SSRG International Journal of Electronics and Communication Engineering Volume 11 Issue 12, 258-271, December 2024
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I12P124
© 2024 Seventh Sense Research Group®

Original Article

Comprehensive Assessment and Optimization of Sentiment Analysis Models for Movie Reviews with Enhanced Movie Recommendation Systems

Manisha Valera[1], Rahul Mehta[2]

[1]*Gujarat Technological University, Ahmedabad, India.*
[2]*Department of Electronics & Communication Engineering, GEC Rajkot, Gujarat, India.*

[1]*Corresponding Author : valeramanisha@gmail.com*

Received: 29 October 2024 Revised: 30 November 2024 Accepted: 17 December 2024 Published: 30 December 2024

Abstract - Sentiment *analysis is the most important process for considering public sentiment towards movies. This study estimates classifiers using TMDb movie reviews, which identify effective sentiment analysis methods. Here, Innumerable classifiers are compared to optimize sentiment classification accuracy. Here, TMDb API is used to fetch movie reviews stored after preprocessing. Classifiers such as Naive Bayes, SVMs, and MLP are evaluated with TfidfVectorizer for text processing. Then, Performance metrics like accuracy and ROC-AUC scores are designed with SVM. Linear SVC proved to be the best classifier, excelling in evaluation measures such as F1-score, accuracy, and ROC-AUC. SVM performance is boosted through GridSearchCV by optimal parameter tuning, which represents vigorous sentiment analysis capability for movie reviews. Automated sentiment analysis is advantageous for critics and review platforms, improving review processing efficiency. The scalable methodology boosts decision-making across industries for marketing and audience engagement strategies for filmmakers. The challenges triggered in movie recommendation systems were reported by a novel approach introduced in this paper, which combines Count Vectorization, Cosine Similarity, and Truncated Singular Value Decomposition (SVD) by utilizing practices likewise Linear SVC for sentiment analysis and Linear Regression for rating prediction by using Hyperparameter tuning and comparison with basic method implementation, This Study exhibited the importance of fine-tuning models for better accuracy. Our method effectively reveals issues like data sparsity and cold start via extensive investigation and evaluation. Our methodology offers a new era of better recommendation accuracy and effective sentiment analysis, improving movie recommendation systems. Ultimately, this research introduces a novel approach that combines sentiment analysis, hyperparameter tuning, and feature scaling to augment movie recommendation accuracy, distinguishing it from present studies.*

Keywords - *Sentiment analysis, Linear SVC, Movie recommendation, Naive Bayes, Machine Learning.*

1. Introduction

Nowadays, people express their views and understandings about products, services, and movies online, resulting in increased user-generated content over social media and review platforms. With masses of manageable online reviews, sentiment analysis has proved vital for extracting and computing the expressive tone from word-based data. This ocean of sentiment-rich information suggests an opportunity for people to boost movie recommendation systems, which usually depend on metadata and collaborative filtering. However, these traditional methods commonly fail to grasp the subtle favourites of users, making it crucial to incorporate sentiment analysis for more tailored and accurate recommendations. Despite the progressions in recommendation algorithms, numerous challenges persist [11],[12]. Traditional systems often have to struggle with the hurdles[13],[14], a cold-start problem, data sparsity, and the incapability to comprehend the contextual significance of user reviews [15],[16]. Recent research [1] highlights the potential of employing deep-learning practices similar to Long-Short-Term-Memory (LSTM) networks and topic modeling approaches [2], such as Latent Dirichlet Allocation (LDA), to augment sentiment analysis. These methods aid a more inclusive understanding of user sentiments and thematic content, leading to better-quality recommendation accuracy. Additionally, state-of-the-art approaches like real-time

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

sentiment analysis, which use AJAX requests [4] and the utilization of micro-blogging data [5], underline dynamic user preferences. Movie recommendation systems face challenges from biases in user reviews and data sources, which can garble recommendations. These biases, which include demographic and genre preferences, also affect model fairness and accuracy.

This research intends to reconnoitre and compare numerous tactics for amalgamating sentiment analysis into movie recommendation systems. By introducing progressive computational techniques, which comprise machine learning procedures like Term-frequency-inverse-document-frequency, the Authors aim to address existing systems' limits and determine the efficacy of sentiment analysis in creating more user-centric recommendations.

The continued portion of this paper is decided as follows: Section 2,3 analyses related work in sentiment analysis and recommendation systems, Section 4,5 frames the projected methodology followed by the investigational results, and Section 6 discusses the findings and future directions. This study introduces a hybrid approach by combining Count Vectorization, Cosine Similarity, Truncated SVD, and Linear models to address sparsity and cold start issues, enhancing recommendation accuracy. This is followed by hyperparameter tuning, which further boosts sentiment classification performance.

2. Related Work

Long-Short-Term-Memory-networks study [1] is employed to conduct sentiment analysis built from movie reviews. Within textual data, the LSTM architecture is exploited to capture intricate sequential dependencies, which is used to increase sentiment classification accuracy. Future improvements could encompass integrating additional NLP practices to polish sentiment analysis outcomes further.

The proposed tactic [2] merges the sentiment analysis and Latent Dirichlet Allocation (LDA). It first assesses user sentiment towards movies and then employs LDA for topic modeling. This system can suggest bespoke recommendations based on thematic similarity. Future research directions might include mixing LDA with collaborative filtering methods to improve recommendation accuracy.

The presented movie recommendation system [3] leverages sentiment analysis plus cosine similarity. Sentiment analysis regulates user preferences from the movie reviews, whereas cosine similarity computes movie resemblances to generate appropriate recommendations. Additional examination could contain hybrid similarity measures and scalability testing with grander datasets.

The developed real-time movie recommendation engine [4] incorporates sentiment analysis via AJAX requests. This approach lets users get dynamic updates of movie suggestions based on ongoing user sentiment during an interaction. Future developments might emphasise refining real-time recommendation updates and the user interface to boost user engagement.

They explored movie recommendation systems [5] using sentiment analysis extracted from microblogging data. By amassing sentiment from diverse social media platforms, the system aims to improve recommendation accuracy. Future work could involve mixing sentiment analysis from a wider collection of social media causes and scaling up to handle larger datasets.

They applied numerous machine learning tactics [6], which are used to analyze sentiments expressed in movie reviews. The study aims to classify user opinions truthfully by applying these various algorithms.

Future enhancements might include incorporating deep learning practices that can boost the precision of sentiment classification across miscellaneous movie genres and user demographics.

They classified movie reviews for which TF-IDF and augmented Machine-learning systems were used [7]. This approach focuses on feature extraction, which uses TF-IDF and applies collaborative learning practices for study to handle larger and more diverse datasets effectively and optimizes for scalability.

They employed machine learning algorithms [8] that analysed movie review sentiment. Their study contributes to the field by evaluating sentiment across movie reviews, potentially paving the way for future applications in multilingual sentiment analysis and broader linguistic datasets.

As suggested in [9], They focused on sentiment analysis for movie reviews, which are outcomes of machine learning practices. Their study aims to augment sentiment classification accuracy through severe feature engineering and model optimization. Future directions could include mingling domain-specific features and context-aware sentiment analysis to get more accurate classifications.

They planned a hybrid recommender system [10] for movie recommendations enhanced with sentiment analysis. This system personalizes recommendations based on user emotions and preferences by integrating sentiment analysis.

Future enhancements might comprise real-time adaptation to user feedback and integration with streaming data sources for dynamic recommendation updates.

