Content Based Movie Recommendation System

Prof. N.R. Zinzurke
Dept. Computer Engineering
JSPM's JSCOE Pune, India
ni3zinzurke4040@gmail.com

Pranav Chobhe
Dept. Computer Engineering
JSPM's JSCOE Pune, India
pranavchobhe162@gmail.com

Sonali Jogdand
Dept. Computer Engineering
JSPM's JSCOE Pune, India
sonalijogdand0015@gmail.com

Sakshi Karle
Dept. Computer Engineering
JSPM's JSCOE Pune, India
ksakshi939@gmail.com

Abstract*— In the era of digital streaming platforms, the sheer volume of available content can overwhelm users, making the efficient discovery of relevant and enjoyable content a critical challenge. Many individuals spend hours and hours of their productive time in searching for the content that best fits their needs and interests. To address this challenge, movie recommendation system has emerged as essential tool. The adoptionof computer-assisted approaches is essential to help in overcoming these constraints. With the advancement of data science, machine learning (ML) models are being used in recommending a type of content to user based on his/her individual interests. In this study, a vectorization and cosine similarity base model is fine-tuned efficiently to study user interest in order to recommend the content. In this paper, we present a movie recommendation engine following a content-based recommendation approach. We use publicly available features such as movie plot, movie genre and ratings to find similarity between the movies and generate a recommendation list. By providing recommendations which are consistent with user interest, the aim is to make the platform personalized.

***Keywords*—Movie recommendation, content-based, vectorization, cosine similarity, movies, recommender system, TMDb.**

I. INTRODUCTION

Digital content present on the internet these days can be overwhelming for user. This content is dynamic and diversely established in nature. These features of the content complicate the process of deriving useful information which helps in providing users with individualized support while searching through the massive amount of accessible data. To address this complexity issues a recommendation system can be used to suggest relevant items like movies, songs, shows, etc. to users. Our recommendation engine for movies is primarily built using machine learning, cosine similarity metrics, and content-based filtering approaches. Based on the user's past behavior or explicit feedback, content-based filtering techniques employ movie features to suggest additional films that are comparable to the user's favorites. Two videos can be viewed as two vectors in m dimensional user space in cosine similarity. The cosine of the angle between the vectors is used to calculate how similar they are to one another. In our system, machine learning is employed to create recommendation models and to retrieve information.

We know that in the content filtering we recommend movies to the user based on the movie details like title, actors etc. and based on the user's history. Recommendation system which are purely based on content filtering have certain drawbacks like there is not enough variety or novelty, Scalability is difficult etc. And in the collaborative filtering, It compiles the user ratings for services like items, movies, etc., finds patterns among users based on their ratings, and generates fresh recommendations for the user based on inter-user comparisons. Recommendation systems which are purely based on the collaborative filtering have certain drawbacks like cold start problem, hard to include side features for services like item, movies etc. The side elements for movie recommendations may include a user's country or age. Including available side features raises the model's caliber. Using a machine learning technique in the movies recommendation will surely help to improve the efficiency of the recommendation system. We have used TMDb dataset for the movie's recommendation. In this paper, we have first studied the dataset properly and then done the exploratory data analysis on the dataset to recognize the patterns and understand it properly. Then we have done preprocessing of the dataset. Creating different machine learning based recommendation models using content and collaborative filtering. Then we have done training and testing of these models. Then using the best recommendation model, we have created the Rest API and then created the recommendation system GUI for the movies. Then in the GUI we need to enter movies details like title etc. and then we will recommending similar movies to the user.

II. OBJECTIVE AND OVERVIEW

Accurate recommendations: Suggesting content based on the user's preferences accurately is the priority when working on this project [5]. The movies search by the users are studied for their genres, movie plot, crew members, etc. to provide a fine-tuned list of recommended movies based on the users' previous search.
(1) Overview of the project objectives:
Data preprocessing: This step involves collecting and cleaning movies data and arrange it in sophisticated form that makes easier for the ML model to study it.

Implementing Bag of words: This step involves extracting and counting the occurrence of similar words present in movie description section of data [3].

Creating vectors: Based on the result given by Bag of words method, n number of vectors are formed in n-dimensions.

Finding similarity: This step involves implementation of cosine similarity method to finding similarity between the movies by examining their respective vectors.

III. TECHNOLOGY IN PROJECT

(1) Flask Framework:

Usage: Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. Lightweight.

(2) AJAX:

Usage: AJAX is a set of web development techniques used to create asynchronous web applications. It allows for updating parts of a web page without requiring a full page reload, providing a smoother and more responsive user experience.

[3] Beautiful Soup:

Usage: Beautiful Soup is a Python library used for web scraping. It allows developers to extract data from HTML and XML files, which is useful for retrieving information from websites, such as movie reviews from IMDb in this case.

[4] Cosine Similarity:

Usage: Cosine similarity is mentioned as the method used to measure similarity between movies for recommendation purposes. This involves comparing the textual details of movies using a cosine similarity metric. Python provides libraries for performing cosine similarity calculations, such as scikit-learn.

