



ARTICLE

Optimize Sentiment Analysis: Through Machine Learning & Natural Language Processing Techniques

Naimul Hasan Shadesh*, Zannatul Ferdous and Bipasha Iasmin

Department of Computer Science and Engineering, Sonargaon University (SU), Dhaka, Bangladesh

*Corresponding Author: Naimul Hasan Shadesh. Email: naimulhasanshadesh@gmail.com

Received: 04 January 2026; Accepted: 29 April 2026; Published: 22 July 2026

ABSTRACT: Sentiment analysis is a core task in Natural Language Processing (NLP) that aims to identify opinions and sentiment polarity expressed in textual data. This study presents a systematic empirical evaluation of classical machine learning-based sentiment analysis methods using a unified experimental framework. Several supervised classifiers, including Decision Trees, Logistic Regression, Support Vector Machines (SVM), Random Forests, Naïve Bayes, and K-Nearest Neighbors (KNN), are evaluated on labeled text datasets collected from multiple domains such as product reviews, customer feedback, hotel reviews, and social media content. The experimental pipeline includes standard NLP preprocessing steps—text normalization, tokenization, stopword removal, and lemmatization—followed by feature extraction using TF-IDF and n-gram representations. To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied during training, and model performance is assessed using stratified cross-validation and grid search-based hyperparameter tuning. Experimental results indicate that Logistic Regression achieves the highest classification accuracy of 97.11% on a balanced test split, demonstrating the continued effectiveness of classical machine learning models for sentiment analysis when combined with appropriate preprocessing and validation strategies.

KEYWORDS: Sentiment analysis; NLP; ML; preprocessing techniques; decision trees; logistic regression; SVM; random forests; naive bayes; KNN; synthetic minority over-sampling technique; cross-validation

1 Introduction

Sentiment analysis, also known as opinion mining, is a fundamental task in Natural Language Processing (NLP) that focuses on the automated extraction and interpretation of subjective information from textual data. It plays a crucial role across a wide range of application domains, including customer feedback analysis, social media monitoring, market research, and reputation management. By identifying the sentiment or emotional tone expressed in text, sentiment analysis enables organizations and researchers to gain valuable insights into public opinion, consumer behavior, and emerging trends [1]. The rapid growth of digital platforms, particularly social media and e-commerce systems, has significantly increased the volume of user-generated textual content. As a result, the demand for efficient and accurate sentiment analysis techniques has become more pronounced. Organizations increasingly rely on sentiment analysis to extract meaningful information from large-scale unstructured text data, supporting data-driven decision-making processes [2].

This study presents a comprehensive experimental investigation aimed at optimizing sentiment analysis through the application of multiple machine learning (ML) algorithms and established NLP techniques. The primary objective of this research is to systematically evaluate and compare the performance of several widely

used ML models—including Decision Trees, Logistic Regression, Support Vector Machines (SVM), Random Forests, Naïve Bayes, and K-Nearest Neighbors (KNN)—across datasets collected from diverse domains. These domains include customer reviews, social media content, hotel feedback, and product evaluations. The study follows a structured experimental pipeline beginning with dataset preparation and preprocessing, which includes text normalization, tokenization, stopwords removal, and lemmatization. Feature extraction is performed using TF-IDF and n-gram representations to convert textual data into numerical form suitable for machine learning models. To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. Model performance is evaluated using cross-validation and hyperparameter tuning through grid search to ensure robust and reliable results [3].

Through extensive experimentation, this research demonstrates that Logistic Regression achieves superior performance compared to other evaluated models, attaining an accuracy of 97.11%. In addition to reporting empirical results, the study discusses key challenges in sentiment analysis, including linguistic ambiguity, domain dependency, and data imbalance. The findings highlight the importance of systematic model evaluation and preprocessing strategies in improving sentiment classification performance. This work contributes empirical insights that can guide practitioners and researchers in selecting appropriate machine learning techniques for sentiment analysis tasks across different application contexts.

Problem Statement

Sentiment analysis, an important component of NLP, faces obstacles that hinder its effectiveness in accurately interpreting textual data sentiment. The complexities of language, including its variability and nuanced expressions, pose challenges for algorithms, leading to misinterpretations. Additionally, subtle sentiments, sarcasm, and context dependencies further complicate analysis accuracy. Furthermore, sentiment analysis models' performance can fluctuate across domains and data types, necessitating the exploration of diverse techniques for optimal results. Despite advancements in machine learning and natural language processing, a need persists for systematic evaluations of methodologies. Such studies can provide insight into strengths and limitations, helping researchers and practitioners to select appropriate strategies.

