Jurnal Kejuruteraan 37(3) 2025: 1593-1605
https://doi.org/10.17576/jkukm-2025-37(3)-40

# Optimized Movie Recommendation via Sentiment Analysis & Hyperparameter Tuning

Manisha Valera[a*] & Rahul Mehta[b]

[a]*Research Scholar, Gujarat Technological University, Ahmedabad*

[b]*Assistant Professor, Department of Electronics & Communication Engineering, GEC Rajkot, Gujarat, India*

[*]*Corresponding author: rdmehta@hotmail.com*

*Received 20 September 2024, Received in revised form 16 November 2024*
*Accepted 1 February 2025, Available online 30 May 2025*

ABSTRACT

*For improving movie recommendation schemes, this study is targeting to overcome dilemmas like data sparsity and cold start issues, which can limit the relevance and accuracy of recommendations. Progressive tactics are introduced by merging Count Vectorization, Cosine Similarity, Truncated SVD, Linear SVC for sentiment analysis, and Linear Regression for rating prediction, whose objective is to tackle the complications faced by conventional movie recommendation systems, finally gaining superior accuracy as well as finesse in both recommendation and sentiment analysis tasks. In this paper, it explores the challenges of hyperparameter tuning in machine learning, particularly the shortfalls of traditional methods -Grid Search and Randomized Search. Here it inspects progressive techniques as well, such as Bayesian optimization and Optuna, which improve model performance by optimizing hyperparameters more proficiently and reducing computational costs. The proposed methodology governs remarkable effectiveness in overcoming data sparsity and cold start anomalies, plus achieving superior accuracy in recommendations and finesse in sentiment analysis. by achieving an accuracy of 99.87%. scores, The Optuna-optimized LinearSVC model reveals outstanding sentiment classification abilities. at macro and micro levels, this performance is highlighted by its flawless precision, recall, and F1-score, which highlights its accuracy and balanced classification across all classes. Same way, showcasing robust regression performance, the Optuna-optimized LinearSVR model attained a Mean Absolute Error (MAE) of 0.596 and a Mean Squared Error (MSE) of 0.7245. These metrics indicate that the model provides accurate continuous predictions with minimal error. This study makes a noteworthy growth in movie recommendation systems, promising notable evolutions in accuracy and sophistication, flagging the way for future developments in the field.*

*Keywords: Movie recommendation; sentiment analysis; singular value decomposition; hyperparameter tuning; optuna optimization; LinearSVC; LinearSVR*

## INTRODUCTION

The ability of Machine Learning (ML) to unconventionally learn data for the Recommender systems, essential across e-commerce, entertainment, and social media, harness leading-edge algorithms like collaborative and content-based filtering, alongside emergent approaches such as hybrid models and deep learning. Context-aware systems further adapt recommendations based on user context. Matrix factorization and reinforcement learning enhance recommendation accuracy, while knowledge-based systems suit to spheres with limited data. Netflix and Amazon are the platforms which employ these techniques, transform the user experiences through tailor-made suggestions. However, challenges persevere, conspicuously the "cold start" event in movie recommendation systems, The relentless pursuit of more dynamic and precise algorithms fuels ongoing research, driving novelty across numerous fields. Though, machine learning plus deep learning models is vital to a wide range of industries in current data-centric world, optimizing these models dregs a challenge due to the need for careful tuning of hyperparameters which are learning rates, regularization factors, and neural network's number of layers. These hyperparameters play a decisive role in defining the model's ability to perform well, and incorrect tuning can produce poor generalization or inept model performance, impeding their effectiveness.

Here, this study reports significant challenges which are existing in movie recommendation systems, particularly data sparsity and the cold start problem, these problems hinder the relevance and adaptability of recommendations. While traditional recommendation methods—such as collaborative filtering and basic content-based approaches—which often struggle with these kinds of issues, where user behavior is complex and rapidly evolving. they also clash with the limitations when applied to large-scale, dynamic real-world systems. for advancing the state of the art, a framework is proposed which uses combination of Count Vectorization and Cosine Similarity for boosted content-based filtering, Truncated SVD- which is used for dimensionality reduction, Linear SVC which is used here for robust sentiment analysis, and Linear Regression for rating prediction. This integrated approach is aimed to achieve advanced accuracy and refinement in recommendation precision and sentiment analysis compared to conventional systems. However, further research will be necessary to fully validate its performance and adaptability in large-scale, real-world settings with highly dynamic user interactions.

