International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Movie Recommendation Chatbot Using Python

Aarushi Singh*[1], Anushka Kashyap*[2]

*[1] Department of Computer Science and Engineering, K.R. Mangalam University, Gurugram, Haryana, India
*[2] Department of Computer Science and Engineering, K. R. Mangalam University, Gurugram, Haryana, India

ABSTRACT

Personalized content distribution is now quite important in the digital environment of today. We present a Mood-Based Movie Recommendation Chatbot created with Python and multiple machine learning tools. Analyzing user input text, the chatbot recommends movies to them depending on their emotional state and tastes. We created a complete movie database using IMDb and TMDB datasets and applied natural language processing methods to grasp user moods. This paper addresses the methodology, design architecture, findings from the real-world testing of the chatbot, difficulties during development, comparison with conventional recommendation systems and the consequences of mood-based advice in improving user experience.

Keywords: Movie Recommendation, Chatbot, Mood Detection, Natural Language Processing, Python, User Experience

I. INTRODUCTION

Selecting the appropriate movie has become difficult for consumers in the era of streaming platforms and great content availability. Conventional recommendation systems sometimes mostly rely on viewing history, ratings, or popularity measures, which might not always fit a user's emotional situation right now. Understanding this discrepancy, our project intends to design a chatbot that detects the mood of the user and recommends suitable movies based on that. By emphasizing natural language processing (NLP) and machine learning, we allow a more personal, intuitive interaction between users and recommendation systems. Extensive research has been done on recommendation systems using approaches ranging from cooperative filtering to content-based filtering is another name for Although efficient, these systems sometimes fail to meet real-time emotional needs. For example, someone who usually watches thrillers could want a lighthearted comedy following a demanding day. Our chatbot dynamically responds to such emotional changes, so improving user pleasure. Furthermore under increasing evidence are mood-centric systems' ability to boost user retention, extend engagement time, and create closer emotional connection with the platform. Thus, including mood analysis into recommender systems presents both a technological difficulty and a potential breakthrough.

II. METHODOLOGY

System Synopsis :

Three stages define the movie recommender chatbot mostly: user input analysis, mood classification, and movie suggestion. The aim is to create a light-weight yet effective conversational interface with emotional awareness and response capability.

Data Construction:

We apply two primary datasets:

- IMDb dataset for movie ratings, genres, and basic descriptions.
- TMDB dataset for additional metadata such as mood-related tags, plot summaries, and user reviews.

Data cleaning involved removing inconsistencies, handling missing values, and standardizing genre tags. Additional mood labels were semi-manually created based on plot summaries and existing genre categorizations.

Natural Language Processing (NLP) for Mood Detection To understand the user's mood from text input, we implemented:

- Tokenization and Stop-word removal using NLTK.
- Lemmatization to normalize words.
- Sentiment analysis using VADER for social media text and TextBlob for general sentiment scoring.

- Classification of input into predefined moods: Happy, Sad, Angry, Romantic, Adventurous, Scared, and Calm.

Sample classification keywords:

- Happy: "excited", "joyful", "celebrate"
- Sad: "heartbroken", "upset", "crying"
- Angry: "furious", "irritated", "angry"

Recommendation Engine

Based on the identified mood, the chatbot filters movies with tags and genres matching the emotional context. Genre mapping was manually curated to ensure high relevance.

Chatbot Development

We developed the chatbot backend using Python. The main libraries used include:

- Flask (for setting up a lightweight server)
- Pandas and NumPy (for data manipulation)
- Scikit-learn (for basic classification tasks)

III. MODELING AND ANALYSIS

Architecture Design

The chatbot follows a modular architecture:

1. **User Input Module:** Captures raw user queries.
2. **Mood Detection Module:** Analyzes emotional content.
3. **Movie Database Module:** Searches relevant movies.
4. **Response Generator:** Sends movie suggestions to the user.

