eISSN: 2976-1190

Journal of Engineering and Sciences

Vol. 4 Issue 1 May 30, 2025

*DOI:*10.3126/jes2.v4i1.70138

Journal homepage: [Journal of Engineering and Sciences (nepjol.info)]

Publisher: Research Management Cell, Pashchimanchal Campus, Lamachour, Pokhara

Sentiment Analysis of IMDb Movie Reviews Using SVM and Naive Bayes Classifier

Dhurba Subedi[1], Nabin Lamichhane[1*], Nabaraj Subedi[1]

[1] *Department of Electronics and Computer Engineering, IOE, Pashchimanchal Campus, Nepal*

**Corresponding author: babunabin@wrc.edu.np*

(Manuscript Received: 24/07/2024; Revised: 21/04/2025; Accepted: 28/04/2025)

Abstract

Sentiment analysis is a powerful tool for understanding public opinion, especially in the entertainment industry. Opinion in the form of text reviews plays a significant role in the success of a movie. Text-based data analysis is extensively used to recognize opinion sentiments. Achieving the proper sentiment for classification is crucial for both consumers and organizations. Handling large and complex data can pose more challenges during classification. This quantitative research is focused on classifying the sentiments of the IMDb movie review dataset using supervised Machine Learning (ML) models such as Naive Bayes (NB) and Support Vector Machines (SVM). The sentiments were classified as positive and negative to identify best-fit models for the large-scale review classification. 50,000 IMDb movie reviews went through preprocessing and feature extraction to transform unprocessed text input into numerical form, deploying Term Frequency-Inverse Document Frequency (TF-IDF). Eventually, the split between positive and negative ratings was even distributed. SVM and NB models were trained and assessed on various train-test splits to ensure robust model evaluation. Precision, Recall, and F1 Score were performance metrics applied to calculate the efficiency of models. Based on the report, the SVM model outperformed Naive Bayes regarding accuracy. SVM achieved an average accuracy of 88%, while Naive Bayes achieved 85%. This research can significantly aid filmmakers in understanding viewer preferences, which is crucial for market strategy and content creation.

Keywords: IMDb Movie Review; Naive Bayes; SVM; TF-IDF

1. Introduction

As the growth of user-generated content on the internet significantly increases, sentiment analysis has become a vital technique for understanding public opinion. The biggest platform, social media, tends to be open source and available to everyone. Different ideas are found in extreme quantities on these networking sites. People share what they feel, ideas, thoughts, and feelings through these platforms. Events regarding politics, current affairs, natural disasters, terrorism, injustice, economy, and other forthcoming issues happen around us (Sailunaz et al., 2018). Sentiment analysis is an application of Natural Language Processing (NLP) that works with data to discover and extract information. For the betterment of organizations, this technology helps to gather information about public sentiment and emotions on different subjects. Review sites such as IMDb have gained popularity within the film industry as places where viewers may voice their thoughts and offer criticism. These reviews are valuable gauges of public opinion and have the power to affect future content production, marketing tactics, and the financial performance of motion pictures. However, manually analyzing this data is impractical due to the massive volume of reviews published daily (Sailunaz et al., 2018).

The wide range of movie reviews on sites like IMDb has increased, making it more challenging to manually evaluate and classify the sentiments. Different automated solutions are required for consumers, marketers, and filmmakers to analyze the audience responses. The main problem is correctly classifying the IMDb movie reviews as positive or negative using machine learning algorithms. Different challenges lie for projects handling large volumes of textual data and developing models that can perform well in unseen reviews (Sundaram et al., 2021).

Different stages encompass sentiment analysis, including review preprocessing, feature extraction as numerical representation, and classification. During the preprocessing, the raw review cleans unnecessary information such as stop words, special characters like tags, HTML tags, etc. Now the processed text goes to feature extraction, or can be called the vectorization of the text using the technique of TF-IDF (Tabassum et al., 2020). The numerical features obtained are then categorized with machine learning classifiers, including SVM and Naive Bayes (Rahat et al., 2019).