As data complexity plus demand for accuracy remain to grow, traditional methods for hyperparameter tuning, which are Grid Search and Randomized Search, competing to meet the need for rapid and scalable solutions. Their incompetence in management of high-dimensional hyperparameter spaces, which is making it difficult to organize machine learning models in time-sensitive circumstances, such as in finance, healthcare, or real-time analytics. To address these kind of restrictions, more sophisticated hyperparameter optimization methods have arisen, which includes Bayesian optimization, Optuna. These types of approaches allow for critical exploration of hyperparameter spaces, which leads to more precise models with less computational effort. This paper digs into these cutting-edge techniques and it also examines how they help to solve the challenges of hyperparameter tuning across various machine learning applications. The algorithm "Novel Hybrid Movie Recommendation System" (Bahl et al. 2023) operates mixture of techniques, which recommend movies to public based on their scores plus movie features. Firstly, it imports the movies and ratings datasets, then initializes a TF-IDF vectorizer to process movie descriptions followed by calculating cosine similarity scores between movies. Afterward, it employs Singular Value Decomposition algorithm which factorizes the ratings matrix and get latent features. then a function "recommendation," which takes a user ID and a movie as input. Using this function, it retrieves movies similar to the input movie based on cosine similarity scores, then calculates the average rating for these similar movies from the ratings dataset, and predicts the input movie's rating for the given user using the trained SVD model too. The algorithm returns a list of recommended movies as an output. Overall, this combines content-based and collaborative filtering tactics for delivering personalized movie commendations. The different approach is presented (Jain and Sarkar 2023) for the movie recommendation systems, targeting to offer personalized proposals for preferred movies of individual users which is not trusting wholly on popularity /ratings. here the proposed hybrid scheme syndicates weighted middling and min-max scaler approaches for movie rating & popularity estimation. By utilizing TF/IDF for vectorization, here the system computes cosine similarity to regulate the similarity between movie vectors. The results include the outstanding 5 commendations for users which also includes forecast rates for specific movies. the paper also acknowledges ongoing research in constructing recommender systems via deep learning or neural networks, for acquisition of high-performance accuracy. The paper (Yogesh and Kumar 2024) presents a systematic approach for building a movie recommendation system. It starts with data collection and preprocessing from the MovieLens dataset, uses count vectorizer and tf-idf vectorizer for feature transformation, reports challenges like missing data, and engages content-based and collaborative filtering techniques. The system is deployed on Heroku, integrating Streamlit and Pickle for enhanced user interface, producing an effective recommendation system aimed at ameliorating user experience. The hybrid engine algorithm (Rajarajeswari et al. 2019) initiates by identifying the movieID corresponding to the given movie title, computes similarity scores with other movies using cosine similarity based on attributes such as 'vote_count', 'year', and 'vote_average', selects 30 similar movies, estimates ratings for these selected movies using SVD, and compares them with actual ratings to calculate RMSE. It then sorts recommended movies grounded on RMSE and returns the top 10 movies. This approach combines content-based and collaborative filtering, leveraging movie features and user preferences for personalized recommendations. The proposed collaborative filtering algorithm (Bhalse and Thakur 2021) employs SVD plus cosine similarity methods. It forecasts an outstanding-n list of recommended products for the active user by building a matrix of user-item utility from rating using movie data, normalizing it, and reducing dimensionality through SVD. user similarities are calculated by Cosine similarity, enabling the choice of nearest neighbors or setting similarity thresholds. Predicted ratings and recommendations are generated on the basis of the nearest neighbor set. Future investigation could involve arbitrating the algorithm using varied metrics. In Scenario 2nd (Nguyen et al. 2020), the emphasis shifts towards utilizing actors and actresses as tags for the recommender system. However, a significant barrier arises due to the absenteeism of actor/actress information within the MovieLens 100K dataset. Thus, observations lacking this vital data are rejected from the analysis to maintain data integrity. The process of curation leads to a decrease in the quantity of training samples accessible for the recommender system, as opposed to Scenario 1, which only measured genre information. With fewer data points to absorb from, the model's ability to accurately predict user preferences may be somewhat negotiated, leading to slightly worse RMSE scores. Moreover, in this the exclusion of observations excluding actor/actress data is impacting computational competence, which slightly slowing down the running times compared to Scenario 1. Despite these kinds of challenges, the incorporation of actor/actress tags still proves very profitable to the system's performance, which is showcasing its adaptability as well as effectiveness in leveraging diverse metadata for movie recommendations. The study encompasses (Pavitha et al. 2022) a Movie Recommendation System plus Sentiment Analysis. For recommendation purpose, the Cosine Similarity algorithm suggests related movies based on factors like genre, overview, cast, and ratings, indicating accurate in severe testing. Here the Sentiment Analysis categorizes reviews as either positive or negative by applying Naïve Bayes (NB) and Support Vector Machine (SVM) algorithms, where SVM is demonstrating slightly superior accuracy compared to NB. Here Restrictions comprise the system's precision dependency which relies on user input corresponding the dataset and linguistic hurdles in sentiment analysis. Nonetheless, the study accentuates the significance of these systems in catering to user preferences and considerate sentiment nuances in reviews.

The paper (Sahu et al. 2022) presents a multi-module approach for forecasting and recommending forthcoming movies on the ground of user favorites. This Approach employs YouTube promotion reviews and significant traits which are used for forecasting ratings, It also produces a selection of similar new movies grounded on user favorites, and combines forecasted rankings with the user's preferred list. Two datasets are employed: one containing upcoming movie information with trailer reviews and intrinsic features, and another from TMDb with movie metadata and user rating data. Future research suggestions comprise exploring assorted social media platforms for data gathering and mingling cross-lingual comments and emojis for sentiment analysis. The paper (Rathore et al. 2023) proposes the development of a movie recommender system leveraging two primary databases: one encompassing movie data up to 2017 sourced from Kaggle and another containing movies from 2018 to 2020 sourced from Wikipedia. Using the TMDB API, detailed movie information is gathered, while web scraping via BeautifulSoup4 is intended to retrieve customer reviews from IMDB for sentiment analysis. The system integrates collaborative and content-based filtering techniques, employing technologies like Naive Bayes for opinion analysis and NLP for sentiment analysis using NLTK and TFIDF vectorizer. Similarity scores, determined through cosine similarity, aid in assessing item similarity. AJAX requests facilitate efficient client-server communication (Chadokar et al. 2023). Overall, the proposed system offers personalized movie recommendations by analyzing user preferences and sentiments expressed in customer reviews. It surveys variety of hyperparameter optimization methods, which includes tactics like Grid Search, Random Search, and Optuna. It lists their advantages, restrictions, and applications across different domains. It presents a taxonomy for these practices and discusses recent advancements (Yadav et al. 2023). This study (Li et al. 2023) provides dissimilarities of Bayesian optimization methods with the other traditional techniques like Grid Search and Random Search. It also assesses the performance of these methods across different datasets and models, which is highlighting the efficacy and benefits of Bayesian approaches. It surveys the benchmark of hyperparameter tuning approaches for deep learning models, which includes Grid Search, Random Search, and Optuna too.in addition, it also offers practical recommendations and insights into optimizing deep learning models effectively (Smith and Moore 2022). This experimental study investigates adaptive hyperparameter optimization techniques which is including Optuna, for neural networks. It also presents comparative results by showing the effectiveness of adaptive methods over traditional approaches which is used for improving model performance (Lee and Kim 2023). It explores metaheuristic algorithms which are used for tuning hyperparameters of Support Vector Machines, It compares these methods with Grid Search and Random Search. It also demonstrates the efficiency and effectiveness of metaheuristic approaches which are used in optimizing SVM parameters (Zhang et al. 2022). They (Sharma et al. 2024) proposed an enhanced movie recommendation system using Self-Organizing Maps (SOMs) with adaptive learning rates and dynamic neighborhood functions which is used to improve user satisfaction. This approach tailors recommendations by adjusting model parameters, which results in more personalized and relevant suggestions. This paper (Roy et al.2022) systematically reviews major practices in recommender systems, collaborative/content-based filtering, while addressing challenges like data sparsity and scalability. In this paper It also examines emerging trends, including deep learning and context-aware recommendations, and it suggests future research areas for enhancing personalization and accuracy.

