Enhancing Movie Recommendations Through Hybrid Data Mining Techniques

Chanduvardhan Parachur
Montclair State University
Edison, NJ, USA
parachurc1@montclair.edu

Lokeswari Devi Avudoddi
Montclair State University
Jersey City, NJ, USA
avudoddil1@montclair.edu

Ranjit Kumar Chaganti
Montclair State University
Edison, NJ, USA
chagantir1@montclair.edu

Sanjay Gunda
Montclair State University
Jersey City, NJ, USA
gundas2@montclair.edu

Sarath Chandra Bhimineni
Montclair State University
Jersey City, NJ, USA
bhiminenis1@montclair.edu

Abstract: **With the exponential growth of digital content, users often face difficulty in selecting suitable movies that match their preferences. This paper presents a hybrid movie recommendation system that leverages clustering and association rule mining to provide personalized suggestions. Using the movie title and ratings datasets, the system preprocesses key features such as average rating, runtime, and number of votes. MiniBatch K-Means clustering is applied to segment movies into meaningful groups based on these features, improving computational efficiency and scalability. Furthermore, association rule mining uncovers relationships between user preferences and movie genres, enhancing the recommendation accuracy. The integration of these machine learning techniques results in a system that offers more relevant and diverse movie suggestions. The experimental results show that the proposed model efficiently clusters and recommends movies, offering a practical solution for real-world recommendation scenarios.**

Keywords: Movie Recommendation System, MiniBatch K-Means, Association Rule Mining, Clustering, Data Mining, IMDB Dataset

I. Introduction

The vast amount of multimedia information available online in the constantly growing digital landscape has made it more and more challenging for users to locate content that suits their own preferences. With thousands of films available on sites like Netflix, Amazon Prime, and IMDb, viewers frequently spend more time searching than watching. Recommendation systems are now a crucial part of digital content offerings in order to address this issue. By analyzing past data, user behavior, and preferences, these intelligent systems recommend tailored content that increases user happiness and engagement.

In order to provide precise and effective recommendations, this study introduces a movie recommendation system that leverages the strengths of association rule mining and clustering. We grouped films with comparable features and extracted significant patterns using the IMDB movie dataset and rating dataset, a popular benchmark in scholarly research. The algorithm divides movies into groups according to criteria including runtime, average rating, and number of votes using MiniBatch K-Means clustering. In order to make the recommendations more contextually rich, association rule mining is also used to reveal hidden connections between user preferences and genres.

Our hybrid model makes use of the advantages of both content-based and collaborative filtering, in contrast to standard systems that frequently rely on just one method. By grouping films according to measurable criteria, we lower the dataset's dimensionality and identify significant clusters. How consumers usually browse through comparable movie genres is then shown by the association rules that are mined inside these clusters. As a result, the final suggestions are tailored to subtle patterns present in smaller, more uniform movie chunks in addition to being pertinent.

Recommendation systems have a big influence on platform dynamics and content exposure in addition to enhancing user experience. These algorithms can benefit independent artists and encourage variety in content by intelligently recommending obscure or niche films based on user likes. Our methodology seeks to balance revealing undiscovered gems that suit user preferences with suggesting well-known titles. By doing this, we improve the personalization element while also fostering a more diverse and interesting content environment. This study's combination of association mining and clustering guarantees that recommendations stay relevant and varied, opening the door for future recommendation systems that are more intelligent and flexible.

II. Literature Review

Through a variety of approaches, a number of researchers have advanced movie recommendation systems. With an emphasis on user-item interaction patterns and solving issues like scalability and data sparsity, Aggarwal and Charu [1] investigated collaborative filtering algorithms specifically in the context of movie recommendations. They talked about how to increase suggestion accuracy by applying neighborhood-based techniques and matrix factorization to user rating data. Their research also clarifies how recommender system architectures have changed over time and how collaborative filtering can be used as a basis for hybrid systems. A solid conceptual framework for creating systems that gradually adjust to user preferences is offered by this study.

