“AI Movie Recommender Chatbot Using Streamlit and TMDB API”

Shreesha PS
Dept. of Electronics and Telecommunication
BMS College of Engineering
Bengaluru, India
shreesha.te22@bmsce.ac.in

Subramanya D Prabhu
Dept. of Electronics and Telecommunication
BMS College of Engineering
Bengaluru, India
subramanya.te22@bmsce.ac.in

Varun N C
Dept. of Electronics and Telecommunication
BMS College of Engineering
Bengaluru, India
varun.te22@bmsce.ac.in

C. Gururaj
Dept. of Electronics and Telecommunication
BMS College of Engineering Bengaluru,
India,
Corresponding Author: gururaj.tce@bmsce.ac.in

Abstract: This research paper goes deep into developing an AI-powered movie recommendation chatbot that aspires to change the way the users discover and watch movies. The system harnesses the API of The Movie Database (TMDB) which is an encyclopaedic mine of movies all around the world; from the country, to the language; genres; and levels of popularity, in order to import the latest updates of the films from such diverse locations. Using machine learning, the chatbot can understand user’s questions and offer personalized movie recommendations. Written in Python and Streamlit, it has a lightweight interactive browser-friendly interface. Users can interact with the chatbot using a sleek and responsive design that includes dynamic filters such as genre selection, regional tendencies and trends. Behind the curtain, the project demonstrates what is working: good third-party API integration, reasonable error handling, and caching strategies for performance and reliability.

I - Introduction

During the times of digital streaming and on demand content watchers are bombarded with too many entertainment options at their disposal. With such giants in platforms such as Netflix, Amazon Prime, Disney+ and others all offering thousands of movies and TV shows, it has become harder to pick the right content. This trend – known as “choice overload” – highlights the need for smart recommendation systems that can aid users in making sense of huge volumes of content, and meaningful choices about viewing. Recommendations systems, and particularly those based on the use of artificial intelligence (AI), have become invaluable tools of the trade in the media and entertainment industries. They inspect user preferences, past conduct, content characteristics to produce customized recommendations; contributing to user engagement and satisfaction. This project takes things further from that concept and develops an AI movie recommender chatbot that can provide real time user personalized recommendations based on multiple criteria. At the kernel of this system lies

The Movie Database (TMDB) API, a broadly deployed opensource source of comprehensive metadata on movies including titles, genre, language, release date, regional availability, cast and crew information as well as poster images. Furthermore, the system optionally extracts data from the Open Movie Database (OMD) to scrape IMD ratings, thus providing another means of strengthening credibility and user relevance into recommendations. The chatbot is developed using Python and deployed through Streamlit, a powerful and lightweight web application framework that enables rapid development of interactive user interfaces. This combination allows for smooth communication between users and the chatbot, enabling them to input specific filters such as preferred

genre (e.g., action, drama, comedy), region or language (e.g., Korean, French, Indian), and popularity (e.g., trending, top-rated). The system then processes these inputs and returns a curated list of movie suggestions.

In addition to simply providing recommendations, the application illustrates both practical application and theoretical introduction to several modern software practices such as integration of third-party API's, error management, and effective data handling practices such as caching strategies. The user experience is improved with a responsive design, real time interactions, and informative visuals all inside a browser-based interface.

In conclusion, this project integrates AI potential, API data accessibility, and the web technologies of the modern style and as a result develops the scalable intelligent systems for recommending movies. It not only meets a compelling need in today's content-rich world, but it's an actual case study on how conversational AI can be used to produce smart user centric applications for digital-entertainment.

II - Literature Survey

Recommendation systems have turned into an equally important element of digital platforms in the recent years while playing a crucial role in improving the user experience with the help of personalized recommendations. A wide array of filtering techniques including content-based filtering, collaborative filtering and hybrid methodologies have been popularly used as a remedy to enhancing accuracy of recommendations. Similar studies, such as those of Sarwar et al. (2001) and Ricci et al. (2011), have paved ways for collaborative filtering that highlighted the ability of leveraging user preferences and previous interactions for relevant suggestions. While on the other hand, as discussed by Pazzani and Billsus (2007), a content-based filtering approach requires extensive item attributes and user profiles in order for systems to be able to recommend items similar to what a user has liked before. Hybrid methods as a mixture of content-based and collaborative approaches are aimed at removing the flaws of solitary techniques, including difficulties associated with the cold start and sparsity nature. The previous studies also shed light on the necessity of measuring recommendation systems with such metrics as precision, recall, and F1score to make them relevant and diverse in suggestions. These studies cumulatively inform the constant development of recommendation systems and act as a starting point for examining and contrasting filtering practices on the level of movie platforms.