TABLE 1. Hyperparameter Tuning Methods

| Method | Definition | Key Features |
|---|---|---|
| Optuna | An open-source framework for hyperparameter optimization using practices like Bayesian optimization and TPE. | Adaptive search, pruning, user-friendly interface. Emphasized for its adaptive and efficient optimization capabilities for neural networks. |
| RandomizedSearchCV | A method which samples a static number of hyperparameter combinations randomly from a distribution. | Efficiency, flexibility, parallelization. Compared as a practical and faster alternative to Grid Search in high-dimensional spaces. |
| GridSearchCV | A comprehensive search over a grid of hyperparameter values, evaluating entirely possible combinations. | Comprehensive, simple implementation, can be computationally expensive. Highlighted for its thorough exploration but noted as less efficient compared to Bayesian methods. |

The novelty of this study lies in its state-of-the-art hybrid approach, this research study brings innovation by incorporating Count Vectorization, Cosine Similarity, Truncated SVD, Linear SVC, and Linear Regression to overcome the sparsity and cold start glitches towards an improvement in the recommendation accuracy and sentiment analysis. Use of Optuna for hyperparameter tuning progressed model performance and efficiency in terms of the accuracy of sentiment classification.

# METHODOLOGY

## COSINE SIMILARITY

This tactic estimates the cosine of the angle between 2 vectors, which is representing how analogous they are in a multi-dimensional space. It is represented with a range of -1 to 1, where 1 implies perfect similarity, -1 perfect dissimilarity, and 0 orthogonality, it's calculated as below:

$$\textit{cosine similarity}(A, B) = A \cdot B \ / \parallel A \parallel \times \parallel B \parallel \quad (1)$$

## COUNT VECTORIZATION

Our database search results cover a number of irrelevant data.in this, It transforms text documents into numerical vectors by counting the occurrences of each word. This method creates a sparse matrix where each row is allied to a document, and each column is linked to a exclusive word, with the cell values representing the word frequencies. It's commonly used in NLP tasks for feature extraction and text analysis. Mathematically, for a document $d$ containing $n$ unique words, the count vectorization process can be represented as:

$$Count\_Vector(d)=[Count(w1,d),Count(w2,d),...,Count(wn,d)] \quad (2)$$

In this context, Count($w$,$d$) signifies the occurrence count of the word $w$ in document $d$, while $n$ denotes the overall number of diverse words in the complete corpus.

## TRUNCATED SVD

Collaborative Filtering (CF) is crucial to recommendation systems, with the approaches like SVD and neighborhood models, it is being prominent. SVD, it is a key dimensionality reduction technique, which decomposes matrices into U, Σ, and V, revealing latent factors that enhance model performance. Truncated SVD, It is a variant allowing personalized output columns, which proficiently handles sparse matrices, which is used for improving feature extraction. This technique, along with regularization and bias terms, It enhances recommendation accuracy by diminishing error as well as capturing nuanced user-item interactions.

in the realm of matrix analysis, the Singular Value Decomposition – SVD method exposes a fundamental formula which is denoted as $M=U\Sigma V^{\top}$

In this equation:

$M$ means the original matrix, which is targeted for decomposition.

$U$ means left singular matrix, where each column comprises the left singular vectors. These vectors are considered by being the eigenvectors extracted from the matrix $MM^{\top}$

Σ denotes a diagonal matrix housing the singular values, which are identical to eigenvalues.

$V$ signifies the right singular matrix, by embodying columns that encapsulate the right singular vectors. Here These vectors are resultant from the eigenvalues of the matrix $M^{\top}M$

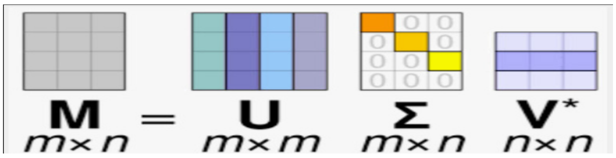


FIGURE 1. SVD matrices

## LINEAR SUPPORT VECTOR REGRESSION

LinearSVR is an SVM variant custom-made for regression tasks, predicting continuous target variables. Implemented with liblinear, it offers flexibility in parameter selection and loss functions. Like LinearSVC, it's highly scalable and seeks the optimal hyperplane to minimize regression errors. Overall, LinearSVR is a potent tool for efficient and effective regression modeling.

### LINEAR SUPPORT VECTOR CLASSIFICATION

LinearSVC is a machine learning algorithm made-to-order for binary classification tasks where data can be separated by a linear boundary. Notably, LinearSVC is highly scalable, improved for efficient handling of large datasets. It operates by recognizing A hyperplane in space with multiple dimensions to maximize class margin and minimize classification errors.

Using Linear SVR and Linear SVC is beneficial because of their effectiveness in high-dimensional data, robustness to outliers, and computational efficiency. It is making them suitable for tasks like sentiment analysis and rating prediction in recommendation systems. Moreover, they provide interpretable linear decision boundaries and allow for hyperparameter tuning, enhancing overall model performance.

## EXPERIMENTAL RESULTS AND DISCUSSION

### DATA COMPILATION AND PREPROCESSING

The recommender system is under development utilizing two primary datasets: one comprising the top 5000 movies up to 2017, and another containing metadata for movies, both obtained from Kaggle. Besides, there's a Wikipedia database listing movies from 2018 to 2020. To enhance movie details, the system leverages TMDB's API , extracting information like titles, genres, ratings, banners, and more (Sinha and Sharma 2023).

This Python script uses the TMDb API to fetch user scores for movies listed in a CSV file called 'main_data.csv'. It initializes the TMDb API with an API key, loads the movie titles from the CSV, and retrieves user scores using the get_user_score function. The script then removes rows (Mechetin 2021) with missing user scores (Zin et al. 2017) and saves the updated data to a new CSV file named 'main_data_with_ratings.csv'. Thus, Preprocessing steps makes suitable data for the further processing. (Başarslan and Kayaalp 2023 ; Danyal et al. 2024).

This script uses the TMDb API , which fetches reviews for movies listed in a CSV file called 'main_data_with_ratings.csv'. It initializes the TMDb API, loads the movie data, and iterates through each movie to retrieve its reviews using the get_movie_reviews function. The script then merges the review data with the original movie data and saves the merged data to a new CSV file named 'main_data_with_reviews_tmdb.csv'.

This script uses TextBlob here for performing sentiment analysis (Gupta et al. 2024) on movie reviews in a CSV file named 'main_data_with_reviews.csv'. Then It loads the data, ensures all reviews are of string type and handles missing values as well (Parlar et al. 2019), followed by classifying each review as 'Good', 'Neutral', or 'Bad' based on sentiment polarity (Bodapati et al. 2019). The sentiment analysis results are added as a new column ('Review_Sentiment') in the DataFrame, which is then saved to a new CSV file of 13.6 MB named 'main_data_with_sentiment_tmdb.csv'.

