MOVIE RECOMMANDATION SYSTEM

A Minor Project Report

in partial fulfillment of the degree

Bachelor of Technology
in
Computer Science & Artificial Intelligence

By

2103A52126 Sanjana Bimireddy

2103A52165 Akshitha Reddy Pailla

2103A52072 Sreenija Danda

Under the Guidance of

Srinivas Aluvala

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE
SR UNIVERSITY, ANANTHASAGAR, WARANGAL
April, 2024.

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

CERTIFICATE

This is to certify that this project entitled **"MOVIE RECOMMENDATION SYSTEM"** is the bonafied work carried out by **B.SANJANA,P.AKSHITHA REDDY,D.SREENIJA** as a Minor Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE** during the academic year 2022-2023 under our guidance and Supervision.

Dr. Srinivas Aluvala
Designation,
SR University,
Ananthasagar, Warangal.

Dr. M.Sheshikala
Assoc. Prof. & HOD (CSE),
SR University,
Ananthasagar, Warangal.

External Examiner

ACKNOWLEDGEMEN

We owe an enormous debt of gratitude to our project guide **Dr. Srinivas Aluvala, Assistant Professor** as well as Head of the CSE Department **Dr. M.Sheshikala, Associate Professor** for guiding us from the beginning through the end of the Minor Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to the project co-ordinators **Dr. P Praveen, Assoc. Prof** for their encouragement and support.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, **Dr. C.V. Guru Rao,** for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

Sanjana Bimireddy

Akshitha Reddy Pailla

Sreenija Danda

ORGANIZATION OF THESIS

1. Title page
2. Certificate
3. Acknowledgement
4. Abstract
5. Table of Contents

The content should be:

1. INTRODUCTION
    1.1. EXISTING SYSTEM
    1.2. PROPOSED SYSTEM
2. LITERATURE SURVEY
3. DESIGN
    3.1. REQUIREMENT SPECIFICATION(S/W & H/W)
4. IMPLEMENTATION
5. CODING AND RESULTS
6. CONCLUSION
7. FUTURE SCOPE

ABSTRACT

The ever-growing library of movies can make choosing what to watch overwhelming. This project aims to develop a movie recommendation system that suggests films tailored to individual user preferences. The system will leverage machine learning techniques to analyze user data, such as past watch history and ratings, to identify patterns and correlations.Social media marketing heavily relies on artificial intelligence. It also covers any intelligence demonstrated by a computer, a robot, or any other machine that resembles human intellect. This study examines how artificial intelligence affects Recommendation systems.They employ machine learning and AI to serve users the material that interests them, identify visuals, recommend tag choices, recognize people in photos, and serve adverts to generate user-specific offers and promotions.Recommendation System is a major area which is very popular and useful for people to take proper automated decisions. It is a method that helps user to find out the information which is beneficial to him/her from variety of data available. When it comes to Movie Recommendation System, recommendation is done based on similarity between users (Collaborative Filtering) or by considering particular user’s activity (Content Based Filtering) which he wants to engage with. To overcome the limitations of collaborative and content based filtering generally, combination of collaborative and content based filtering is used so that a better recommendation system can be developed.

INTRODUCTION

In the vast ocean of cinema, finding the perfect movie to watch can feel like searching for a hidden treasure. We all have our favorite genres, directors, and actors, but sometimes we crave something new or simply can't decide. This is where movie recommendation systems come in, acting as personal guides to cinematic discovery. Machine learning has a subclass known as recommendation engines that often rankor rate people or items. A recommended system, broadly defined, is a system that anticipates the ratings a user would give to a certain item. These predictions will then be ranked and returned back to the user. They're used by various large name companies like Google, Instagram, Spotify, Amazon, Reddit, Netflix etc. often to increase engagement with users and the platform.Information that was relevant to the user's interests and preferences was missing. As a result, recommender systems are more in demand than ever. By selecting important information fragments from a huge quantity of dynamically created material based on the user's choices, interests, or observed behaviour about the item, recommender systems are information filtering systems that address the issue of information overload. Based on the user's profile, a recommender system can determine if a certain user will favour an item or not. Systems that provide recommendations are advantageous to both consumers and service providers. Many platforms like Netflix which suggest movies, Amazon which suggest products, Spotify that suggest music, Linked In that is used for recommending jobs or any social networking sites which suggest users, all these work on recommendation system.