Table 1. Summary of research work done

Dataset	Methodology	Future Enhancement
Movie reviews dataset [1]	LSTM-based sentiment analysis	Boost the LSTM model via attention mechanisms for fine performance.
IMDb movie reviews [2]	Sentiment analysis combined with LDA for topic modeling	Integrate additional features like user profiles and viewing history.
User reviews from multiple sources [3]	Sentiment analysis using cosine similarity	Incorporate neural network-based similarity measures.
User reviews dataset [4]	AJAX-based sentiment analysis for real-time recommendation	Enhance real-time capabilities with more dynamic data sources.
Micro-blogging data (e.g., Twitter) [5]	Sentiment analysis of micro-blogging data for movie recommendation	Advance sentiment analysis by incorporating contextual information.
Movie reviews from various platforms [6]	Machine learning-based sentiment analysis	Extend to multimodal sentiment analysis, including audio and video reviews.
Large movie review datasets [7]	TF-IDF + optimized machine learning algorithms	Discover deep learning representations for better classification accuracy.
Movie reviews dataset in Bangla [8]	Machine-learning techniques for sentiment analysis	Expand to other regional languages and improve language-specific preprocessing.
Diverse movie review datasets [9]	Sentiment analysis (using various machine learning algorithms)	Incorporate hybrid approaches combining machine learning and rule-based methods.
Big data from various movie review platforms [10]	Hybrid recommender system enhanced with sentiment analysis	Integrate with other big data analytics techniques for more robust recommendations.

These summaries mentioned above in Table 1. highlight the varied approaches and policies employed across various studies in the realm of sentiment analysis and movie recommendation systems. This offers insights into recent advancements and probable avenues for additional research.

The algorithm, named the "Novel Hybrid Movie Recommendation System” illustrated in [17], combines multiple techniques for recommendation based on user rating and features of movies, followed by importing datasets like movie and rating, on which TFIDF vectorizer is applied to process the descriptions of movies. It uses SVD-singular value decomposition to factorize the rating matrix and get latent features. Then, the recommendation function is applied to the user ID and movie input, retrieving movies similar to the input movie according to cosine similarity scores. This will calculate the average rating for these similar movies from the dataset named as rating and predict the rating of input movies for the given user, which uses the trained SVD model. The recommended movies are returned in the form of a list. Ultimately, this combines content-based and collaborative filtering tactics to deliver personalized movie commendations.

The scenario [18] highlights the proposed hybrid scheme combining the weighted average and min-max scaler tactics for movie rating and popularity assessment. TF/IDF is used for vectorisation, and the system computes cosine similarity to regulate the similarity between movie vectors. It results in the outstanding 5 commendations of movies for users and prediction rates for given input-specific movies. The paper also highlights the ongoing research via deep learning or neural networks to achieve high-performance accuracy.

The paper [19] presents a methodical approach for a peculiar movie recommendation system. It initiates data collection and preprocessing from the MovieLens dataset and then uses a count vectorizer and TF-IDF vectorizer for feature transformation. It also highlights the known challenges like missing data and engages content-based and collaborative filtering techniques. The outline is deployed on Heroku by integrating Streamlit and Pickle to get a boosted user interface, producing an effective recommendation system to better the user experience.

The hybrid engine algorithm [20] first finds out the movie Id for the given title, then using cosine similarity based on attributes such as 'vote_count', 'year', and 'vote_average' computes the similarity score with the other movies, selects 30 similar movies for which SVD is used for estimating the movie rating. This rating is compared with the actual ratings to count RMSE, and output will be produced as top 10 movies.in this way, movie features and user likings are used for personalized recommendations.

In the paper [22], the proposed innovative algorithm employs a collaborative filtering algorithm that develops SVD and cosine similarity methods. It predicts n list of recommended movies for the current user by building a matrix of user-item utility from rating using movie data. Then, it is normalized, and SVD dimension reduction will be performed. User similarities are calculated by cosine similarity and predicted ratings and recommendations are generated based on the nearest neighbor set. Future investigation can involve analysis of a given algorithm with varied metrics.

In Scenario 2nd [24], importance was given to information like actor /actress as tags for the creation of the recommender system. However, the obstacle to this was the missing actor/actress information within the MovieLens 100K dataset. Therefore, to maintain integrity, this kind of incomplete data was deleted, which led to a decrease in the number of training samples. as opposed to Scenario 1, which only measured genre information. With fewer data points to absorb, the model's ability to accurately predict user preferences may be somewhat negotiated, leading to slightly worse RMSE scores. After all, actor/actress tags still proved advantageous to the system's performance, showcasing its compliance and effectiveness for movie recommendations.

The study [25] incorporates a Movie Recommendation System plus Sentiment Analysis. The recommendation module uses Cosine similarity to suggest movies based on factors like genre, overview, cast, and ratings, attesting accuracy in severe testing. At the same time, sentiment analysis modules comprise Naïve Bayes (NB) plus Support Vector Machine (SVM) algorithms for classifying the reviews as positive or negative. Among them, SVM proved superior in accuracy to NB. It was concluded that the system's precision depends on user input from the corresponding dataset, which faced linguistic barriers.

The paper [26] represented a multimodule approach for forecasting and recommending forthcoming movies based on user choices. It utilizes YouTube promotion reviews and significant traits to predict ratings. It also produces a selection of similar new movies based on user favorites and combines predicted ratings with the user's preferred list. It contains two datasets. Among them, one was about upcoming movie information with trailer reviews and intrinsic features, and another about TMDB with movie metadata and user rating data. It was suggested in future examinations that people can practice social media platforms for data gathering and mingling cross-lingual comments and emojis for sentiment analysis.

The proposed development of a movie recommender system in the paper [27] contained 2 datasets: one was about movie data up to 2017 sourced from Kaggle, and another containing movies from 2018 to 2020 sourced from Wikipedia; detailed movie information is gathered using the TMDB API, while to retrieve customer reviews from IMDB for sentiment analysis BeautifulSoup4 is used. It was a hybrid approach of collaborative and content-based filtering techniques, engaging technologies like Naive Bayes for opinion analysis and NLP for sentiment analysis using NLTK and TFIDF vectorizer. Similarity scores were determined through cosine similarity to assess item similarity. AJAX requests ease efficient client-server communication.

As given above, existing literature explores sentiment analysis and recommendation systems separately, with growing interest in their amalgamation. However, challenges persist in combining sentiment analysis with recommendation techniques to enhance personalization and accuracy.

3. Preliminaries

3.1. Cosine Similarity

Users can use cosine similarity to find the angle between 2 vectors in a multidimensional space. With a range of -1 to 1, where 1 implies perfect similarity, -1 perfect dissimilarity, and 0 orthogonality, it has been calculated as below:

$$Cosine\ Similarity(A, B) = A \cdot B \ / \ \|A\| x \|B\| \tag{1}$$

3.2. Count Vectorization

by enumerating the occurrences of each word. It changes text documents into numerical vectors. This method generates a sparse matrix in which the respective row is related to a document, and each column is related to a unique word, with the cell values on behalf of the word frequencies. It is 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:

$$Count\ Vector(d) = [Count(w1, d), Count(w2, d), \ldots., Count(wn, d)] \tag{2}$$

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.

3.3. Dimensionality Reduction using Truncated SVD

In machine learning, the dimensionality reduction Process is crucial when the datasets have many input variables. To enhance accuracy, focus on relevant variables affecting the output variable and clean up the data. Dimensionality reduction helps a lot. Exploratory Data Analysis (EDA) plays a significant role in identifying and handling missing values, outliers, and irrelevant features. Dimensionality reduction techniques, Principal Component Analysis (PCA), and feature selection methods help with the same.

In summary, dimensionality reduction is an advantageous practice in machine learning, which makes it easy to build the simplest, most effective predictive models that perform well on new data that can be unseen.