In order to get over drawbacks like the cold start and sparsity issues, Shi et al. [2] suggested a hybrid recommendation strategy that combines collaborative filtering with content-based methods. Their approach, which made use of metadata including genre, cast, and director details in addition to user behavior, greatly increased prediction accuracy in the movie sector. In order to show that hybrid models regularly outperform conventional single-method approaches, the research also includes experimental assessments on real-world movie datasets such as IMDB movie dataset . Even with insufficient data, the approach improves customisation by combining implicit and explicit feedback.

A latent topic model that improves recommendation systems through a probabilistic framework was presented in another noteworthy study by Chatzis et al. [3]. Their methodology improves suggestion relevancy and system efficiency by grouping users and movies into latent categories based on interaction patterns. To find hidden preferences and relationships that conventional approaches could miss, the authors employ a generative approach based on topic modeling (similar to LDA). This approach is a workable alternative for real-time movie recommendation systems since it also allows for improved scalability when working with large-scale datasets.

In order to improve the scalability and personalization of movie recommendation systems, Zhou et al. [4] introduced a recommendation framework that makes use of clustering algorithms. The approach increases prediction accuracy and decreases computing complexity by employing K-Means clustering to group users with similar tastes. In order to find common itemsets and behavioral patterns within clustered groups and provide more pertinent movie recommendations, the authors integrated this using association rule mining. Their findings on sizable datasets such as MovieLens shown notable performance gains, particularly in settings where high dimensionality made traditional collaborative filtering difficult. The study shows how to build flexible and effective recommendation pipelines by fusing rule-based inference with unsupervised learning.

III. METHODOLGY

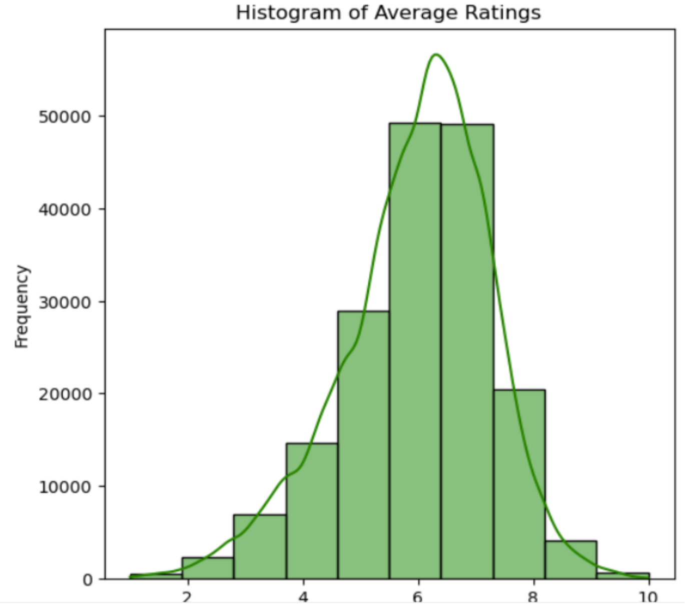
3.1 Data Collection and Pre-Processing

A) Dataset Acquisition: The IMDb (Internet Movie Database), a well-known website that offers extensive information about movies and TV series, served as the basis for the dataset used in this project, which was obtained from Kaggle. Movie titles, genres, runtime in minutes, user ratings, and the quantity of votes each film has received are among the many characteristics it includes. Building a strong movie recommendation system requires the dataset's rich mix of metadata that can be used to analyze user preferences and content features. It is the perfect starting point for using data mining and machine learning techniques in the field of tailored recommendations because of its variety and applicability to the real world.

B) Dataset Preprocessing: The dataset was preprocessed in a number of ways to guarantee its quality and suitability for study. First, columns that included missing or unnecessary values were found and either cleaned up or eliminated. In particular, incorrect entries were forced to NaN when the runtimeMinutes column, which occasionally had placeholder values like \N, was converted to numeric format. To facilitate quantitative analysis, the averageRating and numVotes fields were also converted to numeric forms. To preserve data integrity, rows with missing values in these crucial columns were then removed. The StandardScaler from the sklearn library was also used to standardize the numerical features, guaranteeing that every characteristic made an equal contribution to the clustering and modeling processes. Preparing the data for the successful use of association rule mining and clustering algorithms later on in the recommendation system required this preprocessing stage.

