SCT Proceedings in Interdisciplinary Insights and Innovations. 2025; 3:376
https://doi.org/10.56294/piii2025376

Category: Interdisciplinary Communication and Media Technologies

ORIGINAL

A novel analysis on movie recommendation using machine learning approach

Un análisis novedoso sobre la recomendación de películas mediante el aprendizaje automático

ASVNM Manjusha [1], Pechigulla Ajay [1], Singireddy Sreevani [1], Chatradi Tejaswini [1], K.Padmanaban [1]

[1] Koneru Lakshmaiah Education Foundation, Vaddeswaram,India.

Cite as: ASVSNM M, Pachigulla AK, Singireddy S, Chatradi T, Padmanaban K. A novel analysis on movie recommendation using machine learning approach. SCT Proceedings in Interdisciplinary Insights and Innovations. 2025;3:376.

Submitted: 12-09-2025 **Reviewed:** 27-11-2024 **Accepted:** 03-01-2025 **Published:** 05-01-2025

Editor: Emanuel Maldonado

ABSTRACT

In today's culture, technology plays a crucial role for any kind of recommendations. In this digital World, Algorithms are used in movie recommendation systems to offer films to users based on their watching history or ratings. With the advent of digital services and the massive quantities of information they accumulate on user preferences, these systems have grown in popularity in recent years. This system employs machine learning to do sentiment analysis on movie reviews to improve the user experience. This article also compares NB, SVM, and KNN on metrics such as Accuracy, Precision, Recall, and F1 Score. We present an overview of the many algorithms utilized in the creation of movie recommendation systems, encompassing text categorization, knowledge filtering, and hybrid techniques, in this review. We also examine the problems and limits of these algorithms, as well as possible future research objectives in this subject.

Keywords: Support Vector Machine (SVM), Naive Bayes and K-Nearest Neighbors Algorithm.

RESUMEN

En la cultura actual, la tecnología desempeña un papel crucial para cualquier tipo de recomendación. En este mundo digital, los algoritmos se utilizan en los sistemas de recomendación de películas para ofrecerlas a los usuarios en función de su historial de visionado o sus valoraciones. Con la llegada de los servicios digitales y las ingentes cantidades de información que acumulan sobre las preferencias de los usuarios, estos sistemas han ganado popularidad en los últimos años. Este sistema emplea el aprendizaje automático para realizar análisis de sentimientos sobre las críticas de películas con el fin de mejorar la experiencia del usuario. Este artículo también compara NB, SVM y KNN en métricas como la exactitud, la precisión, la recuperación y la puntuación F1. En esta revisión presentamos una visión general de los muchos algoritmos utilizados en la creación de sistemas de recomendación de películas, que abarcan la categorización de textos, el filtrado de conocimientos y las técnicas híbridas. También examinamos los problemas y límites de estos algoritmos, así como posibles objetivos futuros de investigación en este tema.

Palabras clave: máquina de vectores de soporte (SVM), algoritmo de Naive Bayes y de K Vecinos Más Cercanos.

INTRODUCTION

The Internet has developed swiftly since its inception. Because of the abundance of data accessible via the internet, it is difficult to obtain important information quickly and easily. Fortunately, recommendation systems assists to overcome this difficulty.

Nowadays, Movie optimization techniques have become an essential aspect of the entertainment business in recent years. With the growth of streaming platforms and the massive quantities of data collected on user preferences, these systems have the ability to assist viewers find new films that they may appreciate while also driving income for content producers.

Despite their usefulness, these algorithms have a number of obstacles and constraints that must be addressed. The "cold start" problem, primarily alludes to the difficulty in producing suggestions for clients that have not yet contributed sufficiently data for the system to effectively forecast their preferences, is one of the challenges. Another difficulty is balancing the diversity and originality of recommendations with about there ultimate correctness.

Consequently, the evolution of movie recommendation systems based on algorithms has the potential to significantly improve user serialized dramas while also driving income for content producers.