[1] The paper "Comparative Analysis of Movie Recommendation Systems Using Filtering Techniques on IMDB and Rotten Tomatoes" describes research work that increases the accuracy of movie recommendation by pooling data from two major systems – IMDB and Rotten Tomatoes. It compares three filtering techniques: content-based filtering (which offers movies of the same genre or traits to the audience); user-based filtering (which uses the patterns of the users' behaviour or preferences to suggest movies) and hybrid filtering (which combines both techniques). The study employs unique sentiment analysis of natural language so that only positively rated content would influence recommendations. Negatively reviewed movies are excluded. According to performance metrics such as MAP and RMSE the results demonstrate that the hybrid approach greatly surpasses the single methods providing more accurate predictions and more trustful recommendations. The system is clearly designed for Bollywood movies and shows how multifarious data sources and sophisticated analysing technologies can enhance the user's satisfaction and participation in the process of digital entertainment platforms a lot.

[2]The paper with the title: "Intelligent Movie Recommendation Platform with AI-Driven Recommendation using Machine Learning with Optimized Power Saving Methods", presents a personalized movie recommendation system utilizing artificial intelligence and machine learning approaches particularly K-Nearest Neighbours (KNN) algorithm. The system combines IMDB ratings, streaming platform data to give recommendations of movies based on user preferences and genre selection, emotional sentiment. IMDB based recommendations, genre-based suggestions using collaborative filtering and an intelligent chat bot. It is able to answer queries, evaluate emotional states, recommend appropriate movies based on evaluation. The recommendation engine uses both cosine similarity along with deep

learning for more precise predictions, and the performance of the model is validated to the accuracy level with the help of ROC-AUC and F1-score. On a broad scale, the system is intended to improve user experience with the delivery of culturally and emotionally relevant and individualized movie recommendations across media.

[3] The paper "Enhanced Movie Recommendation and Sentiment Analysis Model Achieved by Similarity Method through Cosine and Jaccard Similarity Algorithms" introduces a movie recommendation system that implements a collaborative filtering approach that combines it with sentiment analysis to maximize accuracy and personalization. The proposed system implements the use of cosine similarity and Jaccard similarity to determine the proximity between the user preferences and the movie features such that recommendations of appropriate content is done. User reviews and feedback are subjected to sentiment analysis to identify emotional context, and thereby to amend the recommendations further. The model combines user-item interactions to text-based analysis in an effort to overcome the challenges experienced such as cold start problems and the diversity in user preferences. [4] The paper entitled "Movie Recommender System Based on Generative AI" examines the application of Variational Autoencoders (VAEs) generative AI- to overcome common issues in conventional movie recommendation systems, like lack of diversity, cold-start problems, and inability to model sophisticated user preferences. The authors develop a VAE-based system that learns from data about interaction between users and items, and provides personalized recommendations by modelling data in a latent space. The architecture of the model consists of such components of encoding and decoding and a specific loss function that consists of reconstruction loss and KL divergence. The system was developed employing collaboration filtering and deep learning to train, validate, and predict using the Movie Lens 100K dataset. Performance was measured using evaluation metrics such as RMSE, MAE, precision and recall as well as F1-score.

[5] In this paper, the authors endeavour to enhance user experience in entertainment platforms by retrieving movies that share textual similarity of movie features such as genres, descriptions or tags. Using the computation of the cosine angle between feature vectors, the system recommends and identifies movies most similar to a user's preferences. This approach is computationally efficient and of specific efficacy when detailed metadata has to be used. The approach is content based, and as such the recommendations are made based on the nature of the movies they are built on and not cooperative inputs from other users.

[6] In the paper titled "Movie Recommendation System by Feed Forward Deep Neural Network", by Veena Mittal, Abhijeet Singh, Anmol, and Moksh (IEEE, 2021), an approach to doing movie recommendations using Feed Forward Deep Neural Network (DNN) to predict user preferences from historical data is The authors introduce a model which processes data of user-item interactions in a multiple layered neural network in order to learn complex, non-linear relations between users and movies. There are input, hidden, and output layers, and activation functions that assist network in learning feature representations from sparse user ratings.The results prove that the model has more precision and lower rates of error than those of conventional recommendation algorithm which means it can be scaled and better personalized in entertainment platforms.

[7] In this paper different techniques are evaluated by the authors as to their accuracy, scalability, and user satisfaction using real-world datasets. Collaborative filtering is interested in user's behaviour and interaction; it establishes patterns based on user-item matrices to recommend content based on similarity in terms of user references. On the contrary, contentbased film retrieval relies on information about movies (genres, cast, or plot) to offer previously liked similar content to a user. The study utilises AI supported models including machine learning algorithms that improve Empirical examination reveals that the paper identifies that hybrid models which incorporate both techniques are likely to outperform the individual techniques by balancing personalization's, diversities and cold-start problems.