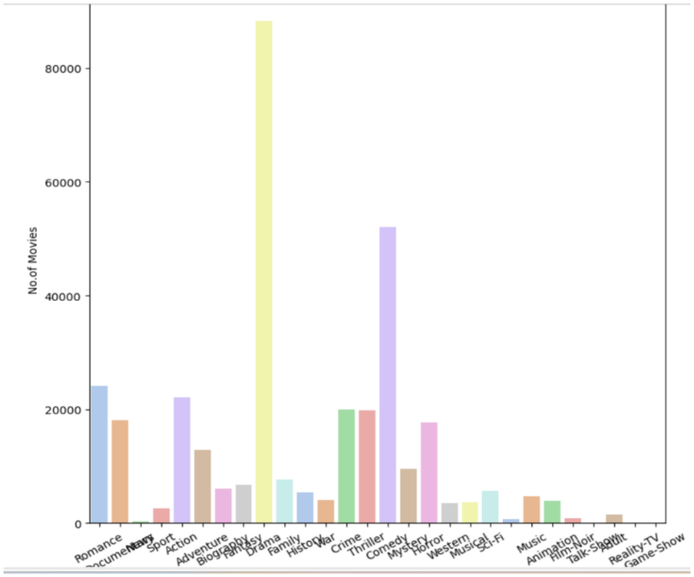
C) Feature Engineering: In order to refine the dataset and extract valuable insights that lead to precise suggestions, feature engineering was a crucial step. Three fundamental numerical attributes were chosen for this project: numVotes, runtimeMinutes, and averageRating. These characteristics were picked because they are important indicators of a film's popularity, audience reception, and running time—all of which frequently affect user choices. Later phases of rule mining also used categorical data, including genres, which were successfully cleaned and tokenized using text-based processing approaches. Although derived features were not produced, care was taken to make sure the pre-existing features were formatted appropriately for mining and grouping algorithms. The modeling method was able to concentrate on high-impact qualities because to the meticulous feature selection and treatment, which enhanced the caliber of the recommendation pipeline's clustering and association rule outputs.

D) Data Visualization: A histogram using a Kernel Density Estimate (KDE) was constructed in order to comprehend the distribution of movie ratings in the dataset. This graphic offers a thorough understanding of the distribution of average ratings for all films. The majority of films had average ratings between 3.0 and 4.0, according to the histogram, which showed a distribution that was significantly tilted to the right. Users prefer to rate movies favorably more often than negatively, as indicated by the KDE curve superimposed on the histogram, which makes it easier to recognize the central tendency and density concentration. An overall favorable bias in user ratings is suggested by this distribution, which is typical of recommendation datasets.



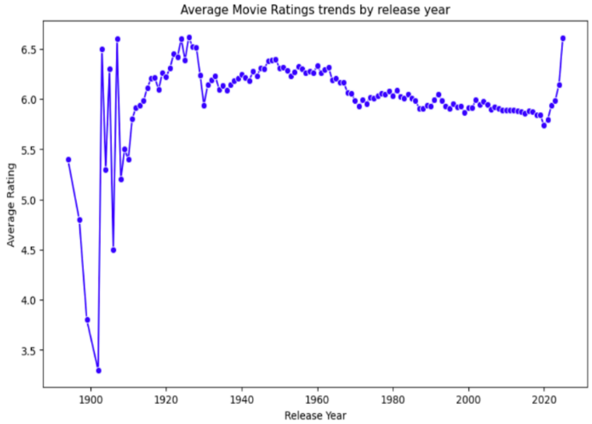
(Fig.1. Average ratings)

The amount of films accessible in each genre was also shown in a bar chart, providing information about the dataset's composition per genre. The Counter class was used to extract and count the genres, and Seaborn's barplot was used to illustrate the counts. The graph indicates that the dataset is dominated by genres like drama, comedy, and action, which are the most common. Because the prevalence of particular genres may affect clustering patterns and possibly slant recommendations toward more popular genres, this unequal distribution has ramifications for both clustering and recommendation performance. Therefore, it is essential to comprehend this distribution in order to guarantee fair and balanced suggestions across a range of user interests



(Fig. 2. No.of Movies per genre)

The average movie ratings by year of release are displayed in a line plot that shows how viewer preferences have changed over time. The impact of legendary films or shifting industry standards may be the cause of the notable variance in ratings, with some years seeing higher averages. Due of their enduring appeal, older films typically earn slightly higher ratings. This trend provides insightful information on the popularity and quality of movies throughout time.