SVD, a key dimensionality reduction practice that can be used for Collaborative Filtering (CF), decomposes matrices into U, Σ, and V, revealing latent factors that boost model performance in the realm of matrix analysis, the Singular Value Decomposition - SVD unveils a fundamental formula denoted as $M=U\Sigma V^{\top}$

In this equation:

- M means the original matrix targeted for decomposition.
- U stands for the 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 akin to eigenvalues.
- V represents the right singular matrix, embodying columns encapsulating the right singular vectors. These vectors are the resultant of the eigenvalues of the matrix $M^{\top}M$

For the sparce data /datasets, SVD is effective, particularly those containing multiple zero values. Sparce data is mainly used for applications such as Recommendation Systems in which some items are unrated by the users, resulting in a zero entry for a matrix, Text Classification which represents an ocean of words or TF/IDF, which leads to sparce matrices due to infrequent word with large vocabularies.

In Figure 1. The Singular Value Decomposition of a real 2×2 matrix M includes the sequential actions of V* (a rotation), Σ (scaling by singular values σ1 and σ2), and U (other rotation) on the given disc and vectors. This process changes the disc by rotating, scaling, and rotating again. Several variations of SVD are given here: Truncated SVD, Partial Least Squares SVD, and Randomized SVD.

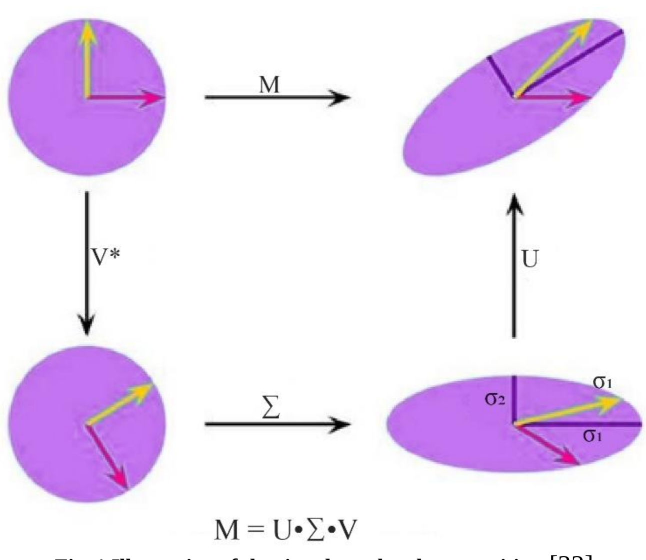

Fig. 1 Illustration of the singular value decomposition [32]

The Truncated SVD differs from Principal Component Analysis (PCA) as it is applied directly to the data matrix, not the covariance matrix. The most significant singular values reduce the data's Dimensionality, their corresponding vectors are retained, and the matrix is effectively compressed. Here, factorization is done for the data matrix so that the number of columns matches the truncation level in the resulting matrices. For example, the value 5.498 can be truncated to 5.5, simplifying the data representation. Truncated SVD, a variant letting personalized output columns, expertly handles sparse matrices, improving feature extraction. This technique and regularization and bias terms upsurge recommendation accuracy by diminishing errors and capturing user-item relations.

3.4. Linear Support Vector Regression

LinearSVR is an SVM variant custom-built for regression tasks, predicting continuous target variables. It is highly scalable and seeks the optimal hyperplane to minimise regression errors. After all, LinearSVR is a powerful tool for efficient and effective regression modeling.

The mathematical formulation for Linear SVR is given below:

$$f(x) = w * x + b \tag{3}$$

The training set(x_i,y_i) in which x_i is known as the feature vector and y_i is known as the target value. In the given equation, w is the weight vector, and b is the bias term. Linear SVR is Effective on High-Dimensional Data and Robust to Outliers.

3.5. Linear Support Vector Classification

LinearSVC is a machine learning algorithm especially made for binary classification processes, where a linear boundary can part data. Particularly, LinearSVC is highly scalable and improved for efficiently handling large datasets. It recognises A hyperplane in space with multiple dimensions to maximize class margin and minimize classification errors. The mathematical formulation for Linear SVR is given below:

$$f(x) = w * x + b \tag{4}$$

In the given equation, w is the weight vector, and b is the bias term. The training set(x_i,y_i) in which x_i is known as the feature vector and $y_i \in \{-1,1\}$, the class label, aims to find a hyperplane that maximizes the margin amid the two classes.

4. Comparative Analysis of Various Classifiers for Sentiment Analysis

The proposed methodology evaluates and optimises numerous classifiers for sentiment analysis on a movie review dataset created by fetching various data from online sources. It uses the TMDb API[21], fetching reviews for movies listed in a CSV file named 'main_data_with_ratings.csv'. It starts with initializing the TMDb API, loads the movie data, and iterates through each movie to retrieve its reviews. And store it in a new CSV file named 'main_data_with_reviews_tmdb.csv'. Then, the process commences by loading and preprocessing this dataset, filling out missing reviews with empty strings. The data is then split into training and testing sets, maintaining an 80-20 split ratio.

The classifiers which are approached in our methodologies are Multinomial Naïve-Bayes[35], Support Vector Machines (SVC and Linear SVC), Logistic Regression, Random Forest, K-Nearest Neighbors, Gradient Boosting, and MLPClassifier. Each classifier is combined with a TfidfVectorizer in a pipeline to alter the text data into mathematical features suitable for machine learning models.

Algorithm: Sentiment Analysis and Classifier Evaluation

1. Initialize TMDb API
2. Load Movie Data
   - Load data from main_data_with_ratings.csv
3. Fetch Movie Reviews
   - For each movie in the dataset:
     - Retrieve reviews using TMDb API
     - Store reviews in main_data_with_reviews_tmdb.csv
4. Load and Preprocess Dataset
   - Load data from main_data_with_reviews_tmdb.csv
   - Fill missing reviews with empty strings
5. Split Data
   - Split data into training set (80%) and testing set (20%)
   - Features (X): Reviews, Target (y): Review Sentiments
6. Define Classifiers
   - Multinomial Naive Bayes, Random Forest, Support Vector Machines (SVC and Linear SVC), Logistic Regression, K-Nearest Neighbours, Gradient Boosting, MLPClassifier
7. Train and Evaluate Classifiers
   - For each classifier:
     1. Create a pipeline with TfidfVectorizer and the classifier
     2. Train the pipeline on the training data (X_train, y_train)
     3. Predict on the testing data (X_test)
     4. Calculate evaluation metrics
     5. Store evaluation metrics
8. Optimize SVC using GridSearchCV
   - Define parameter grid:

Fig. 2 Algorithm of Sentiment Analysis and Classifier evaluation for Movie Review

Here, the classifiers are trained on the training set, and predictions are made through the test set. The evaluation metrics for the classifier are accuracy, precision, recall, F1-score, and ROC-AUC scores, which are calculated to measure their performance. The ROC-AUC score is computed for the classifiers supporting probability predictions, which consider binary and multiclass scenarios using binarized labels when necessary. For the classifiers' performance evaluation, several evaluation metrics are generally used: F1 score, accuracy, recall, and precision. These are essential because of their widespread application. Presenting the confusion matrix is vital for establishing the mathematical formulations for these metrics. The confusion matrix in Figure 4 can be understood as given in the error matrix, which shows 4 quantities: 1. True

Positive (TP), 2. False Positive (FP), 3. True Negative (TN), and 4. False Negative (FN). A matrix row indicates the actual class, whereas each column denotes the predicted class.

TP specifies the positive review, which is predicted by the classifier correctly, and 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. In the 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 specifies the ratio of correct predictions to the overall predictions. Its best value is 1 for 100% accurate prediction, and the least value of 0 for 0% prediction. Accuracy is well-defined hereby given formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

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

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

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

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

The F1 score is measured as a vital parameter for estimating the classifier's performance and is frequently observed as more critical than precision and recall. It describes the balance between precision and recall by incorporating both measures.