### PROPOSED RECOMMENDATION MODEL ENHANCED BY SENTIMENT ANALYSIS & HYPERPARAMETER TUNING

This code extract is an inclusive implementation of sentiment analysis and predicting movie rating using machine learning methodology. The research emphasizes varying dislikes to certain media elements in ethical recommendations. We propose a user-driven ethical recommendation system post-recommendation to aid in selecting morally suitable items. It's posited that user perceptions of item ethics can gauge system trust and credibility. It starts by loading data from a CSV file and prep prediction processing it, including handling absent values and combining relevant columns. Sentiment analysis is conducted using TextBlob, extracting sentiment scores and labels for each review. The reviews are then united with other features, including user scores, and processed through Count Vectorizer for numerical representation. Singular Value Decomposition (SVD) is utilized for dimensionality reduction, followed by the partition of data into training plus testing sets. LinearSVC and LinearSVR models are trained and optimized using hyperparameter tuning tactics like GridSearchCV, RandomizedSearchCV, and Optuna and visualizes ROC curve metrics for the LinearSVC model and MSE distribution for the LinearSVR model across different tuning methods. Optuna provides the highest accuracy for LinearSVC or the lowest MSE for LinearSVR, it directs that Optuna's hyperparameter optimization is more effective in improving the model's performance compared to the further methods.

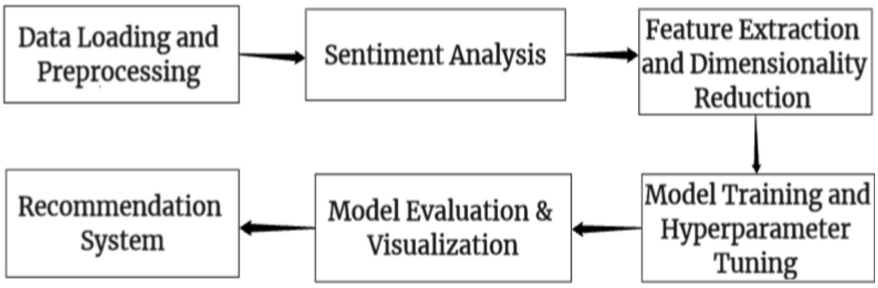


FIGURE 2. Process Flow of Movie Recommendation and Sentiment Analysis & Hyperparameter Tuning

In Table 2. Optuna proves superior performance in hyperparameter tuning compared to rest of tactics. For LinearSVC, Optuna's best C value of 979.12 results in a remarkable accuracy of 99.87%, with perfect macro and micro averaged precision, recall, and F1 scores. For LinearSVR, Optuna achieves an optimal C value of 0.003, which is consequent in a mean squared error (MSE) of 0.72 and a mean absolute error (MAE) of 0.60. This highlights Optuna's effectiveness in optimizing model performance beyond traditional methods as given in FIGURE 3. Additionally, it provides functionality to get top movie recommendations based on similarity scores and predicted ratings using the trained Linear SVR model, including actual and predicted ratings for comparison. The code ends with visualization of the actual vs. predicted ratings using color gradients to interpret prediction accuracy visually.

TABLE 2. The results for Linear SVC and SVR using GridSearchCV, RandomizedSearchCV, and Optuna

| Model | Hyperparameter Tuning Method | Best Hyper-parameters (C) | Accuracy (%) | MSE | ROC AUC Score |
|---|---|---|---|---|---|
| Linear SVC | GridSearchCV | 10 | 99.27 | - | - |
| Linear SVC | RandomizedSearchCV | 9.607 | 99.29 | - | - |
| Linear SVC | Optuna | 27.269 | 99.56 | - | - |
| Linear SVR | GridSearchCV | 0.1 | - | 0.7148 | - |
| Linear SVR | RandomizedSearchCV | 1.660 | - | 0.8534 | - |
| Linear SVR | Optuna | 704.506 | - | -7.6190 | 0.9996 |

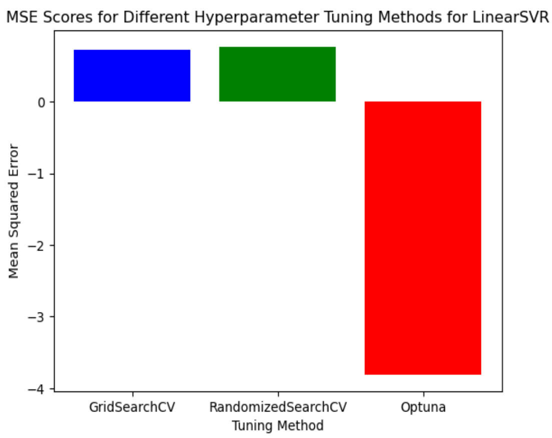


FIGURE 3. MSE score for hyperparameter tuning methods for LinearSVR

These results point to the performance of two different linear models: Linear-Support-Vector-Classification (Linear SVC) and Linear-Support-Vector-Regression (LinearSVR). In short, the LinearSVC model achieved high accuracy and performed well across precision, recall, and F1 score, making it apposite for classification tasks, such as recommendation systems. Whereas the LinearSVR model demonstrated its effectiveness in regression tasks, with low MAE and MSE, indicating its capability to predict continuous target ratings accurately.

For the classifiers' performance evaluation, here several evaluation metrics are generally used: F1 score, accuracy, recall, precision. These are essential because of widespread application of them. Presenting the confusion matrix is vital for establishing the mathematical formulations for these metrics. The confusion matrix given in Figure 4, can be understood as given in error matrix which shows 4 quantities: 1. true positive (TP), 2. false positive (FP), 3. true negative (TN), and 4. false negative (FN). Respective row of the matrix indicates the actual class, whereas each column denotes the predicted class. TP specifies the positive review which is predicted by classifier correctly, & the actual label is also positive. If the review belongs to the negative class, then the review will be considered TN and the actual outcome will be negative too. For a false positive, the review is projected as positive, while the original one is negative. Same way, a review is labelled a false negative if it belongs to the positive class but the classifier predicts it as negative.

The frequently used evaluation measure Accuracy, that specifies the ratio of correct predictions to the over-all predictions. Its best value of 1 for 100% accurate prediction and a least value of 0 for 0% prediction. Accuracy is well-defined hereby given formula:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \tag{1}$$

Precision shows the amount of the absolutely predicted cases which are positive in real. It is defined here by the given formula:

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{2}$$

Recall computes the proportion of correctly predicted positive instances to the total actual positive instances.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{3}$$

The F1 score is measured as vital parameter for estimating the classifier's performance and is frequently observed as more critical than precision as well as recall. It describes the balance between precision and recall by incorporating both measures. The value of F1 score ranges between 0 and 1, with 1 indicating flawless classifier performance. The F1 score is defined here by the given formula:

$$\boldsymbol{F1\ Score} = \mathbf{2} * \frac{\boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision+Recall}} \tag{4}$$

The paper (Sahu et al. 2022) has proposed the model which was giving Mean Absolute Error of 0.7406 and Mean Squared Error of 0.8648 respectively, but here in this Paper the proposed methodology is providing Mean Absolute Error of 0.5960and Mean Squared Error of 0.7245 respectively which shows improvement in predicting the rating.