2. Related Works

Rahat et al. (2019) compared the effectiveness of SVM and Naive Bayes algorithms for sentiment analysis using a dataset of 10,000 airline reviews. This research found that the SVM is more appropriate relative to Naive Bayes across several metrics, achieving an accuracy of almost 83% compared to Naive Bayes' 77%. Poudel et al. (2022) focused on classifying the tweets related to Nepal Tourism or not. The research applied the two classification models, Naive Bayes and SVM classifier. The result indicated that the performance of the SVM algorithm is more appropriate than Naive Bayes, with an accuracy of 91% and 81%. The study also showed that the algorithms perform well for the domain-specific dataset. Tan et al. (2023) surveyed sentiment analysis, examining several techniques such as lexicon-based, machine learning, and deep learning. The study compiled the benefits and drawbacks of each technique, explaining how deep learning models like BERT assist in identifying the complex patterns and information in text. Hemakala and Santhoshkumar (2018) used the advanced classification method proposed for analyzing Twitter data related to airline services. The authors utilized an ensemble technique combining multiple classifiers, such as SVM, Naive Bayes, and Decision Trees. Their ensemble method improved classification performance, providing 84.5% accuracy, outperforming the performance of individual classifiers like SVM or Naive Bayes alone. Dadgar et al. (2016) proposed a novel approach for news classification using text mining techniques based on TF-IDF and Support Vector Machine (SVM). Their approach aimed to enhance the classification performance of news articles and demonstrated that SVM, when coupled with TF-IDF for feature extraction, achieved promising results in terms of accuracy and computational efficiency. Their method outperformed other traditional machine learning algorithms in the classification of news articles, with SVM achieving better accuracy and precision. Vanaja and Belwal (2018) investigated sentiment analysis focused on specific aspects of e-commerce data, which looks at sentiments associated with particular characteristics of a product or service applying the two classification models Naive Bayes and SVM algorithms for the classification of the data. The findings indicated that the efficiency of the Naive Bayes algorithm is more accurate than the SVM achieving accuracy of 90% and 83% respectively.

Saad (2020) conducted opinion mining on Twitter data related to six US airlines, utilizing machine learning techniques to classify tweets into positive, negative, and neutral categories. Among the classifiers tested, Support Vector Machine (SVM) achieved the highest accuracy of 83.31%, outperforming others like Logistic Regression, Naïve Bayes, and Decision Trees. The study employed pre-processing, feature extraction, and the Bag of Words model, followed by K-Fold Cross-Validation to evaluate performance. Chakriswaran et al. (2019) surveyed the studies in the area of emotion AI-driven sentiment analysis. They compared the performance on each domain as ontology-based and lexicon-based, with some machine learning models for the sentiment analysis. Their study found that aspect-based ontology, term frequency approach, and Support Vector Machine-based approach achieved 83%, 85%, and 90% accuracy on ontology-based, lexicon-based, and machine learning models, respectively. Sentiment analysis was conducted on Twitter data related to the 2019 Indonesian presidential candidates using the Naïve Bayes algorithm (Wongkar & Angdresey, 2019). Their study found that the Jokowi-Ma'ruf Amin pair (54.55%) had a slightly lower negative sentiment than the Prabowo-Sandiaga pair (55.68%), with an overall accuracy of the model as 80.1%. Additionally, they compared Naïve Bayes, SVM, and K-NN classifiers, with Naïve Bayes achieving the highest accuracy of 75.58%.

Madhoushi et al. (2015) found that machine learning methods, including supervised, unsupervised, and hybrid techniques, are most commonly used for sentiment analysis tasks like classification and aspect extraction. However, they also identified key challenges, including domain adaptability, insufficient labeled data, and difficulty handling complex sentences and non-English languages. Novilia et al. (2022) found that SVM outperformed Naïve Bayes in smartphone sentiment analysis, achieving higher f-measure values using 6 and 18 n-grams features. The combination of SVM and Query Expansion Ranking (QER) yielded the highest accuracy, with an f-measure value of 0.91, indicating significant improvement in sentiment analysis performance.

Rana and Singh (2016) compared sentiment orientation using SVM, Naïve Bayes, and Synthetic Words approaches on movie reviews, finding that Linear SVM achieved the highest accuracy. Their analysis also revealed that drama reviews yielded the highest accuracy rate among the tested genres. Kristiyanti et al. (2018) compared the performance of SVM and Naïve Bayes Classifier (NBC) for sentiment analysis on 800 Indonesian tweets about West Java governor candidates. The study found that NBC outperformed SVM, achieving 94% accuracy for the Deddy Mizwar-Dedi Mulyadi candidate pair, demonstrating its superior effectiveness for this dataset. Ligthart et al. (2021) conducted a tertiary study on sentiment analysis, synthesizing findings from secondary studies to offer a comprehensive overview of key topics, features, algorithms, and datasets. Their analysis of 112 recent deep learning-based sentiment analysis papers revealed that LSTM and CNN are the most commonly used algorithms in the field.