There are several algorithms for movie recommendation systems, each with its own set of strengths and flaws. Collaborative filtering algorithms generate suggestions based on the watching behaviors of similar users, whereas content-based screening strategies make recommendations utilizing the characteristics of the films themselves. In an effort to increase suggestion accuracy, hybrid techniques that mix adaptive algorithms and evidenced filtering have also been developed.

In this review, we present an overview of the many algorithms utilized in the creation of movie recommendation systems, while also providing a discussion of their problems and limits. We also discuss some potential future areas for study in this area.

LITERATURE REVIEW

In recent years, there have been a substantial amount of study on movie recommendation systems, with many algorithms and techniques being created and evaluated.

One of the most prominent ways is particle swarm optimization, which provides suggestions based on comparable users' viewing behaviors. There are two types of collaborative filtering: sensation collaborative filtering and framework collaborative filtering. These produce values based on the complete user-item dataset, whereas model-based collaborative filtering algorithms develop a latent representation of the data before making recommendations.

Another method is content-based filtering, whereby produces suggestions based on film qualities such as genre, actors, and narrative. These algorithms often offer suggestions based on a feature wavelet coefficients of the movie and approaches such as cosine similarity.

By combining the qualities of both methodologies, these approaches have the ability to increase suggestion accuracy.

The cosine similarity approach is used to recommend movies. The distance function is computed using a normalized popular score, and the K-Nearest Neighbors approach is used to increase accuracy.

KNN has several advantages, including its simplicity, flexibility, and capability to deal with inter and continuous dependent variable. The simplicity of NB, as well as its capability to handle missing data and highly dimensional data, are among its advantages. SVM's advantages include its capability to process high-dimensional data as well as its fluidity via the use of various kernel functions.

DATA ENGINEERING

Movie Database management system, projection methods, user option analysis in cognitive science all have roots in recommender system research. In the mid-1990s, recommender systems developed as a separate study field. The basic goal of Sentiment Classification is to research users' behavior in order to learn their preferences and interests. Existing Recommender Systems are classified into seven fundamental types and their sub - classes, which we will look at in depth in this section.

A. Data Cleaning

Before you start creating your recommendation model, ensure sure your dataset is clean and free of mistakes or inconsistencies. This might include dealing with missing numbers, fixing mistakes, and deleting duplicates.

B. Feature Selection

You must determine which features will be used as input to your recommendation model. This might include movie-specific information (e.g., genre, year of release, actors, etc.) as well as user-specific features (e.g., age, gender, ratings history).

C. Data Filtering

In the actual world, data obtained via explicit feedbacks such as movie ratings might be quite scarce, with most data points coming from really popular goods (movies) and highly engaged people. A large number of lesser-known products (movies) have no ratings at all. Let's plot the frequency distribution of movie ratings.

D. Normalization

Normalizing your characteristics so that they are all on the same scale may be beneficial. This is especially crucial when utilizing algorithms that are sensitive to the magnitude of the input characteristics, such as KNN or SVMs.

E. Train / Test Split:

To analyze the performance of your recommendation model, divide your dataset into training and testing sets for the algorithms.

F. Data Transformation

Depending on the nature of your data, you may need to apply various modifications to your features before using them with your preferred recommendation algorithm. For example, you could wish to apply a log transformation to numerical characteristics that are skewed.

G. Feature Engineering

In order to optimize the recommendation model, you may wish to generate new features by merging or altering current features.

These are just a few examples of pre-processing processes to think about while developing a movie recommendation system with SVMs, KNN, or NB. The particular preparation procedures you'll need to perform will get to optimized for the recommendation algorithm you've chosen.

ANALYSING THE MODEL

In the study, a dataset was utilised to develop the mutual framework, which is employed to recommend titles. The Sentiment Classification algorithms K-Nearest Neighbors (KNN), Naive Bayes (NB), and Support Vector Machine (SVM) were then trained on another dataset.

