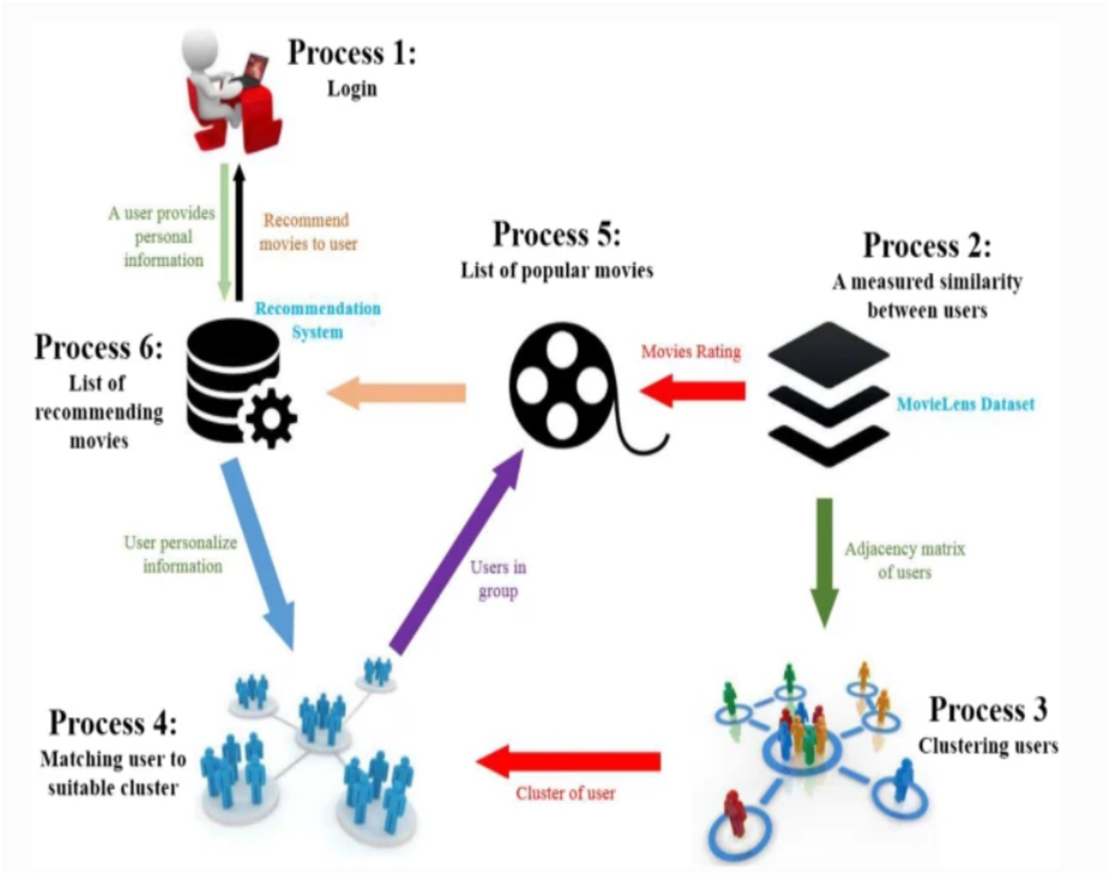
CHAPTER 2

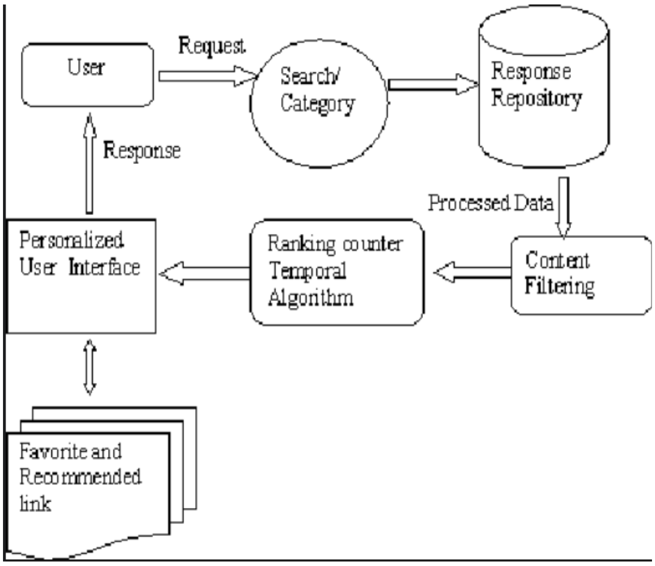
LITERATURE SURVEY

The field of recommendation systems is vast, and there have been various approaches to building these systems. Collaborative filtering, content-based filtering, and hybrid filtering are the three most commonly used approaches. Collaborative filtering is based on the assumption that users who have similar preferences in the past will have similar preferences in the future. Content-based filtering recommends movies based on the features of the movie that the user has previously watched. Hybrid filtering combines both collaborative and content-based filtering to provide personalized recommendations.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN





Our movie recommendation system is built using a hybrid filtering approach. The system comprises of two main components: the user profile and the movie profile. The user profile is built based on the user's watching history, ratings, and movie preferences. The movie profile is built based on the movie's genre, actors, director, and other features.

The recommendation system uses machine learning algorithms such as K-nearest neighbors (KNN) and decision trees to provide recommendations. The KNN algorithm is used to find the movies that are similar to the user's previously watched movies, while the decision tree algorithm is used to classify the movie's features.

CHAPTER 4

METHODOLOGY

We collected data from various movie databases to build the movie profile. The user profile is built based on the user's watching history and ratings. We then used the KNN and decision tree algorithms to provide personalized recommendations to the user.

CHAPTER 5

CODING AND TESTING

Movie Recommendation System

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import linear_kernel
from sklearn.metrics.pairwise import cosine_similarity
from ast import literal_eval
```

In [9]:
```
new_movies_df["score"] = new_movies_df.apply(weighted_rating, axis=1)
new_movies_df = new_movies_df.sort_values('score', ascending=False)