Previously, sentiment analysis research mainly used smaller, domain-specific datasets and concentrated on particular areas, such as social media, travel, and e-commerce. With less focus on using these techniques on bigger, more varied datasets, like movie reviews, this research mainly investigated the efficacy of algorithms like SVM and Naive Bayes in specific circumstances. The use of aspect-level sentiment analysis, primarily limited to product or service reviews, represents another gap. Additionally, prior research frequently concentrates on a single platform or review type without comparing sentiment analysis across different textual data sources. By using sentiment analysis techniques on a sizable and varied dataset of movie reviews, assessing how well various classifiers perform in this novel setting, and investigating the possibility of aspect-level sentiment analysis to understand user perceptions of particular aspects of films, this study aims to close these gaps.

3. Methodology

Different steps are taken to classify the sentiments of IMDb movie reviews as positive or negative, such as data gathering, preprocessing, feature extraction, and machine learning algorithms like SVM and Naive Bayes. Figure 1 shows the method used for sentiment analysis.

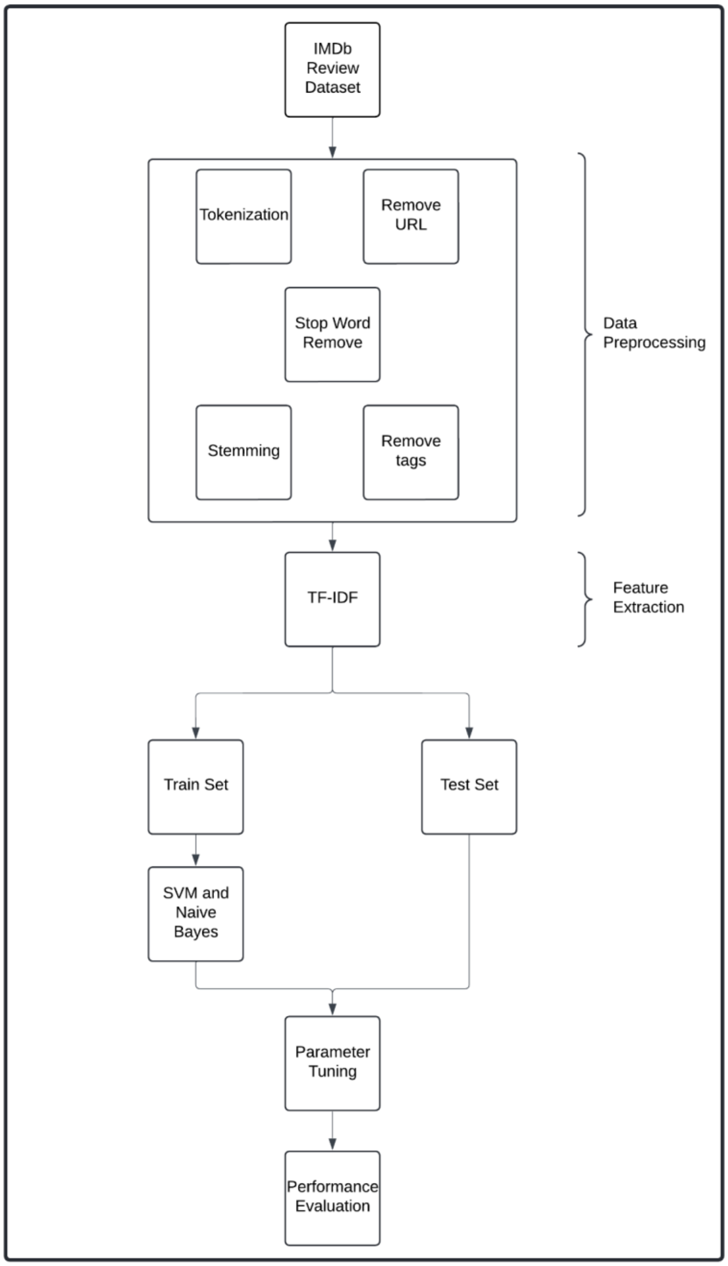


Figure 1. Sentiment Analysis Methodology

3.1. Data gathering

The IMDb movie review dataset collected from the Kaggle source is stored in a data frame (N, n.d.). The data contains 50,000 reviews. As seen in Figure 2, the gathered data is a balanced set with a balanced number of positive and negative reviews, perfect for training the machine learning models.