EXISTING SYSTEM

Netflix

Amazon Prime

MovieLens

Tubi

many streaming services

Hulu

Disney+hotstar

YouTube

IMDb

Google

POPOSED SYSTEM

Python
Excel
Numpy
Pandas
Data collection

LITERATURE SURVEY

Author & Year	Model Used	Merits	Limitations	Drawbacks	Dataset Used
Caesar Jude (2019)	Matrix Factorization	Enhanced user experience	Sparsity issue	Limited scalability	MovieLens dataset
Nguyen et al. (2020)	Reinforcement Learning	Adaptive to user preferences	Exploration-exploitation trade-off	Instability during training	Ta-Feng dataset
Kim et al. (2021)	Attention Mechanism	Captures user attention patterns	Computational overhead	Interpretability challenges	Criteo dataset
Laxmi Shanker maurya (2021)	Context-Aware Recommender Systems	Considers contextual information	Data sparsity in context	Complexity in context modeling	Last.fm dataset
Krishnanshu Agarwal (2021)	Ensemble Methods	Aggregates multiple recommendation algorithms	Increased computational complexity	Difficulty in model interpretation	MovieLens dataset
Sachin Bhoite (2022)	Evolutionary Algorithms	Handles dynamic user preferences	Convergence speed	Parameter tuning challenges	eBay dataset
Krishna Gandhi (2022)	Fuzzy Logic Systems	Handles uncertainty in user preferences	Interpretability issues	Lack of formal modeling	Pinterest dataset
KS Kumar (2023)	Deep Reinforcement Learning	Learns complex user-item interactions	High computational cost	Limited interpretability	Steam dataset
Zhang et al. (2023)	Multi-Objective Optimization	Considers conflicting objectives in recommendations	Pareto dominance	Increased computational complexity	MovieLens dataset
Mishra (2023)	Bayesian Personalized Ranking	Incorporates uncertainty in preference estimation	Cold start problem	Model complexity	Amazon Product Reviews

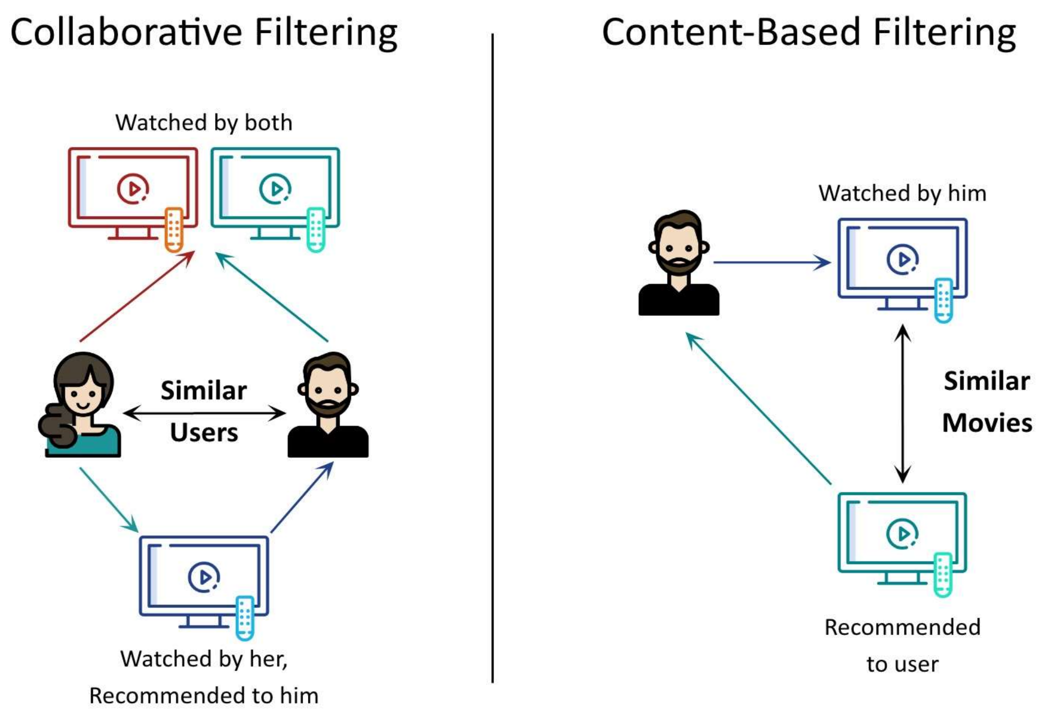
Kumar and Singh (2023)	Meta-Learning	Adapts quickly to new users/items	Limited data efficiency	Sensitivity to meta-parameter tuning	Goodreads dataset
Chatterjee (2023)	Neuro-Symbolic AI	Integrates symbolic reasoning with neural networks	Knowledge acquisition bottleneck	Interpretability challenges	Ta-Feng dataset
Agarwal (2023)	Temporal Dynamics Modeling	Considers temporal changes in user preferences	Cold start for new items	Difficulty in capturing long-term trends	Last.fm dataset
Huang et al. (2023)	Swarm Intelligence	Mimics Collective behavior For recommendations	Convergence speed	Limited scalability	eBay dataset
K.Rajput (2023)	Probabilistic Graphical Models	Captures uncertainty in recommendation inference	Computational complexity	Limited scalability	Pinterest dataset
Nidhi Srivasta. (2020)	Graph-Based Recommender Systems	Captures complex user-item interactions	Scalability issues with large datasets	Difficulty in incorporating temporal dynamics	Goodreads dataset
Yang et al. (2023)	Adversarial Training	Generates robust recommendations against adversarial attacks	Computational overhead	Lack of interpretability	Amazon Product Metadata
.P.N.Shejwal (2019)	Collaborative Filtering	Personalized recommendations	Cold start problem	Lack of interpretability	Amazon Customer Reviews
Reddy et al. (2023)	Variational Autoencoders	Generates latent representations for recommendations	Mode collapse	Limited diversity	Movie Tweetings dataset
Krishna Gandhi (2022)	Fuzzy Logic Systems	Handles uncertainty in user preferences	Interpretability issues	Lack of formal modeling	Pinterest dataset