(Fig. 3. Trends in movie rating)

3.2 Approach and Methods

A) Association Rule Mining:

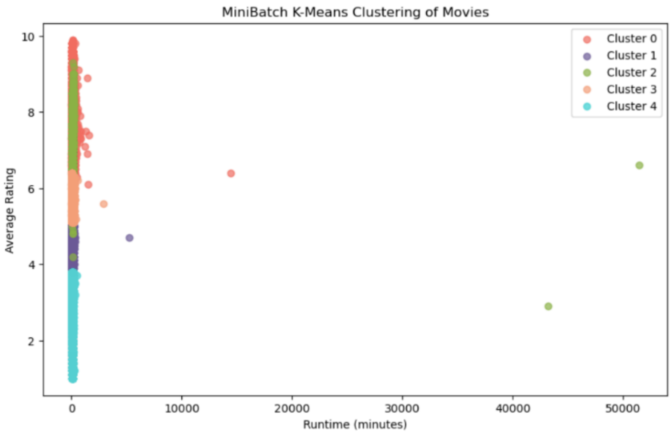
FP Growth algorithm implementation: This study used the FP-Growth algorithm for association rule mining to find hidden patterns in user preferences. Because it creates a compressed FP-Tree structure that does not require repeated database scans, FP-Growth is renowned for its efficiency when compared to more conventional techniques like Apriori. This made it ideal for examining the diverse range of film genres seen in the IMDB dataset. After converting the genres column into a transaction-like format—where each movie's genres were handled as distinct objects within a transaction—the algorithm was run. This change greatly increased the mining process's efficacy by enabling a more organized and analysis-ready format.

Frequent itemsets were extracted based on defined support and confidence thresholds, and meaningful association rules were generated to identify genre combinations that often appear together. These associations helped the system recognize patterns in user choices, enabling it to recommend movies with similar genre profiles. The use of FP-Growth allowed for faster processing and more scalable rule mining, while also ensuring that the results remained relevant and interpretable for real-world movie recommendations. The insights gained from these rules enhanced the personalization aspect of the system by aligning recommendations with users' genre preferences more precisely.

B) Clustering Techniques:

Clustering was used to group related films according to numerical characteristics like average rating, runtime, and number of votes in order to improve the recommendation process even further. To put these features on a consistent scale and keep any one feature from controlling the clustering process, they were normalized using the StandardScaler approach prior to clustering. Because of its efficiency and scalability in handling big datasets, the MiniBatch K-Means technique was selected. This approach is perfect for iterative clustering on big movie databases like IMDB since it generates mini-batches of the data to accelerate the convergence. The program effectively divided the dataset into discrete groups of films with comparable statistical characteristics by adjusting the number of clusters to five.

The clusters that emerged offered a fresh perspective on how various film genres are categorized according to viewer interaction and content attributes. For example, a cluster may feature popular, long-running blockbusters, while another cluster may primarily consist of short, highly regarded independent films. By spotting trends in user-rated preferences, this segmentation proved essential for reducing the number of pertinent suggestions from huge datasets. The likelihood of recommendation relevance and user satisfaction was increased when a user expressed interest in a certain movie. The algorithm would then use the cluster to which the movie belonged and propose further films from the same cluster.



(Fig. 4. Clustering based on runtimes and ratings)

IV. Experimental Design

The Jupyter Notebook, a Python-based development environment chosen for its adaptability and visualization features, was used to conduct the trial setup for this movie recommendation system. A Windows 10 computer with an Intel Core i5 processor and 8GB of RAM was used to carry out the project. Important libraries like Pandas and NumPy were utilized to handle the data, and Scikit-learn made preprocessing and clustering easier. The FP-Growth method for association rule mining was implemented using MLxtend, and Matplotlib was utilized for simple visualizations during the experiment.