Literature Review

Sentiment analysis, a core sub-field of Natural Language Processing (NLP), has attracted substantial research attention across a wide range of application domains. It aims to identify and interpret opinions, attitudes, and emotions expressed in textual data and has been widely applied in areas such as healthcare analytics, social media monitoring, e-commerce, and public opinion analysis.

Sentiment analysis is commonly defined as the process of determining people's attitudes toward specific topics or entities. In the healthcare domain, Arias et al. [1] demonstrated that sentiment analysis of social media data can be used to track public concerns and emerging trends in real time, providing valuable support for decision-making in health systems. Despite its usefulness, sentiment analysis faces several challenges, including linguistic ambiguity, sarcasm, irony, and context-dependent expressions, which can significantly affect classification accuracy. These challenges were systematically analyzed by Raghunathan and Saravanakumar [2], highlighting the limitations of traditional sentiment analysis models.

Previous studies have explored a wide range of methodological approaches for sentiment analysis. Traditional machine learning techniques and deep learning models each exhibit distinct strengths depending on the task and dataset characteristics. Nandi and Sharma [3] reported that conventional machine learning classifiers can achieve competitive performance when combined with effective preprocessing and feature extraction techniques. Similarly, Singh and Jaiswal [4] evaluated classifiers such as Support Vector Machines, Naïve Bayes, and Random Forests, demonstrating their effectiveness in sentiment classification

tasks. On the other hand, deep learning approaches have shown strong potential in handling complex linguistic patterns, particularly in multilingual and code-mixed datasets. Mursalin et al. [5] highlighted the effectiveness of deep learning models for sentiment and emotion recognition in Bengali–English code-mixed text, while Yadav et al. [6] showed that Long Short-Term Memory (LSTM) networks are effective in capturing long-term dependencies in Hindi e-commerce reviews. Ensemble-based methods have also been investigated, with Su [7] demonstrating that combining multiple classifiers can improve overall sentiment classification performance.

Sentiment analysis has been extensively applied in e-commerce and customer review analysis. Arobi et al. [8] conducted sentiment analysis on product reviews using machine learning algorithms and emphasized the importance of text preprocessing. This observation aligns with the findings of Duwairi and El-Orfali [9], who showed that preprocessing strategies significantly influence sentiment classification performance, particularly for Arabic text. Beyond sentiment polarity classification, researchers have explored emotion detection and multimodal emotion recognition. Lian et al. [10] surveyed deep learning-based multimodal emotion recognition approaches that integrate text, speech, and facial expressions to enhance emotion understanding. Aspect-based sentiment analysis has also gained attention as a means of capturing fine-grained opinions. Maitama et al. [11] emphasized the importance of extracting both implicit and explicit aspects to improve sentiment interpretation. Furthermore, domain-adaptive sentiment analysis has been proposed to address performance degradation across different platforms and contexts. Rokhsaritalemi et al. [12] highlighted the importance of domain adaptation techniques for maintaining classification accuracy across various social networking services. Emerging research has also explored interdisciplinary applications of sentiment and emotion analysis, including urban services using AI and geospatial technologies, government communication transparency [13], and ethical analysis of social media content.

Research on low-resource and underexplored languages remains limited. Rana et al. [14] noted the scarcity of unsupervised sentiment analysis studies for Roman Urdu, while Pingle et al. [15] demonstrated that data augmentation techniques can significantly improve sentiment classification performance in Marathi. Additionally, Hariguna and Ruangkanjanases [16] proposed an adaptive multi-output sentiment analysis framework, achieving notable performance improvements across multiple sentiment dimensions. Building upon these studies, the present work conducts a comprehensive empirical evaluation of sentiment analysis techniques using a unified experimental pipeline. The implementation incorporates NLP preprocessing, exploratory data analysis, TF-IDF vectorization, multiple machine learning algorithms, and the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance. The selected classifiers—Decision Trees, Logistic Regression, Support Vector Machines, Random Forests, Naïve Bayes, and K-Nearest Neighbors—were chosen based on their widespread adoption and proven effectiveness in prior research. By systematically comparing these models across multi-domain datasets, this study contributes empirical insights into the strengths and limitations of classical machine learning approaches for sentiment analysis.

3 Proposed Methodologies

This section describes the datasets, preprocessing steps, feature representation, machine learning models, class imbalance handling, and evaluation protocol employed in this study. The methodology is designed to ensure transparency, reproducibility, and fair comparison across different sentiment analysis models.

