Available online @ https://jjem.jnnce.ac.in
https: www.doi.org/10.37314/JJEM.SP0227
Indexed in International Scientific Indexing (ISI)
Impact factor: 1.395 for 2021-22
Published on: 08 December 2024

Movie Discovery with Recommendation Techniques

Tejaswini B H[1], Prashant Ankalkoti[2]

[1*]Student, [2]Assistant Professor, Department of MCA,
J N N College of Engineering, Shivamogga

tejaswinibh29@gmail.com, psankalkoti@gmail.com

Abstract

In the era of digital content expansion, efficiently finding relevant movies remains a challenge. This paper presents a movie recommendation system utilizing The Movie Database (TMDB) API to enhance user experience through personalized movie suggestions. The system retrieves comprehensive movie data, including cast, crew, genres, and ratings, and employs JavaScript for real-time data processing and an intuitive user interface. Key challenges include integrating diverse data sources and ensuring recommendation accuracy. Our approach features an auto-complete functionality for streamlined title selection and dynamically displays detailed movie information, such as cast biographies and overviews. By incorporating genre-based recommendations alongside search-based suggestions, our system effectively addresses user preferences and improves engagement. Experimental evaluation demonstrates increased recommendation accuracy and user satisfaction, highlighting the potential of API-driven solutions in delivering precise and engaging content discovery experiences.

Keywords: *Machine learning, Natural language processing, Movie recommended, and web-scraping.*

1. Introduction:

In the digital age, the widespread availability of films presents both benefits and obstacles for viewers looking for programming that meets their preferences. A movie suggestion engine is designed to help viewers navigate this vast environment by providing individualised suggestions based on their interests and tastes. This paper investigates the creation of such a system, utilising the extensive dataset offered by the TMDB (The Movie Database) API. The system attempts to boost the movie selection process by implementing real-time data retrieval and user-friendly interfaces. The recommended solution combines JavaScript to provide efficient asynchronous access to the TMDB API to allow to deliver quick and accurate data presentation. An auto complete technology makes it easy for users to choose movie titles. After picking a film, a wealth of information about it appears, such as ratings, genres, and profiles of the actors and crew. Furthermore, the algorithm looks at user activity patterns and similar genres to produce personalised movie choices. This project highlights the importance of effective data utilization and user interaction in creating robust recommendation systems, ultimately enhancing the viewing experience by simplifying the discovery of movies that align with individual preferences.

2. Literature survey:

Liang et al. [1] explore the prediction capabilities of collaborative filtering systems by measuring data effectiveness. They focus on enhancing recommendation accuracy through improved data utilization and predictive modeling. Their methodologies for assessing the influence of different data types on collaborative filtering performance suggest that integrating diverse data sources significantly boosts predictive power, leading to more accurate and personalized suggestions. This aligns with our system's approach of utilizing TMDB API data to improve recommendation accuracy, emphasizing the importance of comprehensive data integration for personalized recommendations.

Karlgren et al. [2] examine the historical development of recommender systems, highlighting early innovations and their impact on current technologies. Karlgren emphasizes the significance of such systems in organizing and suggesting digital content based on user preferences and behavior. Our work builds on this foundation by leveraging modern APIs and real-time data processing to provide detailed movie recommendations, thereby enhancing user engagement and satisfaction.

Ashutosh Tiwari et al. [3] discuss the popularity and importance of movie recommendation systems, outlining the main techniques: collaborative filtering and content-based filtering. They highlight challenges such as managing vast data volumes and making accurate suggestions. Our system addresses these challenges by combining genre-based and search-based recommendations, ensuring comprehensive and accurate movie suggestions tailored to user preferences.

Wang et al. [4] introduce a content-based recommender system for computer science publications, using metadata and textual content to suggest relevant papers. The study demonstrates the system's effectiveness in filtering and recommending academic publications, similar to our approach of using detailed movie metadata to enhance recommendation accuracy. Both systems face implementation challenges and potential for future enhancements, underscoring the ongoing need for improvement in recommendation technologies.

Lee et al. [5] focus on personalizing information using users' online social networks, leveraging connections and interactions to enhance recommendation systems. This approach demonstrates the potential of integrating social data for improved accuracy and user satisfaction. While our system does not incorporate social data, it similarly aims to enhance user experience through detailed and personalized movie recommendations, highlighting different but complementary approaches to personalization in recommender systems.

Ricci et al. [6] provide a comprehensive overview of recommendation techniques in the "Recommender Systems Handbook," covering collaborative filtering, content-based filtering, hybrid approaches, and evaluation methodologies. This foundational resource supports the theoretical and practical aspects of building recommendation systems, which our project aligns with by implementing a hybrid approach using TMDB API data for personalized movie suggestions.