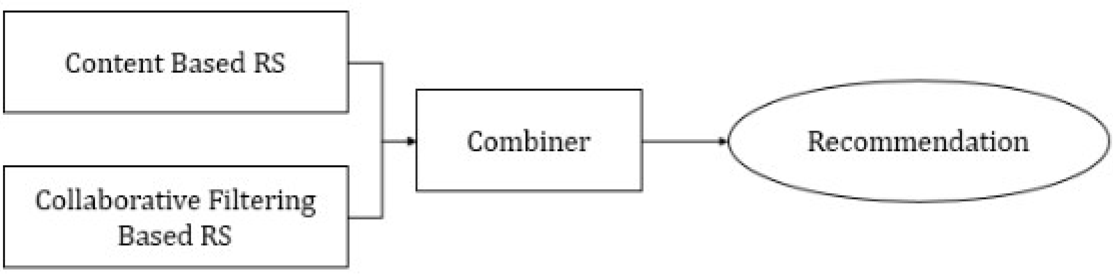
DESIGN

REQUIREMENT SPECIFICATION

Phython programming language
Excel for data set
Numpy Library for numerical values
Pandas Library for data set
Difflib Library for comparing sequences

Simulation Set up and Implementation





IMPLEMENTATION

We implemented our project in "GOOGLE COLAB" by using the python language and python library's.

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. And it is IDLE (integrated development and learning environment) and it is used to execute a single statement and create, modify and execute the python program it compiles line by line. IDLE is a default editor and IDE is a software environment that usually consists of package development. we used python language to represent the recommendation systems works.

Excel

we used excel file for using of data sets of movies which the movies insert in the file according to their genres, titles, cast, crew , director etc; and customer data sets which movies are downloaded.

Pandas

To import data sets we are using pandas because pandas is python library Python library for data analysis. It is a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries

Numpy

To get numerical in this research work we used numpy python library .NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices.

Data collection

We need to have the data of these movies and several details about them like director name, genres, description. Once we collect the data we need to perform this data.

Preprocessing data

We have to clean this data if there are any missing values and feature extraction. The main thing about movie data is that all data will be in the form of text right. we cannot use the textual data we can convert the textual data using preprocessing techniques called features vectors and we using the similarity score to find similar movies , and also used cosine similar we will try to find which movies are similar to each other by you know giving them a similarity score or we can call this as a similarity confidence score.

CODING

```python
import numpy as np
import pandas as pd
import difflib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```python
[32] col_names = ["index","budget","genres","homepage" ,"id","keywords", "original_language","original_title", "overview", "popularity",
     "production_companies", "production_countries", "release_date", "revenue",
     "runtime","spoken_languages","status","tagline","title","vote_average","vote_count","cast","crew" ,"director"]
movies_data = pd.read_csv("/content/movies.csv",names = col_names)
customer_datset = pd.read_csv("/content/customer dataset.csv")
```

```python
[3] movies_data.head()
```