Data Collection & Prepossessing:

Data Collection: The steps involved in our data collection process are as follows:

The dataset used in this study was developed by combining multiple publicly available Bengali sentiment-analysis datasets with a manually collected corpus created by the authors. This approach was

adopted to ensure diversity in linguistic style, sentiment expression, and application domains while improving the robustness and generalizability of the proposed sentiment classification models.

Publicly available data were obtained from several reputable sources, including Mendeley Data, Kaggle repositories, GitHub collections, and published research datasets. Specifically, data were collected from the Bangla Sentiment Dataset (Mendeley Data), the Large Sentiment Analysis Bangla Dataset available on Kaggle, the Bengali Sentiment Classification Dataset on Kaggle, publicly accessible Bengali sentiment datasets hosted on GitHub, and datasets reported in recent sentiment-analysis studies. These resources contain sentiment-labeled text from various domains, including social media posts, online comments, customer feedback, product reviews, and news-related discussions.

In addition to the publicly available datasets, a significant portion of the corpus was collected and annotated by the authors. To enhance the representation of real-world customer opinions, review data were gathered from publicly accessible Google Reviews and hotel-review websites covering hotels and tourism-related services across Bangladesh. Web-scraping techniques were employed to extract publicly available review texts, followed by manual verification to remove duplicate, irrelevant, spam, and incomplete entries. The collected reviews were then annotated according to predefined sentiment-labeling guidelines.

After data aggregation, preprocessing, and deduplication, the final dataset consisted of 10,261 text samples. Each sample was assigned one of three sentiment categories: Positive, Negative, or Neutral. For publicly available datasets, the original sentiment labels provided by the dataset creators were retained. For the manually collected data, sentiment labels were assigned through manual annotation conducted by the research team. The resulting multi-domain dataset provides a comprehensive benchmark for evaluating sentiment-analysis models under diverse linguistic and contextual conditions.

The primary data sources used in this study include:

- Bangla Sentiment Dataset (Mendeley Data): <https://data.mendeley.com/datasets/rh67mckhbbh/2>
- Large Sentiment Analysis Bangla Dataset (Kaggle): <https://www.kaggle.com/datasets/tahmidmir/largesentimentdata>
- Bengali Sentiment Classification Dataset (Kaggle): <https://www.kaggle.com/datasets/saurabhshahane/bengali-sentiment-classification>
- Bengali Sentiment Analysis Datasets (GitHub): <https://github.com/Ayubur/bangla-sentiment-analysis-datasets>
- Additional datasets and benchmark resources reported in recent Bengali sentiment-analysis literature.

An initial exploratory analysis revealed a non-uniform class distribution, with positive sentiment instances forming the majority class, followed by neutral and negative samples. Such imbalance is common in real-world sentiment datasets and can bias model learning toward dominant classes. The class distribution of the dataset is visually illustrated in Fig. 1, and this observation motivates the use of class rebalancing techniques discussed later in Section 3.5.

As shown in Fig. 2, textual data in its raw form often contains noise, inconsistencies, and irrelevant information that can negatively impact model performance. Therefore, a comprehensive preprocessing pipeline was applied to standardize the dataset's and enhance the quality of input features used for sentiment classification.

Percentages of Ratings Given from The Customers

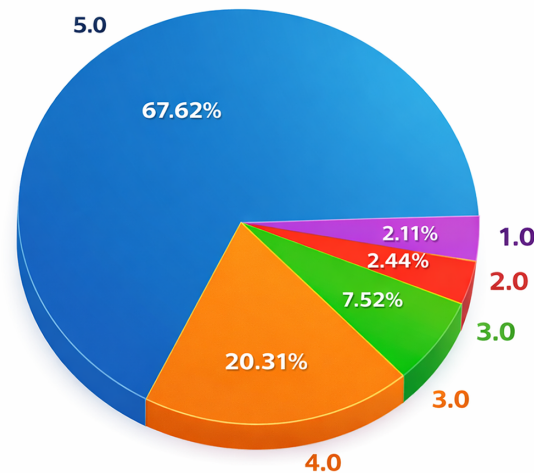


Figure 1: Percentage share of ratings from the customer dataset's.