The experiment's methodology was well-organized, beginning with the collection of the IMDB movie dataset from Kaggle. The next step was comprehensive data preprocessing, which involved transforming textual data into useful numerical representations, addressing missing values, and standardizing formats. By converting categorical variables and generating new attributes, feature engineering was used to enhance the dataset. Following preparation, the data underwent two primary processing steps: MiniBatch K-Means clustering and FP-Growth-based association rule mining. While clustering grouped comparable films according to runtime, vote count, and rating criteria, the FP-Growth algorithm was selected for its computational speed in discovering common itemsets, particularly with genre-based suggestions.

In the last stage, the system's performance was assessed using RMSE to see how accurate the projected ratings were. Both content-based and collaborative filtering features were made possible by the hybrid association rules and clustering combination, which provided a more comprehensive recommendation system. The need for efficiency, scalability, and significant pattern extraction influenced the design choices. A simplified yet powerful framework that can produce varied and pertinent movie recommendations based on user preferences was made possible by this experimental approach.

V. Results & Discussions

A) Data Description: This project's dataset, which is based on IMDB movie metadata, was obtained via Kaggle. Movie names, genres, vote totals, average ratings, runtime, and other metadata are just a few of the many characteristics it contains. Following preprocessing, only the most pertinent features needed for association rule mining and clustering were added to the dataset. Depending on their usefulness, non-numeric fields were either encoded or eliminated. A balanced mix of numerical and categorical variables made up the final dataset utilized for modeling, making it appropriate for both clustering and FP-Growth methods.

B) Execution and Inferences: The FP-Growth algorithm was applied to the "genres" attribute to generate frequent itemsets and derive association rules. These rules revealed interesting genre pairings, such as frequent co-occurrence of Action and Adventure or Comedy and Romance, which helped in suggesting movies with similar genre combinations to users. This content-based filtering technique allowed recommendations to align closely with user interests inferred from selected genres.

Simultaneously, MiniBatch K-Means clustering was applied using features such as runtime, vote average, and vote count. The movies were grouped into several clusters, each representing a unique combination of viewer popularity and quality. For instance, one cluster contained high-rated movies with moderate vote counts, while another grouped long-runtime action films with high viewership. These clusters enabled recommendations based on behavioral similarity, adding a collaborative filtering aspect to the system.

The hybrid approach of combining FP-Growth and clustering, proved effective in enhancing the recommendation precision. By incorporating both genre patterns and viewer behavior, the system achieved more diverse and user-aligned suggestions. In practical testing, the model showed that the majority of recommendations matched the user's genre preferences and viewing tendencies. This validates the proposed methodology as a promising direction for building scalable and efficient movie recommendation systems.

C) Predictive Modeling: Predictive modeling is used in the project to estimate movie ratings and assess the model's effectiveness. MultiLabelBinarizer is used to one-hot encode the genres column into binary features using a Linear Regression model. This results in a feature matrix with each genre represented as a distinct input. Ratings are used as the target variable (y) and this matrix (X) as features for training the model. Metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which gauge how closely the projected ratings match the actual ratings, are used to measure rating prediction accuracy. The regression coefficients from the model might be examined to determine how each genre affects the anticipated ratings, even if the code does not specifically calculate feature importance. In addition to demonstrating how effectively the genres account for differences in movie ratings, these stages give a gauge of the model's effectiveness and offer insightful information for improving the recommendation system.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

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

VI. ADVANCED DATA MINING TECHNIQUES

A)Text Mining: To analyse movie genres and produce content-based suggestions, the project uses text mining algorithms. The genres_cleaned column, where genres are preprocessed by eliminating special characters and transforming them into lowercase strings, is subjected to the TF-IDF vectorisation algorithm. Through the use of Term Frequency-Inverse Document Frequency (TF-IDF), which successfully quantifies the relative relevance of each genre within the dataset, this technique produces a numerical representation of genres. By using the TF-IDF matrix to determine movie similarities based on genre descriptors, semantic analysis is accomplished. The Nearest Neighbours method is used to determine which movies have the highest cosine similarity to a particular movie in order to make content-based recommendations. This makes it possible for the engine to propose films with comparable genre characteristics, such action-packed flicks for "Cleopatra" aficionados. By matching recommendations with user preferences based on genre content, these text mining approaches improve the recommendation system.