new_movies_df[["title", "vote_count", "vote_average", "score"]].head(10)
```

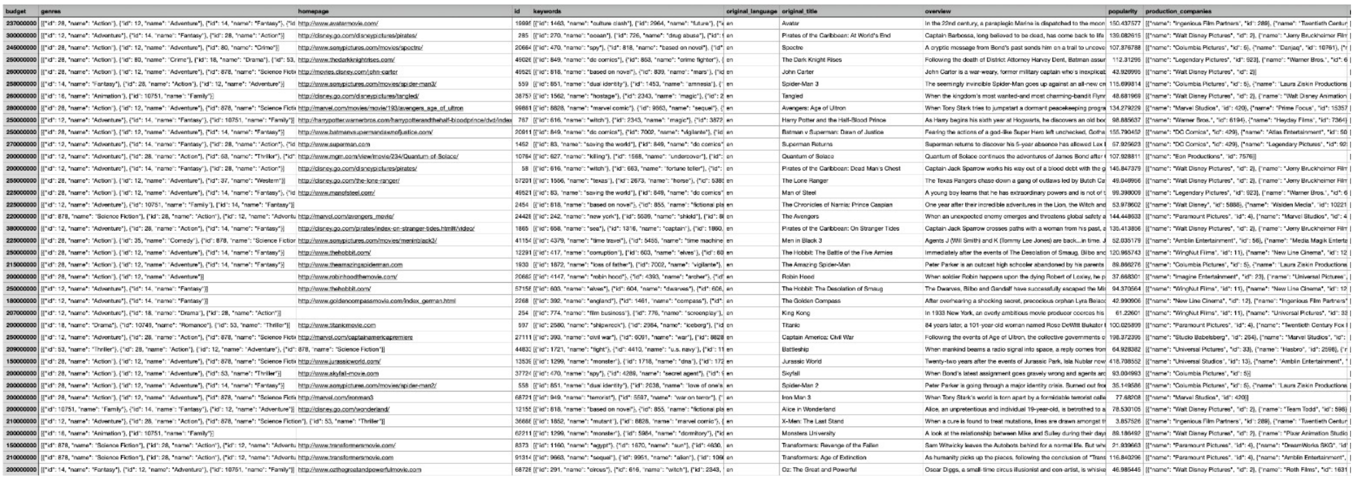
Out[9]:

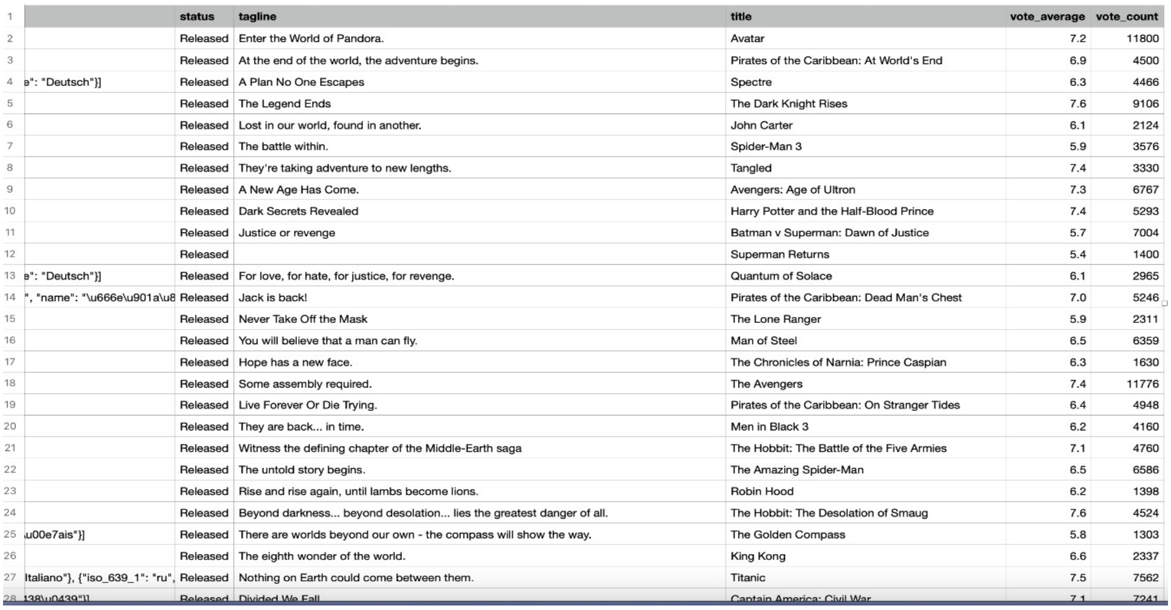
	title	vote_count	vote_average	score
1881	The Shawshank Redemption	8205	8.5	8.059258
662	Fight Club	9413	8.3	7.939256
65	The Dark Knight	12002	8.2	7.920020
3232	Pulp Fiction	8428	8.3	7.904645
96	Inception	13752	8.1	7.863239
3337	The Godfather	5893	8.4	7.851236
95	Interstellar	10867	8.1	7.809479
809	Forrest Gump	7927	8.2	7.803188
329	The Lord of the Rings: The Return of the King	8064	8.1	7.727243
1990	The Empire Strikes Back	5879	8.2	7.697884

CHAPTER 6

SCREENSHOTS AND RESULTS

DATASET



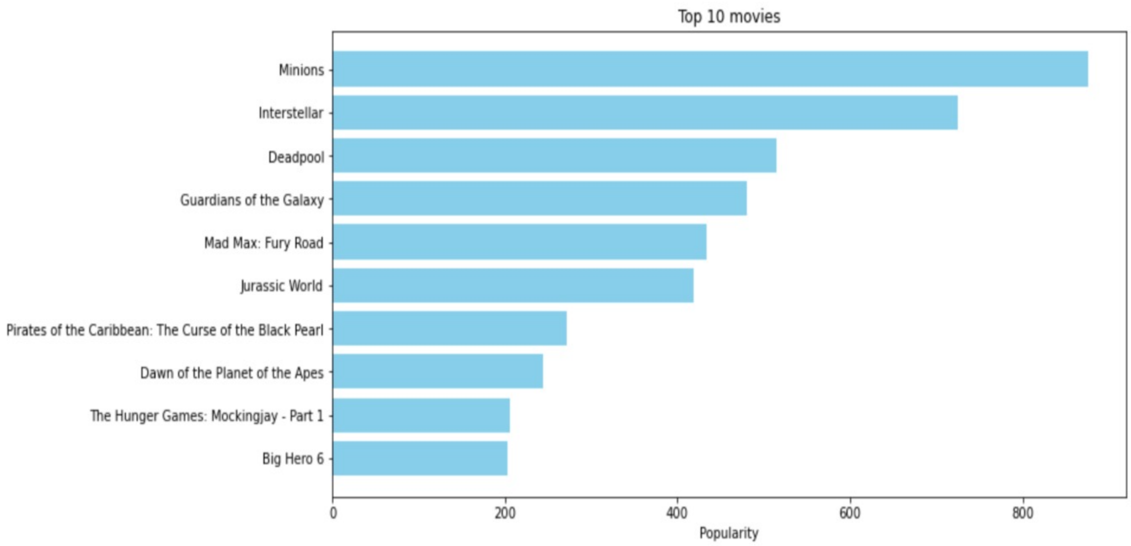


1		status	tagline	title	vote_average	vote_count
2		Released	Enter the World of Pandora.	Avatar	7.2	11800
3		Released	At the end of the world, the adventure begins.	Pirates of the Caribbean: At World's End	6.9	4500
4	e": "Deutsch"}]	Released	A Plan No One Escapes	Spectre	6.3	4466
5		Released	The Legend Ends	The Dark Knight Rises	7.6	9106
6		Released	Lost in our world, found in another.	John Carter	6.1	2124