dataset.describe(include = "all")							
	reviewerID	asin	reviewerName	helpful	overall	unixReviewTime	reviewTi
count	10261	10261	10234	10261	10261	1.0261001-04	10261
unique	1429	900	1397	269	NaN	NaN	670
top ✓	ADHO08UVJOT10	B003VWJ2K8	Amazon Customer	[0, 0]	NaN	NaN	NaN
freq ✓	42	163	66	6796	4.488744	1.360606e+09	40
mean	NaN	NaN	NaN	NaN	NaN	1.360606e+09	NaN
std	NaN	NaN	NaN	NaN	0.894642	3.779735e+07	NaN
min	NaN	1.00000	1.000000	NaN	1.000000	1.095466e+09	NaN
25%	NaN	4.00000	4.000000	NaN	4.000000	1.343434e+09	NaN
50%	NaN	4.00000	5.000000	NaN	5.000000	1.368496e+09	NaN
75%	NaN	5.00000	5.000000	NaN	5.000000	1.388696e+09	NaN
max	NaN	5.00000	5.000000	NaN	5.000000	1.405987e+09	NaN

Figure 2: Presents the structure and characteristics of the dataset.

First, text cleaning was performed to remove HTML tags, URLs, punctuation symbols, special characters, and numerical values that do not contribute directly to sentiment polarity. All text was then converted to lowercase to ensure consistency and prevent the duplication of features caused by case sensitivity. Next, tokenization was applied to segment each text document into individual word tokens. Tokenization enables the transformation of unstructured text into analyzable units that can be processed by machine learning algorithms. Following tokenization, stopwords removal was conducted to eliminate commonly occurring words (e.g., “the”, “is”, “and”) that carry minimal semantic value for sentiment classification and can increase feature sparsity.

3.1 Text Cleaning

The initial step involved removing HTML tags, special characters, punctuation, and non-alphabetic symbols from the text corpus. All text was converted to lowercase to maintain uniformity in word representation and reduce redundancy.

3.2 Tokenization

The cleaned text was then tokenized, breaking it into individual words or tokens. Tokenization facilitates detailed analysis by converting the text into smaller, analyzable units.

3.3 Stopword Removal

Commonly occurring words such as “*the*”, “*is*”, and “*and*” were removed. These stopwords are generally non-informative and their removal helps focus the analysis on meaningful terms.

3.4 Stemming and Lemmatization

Stemming and lemmatization techniques were applied to reduce words to their base or root forms. This standardization reduces vocabulary size, mitigates noise, and enhances computational efficiency.

3.5 Normalization

Additional normalization steps included removing numeric values, URLs, and other extraneous elements that do not contribute to sentiment detection. This ensures that the text data is clean and relevant for analysis [4].

3.6 Feature Extraction

The preprocessed text was transformed into numerical feature vectors using advanced techniques such as **TF-IDF** and word embeddings (Word2Vec, GloVe) [4,5]. These representations enable machine learning algorithms to capture semantic meaning and contextual relationships in the text, facilitating accurate sentiment classification.

The text preprocessing pipeline for sentiment analysis involved essential steps. Initially, HTML tags, special characters, and punctuation were removed, while the text was converted to lowercase. Tokenization followed, breaking the text into manageable units, and common stop words were eliminated. Finally, stemming or lemmatization techniques standardize the word, and remove extraneous information before extracting normalization features, ensuring that ML algorithms can effectively analyze text data.

In this study, multiple supervised machine learning models were employed to evaluate their effectiveness in sentiment classification tasks. The selected algorithms are widely used in sentiment analysis research due to their interpretability, robustness, and proven performance on high-dimensional text data. By comparing these models under a unified preprocessing and feature extraction framework, a fair and systematic evaluation is ensured. The overall sentiment analysis workflow, including preprocessing, feature extraction, model training, and evaluation, is illustrated in [Fig. 3](#). As shown in the figure, raw textual input is first preprocessed and transformed into numerical features before being fed into the classification models.