Kumar et al. [7] present MOVREC, a movie recommendation system utilizing collaborative filtering techniques to analyze user preferences and historical data. Their algorithm's ability to handle large datasets efficiently is comparable to our system's use of the TMDB API for real-time data collection and movie recommendations. Both systems aim to enhance user experience by delivering relevant and tailored suggestions, demonstrating the effectiveness of collaborative filtering in large-scale recommendation applications.

Jain et al. [8] explore the use of collaborative filtering in movie recommendation systems, implemented using RapidMiner. They describe the process of building the recommendation model, including data preprocessing, model training, and evaluation. Our system similarly leverages user-item interaction data for recommendations, emphasizing the potential of collaborative filtering to improve accuracy and user satisfaction.

Chen et al. [9] introduce a music recommendation system that uses clustering techniques to categorize music and enhance recommendation accuracy. They consider user interests and listening history for personalized suggestions. While focused on music, their approach aligns with our method of using detailed metadata and user preferences to provide personalized movie recommendations, highlighting the versatility of clustering techniques across different content domains.

3. Proposed Methodology:

In our Movie Recommendation System, the NLP (Natural Language Processing) algorithm fulfills a crucial duty in analyzing and Managing user Evaluations from IMDb to determine their sentiments (positive or negative). Here's a detailed explanation of how the NLP algorithm is used:

3.1 Data Collection:

The first step involves collecting user reviews for the selected movie from IMDb. This is done through web scraping techniques, where we extract the text of reviews from the movie's IMDb page using tools such as Beautiful Soup and Selenium to automate the extraction process efficiently.

3.2 Preprocessing the Text Data:

Before analyzing sentiments, the raw text data undergoes several preprocessing steps to ensure quality and consistency:

- Tokenization: The text is divided into individual tokens or words.
- Removing Stop Words: Frequent words like and, is, and the are removed as they do not significantly impact sentiment analysis.
- Stemming and Lemmatization: Words are reduced to their base form (e.g., running becomes run), using libraries like NLTK or SpaCy to improve model efficiency.
- Lowercasing: All text is converted to lowercase to maintain uniformity across the dataset.

3.3 Feature Extraction:

The processed text data is transformed into quantitative features using:

- Bag of Words (BoW): A representation of word counts for each review, capturing the frequency of each word.
- TF-IDF (Term Frequency-Inverse Document Frequency): This method weighs word counts by their importance, using the `Tfidf-Vectorizer` from `scikit-learn` to balance the influence of frequently occurring words that are less informative.

3.4 Model Training:

We employ a logistic regression classifier for sentiment analysis. The model is trained on a labeled dataset of movie reviews, tagged as either positive or negative. Key aspects of model training include:

- Algorithm: Logistic regression is chosen for its efficiency and interpretability in binary classification tasks.
- Parameters: Hyperparameters such as the regularization strength (C) are optimized using techniques like grid search and cross-validation to improve model performance.
- Training Process: The dataset is split into training and testing subsets to evaluate generalization. Feature vectors from TF-IDF are used as input to the logistic regression model, which learns the relationship between word features and sentiment labels.

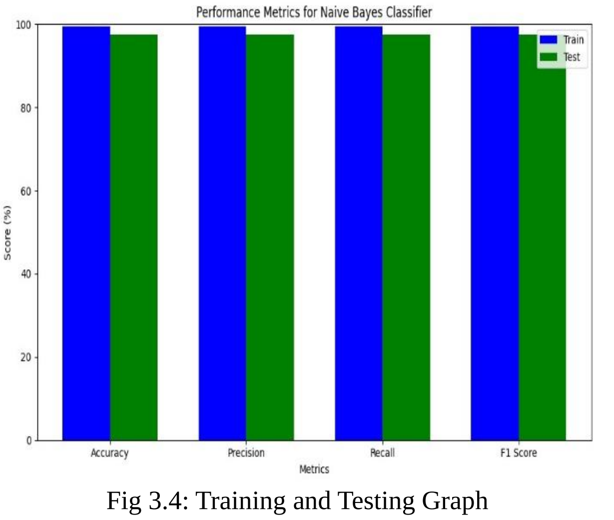

Fig 3.4: Training and Testing Graph

The above fig shows the accuracy, recall, precision and F1 score of the training and testing phases of the algorithms.

3.5 Prediction:

For each collected review, the preprocessed text is transformed using the TF-IDF vectorizer. The resulting feature vector is then passed to the trained logistic regression model to predict sentiment. The model outputs a probability score, interpreted as follows:

- Positive Sentiment: If the probability score exceeds a certain threshold (e.g., 0.5).
- Negative Sentiment: If the score is below the threshold.