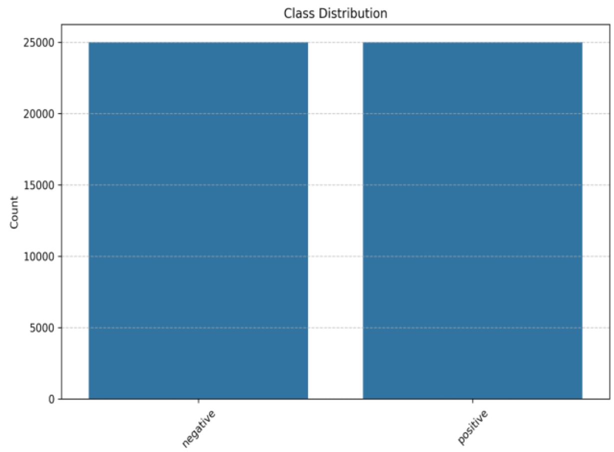


Figure 2. Class Distribution of Movie Reviews

3.2. Data Pre-processing

The preprocessing of the reviews is necessary to make them noise-free and ready for applying the algorithm. The text column will be converted to lowercase; tags like HTML tags are removed; different stop words are removed that do not have significance; stemming is performed to reduce the words to their root form. The sample of the difference between the original review and the pre-processed review is shown in Table 1.

Table 1. Data After Pre-Processed

Original Review	Pre-Processed Review
If you like original gut-wrenching laughter, you will like this movie. If you are young or old then you will love this movie, hell even my mom liked it. Great Camp!!!	like origin gut wrench laughter like movi young old love movi hell even mom like great camp
The Buddy Holly Story is a great biography with a super performance from Gary Busey. Busey did his own singing for this film and he does a great job.	buddi holli stori great biographi super perform gari busey busey sing film great job
I don't know how anyone could hate this movie. It is so funny. It took a unique mind to come up with this storyline. It's not your typical alien movie. These aliens are so stupid and confused. You need to rent it at least once.	dont know anyon could hate movi funni took uniqu mind come storylin typic alien movi alien stupid confus need rent least

This process transforms text data into numerical features using methods like Term Frequency-Inverse Document Frequency (TF-IDF). The TF-IDF converts text to numerical features and helps identify how important a word is to that document.

Term Frequency (TF): It is calculated as the frequency of a word present in the document divided by the total word count. Term frequency can be defined as:

$$TF = \frac{Frequency\ of\ word\ present\ in\ the\ Document}{Total\ word\ count\ in\ that\ document} \text{ ---------------------(i)}$$

TF gives a higher score to terms that appear frequently in the specific document, indicating which term is important to the content of the document.
Inverse Document Frequency (IDF): IDF of each word is computed as the logarithm of the ratio of the total number of documents to the number of documents containing that word.

$$IDF = \log\left(\frac{Total\ number\ of\ documents}{Documents\ containing\ word\ W}\right) \text{ ------------------------- (ii)}$$

IDF gives a higher score to terms that are unique to fewer documents, indicating that the more unique terms provide more information.
The final TF-IDF score is:

$$TF\text{-}IDF = TF * IDF \text{ ---------------------------------------(iii)}$$

The feature vectors for each document are constructed on the TF-IDF score. Each vector represents a document in the feature space, considering each dimension represents a term. Higher dimensions are computationally expensive for the model.

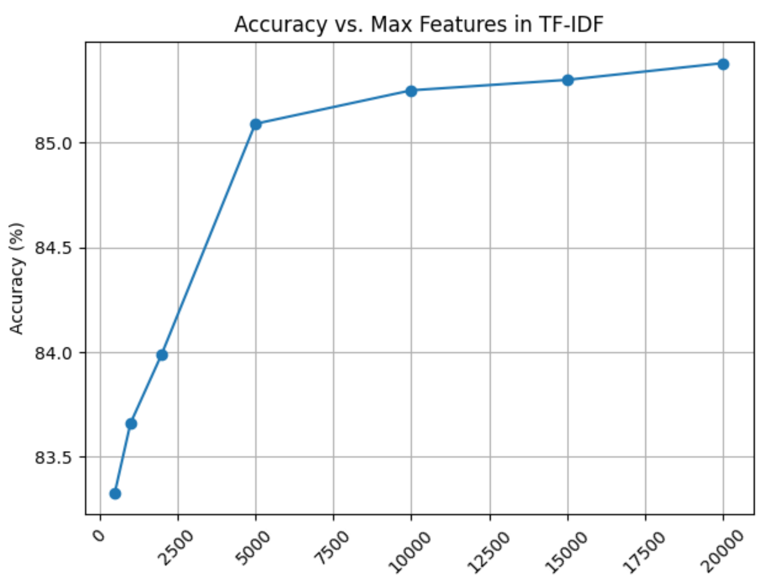


Figure 3. Accuracy vs Max Features

As shown in Figure 3, the extraction of features is compared to the model's efficiency. We observed that the efficiency increased rapidly from 1000 to 5000 features. However, after 5000 features, it increased slightly, indicating that additional features contribute less to model performance. To preserve the accuracy of the model and minimize computational expenses, 5000 maximum features is chosen for the model.

