International Journal of Cloud Computing and Database Management

E-ISSN: 2707-5915
P-ISSN: 2707-5907
Impact Factor (RJIF): 5.4
IJCCDM 2025; 6(2): 90-92
Journal's Website
Received: 15-08-2025
Accepted: 19-09-2025

Dr. E Venkatesan
PG Department of Computer Science, RV Government Arts College, Chengalpattu, Tamil Nadu, India

Dr. V Thangavel
Head, LIRC, St. Francis Institute of Management and Research, Mumbai, Maharashtra, India

Comparative study of naïve Bayes and SVM algorithms for text mining using natural language processing

E Venkatesan and V Thangavel

DOI: https://www.doi.org/10.33545/27075907.2025.v6.i2b.111

Abstract

This study focuses on applying Natural Language Processing (NLP) and text mining techniques for efficient text document analysis. The objective is to compare two machine learning algorithms—Naïve Bayes Classifier and Support Vector Machine (SVM) for accurate text classification and pattern recognition. Essential preprocessing techniques such as tokenization, stop-word removal, stemming, and lemmatization are applied to eliminate noise and improve text quality. Experimental results show that Naïve Bayes performs faster with lower computational cost, while SVM provides higher accuracy for complex datasets. The findings demonstrate that appropriate preprocessing and algorithm selection greatly enhance the effectiveness of NLP-based text mining applications.

Keywords: Natural Language Processing (NLP), text mining, naïve Bayes classifier, Support Vector Machine (SVM), preprocessing, text classification, noise removal

Introduction

The exponential growth of digital text data across social media, scientific publications, news articles, and organisational records has made text mining an essential tool for extracting meaningful insights from unstructured textual information (Manning, Raghavan, & Schütze, 2008) [5]. Text mining involves applying computational techniques to automatically discover patterns, trends, and relationships in text, transforming raw data into actionable knowledge. Common applications include document classification, clustering, sentiment analysis, topic modeling, and information retrieval (Aggarwal & Zhai, 2012) [7]. By uncovering hidden patterns in large datasets, text mining supports decision-making in domains such as healthcare, finance, education, and marketing.

Natural Language Processing (NLP), a subfield of artificial intelligence and computational linguistics, provides the theoretical and algorithmic foundations for text mining (Jurafsky & Martin, 2023) [4]. NLP enables machines to understand, interpret, and generate human language, making it possible to perform tasks such as machine translation, speech recognition, named entity recognition, and automated summarisation. The integration of NLP techniques with text mining allows researchers and practitioners to process large volumes of unstructured text efficiently and accurately.

However, textual data often contains noise, including irrelevant words, spelling errors, inconsistent formatting, and redundancies, which can negatively affect the performance of machine learning models (Aggarwal & Zhai, 2012) [7]. To address this, preprocessing techniques such as tokenisation, stop-word removal, stemming, lemmatisation, and vectorisation are applied to clean and normalise the data (Bird, Klein, & Loper, 2009) [2]. Preprocessing ensures that the input data is structured in a way that improves the efficiency and accuracy of subsequent analytical models.

In this research, two widely used supervised learning algorithms Naïve Bayes Classifier and Support Vector Machine (SVM) are applied to classify and analyse text documents. Naïve Bayes, based on probabilistic inference, is known for its simplicity and computational efficiency, making it suitable for small to medium-sized datasets (Maron, 1961) [6]. SVM, on the other hand, identifies an optimal hyperplane in high-dimensional feature space, providing higher accuracy and robustness for complex and large datasets (Cortes & Vapnik, 1995) [3].

The main objective of this study is to evaluate the performance of these two algorithms in terms of accuracy, precision, recall, F1-score, and runtime efficiency within a text mining framework. By integrating effective preprocessing with NLP-based algorithmic analysis, this

Corresponding Author:
Dr. V Thangavel
Head, LIRC, St. Francis Institute of Management and Research, Mumbai, Maharashtra, India

research aims to enhance the reliability, efficiency, and applicability of automated text document classification systems across multiple domains.

2. Literature Review

This literature review focuses in the rapid advancement of Natural Language Processing (NLP) and text mining has transformed how textual information is processed, analysed, and interpreted. These technologies enable the extraction of meaningful patterns and insights from large collections of unstructured text data. According to Feldman and Sanger (2007) [9], text mining integrates methods from machine learning, statistics, and linguistics to discover hidden structures and semantic relationships in documents. Similarly, Manning, Raghavan, and Schütze (2008) [5] emphasised that NLP provides computational models that make it possible for machines to understand and manipulate natural language for applications such as classification, summarisation, and information retrieval.