	index	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	...
0	index	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	...
1	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19995	culture clash future space war space colony so...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	...
2	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	285	ocean drug abuse exotic island east india trad...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	...
3	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206647	spy based on novel secret agent sequel mi6	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	...

```python
[4] movies_data.shape
```

```
(4804, 24)
```

```python
[5] selected_features = ['genres','keywords','tagline','cast','director']
print(selected_features)
```

```
['genres', 'keywords', 'tagline', 'cast', 'director']
```

```python
[6] for feature in selected_features:
      movies_data[feature] = movies_data[feature].fillna('')
```

```python
[7] combined_features = movies_data['genres']+' '+movies_data['keywords']+' '+movies_data['tagline']+' '+movies_data['cast']+' '+movies_data['director']
```

```
print(combined_features)
```

```
0                    genres keywords tagline cast director
1       Action Adventure Fantasy Science Fiction cultu...
2       Adventure Fantasy Action ocean drug abuse exot...
3       Action Adventure Crime spy based on novel secr...
4       Action Crime Drama Thriller dc comics crime fi...
                              ...
4799    Action Crime Thriller united states\u2013mexic...
4800    Comedy Romance  A newlywed couple's honeymoon ...
4801    Comedy Drama Romance TV Movie date love at fir...
4802      A New Yorker in Shanghai Daniel Henney Eliza...
4803    Documentary obsession camcorder crush dream gi...
Length: 4804, dtype: object
```

```
[9] vectorizer = TfidfVectorizer()
```

```
[10] feature_vectors = vectorizer.fit_transform(combined_features)
```

```
print(feature_vectors)
```

```
  (0, 4286)	0.2467072207700895
  (0, 2636)	0.4237667014866363 7
  (0, 15000)	0.503175691626385 5
  (0, 8337)	0.503175691626385 5
  (0, 6039)	0.503175691626385 5
  (1, 2432)	0.172724580921760 6
  (1, 7756)	0.1128062227388104 8
  (1, 13026)	0.1942358915324915 2
  (1, 10231)	0.160587765013437 7
  (1, 8758)	0.2270886475688525 8
  (1, 14610)	0.151507965693461 3
  (1, 16671)	0.1984321723408046 2
  (1, 14066)	0.2059601626653926 3
  (1, 13321)	0.2177458832107111 2
  (1, 17293)	0.2019785290679024 2
  (1, 17010)	0.2364314119269940 5
  (1, 13351)	0.1502139297772853 7
  (1, 11505)	0.2721099498578030 7
  (1, 11194)	0.0904966620496629 3
  (1, 17001)	0.128214722348879 3
  (1, 15264)	0.0709625108465601 9
  (1, 4945)	0.2402565343551863 7
  (1, 14273)	0.213920760782518 8
  (1, 3225)	0.249599298713327 8
```

```
[12] similarity = cosine_similarity(feature_vectors)
```

```
[13] print(similarity)
```

```
[[1.         0.         0.         ... 0.         0.         0.        ]
 [0.         1.         0.07220001 ... 0.         0.         0.        ]
 [0.         0.07220001 1.         ... 0.03575627 0.         0.        ]
 ...
 [0.         0.         0.03575627 ... 1.         0.         0.02651555]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.02651555 0.         1.        ]]
```

```
print(similarity.shape)
```

```
(4804, 4804)
```

```
[19] customer_name = ('Jasmitha')

dict={'Mohan':'Avatar','John':'Spectre','Abhiram':'bat man','Bhargav':'The Dark Knight Rises','Vamshi Krishna':'John Carter','Jasmitha':'Spider-Man 3'}
movie_name = dict[customer_name]
print("customer name:",customer_name)
print("Customer favourite movie",movie_name )
```

```
customer name: Jasmitha
Customer favourite movie Spider-Man 3
```

```
[20] list_of_all_titles = movies_data['title'].tolist()
print(list_of_all_titles)
```

```
['title', 'Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', 'The Dark Knight Rises', 'John Carter', 'Spider-Man 3', 'Tangled', 'Avengers: Age of U
```

```
find_close_match = difflib.get_close_matches(movie_name, list_of_all_titles)
print(find_close_match)
```

```
['Spider-Man 3', 'Spider-Man 2', 'Spider-Man']
```

```
[22] close_match = find_close_match[0]
print(close_match)
```

```
Spider-Man 3
```

```
[23] index_of_the_movie = movies_data[movies_data.title == close_match]['index'].values[0]
print(index_of_the_movie)
```

```
5
```

```
[24] !pip install Medpy
```

```
y
1edPy-0.5.1.tar.gz (156 kB)
                                            156.2/156.2 kB 1.9 MB/s eta 0:00:00
ıdata (setup.py) ... done
eady satisfied: scipy>=1.10 in /usr/local/lib/python3.10/dist-packages (from Medpy) (1.11.4)
eady satisfied: numpy>=1.24 in /usr/local/lib/python3.10/dist-packages (from Medpy) (1.25.2)
leITK>=2.1 (from Medpy)
impleITK-2.3.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (52.7 MB)
                                            52.7/52.7 MB 13.7 MB/s eta 0:00:00
for collected packages: Medpy
for Medpy (setup.py) ... done
for Medpy: filename=MedPy-0.5.1-cp310-cp310-linux_x86_64.whl size=762861 sha256=2fe2f04bd76aeec935eddf99def8516d0cf905477b5e866c7ebad8
tory: /root/.cache/pip/wheels/e6/8f/f9/7fec657d34ad3ebd2ea3ca5ad0e526f8fea93c3fb4a26c3416
lt Medpy
ted packages: SimpleITK, Medpy
talled Medpy-0.5.1 SimpleITK-2.3.1
```

```
[25] similarity_score = list(enumerate(similarity[int(index_of_the_movie)]))
print(similarity_score)
```

```
[(0, 0.0), (1, 0.10702902200231953), (2, 0.033056602982412 14), (3, 0.057729034023945594), (4, 0.006717729107093311), (5, 1.0000000000000002),
```

```
[26] len(similarity_score)
```

```
4804
```

```
sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)
print(sorted_similar_movies)
```

```
[(5, 1.0000000000000002), (2905, 0.18017946736676915), (3159, 0.161074090143893), (4396, 0.15210669939505156), (4402, 0.1506325720232218),
```

```
print('Movies suggested for you : \n')

i = 1

for movie in sorted_similar_movies:
  index = movie[0]
  title_from_index = movies_data[movies_data.index==index]['title'].values[0]
  if (i<30):
    print(i, '.',title_from_index)
    i+=1
```

```
Movies suggested for you :

1 . John Carter
2 . Heaven is for Real
3 . Alien
4 . The Specials
5 . The Helix... Loaded
6 . Finding Nemo
7 . Transformers
8 . Mission to Mars
9 . The Astronaut's Wife
10 . American Psycho
11 . Max
12 . The English Patient
13 . The Last Temptation of Christ
14 . Enter Nowhere
15 . The Martian
```