The sentiments are also encoded as 1 for positive and 0 for negative. This binary encoding allows machine learning models to be trained more easily and learn and predict sentiments based on the feature vectors.

3.4. Sentiment Analysis and ML algorithm

After the feature extraction, the processed features are split for training and testing data and algorithms such as Naive Bayes and SVM. The outcomes from models are then evaluated and compared with the performance of the two algorithms.
Multinomial Naive Bayes was implemented for classification, a supervised machine learning approach using a probabilistic classifier based on the Bayes theorem. This approach is particularly designed for text documents. It classifies the data based on features and calculates its probability.

$$P(A/B) = \frac{P(B/A).P(A)}{P(B)} \text{------------------------------(iv)}$$

where P(A/B) is posterior probability of class A given features B, P(B/A) is the likelihood, P(B) is probability of class B.
For a more robust classification, we also employ the SVM algorithm. It is a supervised machine learning algorithm that finds the ideal hyperplane to isolate the various classes in the element space. It finds the hyperplane that maximizes the margin. The original training data is transformed into a higher dimension using nonlinear mapping. It finds the linearly optimal separating hyperplane inside this new dimension. The distance between the nearest data point from each class and the hyperplane is known as the margin.
In this scenario, SVM uses kernel functions to convert the original training data into a higher-dimensional space, which allows it to identify and separate complex, nonlinear relationships within the data. This transformation enhances SVM's classification performance by finding the optimal separating hyperplane for the different parameters.

3.5. Parameter Tuning

To maximize the effectiveness of the classification models in machine learning, hyperparameter tuning is a crucial step. The accuracy with which SVM and Naive Bayes algorithms can classify movie reviews depends on an array of parameters that need to be adjusted. GridSearchCV was used to optimize these hyperparameters. To determine the optimal model configuration, these tools conduct an exhaustive search across a predefined grid of hyperparameters. GridSearchCV brute-forces the hyperparameter space, where all the combinations of hyperparameters are tested. For every combination, it performs k-fold cross-validation. It splits training data into k subsets (folds), trains the model on k-1 folds, and evaluates the remaining folds. The process is iterated k times, with a different fold used for validation each time. The best combination is selected based on the average cross-validation score.

4. Results & Discussion

Sentiment classification is done for the IMDb review dataset using machine learning algorithms such as SVM and Naive Bayes. This section outlines the findings of the classification task, discussing key aspects such as hyperparameter tuning, cross-validation performance, model accuracy, confusion matrix analysis, and evaluation metrics like recall, precision, and F1-score.

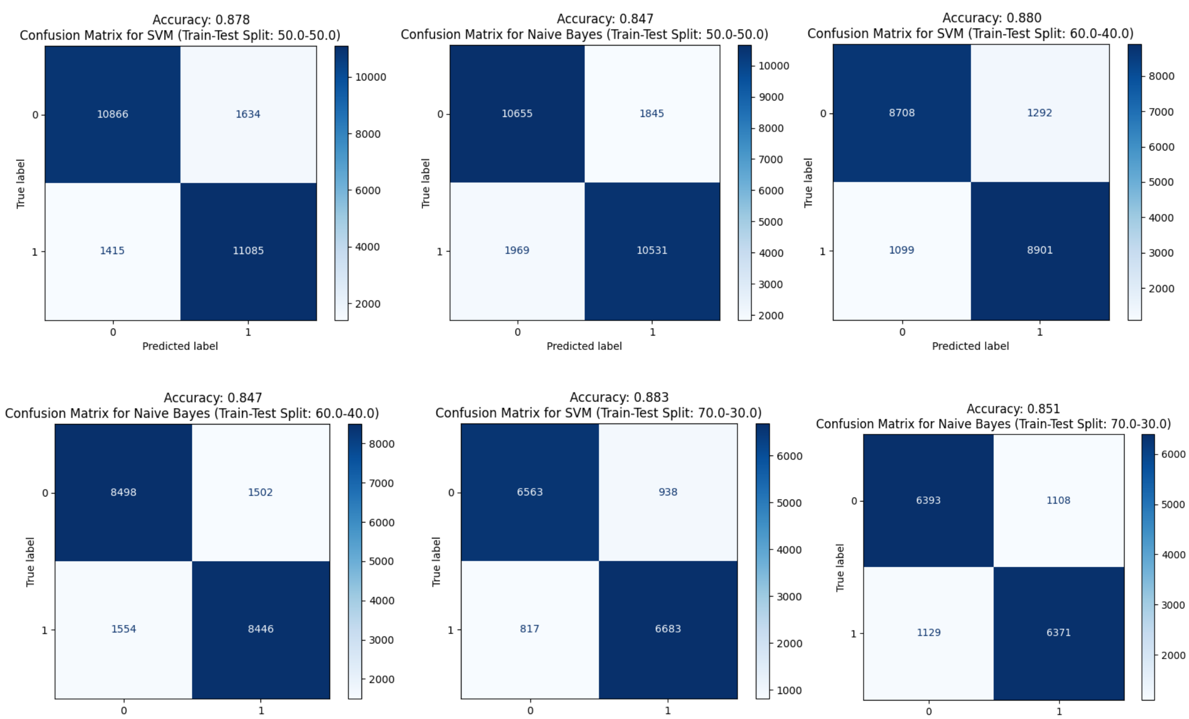
4.1. Hyperparameter Tuning and Cross-Validation