User Input → NLP Engine → Mood Classifier → Movie Filter → Response Generator

Data Cleaning

- Handled missing movie metadata using mean substitution and manual annotation.
- Normalized genre and tag fields.
- Filtered out movies with fewer than 500 user ratings to ensure quality recommendations.

Mood Classification Model

Initially, a basic rule-based approach was applied using sentiment scores. Later, a simple logistic regression model trained on emotion-labeled sentences improved accuracy.

Comparison of Mood Classification Models:

Model	Accuracy
Rule-Based Sentiment	72%
Logistic Regression	81%
Multinomial Naive Bayes	79%

Sample Sentences and Predicted Moods:

User Input	Predicted Mood
"Feeling a bit low today."	Sad
"Let's party tonight!"	Happy
"I want some adventure."	Adventurous

IV. RESULTS AND DISCUSSION

Performance Evaluation

We evaluated the chatbot with a group of 50 users, each asked to describe their mood and rate the suggestions.

User Mood	Avg. Satisfaction Rate
Happy	85%
Sad	82%
Angry	78%

User Mood	Avg. Satisfaction Rate
Romantic	88%
Adventurous	83%
Scared	80%
Calm	84%

Comparison with Traditional Systems

Traditional collaborative filtering systems (like Netflix's basic algorithm) performed adequately in content personalization but lacked emotional adaptability. Our mood-based chatbot achieved a 15% higher satisfaction rate among users who reported fluctuating emotions.

User Feedback

- "Loved the emotional touch!"
- "Sometimes misunderstood my mood, but overall great suggestions."
- "Would be awesome to have voice interaction!"

Challenges

- Ambiguous input (e.g., "I'm not sure") posed classification issues.
- Overfitting to common moods like "happy" during initial model training.
- Limited movie options for rare moods like "calm" or "adventurous".

V. FUTURE SCOPE

- **Voice-based Interaction:** Integrating speech-to-text models for voice-driven conversations.
- **Continuous Learning:** Reinforcement learning to adapt recommendations based on user feedback.
- **Multilingual Support:** Handling inputs in Hindi, Spanish, and other major languages.
- **Rich Context Understanding:** Using transformers like BERT for deeper emotional analysis.

VI. RELATED WORK

Several research studies have explored context-aware recommenders. However, most focus on location, time, or demographic information rather than emotional context. Work by Koren et al. introduced matrix factorization for collaborative filtering, but it lacked emotional dynamics. Recent advances in affective computing highlight the importance of emotion-aware systems, which aligns with our chatbot's objectives.

VII. CONCLUSION

In this research, we developed a chatbot that recommends movies based on the emotional state of the user, moving beyond conventional history-based recommendations. The system demonstrated promising results in matching user moods with appropriate movie choices, offering a personalized and satisfying entertainment experience.

Our findings validate the significance of mood-based recommendations and point towards an exciting frontier in personalized content delivery. With further improvements like voice recognition, multilingual capabilities, and real-time learning, such systems can redefine how users engage with digital entertainment platforms.

ACKNOWLEDGEMENTS

We express our sincere gratitude to our mentors, friends, and family who supported and guided us throughout this project. Special thanks to the open-source Python community for providing incredible resources and tools.

REFERENCES

[1] Koren, Yehuda, et al. "Matrix factorization techniques for recommender systems." Computer 42.8 (2009): 30-37.

[2] Liu, Bing. "Sentiment analysis and opinion mining." Synthesis lectures on human language technologies 5.1 (2012): 1-167.

[3] "IMDb Datasets", available at: https://www.imdb.com/interfaces/

[4] "The Movie Database (TMDb) API Documentation", available at: https://developers.themoviedb.org/3

[5] Hutto, C.J., and Gilbert, Eric. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text." Eighth International Conference on Weblogs and Social Media (ICWSM-14).

[6] Poria, Soujanya, et al. "A review of affective computing: From unimodal analysis to multimodal fusion." Information Fusion 37 (2017): 98-125.