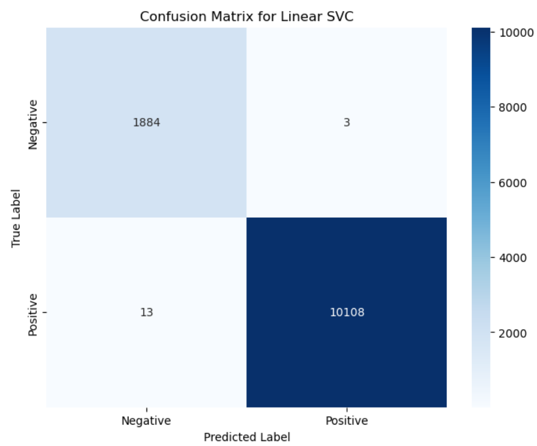


FIGURE 4. Confusion Matrix for Linear SVC

Here is a TABLE3. briefing the results for Linear SVC and SVR using Optuna:

TABLE 3. Hyperparameter Tuning Methods Evaluation Mesures using Hyperparameter tuning method Optuna

| Model | Best Hyperparameters (C) | Accuracy (%) | Macro-Averaged Precision | Macro-Averaged Recall | Macro-Averaged-F1 Score | Micro-Averaged Precision | Micro-Averaged Recall | Micro-Averaged-F1 Score | MAE | MSE |
|---|---|---|---|---|---|---|---|---|---|---|
| Linear SVC | 979.12 | 99.87 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | - | - |
| Linear SVR | 0.00296 | - | - | - | - | - | - | - | 0.5960 | 0.7245 |

Linear SVC (Optuna) attained excellent performance with 99.87% accuracy and perfect macro and micro averaged metrics as given in FIGURE 5.

Linear SVR (Optuna) gave a Mean Absolute Error (MAE) in above table of 0.5960 and Mean Squared Error (MSE) of 0.7245 as given in FIGURE 6.

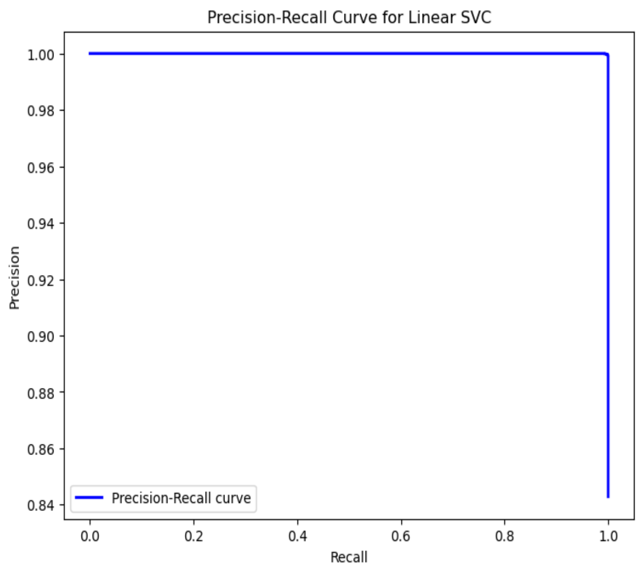


FIGURE 5. Precision Recall Curve for Linear SVC

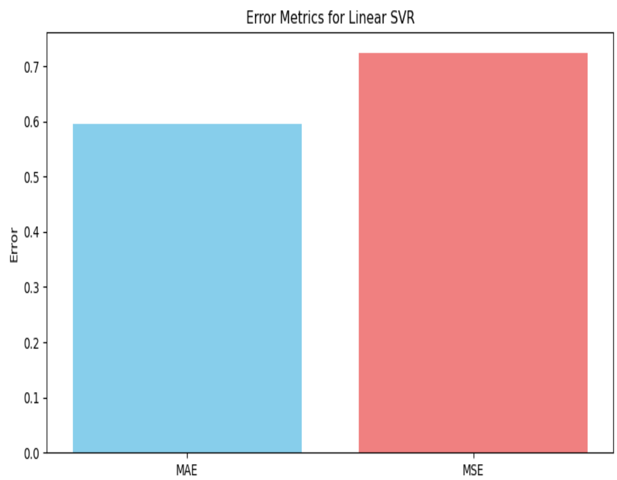


FIGURE 6. Error Metrics for Linear SVR using Optuna

In below FIGURE 7. the scatter plot shows predominantly blue colors, it suggests that the predicted ratings are closer to the actual ratings, indicating a smaller absolute difference between them. This could agree with that the linear SVR model is performing well in predicting ratings that are consistent with the actual values.