Both SVM and Naive Bayes models used hyperparameter tuning using GridSearchCV with k-fold cross-validation to optimize their performance. For SVM, hyperparameters such as the regularization parameter (C) and kernel type were fine-tuned. Similarly, for Naive Bayes, the smoothing parameter alpha was optimized. Cross-validation ensured that the specific partitioning of the training and test data did not bias the model's performance. The best-performing hyperparameters were determined based on average cross-validation scores. The optimal parameter for algorithms is:

- **SVM**:
    - Kernel: “RBF”
    - C: “1”
- **Naive Bayes**:
    - alpha: “5”

4.2. Confusion Matrix

The optimized model is tested on the test set after selecting the hyperparameters. Different tests have been done while splitting the dataset into a training set and a test set. The confusion matrix provides a number for true positives, true negatives, false positives, and false negatives. The tests have been done on the SVM and the Naive Bayes models, as shown below. Class “0” represents the negative sentiment and Class “1” represents the positive sentiment in the confusion matrix shown in Figure 4.



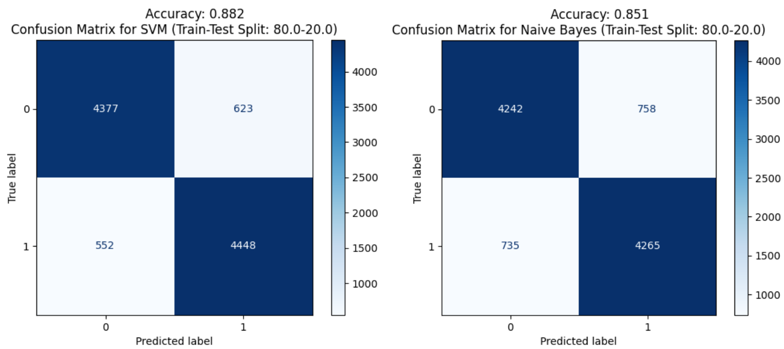


Figure 4. Confusion Matrix for Different Algorithms for Different Splits

As the train data increases from 50-50 to 80-20, we can see from the confusion matrix that the false positives and false negatives decrease. The model learns to identify more patterns within the data as it receives more examples of positive and negative labels as training data grows. With more training examples, the model can generalize better and learn to make more accurate predictions, leading to fewer false positives and false negatives. FP and FN can result from model complexity, noisy data, poor feature representation, and limitations in the model's assumptions. These errors can be decreased by using advanced algorithms, improving feature engineering, or optimizing hyperparameters.

4.3. Comparison of Results

Table 2. Accuracy Comparison of the Model

S.N.	**Train-Test Split**	**Accuracy (%)**			
		SVM		Naive Bayes	
		Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy
1	50-50	87.80	92.80	84.74	86.31
2	60-40	88.04	92.43	84.72	86.14
3	70-30	88.30	92.10	85.1	85.90
4	80-20	88.25	91.90	85.1	85.795

The accuracy of different models on different training and testing sets is shown in Table 2. SVM has an increase in accuracy on the test set and train set as the train set increases, reaching a highest accuracy of 88.30 % with a 70-30 split. The accuracy remains very stable across 70-30 and 80-20 splits. The gap between the test set accuracy and the train set accuracy goes on decreasing on an 80-20 split. This indicates that the 80-20 split has improved performance in reducing overfitting.

Naive Bayes increases in accuracy on the test set and train set as the train set increases, reaching a highest accuracy of 85.1 % with a 70-30 and 80-20 split. The gap between the test set accuracy and the train set accuracy decreases on an 80-20 split. This indicates that the 80-20 split has improved performance in reducing overfitting.