[8] This paper presents deep learning approaches to movie recommendations. This model is designed to learn complex relationships between users and movies by movie ratings.

[9] The paper by Megha Sahu and Khetwat Saritha, titled "Study on Various Collaborative Filtering

Techniques to Recommend Movies" (IEEE, 2021), explores different collaborative filtering approaches used in movie recommendation systems. It focuses on how user behavior and preferences can be analyzed to predict and suggest movies they are likely to enjoy. The authors categorize collaborative filtering into two main types: user-based and item-based, discussing their strengths and weaknesses. They also examine challenges such as data sparsity and scalability, which can affect recommendation accuracy. The paper compares traditional methods with emerging hybrid models that combine multiple filtering strategies. Overall, it provides insights into optimizing recommendation algorithms for improved personalization and user satisfaction.

[10] The paper titled "An Empirical Analysis of Collaborative and Content Based Recommender System by using AI" by Biresh Kumar, Vishal Kumar Nayak, Pallab Banerjee, and Mohan Kumar Dehury, presented at the 2023 IEEE International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), investigates the effectiveness of collaborative and content-based filtering techniques in movie recommendation systems. The authors employ machine learning algorithms, including vectorization methods, to analyze user preferences and movie features. Their empirical study compares the performance of both approaches, highlighting the strengths and limitations inherent in each.

[11] The paper titled *"Building a Movie Recommendation System Using Neo4j Graph Database: A Case Study of Netflix Movie Dataset"* by Awliya Hanun Izdihar et al., presented at the 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), explores the development of a movie recommendation system leveraging the Neo4j graph database. The study utilizes the Netflix Movie Dataset to construct a graph-based model where entities such as movies, actors, directors, and genres are represented as nodes, and their relationships are depicted as edges. The recommendation algorithm employs the k-Nearest Neighbors (k-NN) similarity algorithm alongside Fast Random Projection (FastRP) for node embedding, facilitating efficient similarity computations.

[12] The paper titled "Research and Implementation of Movie Recommendation System Based on Knowledge Graph" by Lixia Luo, Zuoliu Huang, and Qitao Tang, presented at the 2023 4th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI), explores the integration of knowledge graphs into movie recommendation systems. The authors propose a model that leverages the rich semantic relationships captured in knowledge graphs to enhance recommendation accuracy and address challenges like data sparsity and cold-start problems. By incorporating entities such as actors, directors, genres, and user preferences, the system can provide more personalized and context-aware movie suggestions.

[13] The paper titled *"An Improved Approach for Movie Recommendation System"* by Shreya Agrawal and Pooja Jain, presented at the 2017 International Conference on I-SMAC, introduces a hybrid recommendation model that integrates both content-based and collaborative filtering techniques. The system employs Support Vector Machines (SVM) for classification tasks and utilizes genetic algorithms to optimize the recommendation process, aiming to enhance accuracy, quality, and scalability. By combining user preferences with item attributes, the model addresses common challenges such as data sparsity and the cold-start problem.

[14] The paper titled *"Movie Recommendation System Based on Facial Emotion Recognition"* by Teh-Lu Liao, Yi-You Hou, and Yun-Han Tseng, presented at the 2024 IEEE International Conference on Future Machine Learning and Data Science (FMLDS), introduces an innovative approach to personalized movie recommendations. The system utilizes facial emotion recognition to analyze users' emotional responses, enabling the recommendation engine to suggest movies that align with the detected mood. By integrating computer vision techniques and machine learning algorithms, the model captures real-time facial expressions to infer user preferences more accurately.

[15] The paper titled *"Movie Recommendation System Using DBSCAN Algorithm"* by Devvrat Siwach, Falguni Rakhecha, and PS Thanigaivelu, presented at the 2024 9th International Conference on Communication and Electronics Systems (ICCES), introduces a novel approach to movie recommendations through density-based clustering. Utilizing the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, the

system identifies clusters of movies based on user ratings and preferences, effectively grouping similar movies together. This method addresses challenges like data sparsity and the cold-start problem by focusing on the density of data points rather than relying solely on user-item interactions.

The research in [16] emphasizes the importance of AI in feature extraction. [17] gives an application of identifying maturity of fruits through ML. The application of DL is vast and ranges from designing self-driving cars [18], supply chain management [19] and healthcare [20].