The value of the F1 score ranges between 0 and 1, with 1 indicating flawless classifier performance. The F1 score is defined here by the given formula:

$$F1\,Score = 2 * \frac{Precision * Recall}{Precision+Recall} \quad (8)$$

Further, the methodology comprises hyperparameter optimization for the SVC [47], which uses GridSearchCV to determine the model's best parameters. Then, the optimized SVC model's performance is estimated equated with the other classifiers [46].

Here, The best classifier is finalized based on the highest F1 score, and its detailed performance metrics are also reported. Here is a table summarizing the evaluation metrics for each classifier, including the best classifier:

Table 2. The evaluation metrics for given classifiers

Classifier	Accuracy	Precision	Recall	F1-score	ROC-AUC Score
Multinomial Naive Bayes	0.6711	0.6283	0.6711	0.542	0.9337
Random Forest	0.9158	0.9137	0.9158	0.8882	0.9411
Support Vector Machines	0.9211	0.9178	0.9211	0.8991	0.9665
Linear SVC	**0.9321**	**0.9255**	**0.9321**	**0.9258**	**N/A**
Logistic Regression	0.925	0.9196	0.925	0.9099	0.9685
K-Nearest Neighbours	0.7006	0.8013	0.7006	0.6918	0.7528
Gradient Boosting	0.9202	0.9114	0.9202	0.9045	0.9518
MLPClassifier	0.9264	0.9237	0.9264	0.9248	0.9537
Optimized SVC	0.9347	0.9331	0.9347	0.9239	0.9677

Moreover, bar plots in Figure 3. are created to visually compare the performance metrics among all the classifiers, from which highlights the best classifier in Figure 3. ROC-AUC curves for classifiers with available ROC-AUC scores are plotted to represent their discrimination ability visually. The finest classifier is 'Linear SVC', as given in Table 2.

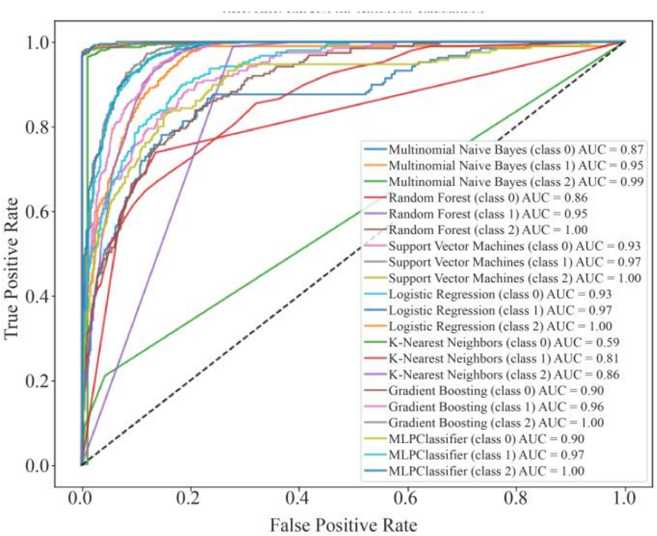

Fig. 3 ROC AUC curves of different classifiers

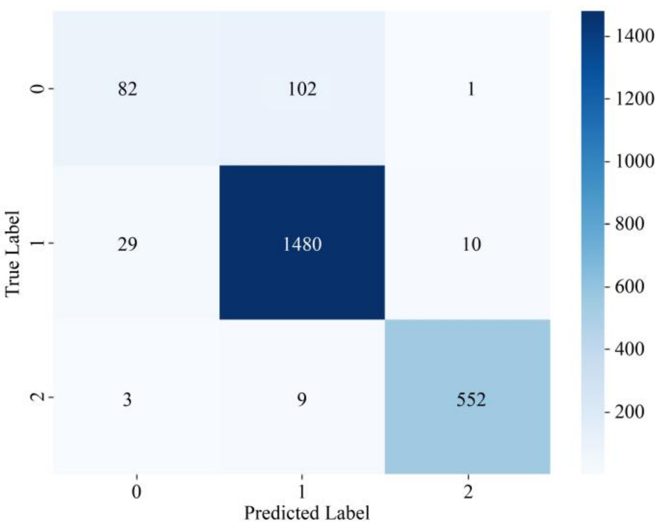

Fig. 4 Confusion Matrix of Linear SVC

This inclusive evaluation and comparison framework provides the most effective classifier for sentiment analysis, enhancing their performance through hyperparameter tuning and providing a strong visualization of their relative strengths and dullness. The Naive Bayes method is computationally competent for high-dimensional text data, making it a strong baseline for sentiment analysis. Whereas SVM handles sparse text features effectually with robust decision boundaries, and MLPClassifier captures non-linear relationships and complex feature interactions, offering a deep learning perspective. These kinds of models provide diverse approaches for all-inclusive evaluation.

5. Experimental Framework

5.1 Data Compilation and Preprocessing

Here, a developed recommender system containing two key datasets [29]from Kaggle[30]: one comprising the top 5000 movies up to 2017 and another with metadata for these movies. Furthermore, it integrates a Wikipedia database listing movies from 2018 to 2020[31],[33],[34]. The system leverages the TMDB API [21] to enhance movie details to extract information such as titles, genres, ratings, and banners. The process is initiated using a Python script, fetching user scores for movies listed in 'main_data.csv'. It initializes the TMDB API, retrieves user scores using the function get_user_score, and eliminates rows with missing scores before saving the updated data to 'main_data_with_ratings.csv'. Another script then fetches reviews for these movies from'main_data_with_ratings.csv', which retrieves reviews using the get_movie_reviews function, then merges the review data with the original movie data, and saves the merged data to 'main_data_with_reviews_tmdb.csv'. Lastly, a sentiment analysis script loads 'main_data_with_reviews.csv', confirms that all reviews are of string type, handles missing values, then classifies each review as 'Good', 'Neutral', or 'Bad' using TextBlob, adds these results as a 'Review_Sentiment' column, and saves the updated data to 'main_data_with_sentiment_tmdb.csv'.

The experimental setup involves a dataset named 'main_data_with_sentiment_tmdb.csv', which is approximately 13.6 MB. The models were implemented using Python and trained on a system with 8 GB of RAM and an Intel Core i5 processor.

5.2 Proposed Enhanced Recommendation Model with Sentiment Analysis

For developing a movie recommendation system, 2 approaches are specified in Figure 5, incorporating sentiment analysis and user ratings. The specified first method began by loading a dataset, handling missing values, and combining several columns into a single string for feature extraction. The preprocessing steps involve cleaning and combining data (e.g., metadata, reviews, and sentiment scores), vectorizing text features using CountVectorizer, and reducing dimensionality with SVD. Sentiment analysis is applied here to extract sentiment scores and labels, combined with other features, and then split the data for training and evaluation. Sentiment analysis was achieved using TextBlob to derive sentiment scores and labels from movie reviews using CountVectorizer. These features were vectorized, and through TruncatedSVD, dimensionality reduction was achieved. The processed features and sentiment scores were then utilized to train a LinearSVC model for sentiment classification and a LinearSVR model for predicting user ratings. Our system used cosine similarity on the SVD features to suggest movies and predict user ratings for these recommendations using the LinearSVR model. The performance of the LinearSVC was mapped using evaluation metrics (accuracy, precision, recall, F1 score), while the LinearSVR was assessed using MAE (Mean Absolute Error), MSE (Mean Squared Error) with a scatter plot visualizing actual versus predicted ratings.