A. K – Nearest Neighbors Model:

https://doi.org/10.56294/piii2025376

A straightforward supervised machine learning technique for classification and regression is K-Nearest Neighbors (KNN). Based on the ratings that other users have given a movie, KNN can be used in the context of movie recommendation to estimate the result that a user would give to that film.

Calculating the distance between the movie for which we want to forecast the rating and every other movie in the dataset is the fundamental concept underpinning KNN. Then, in order to predict the rating for the movie in question, we choose the K movies that are the most similar to the one in question. We do this by looking at the ratings that users have given to those films.

The distance between movies can be calculated in a few different ways, such as utilizing cosine similarity or the Euclidean distance. Cosine similarity is measured as the dot product of the ratings vectors of the two movies divided by the product of their magnitudes, whereas Euclidean distance is determined as the square root of the sum of the squares of the variances between the ratings of the two movies.

$$= \frac{\sum_{p<q}(x(p,q)-x)(t(p,q)-t)}{\sqrt{[\sum_{p<q}(x(p,q)-x]^2[\sum_{p<q}(t(p,q)-t]^2}}$$

where x is the average of x (p, q), t (p, q) is the dendro grammatic distance in between ith and jth observation, and t is the average of t (p, q).

Fundamentally, Minkowski distance is a distance metric that could be employed in recommendation systems to determine the similarity of two individuals or two products. It is characterized as the fth root of the sum of the absolute differences of their coordinates increased to the power of p and is a generalization of both the Euclidean and Manhattan distances. When p equals 1, the Minkowski distance has now become the Manhattan distance, and when p equals 2, the Minkowski distance has become Euclidean distance.

dist(p, q)=(∑r=1d∣pr−qr∣f)1/f

Once the relationship in between movie and all other movies has been discovered, we may choose the K movies that are the closest equivalent to the one that is query and apply user ratings for those movies to forecast the rating for the movie in question.

We can either weight the ratings by the distance between the movies, so that movies that are closer to the movie in question have a stronger influence on the forecast, or we can simply average the ratings of the K nearest neighbor to estimate a movie's rating.

Using the ratings that other users have given to movie A and other movies, for instance, we can estimate the rating that a user will give movie A. To compute the prediction,

we can use the formula shown below:

$$Rating\_prediction = \frac{sum(rating_i * weight_i)}{sum(weight_i)}$$

where weight i is the weight we wish to assign to that rating, which is established according to the distance between movie A and also its ith nearest neighbor, and rating i is the rating that movie A received from its ith nearest neighbor.

A common method for creating a recommendation system is the K-Nearest Neighbors (KNN) approach. A sluggish, non-parametric classification algorithm, the KNN approach trains the dataset whenever a prediction is required. The Psychometric or cosine similarity measurements are used in the KNN technique to identify the correlation users or items.

$$Cosine\ Similarity: sim(i,j) = \frac{\sum_{u=1}^{m} r_{ui} r_{uj}}{\sqrt{\sum_{u=1}^{m} r_{ui}^2 \sum_{u=1}^{m} r_{uj}^2}}$$

Where rxy is the user x-provided rating for item y.

https://doi.org/10.56294/piii2025376

The model's performance is dependent on determining the most optimal value of k; too large a value would be too prohibitively costly and support vector regression, resulting in poor prediction performance. A value for 'k' that is too small results in premature convergence and poor performance on different datasets. To figure out the ideal size of k, we graph the accuracies for a reasonable period of k and select the optimum k as the one that produces the lowest error rate.

Following the specification of k, we compute the expected overall score for target consumer I on item j using their arithmetic mean.

$$\widehat{r_{pq}} = \overline{r_p} + \frac{\sum_r sim(q,s)(r_{ps} - \overline{r_s})}{\sum_s |sim(p,s)|}$$

Where (r_p)¯is the mean rating of user p.
Like this , the KNN model performance with the different datasets to optimise.