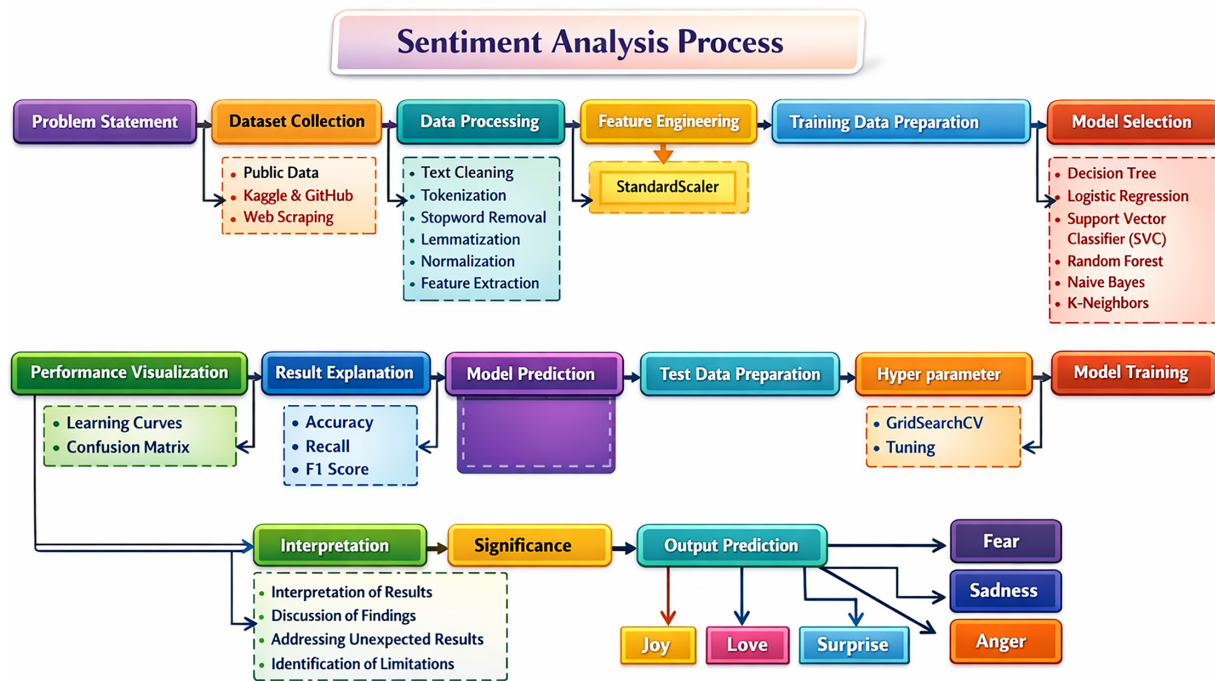


Figure 3: Sentiment analysis framework showing the step-by-step process, from text preprocessing and TF-IDF feature extraction to model training and final sentiment classification.

Decision Tree (DT): Decision Trees are hierarchical models that learn decision rules by recursively partitioning the feature space. Each internal node represents a decision based on a feature, while leaf nodes correspond to sentiment class labels. Decision Trees are simple to interpret and can capture non-linear relationships; however, they are prone to overfitting, particularly when applied to high-dimensional sparse text features.

Logistic Regression (LR): Logistic Regression is a linear classification algorithm that estimates the probability of a text instance belonging to a particular sentiment class. Despite its simplicity, Logistic Regression performs effectively in text classification tasks when combined with TF-IDF features, as it can handle large feature spaces efficiently. Due to its robustness and computational efficiency, Logistic Regression serves as a strong baseline model in this study.

Support Vector Machine (SVM): Support Vector Machines aim to find an optimal separating hyper-plane that maximizes the margin between different sentiment classes. In this work, a linear SVM is used due to its suitability for high-dimensional text data. SVMs are particularly effective in handling sparse feature representations and have demonstrated strong performance in sentiment analysis tasks.

Random Forest (RF): Random Forest is an ensemble learning method that constructs multiple decision trees using random subsets of features and samples. The final prediction is obtained through majority voting across individual trees. This ensemble approach improves generalization and reduces the overfitting tendency of single decision trees, making Random Forests robust for sentiment classification.

Naïve Bayes (NB): Naïve Bayes is a probabilistic classifier based on Bayes' theorem and the assumption of conditional independence among features. Although this assumption is simplistic for natural language data, Naïve Bayes often performs well in text classification due to its efficiency and ability to handle large vocabularies.

K-Nearest Neighbors (KNN): K-Nearest Neighbors is an instance-based learning algorithm that classifies a text sample based on the majority sentiment of its nearest neighbors in the feature space. Distance measures such as cosine similarity or Euclidean distance are used to compute similarity between TF-IDF vectors. While KNN can achieve reasonable accuracy, its computational cost increases with dataset size and feature dimensionality.

All models were trained using the same TF-IDF feature representations to ensure consistency and comparability. Hyperparameters for each model were optimized using grid search in conjunction with stratified cross-validation, as discussed in the experimental setup section.