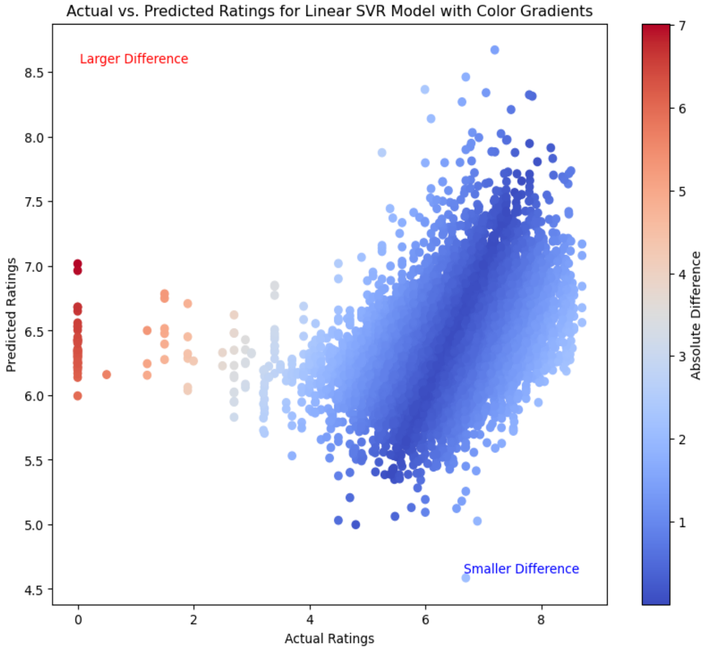


FIGURE 7. Actual Ratings Vs. Predicted Ratings

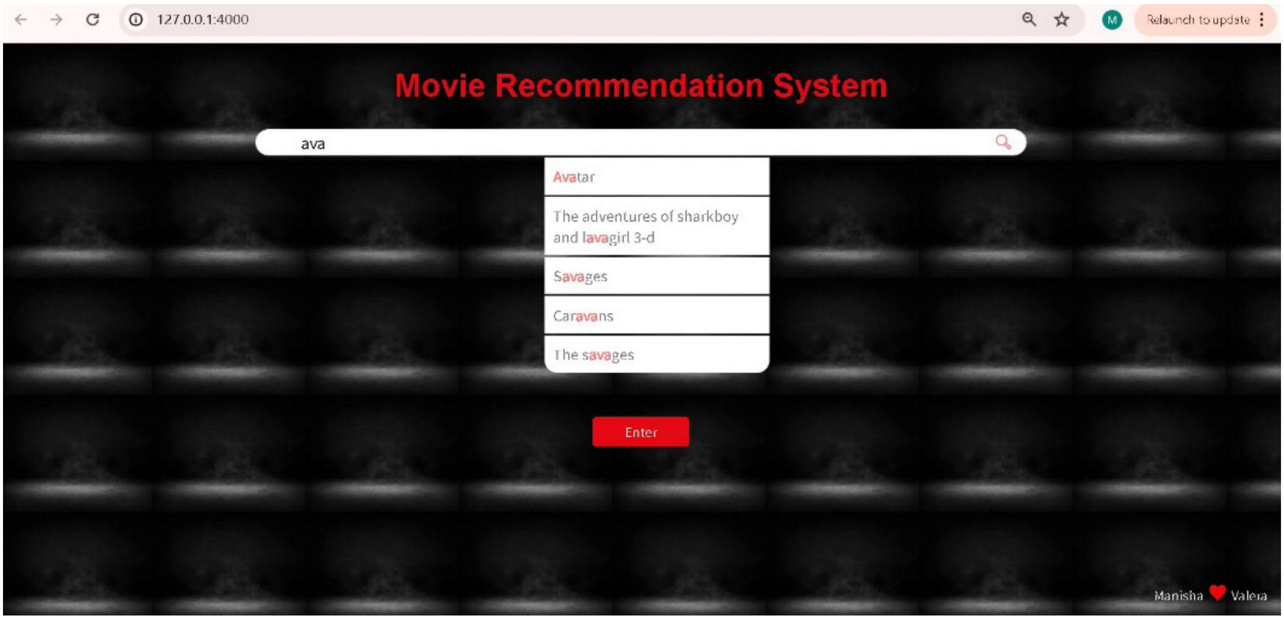


FIGURE 8. Homepage

AJAX, or Asynchronous JavaScript and XML is used for seamless data submission and retrieval, empowering Python data display on the client side through efficient request-response interactions.

In FIGURE 8. The project's homepage integrates a movie data input form, employing AJAX technology for seamless transmission of user-provided details to the backend and dynamic display of processed results.

In FIGURE 9. The search engine displays detailed movie information, including genre, runtime, rating, status, top cast, name, overview, and release year, processed at the backend and presented on the client side.