One of the fundamental challenges in NLP-based text mining is the preprocessing of text data. Raw text often contains redundant or irrelevant information that can negatively affect classification accuracy. Bird, Klein, and Loper (2009) [2] explained that techniques such as tokenisation, stop-word removal, stemming, and lemmatisation are essential to prepare text for analysis. These preprocessing steps reduce noise, improve data quality, and enhance the interpretability of text features for machine learning algorithms.

In the area of text classification, several studies have focused on the comparative performance of traditional machine learning algorithms. The Naïve Bayes Classifier (NBC) has been widely recognised for its simplicity, computational efficiency, and effectiveness in document classification tasks (Maron, 1961) [6]. McCallum and Nigam (1998) [12] demonstrated that the Naïve Bayes approach achieves strong performance even when the independence assumption between features is violated. However, the Support Vector Machine (SVM) algorithm, introduced by Cortes and Vapnik (1995) [3], has been shown to outperform Naïve Bayes in many large-scale text mining problems due to its ability to handle high-dimensional data and find optimal separating hyperplanes between document classes. Joachims (1998) [11] further validated SVM’s superiority in text categorisation through empirical experiments on benchmark datasets.

Another significant area of research involves feature extraction and representation. Traditional models such as the Bag-of-Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF) have been effective for transforming text into numerical vectors (Salton, Wong, & Yang, 1975) [8]. However, more recent studies have introduced word embeddings such as Word2Vec and GloVe, which capture semantic relationships between words and improve algorithmic performance (Mikolov, Chen, Corrado, & Dean, 2013) [13]. These representations, when combined with robust classifiers, can significantly enhance text classification accuracy.

Comparative studies highlight that while Naïve Bayes is faster and requires less computational power, SVM delivers higher accuracy, especially with complex or unbalanced datasets (Aggarwal & Zhai, 2012; Hotho, Nürnberger, & Paaß, 2005) [7, 10]. Additionally, the effectiveness of both algorithms is strongly influenced by preprocessing quality and feature selection techniques. Jurafsky and Martin (2023) [4] noted that integrating advanced NLP preprocessing pipelines with machine learning models leads to more precise and scalable text mining systems.

Overall, existing literature confirms that the combination of effective preprocessing, feature extraction, and algorithm optimisation forms the foundation for high-performing NLP-based text mining applications. This research builds upon prior studies by conducting a comparative evaluation of Naïve Bayes and SVM algorithms with preprocessing techniques to analyse their accuracy, efficiency, and robustness in text document classification.

3. Methodology

The methodology of this study focuses on applying Natural Language Processing (NLP) techniques for text document analysis using two supervised learning algorithms Naïve Bayes Classifier (NBC) and Support Vector Machine (SVM). The research process begins with collecting text datasets from diverse sources and organising them into categories for classification. The collected data undergo several preprocessing steps, including tokenisation, stop-word removal, lowercasing, stemming, lemmatisation, and noise removal to ensure text uniformity and eliminate irrelevant elements. The cleaned data are then transformed into numerical form using the Term Frequency–Inverse Document Frequency (TF-IDF) technique to extract significant features from the text. Both Naïve Bayes and SVM algorithms are implemented and trained on 80% of the dataset, while 20% is reserved for testing. The performance of the models is evaluated using accuracy, precision, recall, F1-score, and runtime efficiency to compare their effectiveness in handling text classification tasks. This methodological framework ensures a fair comparison and helps identify the algorithm best suited for NLP-based text mining applications.

4. Results and discussion.

The experimental evaluation was conducted on a labelled text dataset, preprocessed using tokenisation, stop-word removal, stemming, lemmatization, and TF-IDF vectorisation. Both Naïve Bayes Classifier (NBC) and Support Vector Machine (SVM) were trained on 80% of the data and tested on the remaining 20%. The performance of the mentioned table 1 and figure 1 each algorithm, was measured using accuracy, precision, recall, F1-score, and runtime

Table 1: shows the result of text dataset analysis two Algorithms' performance.

Metric	Naïve Bayes	SVM
Accuracy	85.2%	91.7%
Precision	84.5%	92.1%
Recall	83.9%	91.3%
F1-Score	84.2%	91.7%
Runtime (s)	12	35