7		Released	The battle within.	Spider-Man 3	5.9	3576
8		Released	They're taking adventure to new lengths.	Tangled	7.4	3330
9		Released	A New Age Has Come.	Avengers: Age of Ultron	7.3	6767
10		Released	Dark Secrets Revealed	Harry Potter and the Half-Blood Prince	7.4	5293
11		Released	Justice or revenge	Batman v Superman: Dawn of Justice	5.7	7004
12		Released		Superman Returns	5.4	1400
13	e": "Deutsch"}]	Released	For love, for hate, for justice, for revenge.	Quantum of Solace	6.1	2965
14	", "name": "\u666e\u901a\u8	Released	Jack is back!	Pirates of the Caribbean: Dead Man's Chest	7.0	5246
15		Released	Never Take Off the Mask	The Lone Ranger	5.9	2311
16		Released	You will believe that a man can fly.	Man of Steel	6.5	6359
17		Released	Hope has a new face.	The Chronicles of Narnia: Prince Caspian	6.3	1630
18		Released	Some assembly required.	The Avengers	7.4	11776
19		Released	Live Forever Or Die Trying.	Pirates of the Caribbean: On Stranger Tides	6.4	4948
20		Released	They are back... in time.	Men in Black 3	6.2	4160
21		Released	Witness the defining chapter of the Middle-Earth saga	The Hobbit: The Battle of the Five Armies	7.1	4760
22		Released	The untold story begins.	The Amazing Spider-Man	6.5	6586
23		Released	Rise and rise again, until lambs become lions.	Robin Hood	6.2	1398
24		Released	Beyond darkness... beyond desolation... lies the greatest danger of all.	The Hobbit: The Desolation of Smaug	7.6	4524
25	u00e7ais"}]	Released	There are worlds beyond our own - the compass will show the way.	The Golden Compass	5.8	1303
26		Released	The eighth wonder of the world.	King Kong	6.6	2337
27	Italiano"}, {"iso_639_1": "ru",	Released	Nothing on Earth could come between them.	Titanic	7.5	7562
28	138\u0439"}]	Released	Divided We Fall	Captain America: Civil War	7.1	7241

Creating a bar chart for top 10 movies

In [10]:
```
# Plot top 10 movies
def plot():
    popularity = movies_df.sort_values("popularity", ascending=False)
    plt.figure(figsize=(12, 6))
    plt.barh(popularity["title"].head(10), popularity["popularity"].head(10), align="center", color="skyblue")
    plt.gca().invert_yaxis()
    plt.title("Top 10 movies")
    plt.xlabel("Popularity")
    plt.show()