In the 2nd scheme, the first approach was enhanced by incorporating hyperparameter tuning through GridSearchCV for both LinearSVC and LinearSVR, aiming to achieve optimum performance by selecting the best parameters based on cross-validation results. This tuning led to far accurate models for rating prediction and sentiment classification. The improved models provided more accurate and reliable recommendations and predictions. The critical role of hyperparameter tuning was highlighted in This approach for enhancing model performance and achieving more dependable outcomes in both classification and regression tasks. Hyperparameter tuning or hyperparameter optimization is the procedure of selecting the best set of hyperparameters for a machine learning model. Before the learning process commences, Hyperparameters can be set, and they control the training method and the structure of the given model itself.

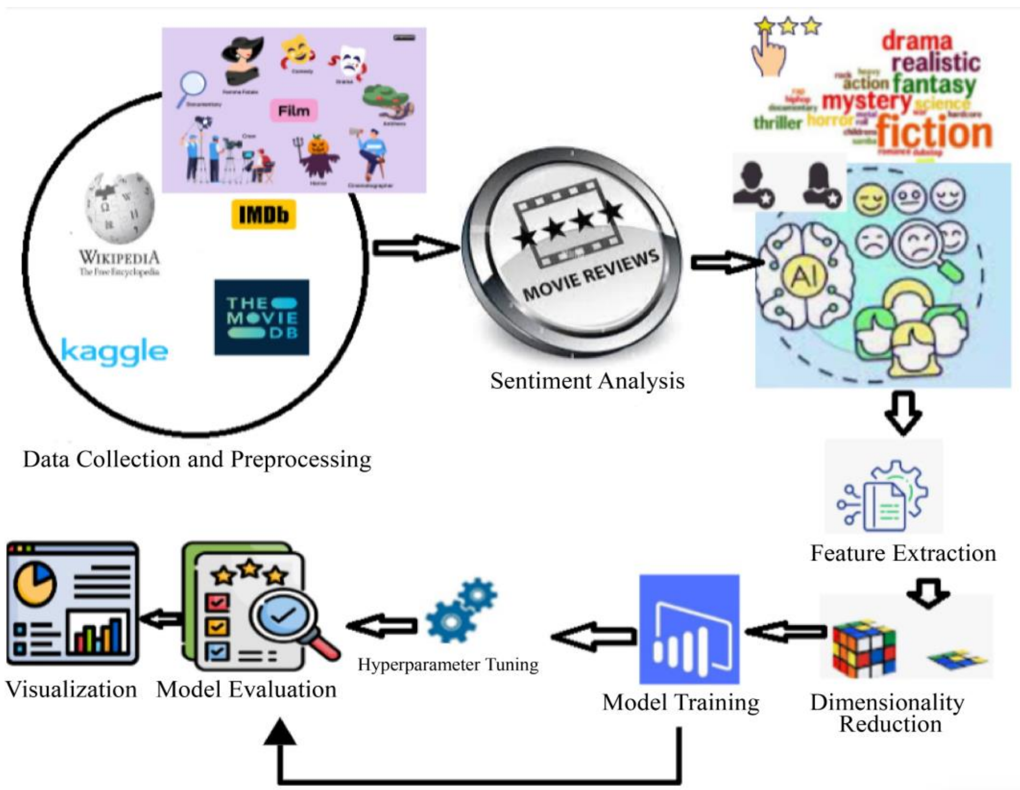

Fig. 5 Process flow of movie recommendation and sentiment analysis with and without hyperparameter tuning

Table 3. Evaluation metrics comparison

Metric	Method 1 (Without Hyperparameter Tuning)	Method 2 (With Hyperparameter Tuning)
Accuracy (SVC)	98.72%	98.67%
Precision (SVC)	99.50%	98.85%
Recall (SVC)	98.97%	99.58%
F1 Score (SVC)	99.24%	99.21%
MAE (SVR)	0.787	0.621
MSE (SVR)	1.097	0.761

Method 1 is implemented without hyperparameter tuning, whereas Method 2 comprises hyperparameter tuning, heightening the model's performance in Table 3. Evaluation Metrics comparison is given. Overall, both methods perform mostly similarly in accuracy for the Linear SVC model, in which Method 1 slightly outperforms Method 2. However, Method 2 showed obtainable improvements in recall and achieved a balanced F1 score. For the model named Linear SVR, Method 2 proved significantly better performance. In the context of lower Mean Absolute Error and Mean Squared Error, it suggests more accurate predictions of user ratings. Overall, Method 2 - Hyperparameter tuning provides more accurate and sophisticated predictions for the movie recommendation system. The GridSearchCV tested the specified values of C: [0.1,1,10] and acknowledged C=0.1 as the best hyperparameter for the LinearSVC model. Here, the best_estimator_ corresponds to the model trained with this optimal parameter.

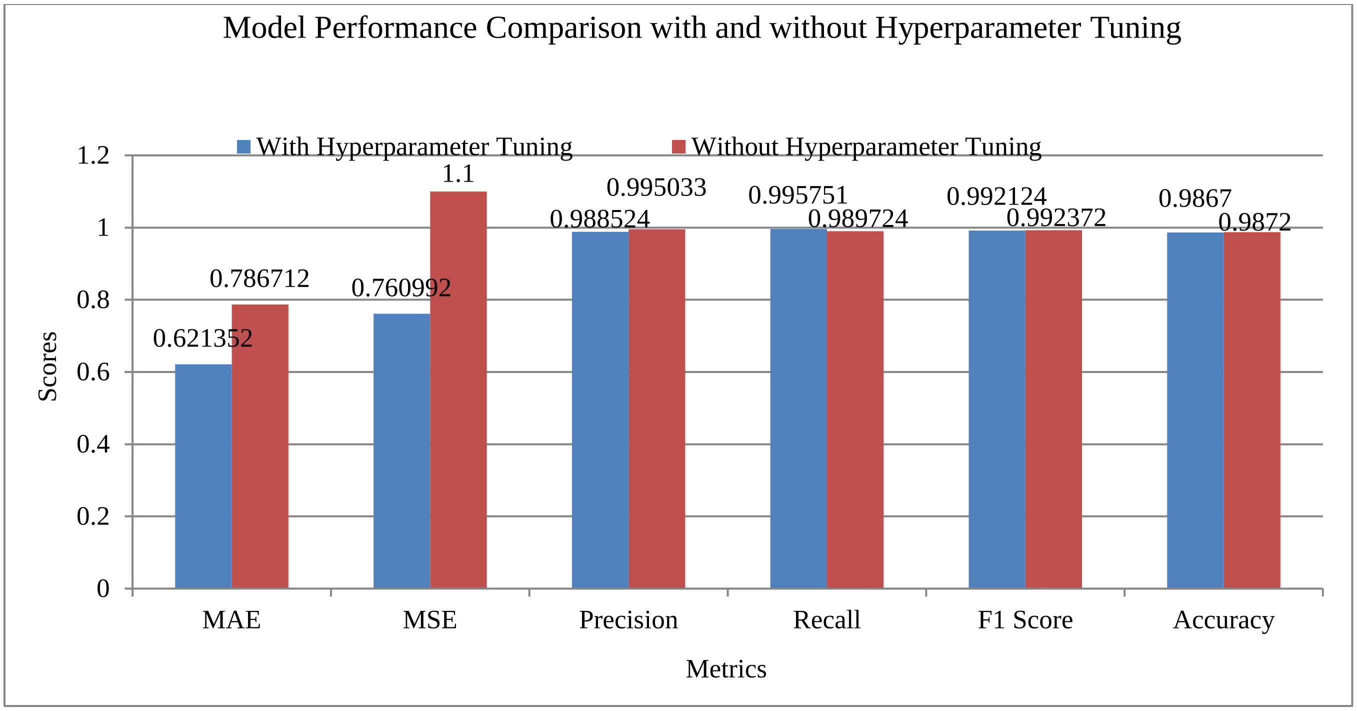

Fig. 6 Evaluation measure comparison of movie recommendation and sentiment analysis with and without hyperparameter tuning