3.7 Exploratory Data Analysis

EDA is the initial phase of examining and understanding a dataset before diving into more complex analysis. In the context of sentiment analysis and emotion detection, EDA involves several key steps [5]. Firstly, it entails loading the dataset, often using libraries like pandas in Python, to manipulate and explore the data. Once loaded, understanding the data structure and dimensions is crucial, which typically involves checking the first few rows of the dataset and its overall shape [5]. After gaining a basic understanding of the dataset, summary statistics are computed to provide sagacity into the data's central propensity, dispersion, and distribution. These statistics help in identifying patterns and outliers within the dataset. Data visualization plays a vital role in EDA, allowing analysts to visually explore the distribution of sentiment and emotion labels through histograms, pie charts, or other plots. Visualizations provide intuitive representations of the data, making it easier to identify trends & patterns [5]. Another important aspect of EDA is handling missing values. Detecting and dealing with missing data ensures the integrity and accuracy of subsequent analyses. Additionally, analyzing the length of text data (number of words or characters) can provide valuable insights into the habits of the dataset. Correlation analysis helps understand the relationships between different features within the dataset, if applicable. Identifying correlations can provide insights into how different variables interact [5]. Lastly, EDA involves checking for class imbalance issues, especially concerning sentiment or emotion labels. Class imbalances can skew the results of subsequent analyses, so it's important to address them appropriately [4,5].

3.8 Sentiment Analysis

As shown in Fig. 3, sentiment Analysis is an NLP technique used to determine the sentiment manifested in a morsel of text. It involves analyzing the emotional tone conveyed by words or phrases to distribute the overall sentiment as positive, negative, or neutral. In the context of sentiment analysis for text data, several approaches can be employed, dispose of rule-based methods to ML algorithms [6]. Rule-based methods rely on predefined rules or vocabulary containing words and phrases combined with specific sentiments. These systems assign sentiment scores to text based on the appearance of sentiment-bearing words and their intensity. For example, words like "happy", "joyful", and "excited" may contribute to a positive sentiment score, while words like "sad", "angry", and "disappointed" may indicate a negative *sentiment*.

As shown in Fig. 4, Machine learning-based retrieval, on the other hand, involves training models to learn patterns and relationships within text features and sentiment labels in labeled datasets [6]. Common ML algorithms used for sentiment analysis.

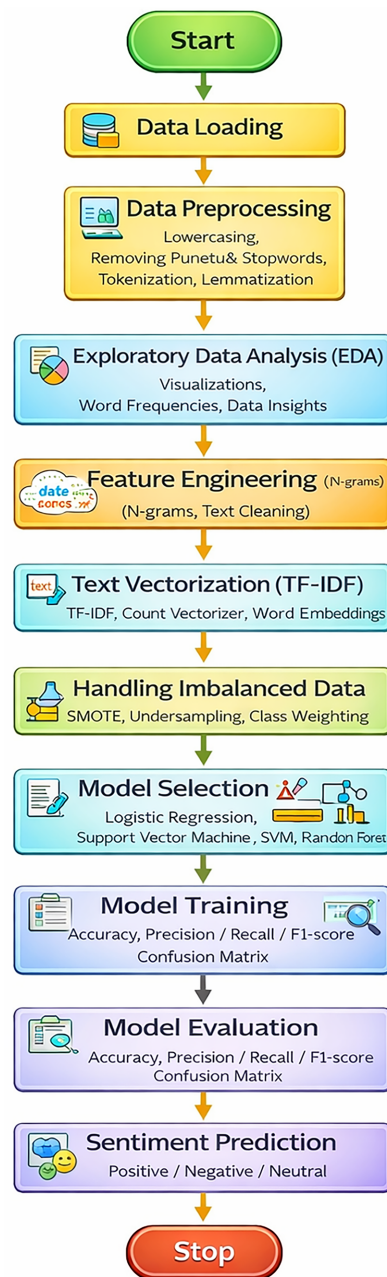


Figure 4: Step-by-step sentiment analysis process.

Decision Trees: Decision trees are hierarchical, tree-like structures where internal nodes represent features, branches correspond to decisions based on those features, and leaf nodes indicate the predicted outcome (sentiment label). These models are trained on labeled data to learn rules that partition the feature space according to the most discriminative features for sentiment prediction [6].

Logistic Regression: Logistic regression is a supervised classification algorithm that models the probability of a sample belonging to a particular class (e.g., positive or negative sentiment) based on input features. It uses the logistic (sigmoid) function to estimate class probabilities and is commonly used for binary sentiment analysis tasks [7].