B. Naïve Bayes Model (NB):

One of the supervised learning technique is known as naive Bayes algorithm and it works on the principle of Bayes Theorem.

It is mostly involved in deep learning with a large training set. Being a classification algorithm, it makes predictions on the probability that an item will occur.

When dealing with large amounts of data, the naive Bayesian classification technique has highly accurate results, and the algorithm is primarily used to predict unrated data. The trouble of statistical likelihood classification is the inspiration for Naive Bayes.

The items to be classified are chosen to give under the condition that the subscriber feature items are satisfied, and even the probability of an event of every segment there under affliction of that kind of event that occurs is obtained, then in order to achieve the classification items are classified through into categories with a high possibility of people who seek, and are also used as the result of prediction.

For example, based on historical data of a specific user watching a movie in a movie theatre or on the Internet, it is discovered that five movies appease the circumstance there under presumption of accomplishing adverse parameters of something like the user, and the first movie has the highest probability of an event, recommend the first movie with the highest chance of occurring to this destination use, depending on the Naive Bayesian algorithm.

In the large number theorem, if the sample data are independent of each other and susceptible to the identical allocation of conditional probability events, the recurrence of the happening can sometimes be used rather than the probability of its occurrence.

Furthermore, under the circumstance of naive Bayesian theory, that seems to be, the quantities amongst each feature of the specimen are irrespective of each other.

The challenge is focused on calculating the conditional probability P(A|B). Because the random sample data A does not comprise a single pendant there in large dataset, the whole report includes the Movie reviews statistical model, A exemplifies the user's relevant data, and indeed the user's necessary data includes not only the name character trait, but also data such as gender, age, employment, and favorite movie type, that seems to be, A embodies a plurality of data attributes and signifies a user's attribute vector. In order to calculate the prediction error, the sample must contain all possible permutations of characteristics. If there are no attribute criteria that are independent of one another, all correlation values are zero.

Individual attribute values are independent of one another in this formula.

$$P(p|q) = \frac{P(p)P(q|r)}{P(q)} = \frac{P(p)}{P(q)} \prod_{i=1}^{d} P(q_i|p)$$

https://doi.org/10.56294/piii2025376

- Performance of Evaluation Index:

A variety of recommendation algorithms, such as search engine recommendation, collaborative filtering-based selection, or naive Bayes-based recommendation system employing ML, can be applied in the customized recommendation system. The performance of the network will not be the same for the classification method implemented in various algorithms, as will the recommended results and the system operating efficiency.

The system's performance must then be analyzed in order to pick the suggested algorithm that is appropriate for the system, so considerably improving the efficiency and accuracy of the suggestion. In this study, the system's performance is primarily assessed using four evaluation indices: Accuracy, Precision, Recall Rate, and F1-score.

The two-dimensional multiple regression is the most generally used assessment model prediction ability prior to the evaluation index computation, in accordance with the general approach of data mining theory.

Actual Class and The Prediction Class:

	POSITIVE	**NEGATIVE**
POSITIVE	True Positive (TP)	False Negative (FN)
NEGATIVE	False Positive (FP)	True Negative (TN)

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

Negative Prediction Value = $\frac{TN}{(TN+FN)}$

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

Specificity = $\frac{TN}{(TN+FP)}$

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

Positive and negative categorization accuracy did not attain comparable levels, but instead differed in comparison with the original research. As a result, the average categorization accuracy suffers. Even with the café and feedback review that serves as a specific topic in this assertion, when viewpoint analysis is executed using a conventional supervised Machine learning algorithm such as SVM and NB, the mean approximate rate decreases by demonstrating a variation classification. Thus, in this thesis, it enhanced NB algorithms are presented to reduce the gap for the approximation rate of classifications and its mean.

C. Support Vector Machine (SVM):

SVMs are a sort of Supervised Machine Learning technique that may be used for classification and regression applications. In the case of movie recommendation, an SVM might be trained to anticipate whether a user would appreciate a specific movie based on factors like as genre, cast, or director.