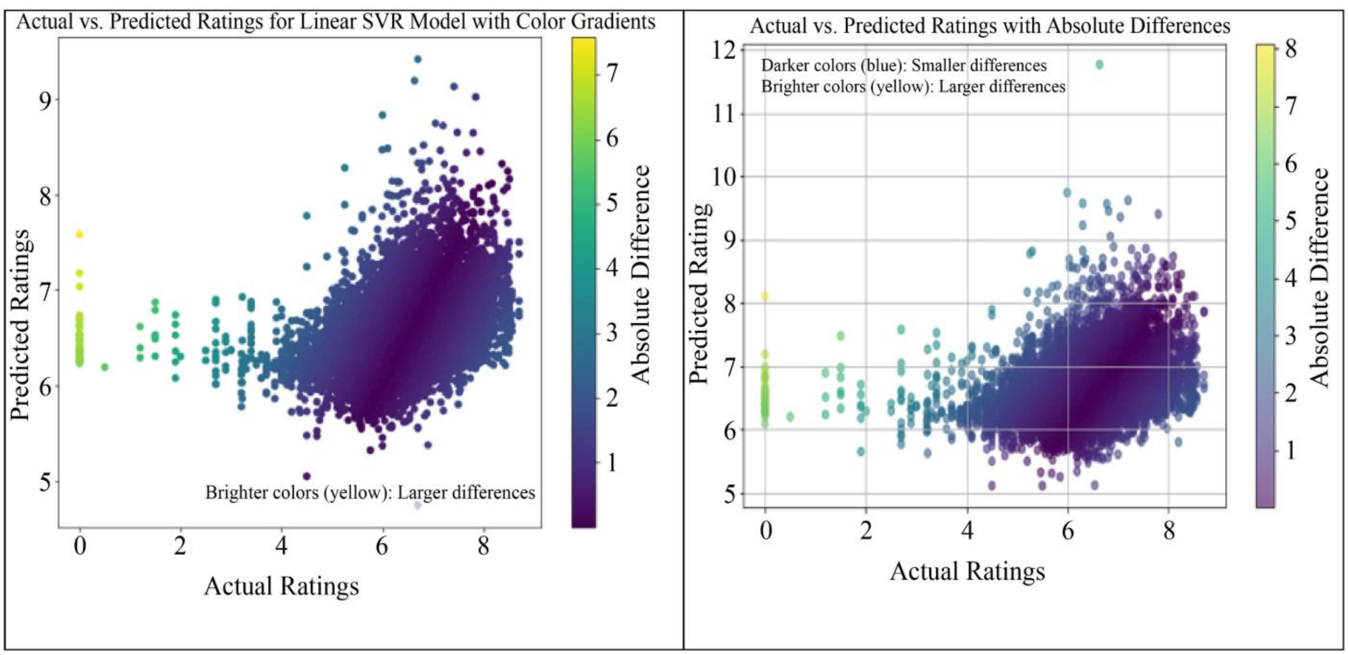

Fig.7 Actual Ratings Vs. Predicted Ratings Of the Linear SVR model(1) with Hyperparameter tuning(2)without Hyperparameter tuning

If the scatter plot shows mostly purple/dark blue colors, it suggests that the predicted ratings are closest to the actual ratings, which indicates a smaller absolute difference between them. This suggests that the linear SVR model performs fine in predicting ratings consistent with the actual values. The color gradient helps to visualize speedily where the model's predictions are more accurate (darker colors like a purple and dark blue) and where they are less accurate (brighter colors like yellow). The plot with more purple colors implies higher prediction accuracy or smaller errors between the model's predictions and the actual ratings. A plot with fewer purple colors and more yellow or brighter colors would indicate larger absolute differences and possibly less accurate predictions in Figure 7. Linear SVR with Hyperparameter tuning shows higher prediction accuracy than Linear SVR without Hyperparameter tuning.

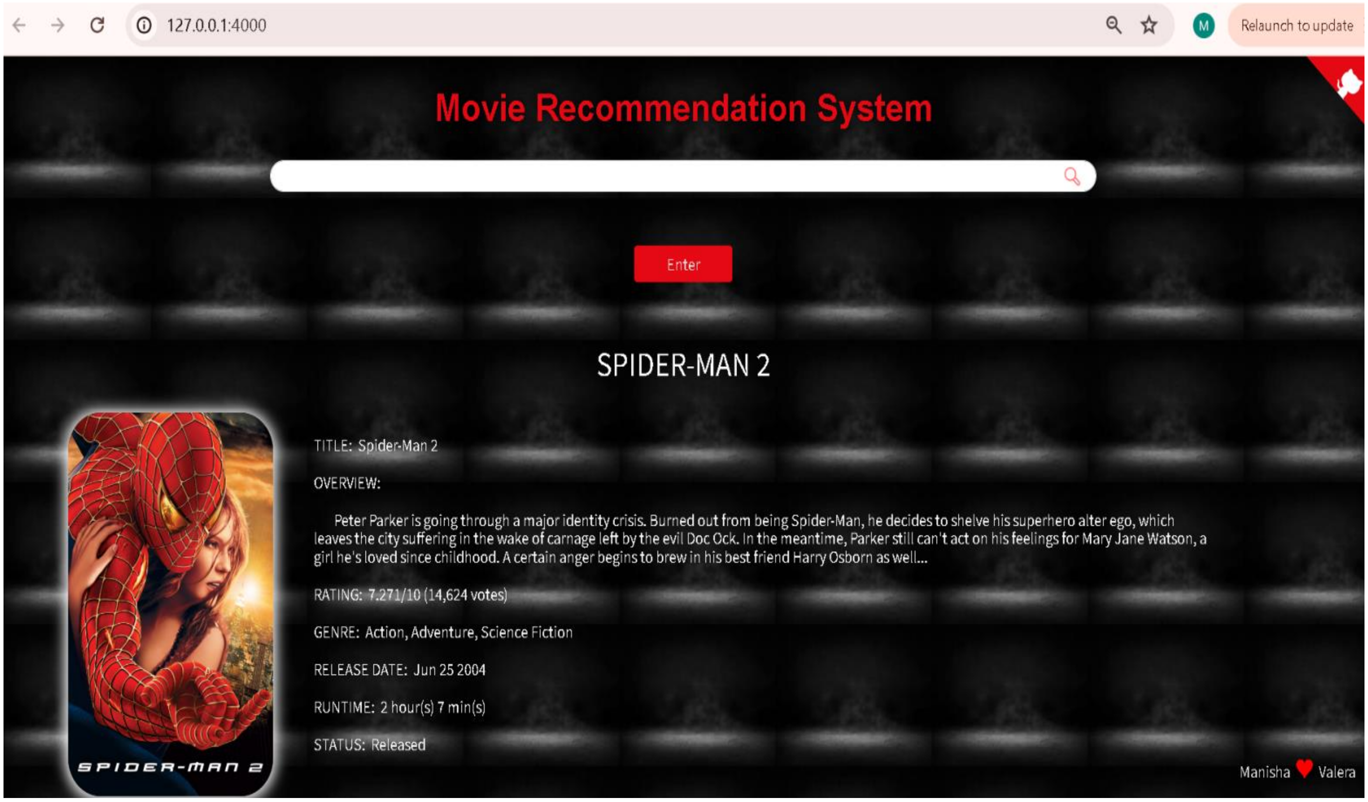

Fig. 8 Displaying movie details using web scraping

The project uses AJAX to send user details to the backend efficiently and dynamically display results.in Figure 8. The search engine offers detailed movie information, including genre, overview, runtime, rating, status, top cast, name, and release year, which is processed on the backend and shown on the client side. Figure 9. focuses on sentiment analysis, organizing user reviews as positive /negative or neutral to gain an understanding of audience sentiments.