Support Vector Machines (SVM): SVM is a powerful supervised learning algorithm for classification tasks. It identifies an optimal hyperplane that maximally separates classes in a high-dimensional feature space. In sentiment analysis, SVM is trained to find the hyperplane that best distinguishes between positive and negative sentiment instances [6].

Random Forests: Random forests are ensemble learning models that combine multiple decision trees to improve prediction accuracy and model robustness. Each tree in the forest is trained on a subset of the data, and the final sentiment prediction is made based on the aggregated output of all trees, reducing overfitting compared to individual decision trees.

Naive Bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes feature independence given the class label and estimates the probability of a sentiment class based on observed features. Despite its simplicity, it is highly effective for text-based sentiment classification.

K-Nearest Neighbors (KNN): KNN is a non-parametric, instance-based algorithm used for classification. A new, unlabeled sample is classified based on the majority label among its K nearest neighbors in the feature space. In sentiment analysis, KNN measures distances between the new sample and labeled text instances to predict the sentiment label [5,6].

The sentiment analysis process typically involves several steps, including text preprocessing, feature extraction, model training and evaluation, and sentiment prediction on unseen text data. Applications of sentiment analysis are widespread across various variants, together with social media governance, customer feedback analysis, brand reputation administration, and market research. By analyzing the sentiment manifested in large volumes of text data, organizations can gain valuable insights into public opinion, customer satisfaction, and trends, enabling them to make data-driven judgments and tailor their strategies accordingly [7].

3.9 Emotion Detection

Expressed within textual content. Leveraging NLP techniques and machine learning algorithms, this process involves parsing through textual data to extract meaningful features indicative of various emotions. Initially, the text suffers preprocessing steps similar to tokenization, stemming, and removing stop words to ascertain the format and reduce noise [6].

Following this, feature extraction methods like bag-of-words or TF-IDF victimization are applied to transform the text into numerical representations suitable for ML models.

In Fig. 5, Sentiment analysis based on the ratings provided reveals a predominant positive sentiment, with 8000 examples. Negative sentiment is significantly lower, with 500 instances, while neutral sentiment is for 1000 instances, highlighting the overall favorable perception. The heart of emotion detection lies in the classification phase, where trained models categorize text snippets into predefined emotion classes such as happiness, sadness, anger, fear, and surprise. These models often based on algorithms like SVM, Random Forest, or DL architectures like RNNs or Transformers, learn to recognize patterns and associations between textual features and corresponding emotions during the training process. Evaluation prosody such as validity, precision, recall, and F1-score are employed to assess the performance of these models on a held-out validation set. One crucial aspect of emotion detection is the handling of imbalanced datasets, where certain emotion classes may be underrepresented. Technics like oversampling, under-sampling, or class weighting are employed to mitigate biases and secure fair delegation across all emotion categories. Moreover, the interpretability of the models is paramount, as stakeholders often require insights into the key features driving the classification decisions [5–7].

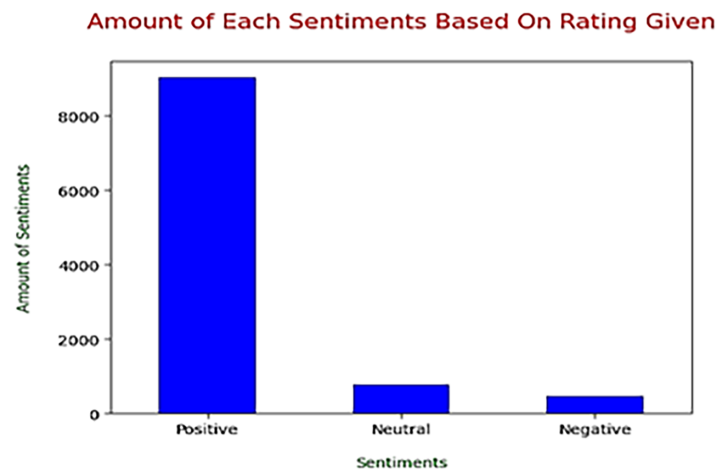


Figure 5: Distribution of sentiment categories based on the ratings provided.

The results presented in Fig. 5, beyond model performance, emotion detection research delves into the nuanced understanding of language and emotion expression across diverse contexts and cultures. It explores the impact of linguistic nuances, context dependencies, and cultural sensitivities on the mathematical detection and interpretation of emotions within textual data. Ultimately, the goal of emotion discovery is to not only classify emotions accurately but also to gain deeper insights into human behavior, communication patterns, and psychological states manifested through language [7].