[5] Frontend Technologies:

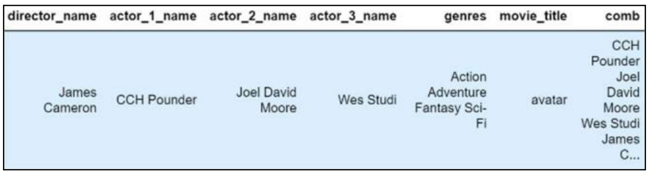
Usage: HTML, CSS, and JavaScript are used for building the user interface and handling interactions on the client-side.

IV. IMPLEMENTATION

(1) DATASET: 1 IMDB 5000 Movie Dataset
2. The Movies Dataset

(2) Data preprocessing:

5000 movies and data like director_name, actor_1_name, actor_2_name, actor_3_name, genres, movie_title, etc attributes of the datasets are consider in data processing [8]. Extracting features such as titles, directors, and actors, and combining it with an existing dataset of movies. So, at this point our data consist of 7 columns, director_name, actor_1_name, actor_2_name, actor_3_name, genres, movie_title, and comb which is a combination of all the attributes related to the movies.



director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb
James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action Adventure Fantasy Sci-Fi	avatar	CCH Pounder Joel David Moore Wes Studi James C...

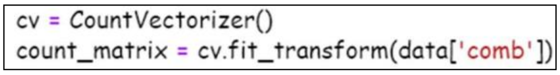
(3) Building machine learning model:

1. Feature extraction:

It is a process, by using with count vectorizer, the words in the comb attributes are converted in to token counts. This results in, calculating the number of occurrence of a particular word in the whole data of 'comb' attribute in each item.

2. Vectorization:

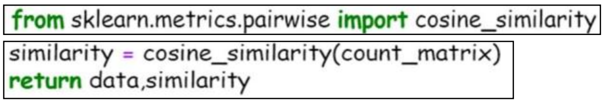
The token generated in the above process is converted in to vectors, in which a single vector represents a single movie tuple. These N number of tuples are arranged in N dimensions.



```
cv = CountVectorizer()
count_matrix = cv.fit_transform(data['comb'])
```

3. Identifying similarity:

Similarity between any two movies is calculated based on the cosine of angle between vectors of those two movies. This is calculated using method like 'cosine similarity' from 'sklearn.metrics.pairwise' module [3]. In this case, lesser the cosine of angle between the two vectors, more similar are the movies associated with those vectors.



```
from sklearn.metrics.pairwise import cosine_similarity
similarity = cosine_similarity(count_matrix)
return data,similarity
```

(4) Web scraping:

Beautiful soup is used for fetching user reviews from IMDB website, so by reading the reviews user can get more idea about the particular movie.

(5) Testing and training:

We must train and test the model when it has been generated. The dataset has been divided in half, 80:20. 20% of the dataset is used to test the model, while the remaining 80% is utilized to train the model [9]. For evaluating the models, we have used metrics like RMSE (Root Mean Square Error) and MAE (Mean Absolute Error). Root Mean Square Error is a statistic that reveals how far, on average, a model's projected values and observed values differ from one another. Mean Absolute error in the context of machine learning refers to the size of the discrepancy between the forecast of an observation and its actual value.

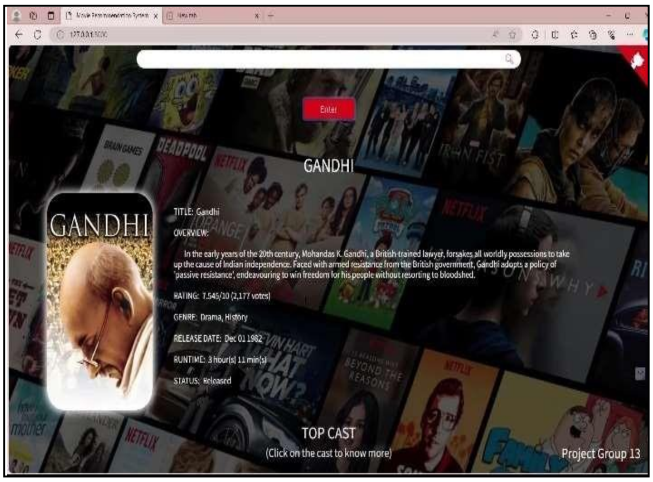
V. MODEL EVALUATION AND RESULTS:

This section contains a discussion of the outcomes from our experimentation and implementation of various machine learning-based recommendation algorithms [9]. We have used two metrics namely RMSE (Root Mean Square Error) and MAE (Mean Square Error) for evaluating various recommendation models.

Recommendation Models	RMSE	MSE
Collaborative + Singular Value Decomposition	0.8675	0.6729
Collaborative + K-Nearest Neighbors	0.9552	0.7257
Collaborative + Co-Clustering	0.9544	0.7249
Collaborative + Alternating Least Square	0.8219	0.7632
Content + Cosine Similarity	0.7481	0.6316

Table -1: Comparison of recommendation model.