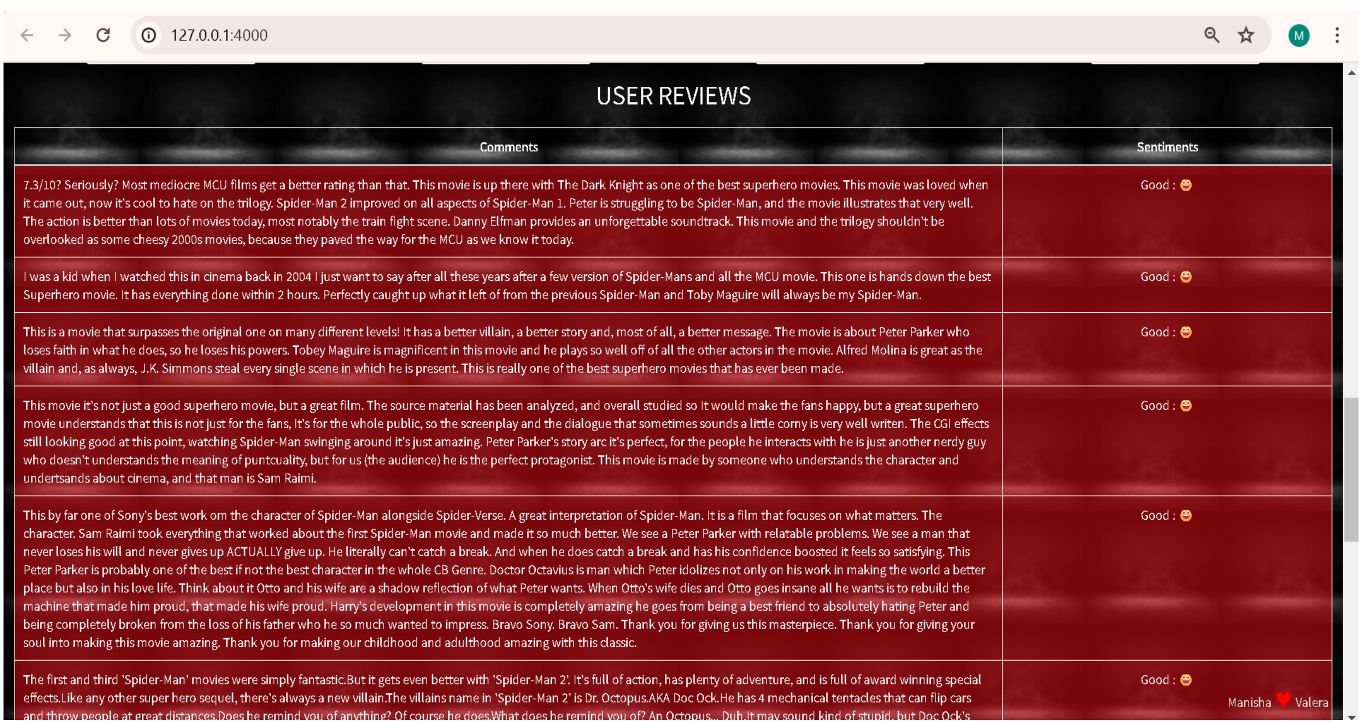

Fig. 9 Sentiment analysis based on user reviews

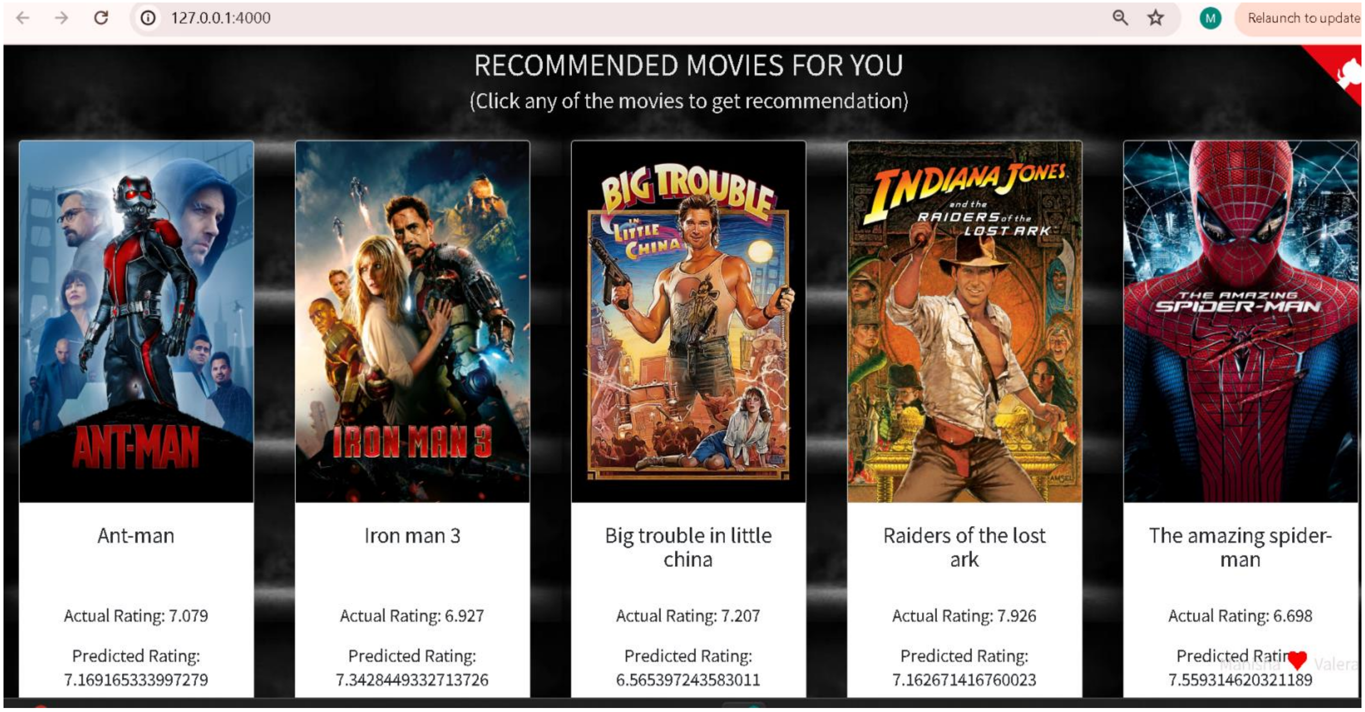

Fig. 10 5 Recommended movies with the least difference between actual and predicted ratings using Linear SVR

The system in Figure 10 shows movie recommendations using sentiment analysis and Linear SVR to rank movies with minimal rating prediction difference.

The paper [26] has proposed a model with a Mean Absolute Error of 0.7406 and a Mean Squared Error of 0.8648, respectively. However, in this paper, the proposed methodology using hyperparameter tuning provides a mean absolute error of 0.621 and a mean squared error of 0.761, respectively, which shows improvement in predicting the rating.

This research can boost movie recommendations and customer engagement in the film industry while encouraging sectors like e-commerce and marketing through improved personalized services.

Better results are achieved by integrating sentiment analysis with advanced feature extraction techniques like TF-IDF and SVD, combined with hyperparameter tuning for models, which allowed for more accurate and personalized predictions compared to existing methods in the literature.