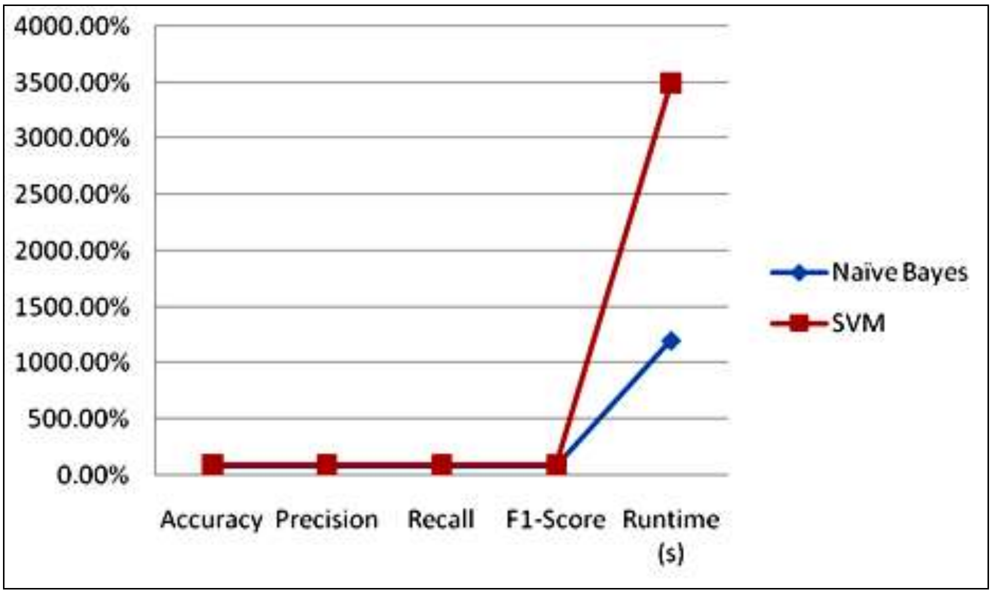

Fig 1: shows the result of two Algorithms' performance of text dataset analysis.

The results indicate that while Naïve Bayes is faster and computationally less expensive, SVM outperforms Naïve Bayes in all accuracy-related metrics. This difference is attributed to SVM's ability to handle high-dimensional data effectively and construct optimal separating hyperplanes in feature space. In contrast, Naïve Bayes relies on the assumption of feature independence, which can limit its performance when term dependencies exist in the dataset.

Preprocessing significantly improved the classification performance of both models by reducing noise and standardising text input. TF-IDF vectorisation further enhanced the models' ability to focus on important terms, increasing precision and recall. Overall, the findings suggest that SVM is more suitable for complex or large-scale text classification tasks, while Naïve Bayes offers a faster alternative for smaller datasets where computational efficiency is a priority.

These results highlight the trade-off between accuracy and runtime efficiency in selecting an algorithm for NLP-based text mining applications. By applying systematic preprocessing and appropriate feature extraction, both algorithms achieve reliable performance, demonstrating the importance of data preparation in text analysis.

5. Conclusion

This study demonstrates that combining preprocessing techniques with machine learning algorithms enhances the performance of NLP-based text mining systems. Naïve Bayes offers computational efficiency, making it suitable for smaller or less complex datasets, while SVM achieves higher accuracy and reliability for large-scale text classification. Effective preprocessing, including tokenisation, stop-word removal, stemming, lemmatisation, and TF-IDF vectorisation, was essential in improving model performance. The research highlights the importance of balancing accuracy, runtime, and computational cost when selecting an algorithm for text mining applications. Overall, integrating systematic preprocessing with appropriate algorithm choice provides a robust framework for efficient and accurate text document classification, laying a foundation for future work in sentiment analysis, topic modelling, and automated text analytics.

Reference

1. Aggarwal CC, Zhai C. Mining text data. Berlin: Springer; 2012.
2. Bird S, Klein E, Loper E. Natural language processing with Python. Sebastopol (CA): O'Reilly Media; 2009.
3. Cortes C, Vapnik V. Support-vector networks. Machine Learning. 1995;20(3):273–297.
4. Jurafsky D, Martin JH. Speech and language processing. 4th ed. London: Pearson; 2023.
5. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. Cambridge: Cambridge University Press; 2008.
6. Maron ME. Automatic indexing: An experimental inquiry. Journal of the ACM. 1961;8(3):404–417.
7. Aggarwal CC, Zhai C. Introduction to text mining. In: Aggarwal CC, Zhai C, editors. Mining text data. Berlin: Springer; 2012. p. 1–34.
8. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Communications of the ACM. 1975;18(11):613–620.
9. Feldman R, Sanger J. The text mining handbook: Advanced approaches in analyzing unstructured data. Cambridge: Cambridge University Press; 2007.
10. Hotho A, Nürnberger A, Paaß G. A brief survey of text mining. LDV Forum. 2005;20(1):19–62.
11. Joachims T. Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of the European Conference on Machine Learning; 1998. p. 137–142.
12. McCallum A, Nigam K. A comparison of event models for Naive Bayes text classification. In: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization; 1998. p. 41–48.
13. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv. 2013;1301.3781.