The SVM used to train the data for the Movie Recommendation System. This training procedure involves tweaking SVM parameters such as kernel type and regularization term to attain the best

https://doi.org/10.56294/piii2025376

efficiency on training data. Once trained, the SVM could be utilized to determine whether a new, previously unseen movie would be enjoyed by a certain user.

The "kernel technique," which entails transferring the user input data so that it becomes linearly separable, is one method for mathematically representing an SVM. This is accomplished by the use of a kernel function, which might be a linear separable. The kernel used will be determined by the characteristics of the data and the intended model complexity.

Assume we do have dataset of movies, each of which is represented by a feature vector x Rn, where n indicates the number of variables (e.g., genre, cast, director). We also include the binary label y 0-1 that indicates if the user enjoyed or hated each movie. Following that, the SVM model is established as follows:

$$f(x) = w\ \hat{}Tx + b$$

where w Rn denotes the weight matrix and b R denotes the bias term. The SVM searches the higher dimensional space hyperplane that best separates positive and negative samples, as specified by the labels p.

We may use a technique called "big margin classification" to optimize the SVM model, which entails maximizing the range in between hyperplane and the closest instances (called the "support vectors"). This is accomplished by dealing with the following optimization method:

Minimize ||*weight*||^2 subject p *a (w ^T a x + b) ≥ 1* for all x

where ||weight|| is the weight vector's Euclidean rule and a x is the multidimensional array for the a-th movie. The linearization component refers to the desired function ||weight||2, which assists in avoiding overfitting by punishing models with excessive weights.

After training the SVM, we can utilize it to predict whether a new movie x new would be liked by a certain user by computing the function f(x new). If f(x new) is more than zero, we anticipate that the user will enjoy the film; if f(x new) is less than zero, we estimate that perhaps the user will despise the film.

Hierarchy Method for Recommendation using SVM:
The suggested method is separated into four steps:

- Create a One-Class SVM model:

Construct a 1-Class SVM user preferences schema for user utilising webpages set Q = {q1, q2, q3, …qz} that have already been read. Try the optimization method below:

$$min\frac{1}{2}\|w_u\|^2 + \frac{1}{vz}\sum_{i=1}^{z}\varepsilon_i - \rho$$
$$s.t\langle w_{u,}\emptyset_{q_i}\rangle \geq \rho - \varepsilon_i, \varepsilon_i \geq 0, i = 1,2, \ldots..,$$

Wu as the vector of attributes for the user.

- Determining the similarity in between the user model and each domain model:

Websites in the domain are represented as P(i) = {p1(i), p2(i),…..pi(i)}. And l1+l2+l3+….+lk = n. n is the entire websites we have. Let's say that we having k domains and subdomains. The domains collection can indeed be written as D = {d1, d2, d3, ….dk} Create a One-Class SVM domain model on every domain. Try the optimization problem below:

$$min\frac{1}{2}\left\|w_{d_i}\right\|^2+\frac{1}{vl_i}\sum_{j=1}^{l_i}\varepsilon_j-\rho$$

$$s.t\left\langle w_{d_i},\emptyset_{p_{t_i}^{(i)}}\right\rangle\geq\rho-\varepsilon_j,\varepsilon_j\geq0,j=1,2,\ldots\ldots,z$$

- Domains with similarities and compute the relevance:

The identity in between domain Di and user might be measured as:

$$sim(d_i,u)=\frac{w_{d_i}.w_u}{\left\|w_{d_i}\right\|\left\|w_u\right\|}$$

- Choosing webpages with identical and for the user preferences:

Sort the user preferences similarities in an order. Choose the top number of domains that most closely match the user's interests. Sort the commonalities in the domain's websites and user interest. The similarity may be computed as follows:

$$sim(p_i,u)=<w_u,\emptyset_{p_i}>-p_u$$

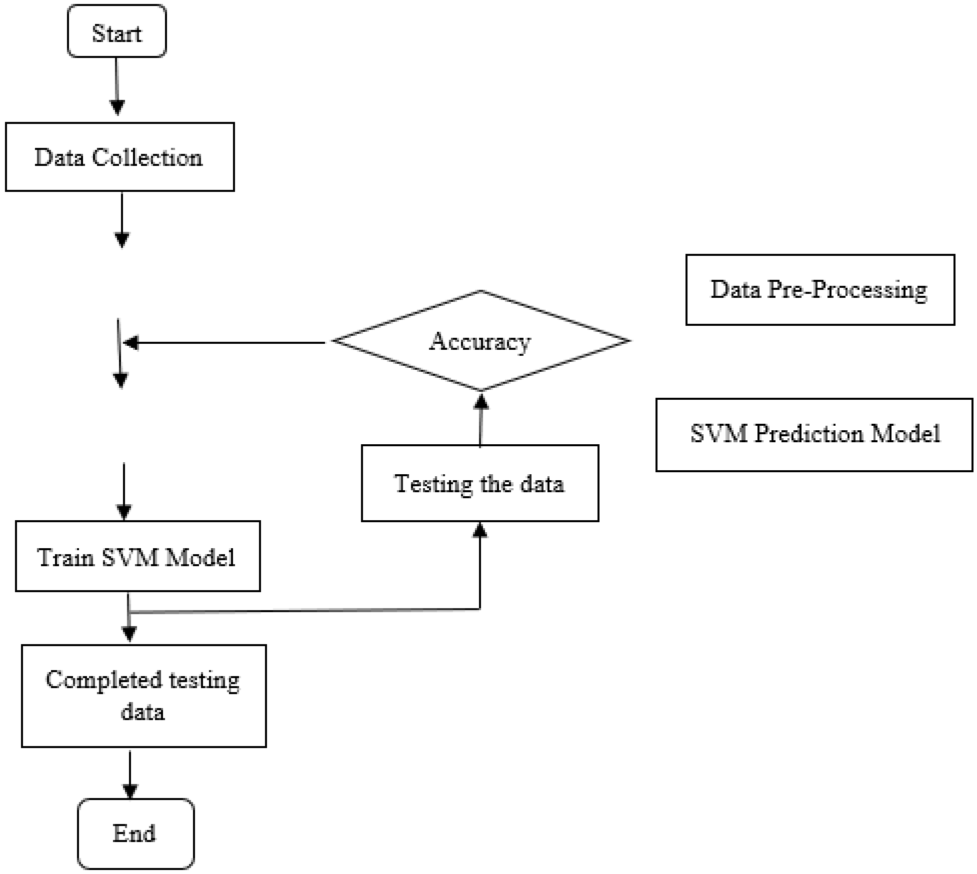
Then, in order of importance of resemblance, sort these webpages. Choose the best number of webpages and promote them to the user.

FINDINGS AND RESULTS

The motive of this paper is to identify the most reliable algorithm to integrate with our models so that our outcome to be optimized solution. We have conducted a research regarding the work procedure and limitations for each algorithm for which we considered, from various sources and gained various insights od training data sets for the algorithms of Machine learning.

There is a process to assemble the information of SVM algorithm for testing the data. First, data is collected and process it after processing train the SVM model and test the data. If error arises when training the model again find the error and test the data for retrieving better results the algorithm.

The flow chart describes a bout the KNN, NB and SVM algorithms which the training datasets gives the outputs in a format to give the Optimized solution for the Movie Recommendation System as illustrated in the fig[1].