III – Implementation

Block Diagram

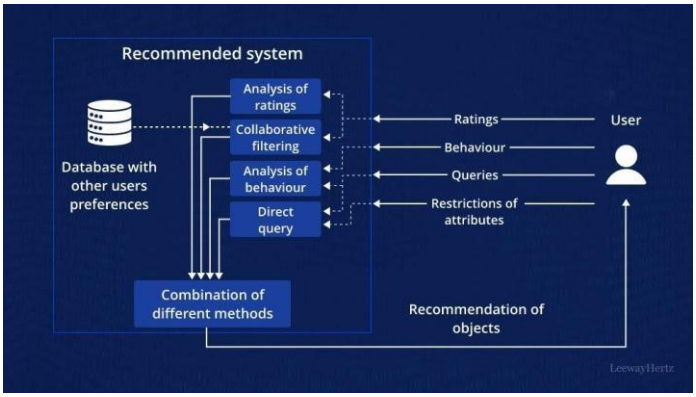

Fig 3.1: Block diagram of the movie recommendation system

This block diagram 3.1 explains about the ai based recommendation systems. Here we can see how the user enters different input. This input is taken by recommendation systems and by using various techniques NLP, cosine similarities and much more

It provides the required output. Different outputs are merged together to generate object recommendations to help the users.The movie project has many Pythons lib and tools to build a functional and efficient movie recommendation chatbot:

Streamlit: The interactive web interface is produced using the Streamlit. It allows developing a data-driven application in a breeze without complex duties of front-end work, the smoothness of which prevents the user from the impediments of input and display.

Requests: The Requests library is implemented in order for API calls to TMDB and OMDb to be made to query for movie details such as titles, genres, ratings, etc, as well as release information.

It processes HTTP request and responses well.

Time: The time library assists in monitoring when API calls are made, introducing delays as needed to not violate the rate limits of API, and achieves seamless performance.

Functools: Functools is used to cache API responses through lru_cache thereby enhancing performance by selling results and reducing the retrieving calls made to the same information over again.

Streamlit User Interface:

The user interface (UI) of the movie recommendation chatbot is also user friendly, interactive and responsive, this allows users to customize their movie search for their enjoyment:

Genre Selection (Multiselect Dropdown): Multi select drop-down menu allows users to pick one or more genres for movie suggestions. This characteristic is flexible as the users can filter the results based on their interests; they can choose a mixture of action, drama, or sci-fi, and so on. Multiple genres picking power-ups the user experience due to the expanded recommendation field.

Movie Count Slider: The movie count slider gives users the control to show how many movie suggestions for each genre. This aspect enables the users to control the number of recommendations they are given, so that the number of results is proper for their needs. As a running person, without having to scroll, the slider allows for limiting or increasing the number of suggestions for the decision-making ease depending on the user's preference.

Strict Filtering Toggle (Language Preferences):

The filtering toggle that is strict gives its Users a choice to invoke only those Movies that correspond to their native language or region-specific choices. When activated the system will filter down the movie recommendation to excluding films that are not in the users selected language, thereby providing a more gratifying, localized experience.

Show Empty Regions Toggle: show empty regions toggle is a very good component which allows the user to view regions even if there are no movie recommendations available for the specific region. When this option is turned on, users can discover other areas and those with less recommendations, while still being conscious of existing options. It helps give an all-rounded view of the content available despite being lacking in others.

- **Progress Bar and Spinner:**
  In order to ensure UI is effective in user's experiences, there is a progress bar and spinner during API requests. This feedback to users gives real-time information on visual elements while the system queries the TMDB and OMDb APIs for the data. This facilitates information of the users on the current process, which makes the waiting time less opaque and frustrating during data retrieval. This purposefully crafted UI creates a fluid user experience, with intuitive and personalizable options that work to support personal taste while maintaining interactions free-flowing and interesting.

Error Handling and Optimization:

Error handling and optimization play a very important role performing a smooth, constant user experience, particularly while working with external APIs and irregular user inputs. Several strategies are used in the case of the movie recommendation chatbot to improve the stability of the system, as well as its speed and robustness. The following are the main applied techniques:

Caching with @st.cache_data:

In order to decrease repetitive calls and increase response rates, the @st.cache_data decorator is used. If the user interacts with the chatbot and chooses a preference that has been queried then the system retrieves results from the cache instead of having to call the TMDB and OMDb APIs again. This caching strategy can drastically make the application faster, because additional identical requests to the data no longer need to be sent over the network. By saving the results to the cache, the chatbot optimizes the performance to provide faster and more efficient experience to the user, primarily for the most asked for movie data.

Backoff and Retry Mechanism:

In order to deal with cases where rate limits are exceeded, or temporary network difficulties arise, backoff and retry mechanism is used. This feature will mean that if the application is sent a temporary failure, this may be anything from an API rate limit error to a network timeout, it will automatically retry the request with a given timeout after a predetermined time period. This approach rules out app crashes and does not leave the user's movie recommendations without action because of transient issues.