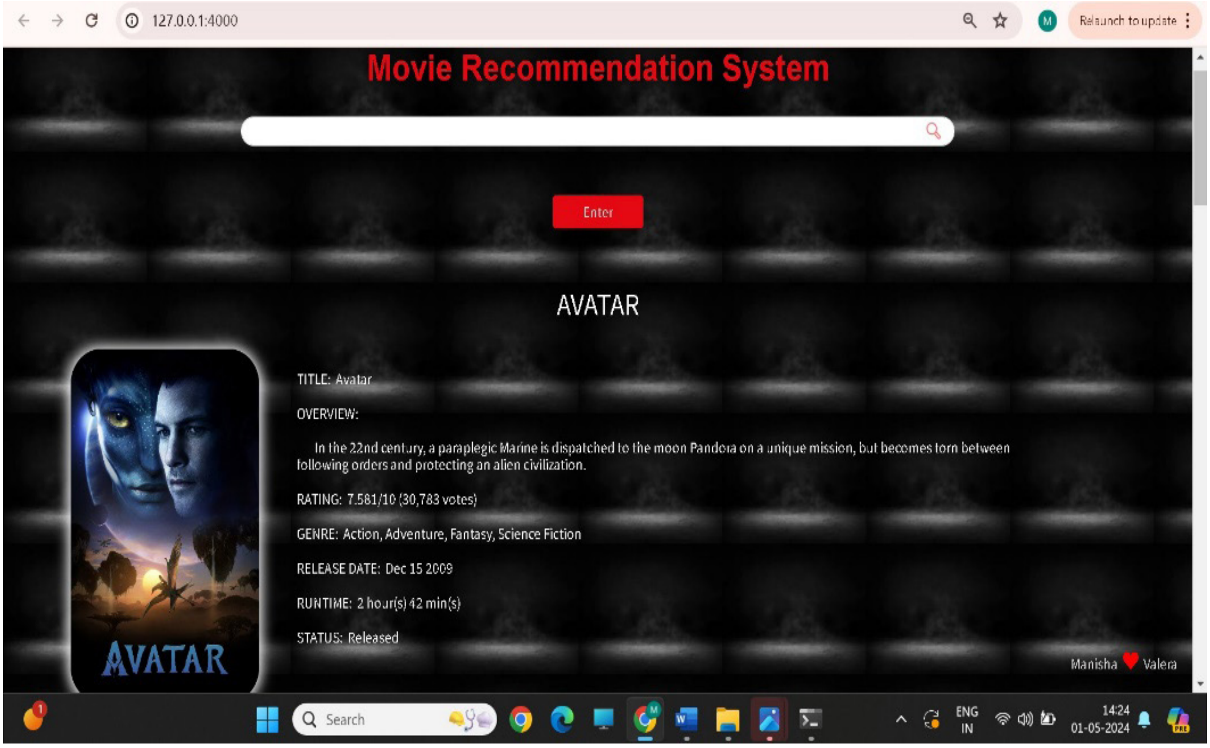


FIGURE 9. Displaying Movie details using Web Scraping

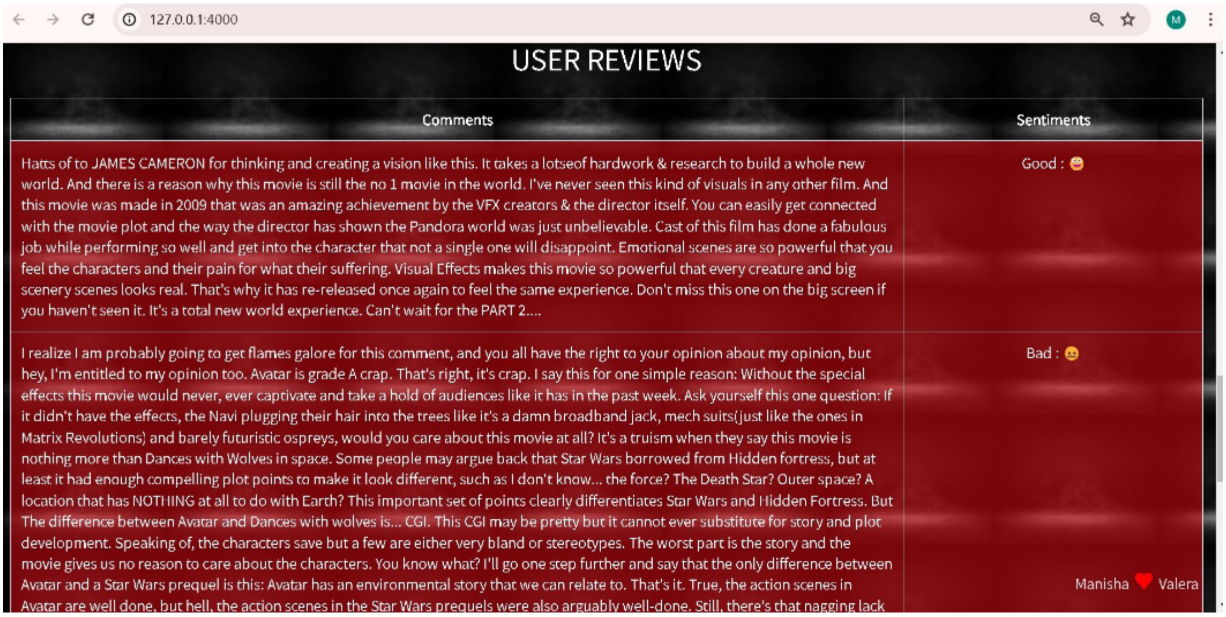


FIGURE 10. Sentiment Analysis based on User Reviews

This project section specializes in sentiment analysis, categorizing user reviews as either positive/negative, or neutral to provide observations into audience views in FIGURE 10.

Utilizing sentiment analysis, the system tailors movie recommendations based on genre, actors, directors, and other factors, showcasing films with the least difference between actual and predicted ratings, determined by Linear SVR in FIGURE 11.

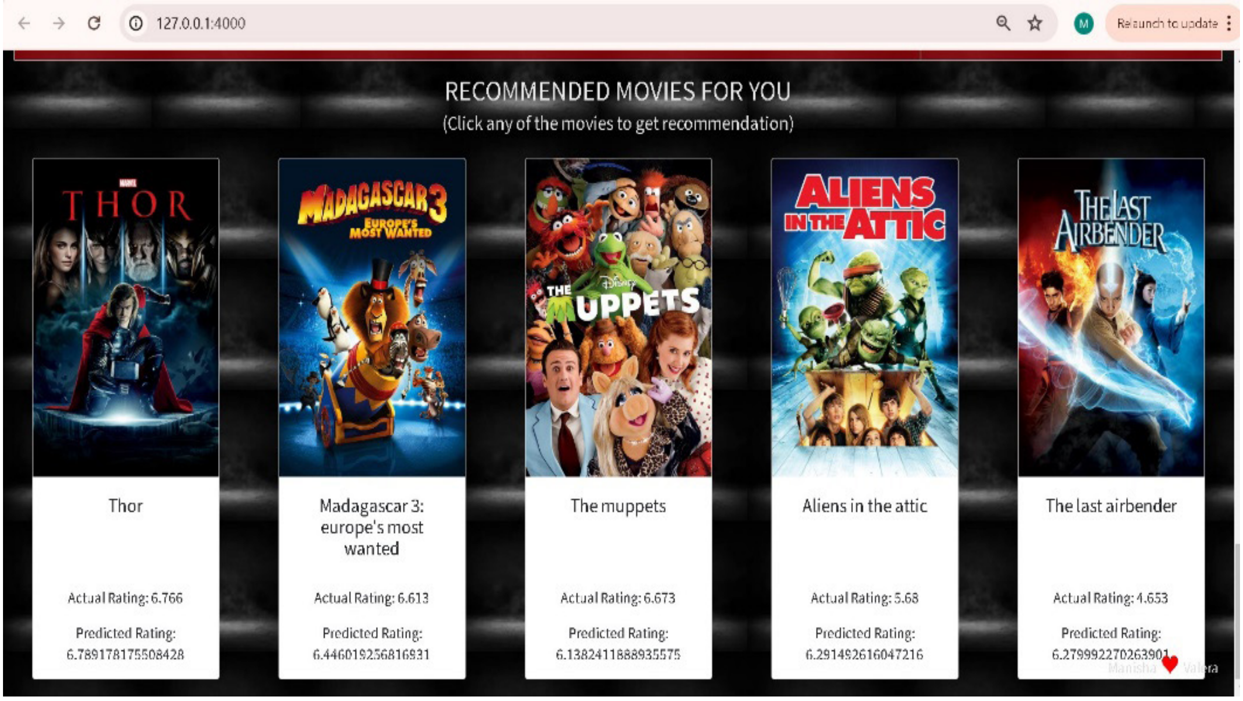


FIGURE 11. Top 5 Recommended Movies with the least difference between actual and predicted ratings using Linear SVR

## CONCLUSION

In summary, this paper launches a well-designed channel for sentiment analysis and movie recommendation, utilizing TextBlob sentiment analysis, Count Vectorizer, and SVD for model optimization. Robust evaluation metrics validate the effectiveness of the approach, allowing personalized movie recommendations based on sentiment analysis and predicted ratings. For LinearSVC(Optuna), achieving an accuracy of 99.87% echoes robust sentiment classification performance by achieving perfect precision, recall, and F1 scores on both macro and micro levels. Linear SVR (Optuna) had a MAE of 0.596 and MSE of 0.725, presenting solid regression performance. Hyperparameter tuning is key for enlightening the performance and generalization of machine learning models. In movie recommendation systems, it boosts accuracy and personalization, making predictions more precise and reliable. Future endeavours could entail refining sentiment analysis methods, incorporating user feedback, and investigating deep learning models to enhance accuracy.

## ACKNOWLEDGEMENT

Not Applicable.