4.4. Evaluation Metrics: Precision, Recall, and F1-Score

Different metrics, such as Precision, Recall, and F1-Score, are evaluated on SVM and Naive Bayes algorithms to assess their performance across different train-test splits in the model. In Tables 3 and 4, Class 0 represents negative sentiments, and Class 1 represents positive sentiments.

Table 3. Evaluation For SVM

S.N.	**Train-Test Split**	**Precision**		**Recall**		**F1 Score**	
		Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
1	50-50	0.88	0.87	0.86	0.88	0.877	0.879
2	60-40	0.887	0.873	0.870	0.890	0.879	0.881
3	70-30	0.889	0.876	0.874	0.891	0.882	0.883
4	80-20	0.888	0.8771	0.875	0.889	0.881	0.883

Table 4. Evaluation For Naive Bayes

S.N.	**Train-Test Split**	**Precision**		**Recall**		**F1 Score**	
		Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
1	50-50	0.84	0.85	0.85	0.84	0.848	0.846
2	60-40	0.845	0.849	0.849	0.844	0.847	0.846
3	70-30	0.849	0.851	0.852	0.849	0.851	0.850
4	80-20	0.852	0.849	0.848	0.853	0.850	0.851

SVM outperforms Naive Bayes in most cases, with higher precision, recall, and F1 Scores for both classes. It maintains a better balance between the two classes, especially with larger training data (80-20 and 70-30 splits). Naive Bayes shows stable performance on different training splits.

The F1 score curve shown in Figure 5 outlines the balance between precision and recall across different thresholds for both the SVM and Naive Bayes models. The SVM model shows a slightly higher peak F1 score than Naive Bayes, denoting superior performance in achieving a balance between precision and recall. The Precision-Recall curve in Figure 6 reveals the trade-off between precision and recall for the SVM and Naive Bayes models. The SVM curve consistently shows higher precision at all recall levels than Naive Bayes. Based on the above curves, SVM performs better in both metrics, establishing it as a more reliable option for tasks that demand accurate classification.

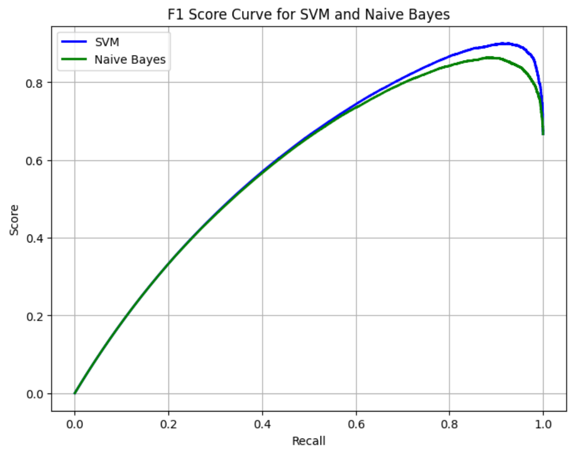


Figure 5. F1 Score Curve

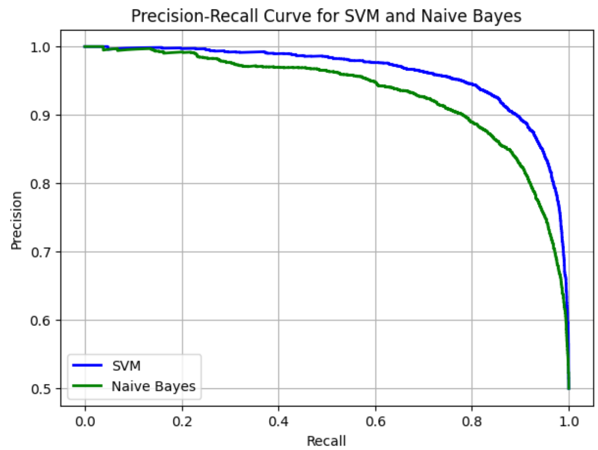


Figure 6. Precision-Recall Curve

5. Conclusion & Future Work

The above results indicate that SVM outperforms Naive Bayes on all evaluation metrics, including accuracy, recall, precision, and F1 score. One of the significant findings of this study is the relationship between the size of the training data and model performance. The study observed that increasing the training data ratio improves the performance of both models. This improvement is especially seen in SVM, where increasing data sets result in lower false positive and false negative rates. We generalized that in the small training ratio, the SVM provided signs of overfitting as the gap between them increased, but the performance improved as the gap decreased on larger splits of data. Naive Bayes showed more stable and consistent performance with a small gap between train and test accuracy, indicating less overfitting. When dealing with smaller datasets or overfitting is a concern, Naive Bayes is an excellent choice due to its stability, even though it cannot achieve the same level of accuracy as SVM.