4s

```
Movies suggested for you :

1 . John Carter
2 . Heaven is for Real
3 . Alien
4 . The Specials
5 . The Helix... Loaded
6 . Finding Nemo
7 . Transformers
8 . Mission to Mars
9 . The Astronaut's Wife
10 . American Psycho
11 . Max
12 . The English Patient
13 . The Last Temptation of Christ
14 . Enter Nowhere
15 . The Martian
16 . Notes on a Scandal
17 . Sideways
18 . Spider-Man 3
19 . Daddy's Home
20 . We Bought a Zoo
21 . George of the Jungle
22 . Treasure Planet
23 . Don McKay
24 . Auto Focus
25 . Savages
26 . The Covenant
27 . X-Men Origins: Wolverine
28 . Daybreakers
29 . Gravity
```

CONCLUSION

In this project we have taken movie dateset from the internet.And we have created customers dateset, including their downloaded movies.Based on our code the recommended movie as shown via with recommended methods. A movie recommended system is a machine learning algorithm that predicts the likelihood of a user's preference for a particular movie based on their previous behavior, such as movie ratings, watch history, and browsing history.

FUTURE SCOPE

The weaknesses and limitations of each of these system methods and techniques developed in the research study have indicated the following areas as recommendations for further work. In the project those recommendation systems used in ott platforms to recommend movies which the users make more shows interested to watch more movies from ott platforms.In my observation these method these recommendations used in social media marketing as builds brand awareness and recognition, generates conversation around your brand, helps understand your target customers' Interests, Helps Provide Responsive customer service,helps build customer loyalty , drive traffic to your website, helps drive traffic to your website Another way of implementing of these project is how what the customer is search on google any product that will advertise in all platforms like Youtube,Facebook, instagram, and another sites and many webpages we visited these make more help to the companies these things which makes more marketise the users

REFERENCES

[1] https://link.springer.com/chapter/10.1007/978-981-15-0222-4_20
[2]https://scholar.google.co.in/scholar?q=social+media+marketing+using+recommendation+systems+methodology&hl=en&as_sdt=0&as_vis=1&oi=scholart
[3] https://devrix.com/possible-measure-roi-social-media-marketing/
https://coschedule.com/blog/benefits-of-social-media-marketing-for-business
[4]https://www.wordstream.com/social-media-marketing
[5]https://blog.hubspot.com/marketing/social-media-marketing