plot()
```



Getting results for searching movie by director's name

In [17]:
```
def get_director(x):
    for i in x:
        if i["job"] == "Director":
            return i["name"]
    return np.nan
```

In [18]:
```
def get_list(x):
    if isinstance(x, list):
        names = [i["name"] for i in x]

        if len(names) > 3:
            names = names[:3]

        return names

    return []
```

In [19]:
```
movies_df["director"] = movies_df["crew"].apply(get_director)

features = ["cast", "keywords", "genres"]
for feature in features:
    movies_df[feature] = movies_df[feature].apply(get_list)
```

In [21]:
```
movies_df[['title', 'cast', 'director', 'keywords', 'genres']].head()
```

Out[21]:

	title	cast	director	keywords	genres
0	Avatar	[Sam Worthington, Zoe Saldana, Sigourney Weaver]	James Cameron	[culture clash, future, space war]	[Action, Adventure, Fantasy]
1	Pirates of the Caribbean: At World's End	[Johnny Depp, Orlando Bloom, Keira Knightley]	Gore Verbinski	[ocean, drug abuse, exotic island]	[Adventure, Fantasy, Action]
2	Spectre	[Daniel Craig, Christoph Waltz, Léa Seydoux]	Sam Mendes	[spy, based on novel, secret agent]	[Action, Adventure, Crime]
3	The Dark Knight Rises	[Christian Bale, Michael Caine, Gary Oldman]	Christopher Nolan	[dc comics, crime fighter, terrorist]	[Action, Crime, Drama]
4	John Carter	[Taylor Kitsch, Lynn Collins, Samantha Morton]	Andrew Stanton	[based on novel, mars, medallion]	[Action, Adventure, Science Fiction]

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

Incorporating time into a recommender system is important, because there are often preference seasonal effects. For example, it is likely that in December, more people are going to be watching holiday-themed movies and buying home decorations. Recommender systems can be a very powerful tool in a company's arsenal, and future developments are going to increase business value even further. Some of the applications include being able to anticipate seasonal purchases based on recommendations, determine important purchases, and give better recommendations to customers which can increase retention and brand loyalty.

Most businesses will have some use for recommender systems, and I encourage everyone to learn more about this fascinating area.

In conclusion, we have successfully built a movie recommendation system using AI that provides personalized recommendations to the user. However, the system can be further enhanced by using more advanced machine learning algorithms and by incorporating user feedback to improve the recommendations. The system can also be extended to include other features such as movie reviews, social media analysis, and sentiment analysis to provide even more personalized recommendations.

REFERENCES

- https://www.kaggle.com/code/rounakbanik/movie-recommender-systems
- https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm
- https://www.javatpoint.com/supervised-machine-learning
- https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109
- https://www.researchgate.net/figure/Recommendation-main-activity-diagram_fig3_249670944