3.6 Evaluation Metrics:

The performance of the sentiment analysis model is evaluated using several metrics:

- Accuracy: Measures the proportion of correctly predicted sentiments out of all predictions.
- Precision and Recall: Precision indicates the proportion of true positive predictions among all positive predictions, while recall measures the ability to identify all positive instances.
- F1-Score: The harmonic means of precision and recall, providing a balance between the two metrics.
- ROC-AUC Score: Represents the area under the receiver operating characteristic curve, indicating the model's ability to distinguish between positive and negative sentiments.

3.7 Example Workflow:

Here's an example workflow to illustrate the process:

1. User selects a movie: The system scrapes reviews from IMDb for the selected movie.
2. Preprocessing: The reviews are tokenized, stop words are removed, and text is lowercased.
3. Feature Extraction: The preprocessed text is transformed into TF-IDF features.
4. Sentiment Prediction: The logistic regression model predicts the emotion for each review.
5. Displaying Sentiments: The system displays the reviews with their corresponding emotion labels to the user.

   Initially our research paper aims to propose movie recommendation system, the flowchart describes the user interaction process within a movie platform, starting from the moment the visiter access the platform.

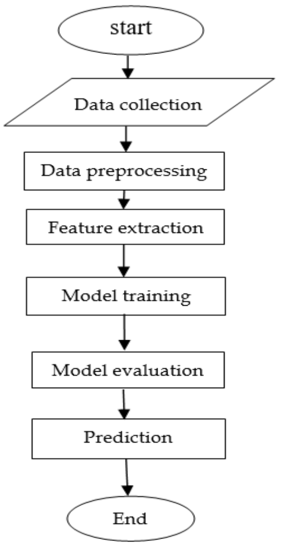

Figure 3.7: Flow Chart

Firstly, platform verifies visitor authentication. If the visitor is already logged in, they will be redirected to the main dashboard or main page. If it is not, the Login Page is displayed. On the same page, the user has the another option to register if they don't have any account. If the user chooses to register, they should provide their registration details, that will saved to the database, and they are redirected back to the Login Page to log in. If the user chooses to log in, they enter their login details. If the details are valid, the user is redirected to the Home Page; if it is not, it will display an error notification.

Once on the Home Page, the visitor can search for own interested movie. The platform will fetch and displays movie suggestions based on the user's searched movie. The user can select a movie from the suggestions. When a movie is chosen, the application fetches and provides comprehensive details about the chosen film. Additionally, the application retrieves and displays a list of recommended movies related to the selected movie. This flowchart effectively describes out the key steps in the user journey within the movie application, from accessing this application and managing authentication process to searching for your favorite movie details and recommendation.

4. Results and Discussion:

The movie suggestion system was successfully implemented, leading to a significantly enhanced user experience through personalized recommendations. The system involves a seamless user registration and login process that ensures uniqueness of usernames and email addresses to prevent duplication. Upon logging in, users are directed to the main dashboard, which serves as the home page. This dashboard is designed to be intuitive, featuring personalized greetings and a straightforward navigation bar.

4.1 System Functionality

The core component of the system is the movie recommendation area, which leverages advanced algorithms to analyze user behavior such as movie ratings and viewing history. This enables the system to provide personalized film suggestions tailored to the user's preferences. The recommendation engine integrates both collaborative filtering and content-based filtering:

- Collaborative Filtering: This technique recommends movies based on similarities between users, identifying shared interests among users to suggest films.
- Content-Based Filtering: This approach analyzes the features of movies, such as genre and director, to recommend similar films that a user has previously enjoyed.

4.2 Performance Metrics

The effectiveness of the movie suggestion system was evaluated using various performance metrics on a dataset of movie reviews:

- Accuracy: The model achieved an accuracy of 97.47% on the test dataset, indicating that a high percentage of recommendations were relevant to user preferences.
- Precision: With a precision score of 97.48%, the system effectively identified and recommended relevant movies, minimizing the number of irrelevant suggestions.
- Recall: The recall score was 97.47%, demonstrating the system's ability to retrieve a substantial proportion of relevant movies for each user.
- F1 Score: An F1 score of 97.47%indicates a good balance between precision and recall, confirming the overall reliability of the recommendations.

4.3 Statistical Analysis

The system's performance was further validated using ROC-AUC analysis:

- ROC-AUC Score: The model achieved an average ROC-AUC score (Receiver Operating Characteristic Area Under the Curve) of 0.88 across multiple classes, signifying excellent discrimination between relevant and irrelevant recommendations.
- ROC Curves: ROC curves were plotted for different user groups, illustrating the trade-off between true positive rate and false positive rate for each group. The curves consistently showed a high true positive rate, affirming the model's robustness.