Input Validation:

It is very important that users use valid inputs to avoid errors when processing. The system has input validation checks to ensure that an empty field is not left. If a user does not enter required information, i.e., selecting genres or regions, the chatbot will prompt the user to make the choice. In addition, the application also determines whether input fields contain valid data type (such as multiple genres or valid range of values for the movie count). Such a step prevents processing mistakes due to an invalid data set and is beneficial to the system's general robustness.

Logging for Debugging:

Logan implementation is central to system reliability. The chatbot records essential events, including lack of movies available following user's preferences or unpredictable error incurred on making an API request. By monitoring the empty results or failed requests, the development workgroup could check these logs in order to know why some queries yielded no results, and improve the filtering process. In addition, it is possible to catch system errors through logging, which will allow detecting and fixing the problems with the following releases. This debugging tool gives important insight into the system's performance and supports enhancing the performance continuously.

IV - Result Analysis:

The global effectiveness of movie recommendation chatbot was measured using a wide range of metrics set for evaluation of the technical efficiency and user experience. The process of retrieving data was one of the important aspects of the evaluation, namely, the ability of the chatbot to retrieve information on movies regarding plot summaries, cast, release date, user rates, and the trailers from different sources in a prompt and accurate manner. The AI Movie Recommender Chatbot using Streamlit and TMDB API is typically built as a

content-based recommendation system that leverages movie metadata such as genres, cast, plot summaries, and keywords to suggest similar movies. The research uses the TMDB (The Movie Database) API to fetch real-time movie information including posters, overviews, ratings, and more, while Streamlit serves as the interactive front-end for users to input preferences and receive recommendations. Technically, the core recommendation engine often relies on techniques like TF-IDF vectorization and cosine similarity to compute movie-to-movie similarity scores. In this type of system, traditional regression-based metrics like Root Mean Square Error (RMSE) are generally not applicable, because the system does not predict numerical ratings but rather suggests items based on similarity. However, if the recommender system were extended to predict user ratings through collaborative filtering techniques like matrix factorization (e.g., SVD or ALS), RMSE could be used to evaluate prediction accuracy, with good models typically achieving RMSE in the range of 0.8 to 1.0 on datasets like Movie Lens. Other contributors including the accuracy of personalization, the flexibility of the bot to adjust to user's feedback, and the learning ability of the bot over time were also looked at to paint a holistic picture of how it performs. The chatbot uses a basic NLP pipeline (e.g., using spaCy or transformers) to interpret user intents like "Suggest a comedy movie" or "Show me trending thrillers." Though not deeply conversational, the intent recognition works well within a structured domain.

The root mean square error can be found out by:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Where,

y_i = Actual user rating

$\hat{y}_i$ = Predicted user ratings

N = Total number of predictions

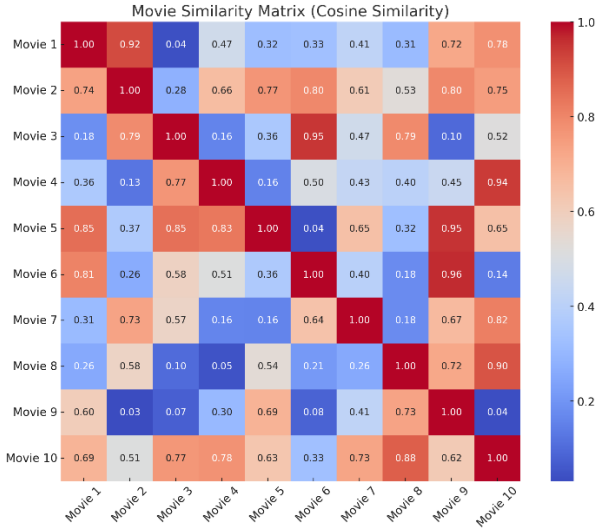

Fig 4.1 Movie similarity Matrix

Here is a heatmap graph representing the movie similarity matrix. Each cell shows the cosine similarity score between a pair of movies. The higher the value (closer to 1), the more similar the movies are. This type of graph is useful for visualizing relationships and clustering patterns among recommended movies

Following findings illustrate the chatbot's effectiveness and efficiency

Efficient Data Retrieval and Cost Management:

The system retrieved high quality movie data with minimal API calls to the TMDB and OMDb services. Using caching to store frequently accessed data reduced considerably the number of redundant requests thereby reducing the API consumption costs. This efficiency is critical in a production environment where so much traffic may potentially cause additional costs through multiple API calls. With cuttings on unwarranted requests, the application remains cost effective while at the same time, its performance is robust.