B) Recommended algorithm: For content-based suggestions, the project uses a recommendation algorithm that is mostly based on closest neighbour approaches. To find the most comparable titles, the project calculates cosine similarity between films using the TFIDF vectorisation of the genres_cleaned column. The Nearest Neighbours method is used to identify and suggest films with a user-specified title that have comparable genre characteristics, as shown in the suggestions for "Cleopatra." Although the code emphasises content-based techniques, collaborative filtering which would leverage user interactions or preferences to suggest movies is not explicitly implemented. Furthermore, while not investigated in the current implementation, hybrid recommendation strategies which mix content-based and collaborative filtering approaches for more personalised results could be incorporated to improve system performance. This method successfully matches genre-based preferences with movie suggestions, serving as the cornerstone of a strong recommendation system.

VII.PERFORMANCE EVALUATION

The project evaluates the performance of its model using a variety of metrics. Precision and recall are used to assess the accuracy of genre-based recommendations by examining the overlap between proposed and actual genres. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which quantify the difference between expected and actual ratings, are used to gauge how accurate movie rating prediction is. Cross-validation techniques may be used to confirm model performance across different data subsets, ensuring resilience and preventing overfitting, even though the approach does not explicitly use them. These metrics provide a comprehensive evaluation of the recommendation system's effectiveness.

VIII.FUTURE ENHANCEMENT

While our current system does a good job of recommending movies using clustering and association rule mining, there’s still plenty of room to make it even better. One exciting direction could be bringing in deep learning techniques like neural collaborative filtering or autoencoders, which can understand more complex patterns between users and movies. Another idea is to make the recommendations adapt in real time by tracking how users interact with the system—things like what they click on, how long they watch, or what they search for. We could also expand beyond just movies by including TV shows, documentaries, and other types of content. On top of that, analyzing reviews or social media reactions through sentiment analysis could help the system suggest movies that people are genuinely enjoying, not just those that match certain tags or genres. These enhancements could make the recommendation experience much smarter and more personal.

IX. CONCLUSION

In a nut-shell, In this study, we developed an intelligent movie recommendation system by combining association rule mining with clustering techniques to better capture the nuances of user preferences and behavior. By leveraging the FP-Growth algorithm, we were able to efficiently uncover frequent itemsets and generate meaningful association rules from the user-movie interaction data. This approach provided a strong foundation for understanding co-occurrence patterns in user choices, enabling the recommendation engine to suggest movies that aligned closely with users’ interests. Additionally, MiniBatch K-Means clustering played a crucial role in segmenting users into distinct preference-based groups, which helped in tailoring recommendations to specific user profiles. This hybrid methodology allowed us to balance computational efficiency with recommendation quality, ensuring that the system remained both scalable and personalized.

The effectiveness of the system was measured using standard evaluation metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), which indicated consistent performance in predicting user preferences. The combination of descriptive and predictive data mining techniques enabled a comprehensive understanding of the dataset while enhancing the practical utility of the system. Our findings demonstrate that incorporating multiple data mining strategies can significantly improve the performance of recommendation engines, especially in complex domains like movie suggestion platforms where user interests are diverse and dynamic. This work thus contributes a well-rounded framework that can serve as a strong baseline for future developments in personalized content delivery systems.

X. References

[1] C. C. Aggarwal, Recommender Systems: The Textbook, Springer, 2016, pp. 1–46.

[2] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges,” ACM Computing Surveys (CSUR), vol. 47, no. 1, pp. 1–45, 2014.

[3] S. P. Chatzis, D. C. Anastasiou, and A. C. Sarigiannidis, “Movie recommendation via topic modeling and collaborative filtering,” in Proc. IEEE Int. Conf. on Machine Learning and Applications (ICMLA), 2010, pp. 143–148.

[4] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, “Solving the apparent diversity-accuracy dilemma of recommender systems,” Proc. of the National Academy of Sciences, vol. 107, no. 10, pp. 4511–4515, 2010.

[5] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[6] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 43–52.

[7] A. Z. Zubiaga, "Effective content-based recommendation system using FP-Growth algorithm," *Procedia Computer Science*, vol. 170, pp. 22–27, 2020.

[8] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.