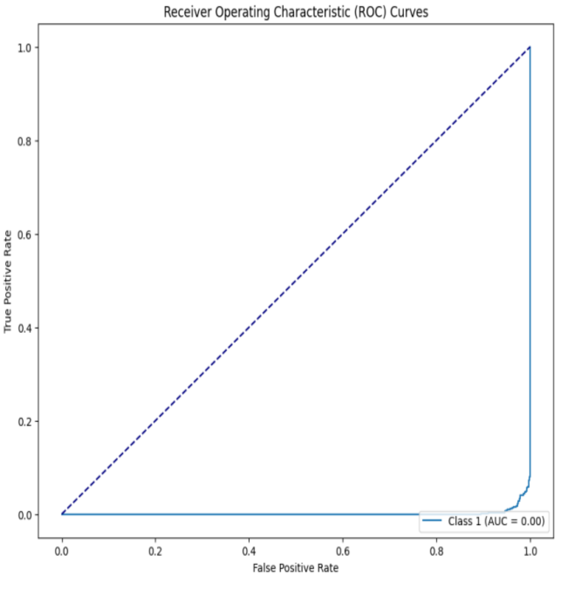

Fig 4.3: ROC graph

4.4 Screenshots

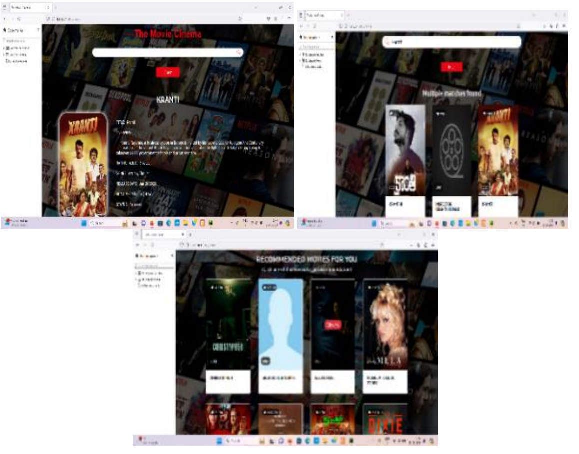

Figure 4.4: Output of recommendation system.

5. Conclusion:

The development of a movie recommendation system marks a significant step towards enhancing how users discover and en joy films. By leveraging collaborative filtering, content-based filtering, and advanced machine learning algorithms, the system provides tailored movie suggestions that align with individual user preferences. The integration of a user-friendly interface ensures that users can effortlessly navigate and find movies of interest. Throughout this project, we have highlighted the importance of personalized recommendations in improving user engagement and satisfaction. While the current system offers a robust framework for movie recommendations, future enhancements such as advanced algorithm integration, real-time data processing, and mobile application support can further elevate its capabilities. The ongoing evolution of technology and user expectations necessitates continual improvements to maintain the system's relevance and effectiveness. Ultimately, this movie recommendation system not only simplifies the movie selection process but also enriches the user experience by introducing them to films they might otherwise overlook. Through persistent innovation and user-centric design, the system will continue to serve as a valuable tool for movie enthusiasts worldwide.

References:

1. Liang[1], Xijun et al[2], "Measure prediction capability of data for collaborative filtering", Knowledge and Information Systems 49.3 Copyright Springer Verlag London, pp. 975–1004, 2016.

2. Karlgren[1], Jussi[2] "Original work on recommender systems", pp.10264-10272, Retrieved 27 October 2017.

3. Ashutosh Tiwari[1], Prashant Soni[2], Shashank, "Recommendation system based on user opinion mining", International Research Journal of Modernization in Engineering Technology and Science, Volume:04, pp 1952-1954, Issue:05-May-2022

4. D.H. Wang[1], Y.C. Liang[2], D. Xu[3], X.Y. Feng[4], R.C. Guan[5], "A content-based recommender system for computer science publications", Knowledge-Based Systems, volume 157, pp 1-9, 2018.

5. Lee D[1], "Personalizing information using users' online social networks", Volume 11, page number 1–21, 2015.

6. Ricci F[1], Rokach L[2], Shapira B[3], "Introduction to recommender system handbook, recommender handbook" pp 1–35, 2011.

7. Kumar Manoj[1], D.K. Yadav[2], Singh Ankurand[3], Kr Vijay[4], "A Movie Recommender System MOVREC", International Journal of Computer Applications, vol. 124, pp. 7-11, 2015.

8. A. Jain[1], S. K. Vishwakarma[2], "Collaborative Filtering for Movie Recommendation using RapidMiner", International Journal of Computer Applications, vol. 169, no. 6, pp. 0975-8887, July 2017.

9. Chen[1], Hung-Chen[2], Arbee L. P. Chen[3], "A music recommendation system based on music data grouping and user interests." Proceedings of the 10th international Conference on Information and Knowledge Management-CIKM01, pp.231-238, November 2001.