In the above table, the model which have low value of RMSE and MSE is considered as best model as it is having less error. So, the model using content and cosine similarity is best as compared to the other models [5]. For creating movies recommendation GUI, we have used PyCharm as IDE and created the API for the best machine learning based recommendation model. Below is the screenshot for recommending movies to the user using cosine similarity [5]. In this case, lower the error, more accurate the model is. So we compared different permutations of possible approaches and algorithms to build the model, we noticed that, the approach with content based method with cosine similarity provides less error. So this approach is best for our model.



GANDHI

TITLE: Gandhi

OVERVIEW:

In the early years of the 20th century, Mohandas K. Gandhi, a British-trained lawyer, forsakes all worldly possessions to take up the cause of Indian independence. Faced with armed resistance from the British government, Gandhi adopts a policy of 'passive resistance', endeavouring to win freedom for his people without resorting to bloodshed.

RATING: 7.546/10 (2,177 votes)

GENRE: Drama, History

RELEASE DATE: Dec 01 1982

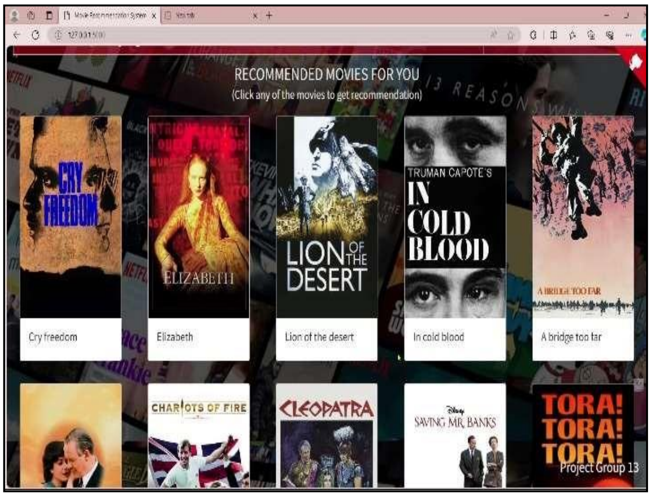
RUNTIME: 3 hour(s) 11 min(s)

STATUS: Released

TOP CAST

(Click on the cast to know more)

Project Group 13



RECOMMENDED MOVIES FOR YOU

(Click any of the movies to get recommendation)

Cry freedom

Elizabeth

Lion of the desert

In cold blood

A bridge too far

Project Group 13

VI. CONCLUSION

In this paper, we have implemented various recommendation models using content and collaborative filtering based on different machine learning techniques to improve the user recommendation in the movies recommendation system. After studying, comparing, and experimenting various recommendation model we have realized that model based on content filtering and cosine similarity was better as compared to the other models. Using the best model, we have created API and using it in movies recommendation GUI. Finally, we are recommending similar movies to the user. In order to boost user satisfaction, our suggested solution would enable the system to make a recommendation to the user that is more accurate. In future we can implement recommendation system which can work on real time information of users. Also, we can try to implement cross domain recommendation system in future.

VII. REFERENCES

[1] M Viswa Murali, Vishnu T G, Nancy Victor,” A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research”,2019, (ICACCS),DOI:10.1109/ICACCS.2019.8728409

[2] Ramni Harbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula, Gaurav Srivastav,” Movie Recommendation System using Cosine Similarity and KNN”,2020,((IJEAT), DOI: 10.35940/ijeat.E9666.069520

[3] Shivganga Gavhane,Jayesh Patil,Harshal Kadwe,Projwal Thackrey,Sushovan Manna, “Recommendation System using KNN and Cosine Similarity”,2020,

[4] Shubham Pawar, Pritesh Patne, Priya Ratanghayra, SimranDadhich, Shree Jaswal, "Movies Recommendation System using Cosine Similarity", (IJISRT), Volume 7, Issue 4, April –2022, 342-346, April 2022.

[5] A. A. Ewees, Mohamed Eisa, M. M. Refaat, “Comparison of cosine similarity and k-NN for automated essays scoring”,(2014),(IJARCCE), DOI10.17148/IJARCCE

[6] Y.H Zhou, D. Wilkinson, R. Schreiber, “Large scale parallelcollaborative filtering for the Netflix prize,” In Proceedings of4th International Conference on Algorithmic Aspects in Information and Management (pp. 337–348). Shanghai: Springer,2008

[7] Tiantian He , Yang Liu, Tobey H. Ko , Keith C. C. Chan , and Yew-Soon Ong “Contextual Correlation PreservingMultiview Featured Graph Clustering”,(2019),(IEEEtransactions)

[8] Zhiheng Wu,Jinglin Li,Qibo Sun,Ao Zhou,“Service recommendation with context-aware user reputation evaluation”,(2017),(IEEE conf)

[9] Khamael Raqim Raheem; Israa Hadi Ali, “Content-based Recommender System Improvement using Hybrid Technique”, (2020) (IEEE Xplore)

[10] Shailesh Kalkar, Prof. Pramila Chawan, “A Survey on Recommendation System based on Knowledge Graphand Machine Learning”, (2022) (IRJET), Volume: 09 Issue: 06 | Jun2022