## DECLARATION OF COMPETING INTEREST

None

## DATA AVAILABILITY

IMDb. *IMDb*. [Online]. Available: https://www.imdb.com/

Kaggle. *IMDB 5000 Movie Dataset*. [Online]. Available: https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset

Kaggle. *The Movies Dataset*. [Online]. Available: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset

Wikipedia. *List of American films of 2018*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2018

Wikipedia. *List of American films of 2019*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2019

Wikipedia. *List of American films of 2020*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2020

The Movie Database (TMDB). *The Movie Database.* [Online]. Available: https://www.themoviedb.org/

## REFERENCES


A. Mechetin. 2021. Review the performance of the Bernoulli Naïve Bayes classifier in intrusion detection systems using recursive feature elimination with cross-validated selection of the best number of features. *Procedia Computer Science* 190: 564–570.

Bahl, D., Kain, V., Sharma, A., & Sharma, M. 2020. A novel hybrid approach towards movie recommender systems. *Journal of Statistics and Management Systems* 23(6): 1049–1058.

Bhalse, N., & Thakur, R. 2021. Algorithm for movie recommendation system using collaborative filtering.

Bodapati, J. D., Veeranjaneyulu, N., & Shaik, S. 2019. Sentiment analysis from movie reviews using LSTMs. *Ingenierie des Systemes d'Information* 24(1): 125-129.

Chadokar, S., Jain, N., & Thakre, A. 2023. Movie recommendation engine with sentiment analysis using AJAX request. *IJIRMPS* 11(1).

Chen, V. X., & Tang, T. Y. 2019. Incorporating singular value decomposition in user-based collaborative filtering technique for a movie recommendation system: A comparative study. In *Proceedings of the 2019 International Conference on Pattern Recognition and Artificial Intelligence*: 12–15.

Danyal, M. M., Khan, S. S., Khan, M., Ullah, S., Ghaffar, M. B., & Khan, W. 2024. Sentiment analysis of movie reviews based on NB approaches using TF–IDF and count vectorizer. *Social Network Analysis and Mining* 14(2): 167-175.

Gupta, A., Pandey, S., Behera, M. P., Darshana, S., & Dash, A. 2024. Sentiment analysis of movie review using machine learning. *Educational Administration: Theory and Practice* 30(5): 7344-7354.

IMDb. *IMDb*. [Online]. Available: https://www.imdb.com/

Jain, S., & Sarkar, M. 2023. Efficient hybrid movie recommendation system framework based on a sequential model. *International Journal of Intelligent Systems and Applications in Engineering* 11(9s): 145–155.

Kaggle. *IMDB 5000 Movie Dataset*. [Online]. Available: https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset

Kaggle. *The Movies Dataset*. [Online]. Available: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset

Lee, J. C., & Kim, D. H. 2023. Adaptive hyperparameter optimization for neural networks: An empirical study. *Journal of Computational Intelligence and Neuroscience* 2023(2): 89-105.

Li, L., Liu, Y., & Wang, C. 2023. Hyperparameter optimization using bayesian methods: A comparative study. *IEEE Transactions on Neural Networks and Learning Systems* 34(6): 1205-1217.

M. M. Danyal, S. S. Khan, M. Khan, S. Ullah, F. Mehmood, and I. Ali. 2024. Proposing sentiment analysis model based on BERT and XLNet for movie reviews. *Multimedia Tools and Applications*, 2024.

M. S. Başarslan and F. Kayaalp. 2023. MBi-GRUMCONV: A novel multi bi-GRU and multi CNN-based deep learning model for social media sentiment analysis. *Journal of Cloud Computing* 12(5.

Nguyen, Q. N., Duong-Trung, N., Le Ha, D. N., Ha, X. S., Phan, T. T., Pham, H. X., & Huynh, H. X. 2020. Movie recommender systems made through tag interpolation. In *Proceedings of the 4th International Conference on Machine Learning and Soft Computing* (pp. 154–158).

Parlar, T., Özel, S. A., & Song, F. 2019. Analysis of data pre-processing methods for sentiment analysis of review. *Computer Science* 20(1): 123-141.

Pavitha, N., Pungliya, V., Raut, A., Bhonsle, R., Purohit, A., & Shashidhar, R. 2022. Movie recommendation and sentiment analysis using machine learning. *Global Transitions Proceedings* 3(1): 279–284.

Rajarajeswari, S., Naik, S., Srikant, S., Sai Prakash, M. K., & Uday, P. 2019. Movie recommendation system. In *Emerging Research in Computing, Information, Communication and Applications: ERCICA 2018, Volume 1* (pp. 329–340). Springer, Singapore.

Rathore, S., Narvare, S., & Chadokar, S. 2023. Movie recommendation engine with sentiment analysis: A survey. *IJIRMPS* 11(3).

Roy, Deepjyoti, and Mala Dutta. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data* 9(59).

Sahu, S., Kumar, R., MohdShafi, P., Shafi, J., Kim, S., & Ijaz, M. F. 2022. A hybrid recommendation system of upcoming movies using sentiment analysis of YouTube trailer reviews. *Mathematics* 10(9): 1568.

Sharma, S., Dubey, G. P., & Shakya, H. K. 2024. Optimizing user satisfaction in movie recommendations using variable learning rates and dynamic neighborhood functions in SOMs. *International Journal of Experimental Research and Review* 41(spl.): 130-145.

Sinha, S., & Sharma, T. 2023. Content-based movie recommendation system: An enhanced approach to personalized movie recommendations. *International Journal of Innovative Research in Computer Science and Technology (IJIRCST):* 11(3): 67-71.

Smith, A. G., & Moore, H. L. 2022. Optimizing deep learning hyperparameters: A survey and benchmark. *International Journal of Data Science and Analytics* 10(4): 234-256.

The Movie Database (TMDB). *The Movie Database.* [Online]. Available: https://www.themoviedb.org/

Wikipedia. *List of American films of 2018*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2018

Wikipedia. *List of American films of 2019*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2019

Wikipedia. *List of American films of 2020*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2020

Yadav, R. R., Alotaibi, M. S., & Khan, M. A. 2023. A comprehensive review of hyperparameter optimization: Techniques and applications. *Journal of Machine Learning Research* 24(1): 123-150.

Yogesh, S., & Kumar, S. G. Machine learning based movie recommender application with deployment.

Zhang, F., Yang, X., & Zhang, L. 2022. Efficient hyperparameter tuning for support vector machines using metaheuristic algorithms. *Knowledge-Based Systems* 232: 107-120.

Zin, H. M., Mustapha, N., Murad, M. A. A., & Sharef, N. M. 2017. The effects of pre-processing strategies in sentiment analysis of online movie reviews. In *AIP Conference Proceedings* 1891(020089).