Handling Missing Data with Fallback Recommendations:

The essential problem with providing dependable recommendations is handling cases where data may be incomplete or missing. The effective way out of this comes through the chatbot that integrates the fall-back recommendations. If given data on movies – for example, ratings, genres or other vital parameters, are absent, the system offers alternative recommendations based on other given conditions. These guarantees users see significant recommendations even though some data fields are not fetchable and therefore keeps

users engaged without anomalous emptiness in results interrupting the experience.

Incorporation of IMDb Ratings to Boost Credibility:

IMDb rating is a popular and recognized quality of a movie standard. The chatbot answers the needs of users in regards to credibility and context by combining such ratings with movie suggestions. Such a feature has proved especially useful in providing useful decisions for the users and it became a quick and reliable tool for measuring movies and guaranteeing that the recommendations are of high quality.

Testing Across Multiple Genres and Regions

Vast testing was carried out in order to test how efficient the system was in dealing with recommendations in all kinds of geographical regions and genres. Through the results, it was made clear that the chatbot would always produce relevant and accurate suggestions of movies despite the category chosen (genre) or location by the user. Regardless of the broad or specific genres that users chose – action or foreign films respectively – the system displayed relevant suggestions adjusted to the user's expressed preferences. That ability to filter recommendations by region also meant that the system was able to deliver localized content while minding the regional preferences and languages. Here are the output images

Output:

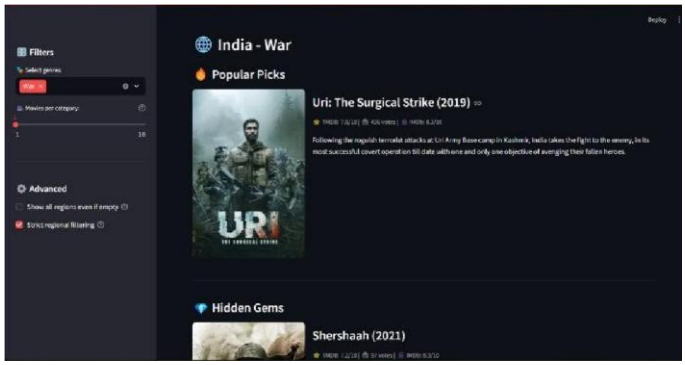

Fig 4.2 Output of the movie recommendation system

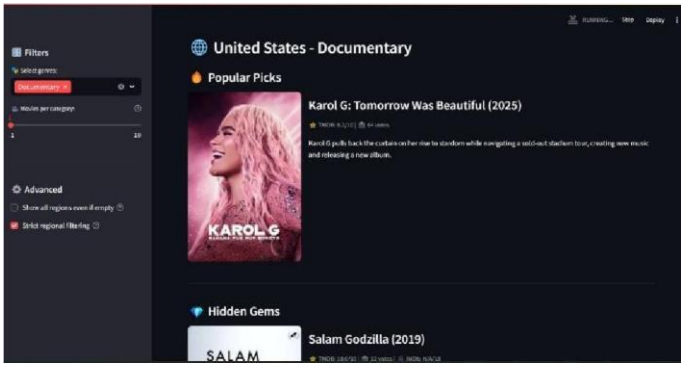

Fig 4.3 Output of the movie recommendation system

Different Genres of Movie of different regions available are:

Action
Adventure
Horror
Romance
War
Drama
Fantasy
Crime
Comedy

Maximum Number of movies this system can recommend, ranges from 1 to 10

V - Future Enhancements:

Although the movie recommendation chatbot has already demonstrated an efficient performance, there are several aspects of its improvement: more personalized, flexible, and comprehensive service.

- **Personalized Recommendations Based on User History:**

  One of the main future changes would be the deployment of personalized recommendations for user interaction history. Through studying the previous tastes like genres, certain movies or topics that the user had engaged, the chatbot could make its ideas more precise. This would not only be making the recommendations useful but it will also aid the system to know what the user wants next. Personalization establishes closer rapport with the user that makes the chatbot more intuitive and innovative according to the user's preferences.