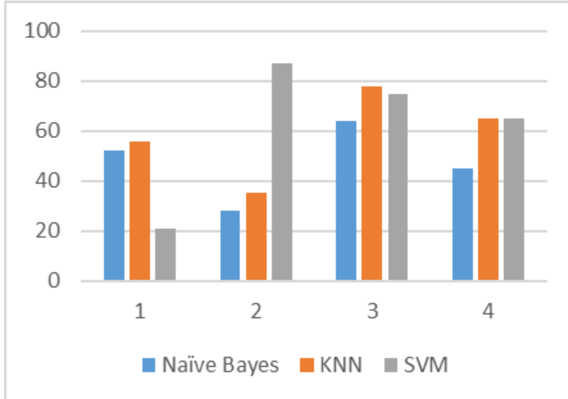


Fig[1]: Flow Chart

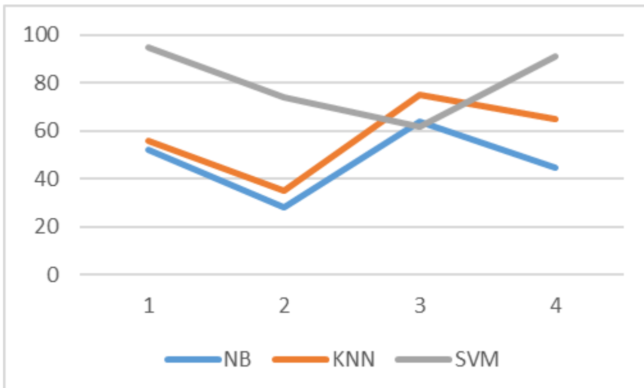
The main goal for the article is that SVM is the viability of Movie Recommendation System for the best algorithm. A useful method for the Movie Recommendation System is Support Vector Machine which has the highest accuracy among the K-Nearest Neighbor Algorithm, Naïve Bayes Algorithm and the Support Vector Machine Algorithm of having of 96.78%. Which the table and the graph shows about the accuracy, suspicious, legitimate, phishy and average that ultimately describes about the Supervised learning Models of Machine learning as shown in the fig[2] and fig[3].

S. No:	Values	NB	KNN	SVM
1	Suspicious	52%	56%	21%
2	Legitimate	28%	35%	87%
3	Phishy	64%	78%	75%
4	Average	45%	65%	65%

Fig[2]: Tabular Form



Fig[3]: Graph



Fig[4]: Line Graph

CONCLUSIONS

This paper is split up into two basic categories. One focuses on the movie recommendation system, whereas the other concentrates on sentiment analysis. The paper examines both systems in depth and draws some noteworthy implications. The Cosine Similarity algorithm was utilized in the Movie Recommendation System to propose relevant of the movie posting by the user based on several parameters such as the genre of the movie, overview, the cast, and the ratings provided to the movie. Even after multiple testing, Cosine Similarity produced reasonable findings and was fairly accurate in recommending movies.

It is challenging to determine definitively which of the Support Vector Machines (SVM), K- Nearest Neighbors (KNN), or Naive Bayes (NB) algorithms is the best choice for developing a movie recommendation system, because the performance of these algorithms can vary depending on a variety of factors such as the nature of the data, the specific requirements of the recommendation task, and the computational resources available.

A recommendation including the size and complexity of the dataset, the unique needs of the recommended task, and the computational resources available. It is critical to thoroughly assess the performance of several algorithms on your unique dataset to discover which one performs best.

Having said that, SVMs may be a viable choice for movie suggestion in some cases. If you have a huge and complicated dataset with numerous characteristics, for example, an SVM may be able to train a more accurate decision boundary than a simpler method like KNN or NB. SVMs, on the other hand, may

https://doi.org/10.56294/piii2025376

be more computationally intensive and need more time and resources to train, requiring you to evaluate these factors against the requirements of Movie recommendation System.
Future Work:

A movie recommendation system might be improved by using a broader range of data sources. It may be improved by tailoring suggestions further. There are several algorithms that might be used to generate suggestions. Collaborative filtering, matrix factorization, and deep learning are three of the most successful ways. The recommendation system might be improved by adding user input and making it more user-friendly by upgrading the user interface.