Future studies should explore how both models perform under conditions of class imbalance, which is common in real-world datasets. The role of feature engineering can be expanded by incorporating additional techniques like word embeddings to capture semantic relationships between words, which may further enhance the model’s ability to classify sentiments accurately. Employing ensemble classifiers or deep learning approaches could generate higher classification accuracy.

References

[1] Chakriswaran, P., Vincent, D. R., Srinivasan, K., Sharma, V., Chang, C.-Y., & Reina, D. G. (2019). Emotion ai-driven sentiment analysis: A survey, future research directions, and open issues. *Applied Sciences*, *9*(24), 5462. https://doi.org/10.3390/app9245462

[2] Dadgar, S. M. H., Araghi, M. S., & Farahani, M. M. (2016). A novel text mining approach based on TF-IDF and Support Vector Machine for news classification. *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, 112–116.

https://doi.org/10.1109/icetech.2016.7569223

[3] Hemakala, T., & Santhoshkumar, S. (2018). Advanced Classification Method of Twitter Data using Sentiment Analysis for Airline Service. *International Journal of Computer Sciences and Engineering*, *6*(7), 331–335. https://doi.org/10.26438/ijcse/v6i7.331335

[4] Kristiyanti, D. A., Umam, A. H., Wahyudi, M., Amin, R., & Marlinda, L. (2018). Comparison of SVM & naïve bayes algorithm for sentiment analysis toward West Java governor candidate period 2018-2023 based on public opinion on twitter. *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, 1–6. https://doi.org/10.1109/citsm.2018.8674352

[5] Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Systematic reviews in sentiment analysis: A tertiary study. *Artificial Intelligence Review*, *54*(7), 4997–5053. https://doi.org/10.1007/s10462-021-09973-3

[6] Madhoushi, Z., Hamdan, A. R., & Zainudin, S. (2015). Sentiment analysis techniques in recent works. *2015 Science and Information Conference (SAI)*, 288–291. https://doi.org/10.1109/sai.2015.7237157

[7] N, L. (n.d.). *IMDB dataset of 50K movie reviews*. Kaggle. Retrieved September 2024, from https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[8] Novilia, G., Adriman, R., & Abidin, T. F. (2022). Comparison of classification methods using feature selection for smartphone sentiment analysis. *2022 International Conference on Electrical Engineering, Computer and Information Technology (ICEECIT)*, *XXX*, 44–48. https://doi.org/10.1109/iceecit55908.2022.10030703

[9] Poudel, B., Devkota, B., & Lamichhane, N. (2022). Automatic Identification of Nepalese Tourism Related Tweets using Machine Learning Algorithms. *Journal of Engineering and Sciences*, *1*(1), 37–45. https://doi.org/10.3126/jes2.v1i1.58398

[10] Rahat, A. M., Kahir, A., & Masum, A. K. M. (2019). Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset. *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*, 266–270. https://doi.org/10.1109/smart46866.2019.9117512

[11] Rana, S., & Singh, A. (2016). Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 106–111. https://doi.org/10.1109/ngct.2016.7877399

[12] Saad, A. I. (2020). Opinion mining on US airline twitter data USing machine learning techniques. *2020 16th International Computer Engineering Conference (ICENCO)*, 59–63. https://doi.org/10.1109/icenco49778.2020.9357390

[13] Sailunaz, K., Dhaliwal, M., Rokne, J., & Alhajj, R. (2018). Emotion detection from text and

speech: A survey. *Social Network Analysis and Mining*, *8*(1). https://doi.org/10.1007/s13278-018-0505-2

[14] Sundaram, V., Ahmed, S., Muqtadeer, S. A., & Ravinder Reddy, R. (2021). Emotion Analysis in Text using TF-IDF. *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 292–297. https://doi.org/10.1109/confluence51648.2021.9377159

[15] Tabassum, A., Dr. Rajendra, & R. Patil. (2020). A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing. *International Research Journal of Engineering and Technology (IRJET)*, *07*(06), 4864–4867.

[16] Tan, K. L., Lee, C. P., & Lim, K. M. (2023). A survey of sentiment analysis: Approaches, datasets, and future research. *Applied Sciences*, *13*(7), 4550. https://doi.org/10.3390/app13074550

[17] Vanaja, S., & Belwal, M. (2018). Aspect-Level sentiment analysis on e-commerce data. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1275–1279. https://doi.org/10.1109/icirca.2018.8597286

[18] Wongkar, M., & Angdresey, A. (2019). Sentiment analysis using naive bayes algorithm of the data crawler: Twitter. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 1–5. https://doi.org/10.1109/icic47613.2019.8985884