- **Sorting and Filtering by IMDb Ratings or Release Year:**

  To improve the control of the users and to make the searching for a perfect movie more effective, it would be reasonable to sort (or filter) the resulting movies by certain properties, for example, by the IMDb score or a release year. Users could turn on listing of high-rated only movies or filter the content only by the movies recently released. This would not only speed up the decision-making process of the users, but also when they are searching for popular or recently updated content, but want to find critically acclaimed movies. Also, this feature would enable users to quickly home in on what they care most about thus enhancing overall user interaction due to the dynamicity and customizability that would be imparted to the user interaction

- **Expanding Language Support and Adding Streaming Availability Data:**

  The next major improvement, then, would be the broader language support for the system, so that it may serve a worldwide audience. The next major improvement, then, would be the broader language support for the system, so that it may serve a worldwide audience. By including more languages in the movie recommendations, the system could better serve users from different linguistic backgrounds, providing localized movie choices based on their region. Furthermore, adding streaming availability data to the chatbot would significantly increase its utility. Users would be able to know immediately where a movie is available for streaming— whether on Netflix, Amazon Prime, or other platforms—making the chatbot more practical and helping users avoid unnecessary searches across various streaming services. This feature would bridge the gap between recommendation and accessibility, enhancing user satisfaction by simplifying their viewing decisions.

- **Offline Fallback Using TMDB Dataset Caching:** For users in areas with limited internet access or in situations where they temporarily lose connectivity, the chatbot

  Using more language inclusions, the system could be more helpful to those using a different language as it would present localized movie choices made according to their region. Additionally, the streaming availability data would represent a major addition to the chatbot in terms of services provided. The chatbot would be useful in the sense that users would know the next move instantly; i.e. whether a movie is up for streaming on Netflix, Amazon Prime or any other streaming services, thus saving a user time from performing a needless quest on different streaming services to verify which service offers streaming privileges on This feature would close the gap between recommendation and accessibility making user satisfaction easier with an uncomplicated choice for their viewing.

VI- CONCLUSION:

For users in locations with poor internet access, or when connection is lost, the chatbot might offer an offline graceful degradation. If the system cached some part of the TMDB dataset, it could recommend movies, even disconnected from the internet. Even though the recommendations would only extend to the previously cached data, the users could browse via flick beats by using the data stored in their device. This improvement would guarantee the system is still usable in low-connectivity settings, making it more robust, and accessible – and particularly important for people who might not always be able to access reliable internet connectivity. The optimization of the API integration and the webbased deployment so that the project achieves functional and user-oriented effects within the movie recommendation system is the focus of this project. The chatbot combines TMDB and OMDb APIs and, because of it, is able to deliver to users the up-to-date, real-time movie data on demand. The smooth integration between these APIs and the frontend, developed on Streamlit, is a lesson on how web technologies can be utilized to generate an interactive platform that is equally responsive and interactive to the users. The chatbot has been made flexible in various areas with many customizable options including genre selection, movie count alterations, and language options that can be configured to let users whittle down suggestions to their preferences. Apart from its flexible interface, the system is strong and stable. It features smart solutions for missing data, uses IMDb ratings for

credibility and has fallback mechanisms that will guarantee any user will get something useful even when the information is lacking. Such features enhance effortless use that guarantees consistent attention and satisfaction. This project also doubles up as a good example of the construction of web-based applications for practical applications in the real world, for example, personalized media discovery. It's easy to understand interface as well as its scaled architecture places it favourably in contexts where ease of usage and performance is of essence. Moreover, the fact that the app can be extended to gain new functionality – personalized recommendations, sorting according to IMDb score, and even a capability for offline viewing – prepares it for expansion and optimization. Finally, the project demonstrates the way combining thirdparty APIs with current web frameworks such as Streamlit can lead to capable, interactive applications. Providing personalized and flexible movie recommendations, the chatbot addresses various users and provides an efficient and pleasurable movie discovery procedure. With future improvements in mind, this system can become a very useful tool for users, who want to get custom movie suggestions.

VII - REFERENCES:

[1] Santosh Kumar Majhi, Preetipadma Sahani, Rosy Pradhan, Biswaranjan Acharya, Foteini Grivokostopoulou , Andreas Kanavos and Vassilis C. Gerogiannis, 'Comparative Analysis of Movie Recommendation Systems Using Filtering Techniques on IMDB', Department of Computer Science and Information,Technology Guru Ghasidas Vishwavidyalaya (Central University), Bilaspur, India. (IEEE 2024)

[2] A. Deenu Mol, Sundaram Arumugam, B.Rithikaa 'Intelligent Movie Recommendation Platform with AI-Driven Recommendation using Machine Learning with Optimized Power Saving Methods' , Department of Information Technology Kongu Engineering College Erode, India (IEEE 2022)

[3] Carmen Pei Ling Tung, Su-Cheng Haw, Wan-Er Kong and Palanichamy Naveen, 'Movie Recommender System Based on Generative AI' , Faculty of Computing and Informatics Multimedia University 63100, Cyberjaya, Malaysia. (IEEE 2024).