6. Conclusion

Considering the future enhancements for this sentiment analysis and classifier evaluation methodology outlined, several ways could deliberately advance its capabilities. Firstly, mixing deep learning models such as Transformers, which can excel in capturing complex semantic relationships, can enhance classification accuracy by utilizing pre-trained language representations such as BERT or RoBERTa and secondly, discovering ensemble methods that can combine predictions from numerous classifiers that could potentially improve robustness. Lastly, scaling infrastructure uses cloud computing to handle larger datasets and computational demands. It would enable more inclusive analyses and quicker model iterations. These augmentations aim to raise the tactics' effectiveness, scalability, and interpretability while performing sentiment analysis on movie reviews.

Using tactics named Linear SVC for sentiment classification and Linear SVR for rating prediction. The authors have compared two methods: a basic implementation and one with hyperparameter tuning, which remarkably improved the regression model's accuracy, evident in lower MAE and MSE values.

Overall accuracy remained consistent, Although the classification model saw only marginal gains in recall and F1 score. It is noted that Hyperparameter tuning is vital for refining machine learning models, especially in complicated tasks like recommendation systems, enhancing prediction precision and user experience. It is concluded that

Future improvements could explore advanced algorithms like neural collaborative filtering and ensemble methods to boost recommendation accuracy and diversity. Unlike existing literature, this work addresses the limitations of traditional recommendation models by providing more truthful and fair predictions. This approach opens new avenues for future studies in personalized recommendations and model optimization methods.

References

[1] Jyostna Devi Bodapati, N. Veeranjaneyulu, and Shareef Shaik, "Sentiment Analysis from Movie Reviews Using LSTMs," *Information Systems Engineering*, vol. 24, no. 1, pp. 125-129, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yilin Zhang, and Lingling Zhang, "Movie Recommendation Algorithm Based on Sentiment Analysis and LDA," *Procedia Computer Science*, vol. 199, pp. 871-878, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Raghav Mehta, and Shikha Gupta, "Movie Recommendation Systems using Sentiment Analysis and Cosine Similarity," *International Journal for Modern Trends in Science and Technology*, vol. 7, no. 1, pp. 16-22, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Satish Chadokar, Naman Jain, and Ayush Thakre, "Movie Recommendation Engine with Sentiment Analysis Using AJAX Request," *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, vol. 11, no. 1, pp. 1-8, 2023. [Publisher Link]

[5] K. Revanth Sai, K. Gopi Chand, and Mr.E.Sankar, "Movie Recommendation Using Sentiment Analysis From Micro Blogging Data," *International Journal of Creative Research Thoughts*, vol. 11, no. 3, pp. 335-342, 2023. [Google Scholar] [Publisher Link]

[6] Harsh Sharma et al., "Sentimental Analysis of Movie Reviews Using Machine Learning," *2nd International Conference on Data Science and Intelligent Applications*, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Muhammad Zaid Naeem et al., "Classification of Movie Reviews Using Term Frequency-Inverse Document Frequency and Optimized Machine Learning Algorithms," *PeerJ Computer Science*, vol. 8, pp. 1-, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Atiqur Rahman, and Md. Sharif Hossen, "Sentiment Analysis on Movie Review Data Using Machine Learning Approach," *2019 International Conference on Bangla Speech and Language Processing*, Sylhet, Bangladesh, pp. 1-4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Anshika Gupta et al., "Sentiment Analysis of Movie Review Using Machine Learning," *Educational Administration: Theory and Practice*, vol. 30, no. 5, pp. 7344-7354, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Yibo Wang, Mingming Wang, and Wei Xu, "A Sentiment-Enhanced Hybrid Recommender System for Movie Recommendation: A Big Data Analytics Framework," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, pp. 1-9, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ishan Rathi, and Manoj Sethi, "A Collaborative Filtering-Based Recommender System Alleviating Cold Start Problem," *International Journal of Scientific & Engineering Research*, vol. 10, no. 9, pp. 27-34, 2019. [Google Scholar] [Publisher Link]

[12] Anshu Sang, and Santosh K. Vishwakarma, "A Ranking Based Recommender System for Cold Start & Data Sparsity Problem," *2017 Tenth International Conference on Contemporary Computing*, Noida, India, pp. 1-3, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Antiopi Panteli, and Basilis Boutsinas, "Addressing the Cold-Start Problem in Recommender Systems Based on Frequent Patterns," *Algorithms*, vol. 16, no. 4, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mehrnaz Mirhasani, and Reza Ravanmehr, "Alleviation of Cold Start in Movie Recommendation Systems Using Sentiment Analysis of Multi-Modal Social Networks," *Journal of Advances in Computer Engineering and Technology*, vol. 6, no. 4, pp. 251-264, 2020. [Google Scholar] [Publisher Link]

[15] Satya Keerthi Gorripati, and Valli Kumari Vatsavayi, "Community-Based Collaborative Filtering to Alleviate the Cold-Start and Sparsity Problems," *International Journal of Applied Engineering Research*, vol. 12, no. 15, pp. 5022-5030, 2017. [Google Scholar] [Publisher Link]

[16] Jinal Panchal, and Sandeep Vanjale, "A Deep Learning Approach towards Cold Start Problem in Movie Recommendation System," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 8S, pp. 81-85, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Dushyant Bahl et al., "A Novel Hybrid Approach Towards Movie Recommender Systems," *Journal of Statistics and Management Systems*, vol. 23, no. 6, pp. 1049-1058, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] R.N. Ravikumar, Sanjay Jain, and Manash Sarkar, "Efficient Hybrid Movie Recommendation System Framework Based on a Sequential Model," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 9S, pp. 145-155, 2023. [Google Scholar] [Publisher Link]

[19] S. Yogesh, and S. Ganesh Kumar, "Machine Learning Based Movie Recommender Aplication with Deployment," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, no. 4, pp. 6771-6776, 2023. [CrossRef] [Publisher Link]

[20] S. Rajarajeswari et al., "Movie Recommendation System," *Emerging Research in Computing, Information, Communication and Applications*, vol. 1, pp. 329-340, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] TMDB Website. [Online]. Available: https://www.themoviedb.org/

[22] Nisha Bhalse, and Ramesh Thakur, "Withdrawn: Algorithm for Movie Recommendation System Using Collaborative Filtering," *Materials Today Proceedings*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Vito Xituo Chen, and Tiffany Ya Tang, "Incorporating Singular Value Decomposition in User-Based Collaborative Filtering Technique for a Movie Recommendation System: A Comparative Study," *Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence*, Wenzhou China, pp. 12-15, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Quynh Nhut Nguyen et al., "Movie Recommender Systems Made through Tag Interpolation," *Proceedings of the 4th International Conference on Machine Learning and Soft Computing*, Haiphong City Viet Nam, pp. 154-158, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] N. Pavitha et al., "Movie Recommendation and Sentiment Analysis using Machine Learning," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 279-284, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Sandipan Sahu et al., "A Hybrid Recommendation System of Upcoming Movies using Sentiment Analysis of Youtube Trailer Reviews," *Mathematics*, vol. 10, no. 9, pp. 1-22, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Sonali Rathore, Seema Narvare, and Satish Chadokar, "Movie Recommendation Engine with Sentiment Analysis: A Survey," *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, vol. 11, no. 3, pp. 1-7, 2023, [Publisher Link]

[28] IMDB Website. [Online]. Available: https://www.imdb.com/

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

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

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

[32] Beginners Guide to Truncated SVD for Dimensionality Reduction, Analytics India Magazine, 2021. [Online]. Available: https://analyticsindiamag.com/deep-tech/beginners-guide-to-truncated-svd-for-dimensionality-reduction/

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

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

[35] Sida Wang, and Christopher D. Manning," Baselines and Bigrams: Simple, Good Sentiment and Topic Classification," *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 90-94, 2012. [Google Scholar] [Publisher Link]