REFERENCES

1. Katarya, R., & Verma, O. P. (2017). An effective collaborative movie recommender system with cuckoo search. Egyptian Informatics Journal, 18(2), 105-112.
2. Nassar, N., Jafar, A., & Rahhal, Y. (2020). A novel deep multi-criteria collaborative filtering model for recommendation system. Knowledge-Based Systems, 187, 104811.
3. Pavitha, N., Pungliya, V., Raut, A., Bhonsle, R., Purohit, A., Patel, A., & Shashidhar, R. (2022). Movie Recommendation and Sentiment Analysis Using Machine Learning. Global Transitions Proceedings.
4. Zhang, C., Duan, X., Liu, F., Li, X., & Liu, S. (2022). Three-way naive Bayesian collaborative filtering recommendation model for smart city. Sustainable Cities and Society, 76, 103373.
5. Kang, H., Yoo, S. J., & Han, D. (2012). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews.
6. Expert Systems with Applications, 39(5), 6000-6010.
7. Gonzales, R. M. D., & Hargreaves, C. A. (2022). How can we use artificial intelligence for stock recommendation and risk management? A proposed decision support system. International Journal of Information Management Data Insights, 2(2), 100130.
8. Wang, Z., Ren, G., Sun, M., Ren, J., & Jin, J. S. (2015). Ranking highlight level of movie clips: A template based adaptive kernel SVM method. Journal of Visual Languages & Computing, 27, 49-59..
9. Min, S. H., & Han, I. (2005, July). Recommender systems using support vector machines. In International Conference on Web Engineering (pp. 387-393). Springer, Berlin, Heidelberg.
10. Li, H., Cui, J., Shen, B., & Ma, J. (2016). An intelligent movie recommendation system through group-level sentiment analysis in microblogs. Neurocomputing, 210, 164-173.
11. Cui, L., & Shi, Y. (2014). A method based on one-class SVM for news recommendation. Procedia Computer Science, 31, 281-290.
12. Wu, C. S. M., Garg, D., & Bhandary, U. (2018, November). Movie recommendation system using collaborative filtering. In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) (pp. 11-15). IEEE.
13. Phorasim, P., & Yu, L. (2017). Movies recommendation system using collaborative filtering and k-means. International Journal of Advanced Computer Research, 7(29), 52.
14. Ruotsalo, T., Weber, S., & Gajos, K. Z. (2022). Active tag recommendation for interactive entity search: Interaction effectiveness and retrieval performance. Information Processing & Management, 59(2), 102856.
15. Marjanović, M., Kovačević, M., Bajat, B., & Voženílek, V. (2011). Landslide susceptibility assessment using SVM machine learning algorithm. Engineering Geology, 123(3), 225-234.
16. Pradhan, A. (2012). Support vector machine-a survey. International Journal of Emerging Technology and Advanced Engineering, 2(8), 82-85.

https://doi.org/10.56294/piii2025376

17. Feng, W., Sun, J., Zhang, L., Cao, C., & Yang, Q. (2016, December). A support vector machine based naive Bayes algorithm for spam filtering. In 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC) (pp. 1-8). IEEE.
18. Peling, I. B. A., Arnawan, I. N., Arthawan, I. P. A., & Janardana, I. G. N. (2017). Implementation of Data Mining To Predict Period of Students Study Using Naive Bayes Algorithm. Int. J. Eng. Emerg. Technol, 2(1), 53.
19. Rahat, A. M., Kahir, A., & Masum, A. K. M. (2019, November). Comparison of Naive Bayes and SVM Algorithm based on sentiment analysis using review dataset. In 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART) (pp. 266-270). IEEE.
20. Pop, I. (2006). An approach of the Naive Bayes classifier for the document classification. General Mathematics, 14(4), 135-138.
21. Sathyadevan, S., Sarath, P. R., Athira, U., & Anjana, V. (2014, August). Improved document classification through enhanced Naive Bayes algorithm. In 2014 International Conference on Data Science & Engineering (ICDSE) (pp. 100-104). IEEE.

FINANCING
None.

CONFLICT OF INTEREST
None.