[4] Karan Shah;Bhavna Arora;Aliraza Shinde;Shreyas Vaghasia, AI in Entertainment Movie Recommendation using cosine similarity IEEE 2022)

[5]Veena Mittal;Abhijeet Singh;Anmol;Moksh, Movie Recommendation System by Feed Forward Deep Neural Network (IEEE 2021)

[6] Ravikumar R N;Sanjay Jain;Manash Sarkar, Personalized Movie Recommendation Based on User Preferences Using Optimized Sequential Transformer Model , IEEE 2022

[7] Rupal Verma;Anjali Diwan , Movie Recommendation System by Using Collaborative Filtering and Louvain Algorithm, March 2023.

[8] Yihan Xu;Nan Xie;Xinjun Hu;Weimin Chen , Development of a Movie AI Question Answering System Based on Knowledge Graph , IEEE 2024

[9] Megha Sahu;Khetwat Saritha, Study on Various Collabrative Filtering Techniques to Recommend Movies, (IEEE 2021)

[10] Biresh Kumar;Vishal Kumar Nayak;Pallab Banerjee;Mohan Kumar Dehury An Empirical Analysis of Collaborative and Content Based Recommender System by using AI ,International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) (IEEE 2023).

[11] Awliya Hanun Izdihar;Nazriyah Deny Tsaniyah;Faraz Nurdini;Belva Rizki Mufidah;Nur Aini Rakhmawati,Buildinga Movie Recommendation System Using Neo4j Graph Database: A Case Study of Netflix Movie Dataset, 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS).

[12] Lixia Luo;Zuoliu Huang;Qitao Tang, Research and Implementationof Movie Recommendation System Base d on Knowledge Graph, 2023 4th International Conference on Computer, Big Data and Artificial
Intelligence (ICCBD+AI).

[13] Shreya Agrawal;Pooja Jain, An improved approach for movie recommendation system, 2020 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC).

[14] Teh-Lu Liao;Yi-You Hou;Yun-Han Tseng, Movie Recommendation System Based on Facial Emotion Recognition, 2024 IEEE International Conference on Future Machine Learning and Data Science (FMLDS)

[15]Devvrat Siwach;Falguni Rakhecha;PS Thanigaivelu, Movie Recommendation System using DBSCAN Algorithm, 2024 9th International Conference on Communication and Electronics Systems (ICCES).

[16] C Gururaj, Satish Tunga, "AI based Feature Extraction through Content Based Image Retrieval", Journal of Computational and Theoretical Nanoscience, February 2020, volume 17, Issue 9-10, pp. 4097-4101, ISSN: 1546-1955 (Print): EISSN: 1546-1963 (Online), DOI: 10.1166/jctn.2020.9018

[17] Veena Nayak, Sushma P. Holla, K.M. Akshayakumar, C. Gururaj, "Machine Learning Methodology toward identification of Mature Citrus fruits", Computer Vision andRecognition Systems using Machine and Deep Learning Approaches IET Computing seriesvol. 42, Chiranji L.C., Mamoun A., Ankit C., Saqib H. and Thippa R.G., Eds, London, TheInstitution of Engineering and Technology, November 2021, Ch. 16, ISBN: 978-1-83953-323-5,pp.385-438,DOI: 10.1049/PBPC042E_ch16

[18] MK Rahul, Praveen L. Uppunda, Sumukh B, Raju S Vinayaka, C. Gururaj, "Simulationof Self-Driving Cars Using Deep Learning", Fundamentals and Methods of Machine andDeep Learning: Algorithms, Tools and Applications, Pradeep Singh Ed, February 2022, Ch.16, Scrivener Wiley Publications, ISBN: 9781119821908, ch. 16, ISBN: 9781119821250,pp379 – 396, DOI: 10.1002/9781119821908.ch16

[19] KS Srujana, Sukruta N Kashyap, G Shrividhiya, C. Gururaj, KS Induja, "Supply Chain
Based Demand Analysis of Different Deep Learning Methodologies for Effective Covid-19Detection", Innovative Supply Chain Management via Digitalization and ArtificialIntelligence, SpringerPublishers, Kumaresan Perumal, Chiranji Lal Chowdhary, Logan Chella Eds, April 2022, Chapter 9, ISBN: 978-981-19-0239-0, pp 135-170, DOI: 10.1007/978-981-19-0240-6_9

[20] Avani KVH, Deeksha Manjunath, C. Gururaj, "Deep Learning Based Detection of
Thyroid Nodules", Multidisciplinary Applications of Deep Learning-Based ArtificialEmotional Intelligence, IGI Global Publishers, Chiranji Lal Chowdhary Ed., November 2022,
Chapter 8, ISBN: 9781668456736, pp 107 – 135, DOI: 10.4018/978-1-6684-5673-6.