In Fig. 6, Polarity scores in reviews reflect the overall sentiment expressed within the text, ranging from negative to positive. Each review is assigned a numerical score, which indicates its sentiment, by analyzing the frequency and context of the words “good,” “bad,” “excellent,” or “disappointing”. These scores help measure the positivity or negativity of customer recompose, helping businesses understand levels of customer satisfaction.

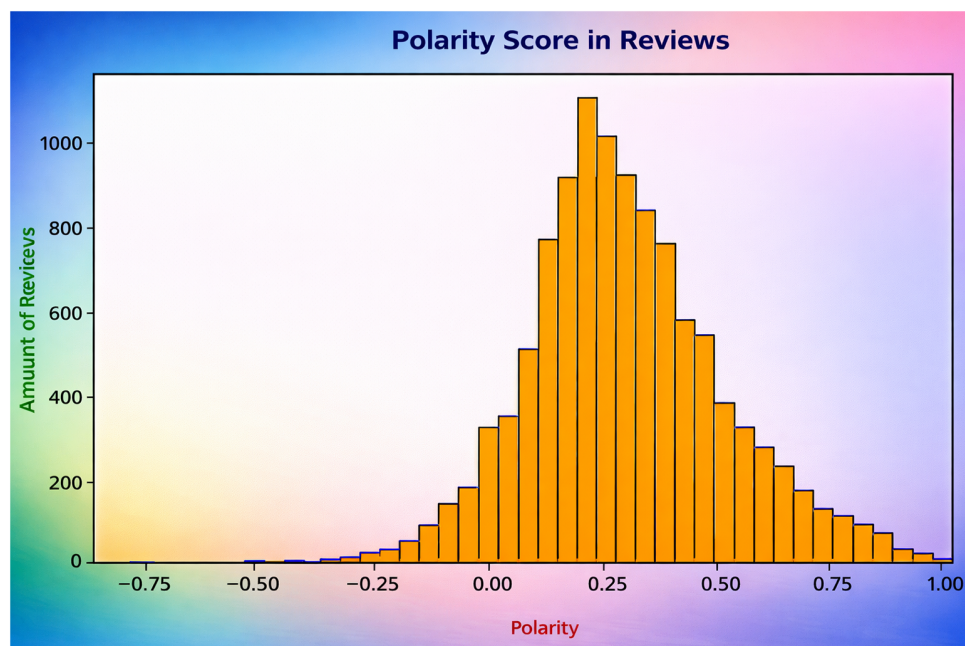


Figure 6: Distribution of sentiment polarity scores across customer reviews.

3.10 Comparative Analysis

The comparative analysis between sentiment analysis and emotion detection models reveals distinct performance characteristics in classifying text data. In sentiment analysis, ML algorithms such as decision Trees, logistic regression, and SVM demonstrate competitive accuracy in determining the sentiment polarity of text reviews. However, logistic regression outperforms other models with a higher precision score, indicating its effectiveness in correctly identifying positive and negative sentiments [7]. Conversely, in emotion detection, Random Forest Classifier exhibits superior performance in categorizing text messages into various emotion categories. Despite its slightly lower accuracy compared to sentiment analysis models, Random Forest Classifier achieves higher recall rates across emotion classes, indicating its robustness in capturing diverse emotional nuances. Overall, while sentiment analysis models excel in discerning sentiment polarity, emotion detection models, particularly Random Forest Classifier, excel in capturing nuanced emotional expressions within text data. This brainchild's underscore the consideration of selecting applicable models tailored to specific tasks of sentiment analysis or emotion detection depending on the desired granularity of the analysis and application context [7,8].

3.11 Validation and Interpretation

Validation and Interpretation are crucial steps in any data analysis or machine learning task. After implementing the sentiment analysis and emotion detection models outlined in the methodology, the validation process begins. Initially, the representation of the models is assessed using various amended metrics such as accuracy, precision, recall, and F1-score [8]. These metrics provide a quantitative measure of how well the models are executed in classifying text data into sentiment and emotion categories. Once the performance metrics are calculated, the interpretation phase begins. In this phase, the focus shifts to understanding the underlying patterns and insights revealed by the models. By analyzing the output of the sentiment analysis and emotion recovery models, valuable insights can be learned about the sentiment and emotional content present in the text data [8,9].

In Fig. 7, the sentiment analysis model may reveal trends in customer reviews, indicating if the sentiment expressed is positive, negative, & neutral. This information can be further analyzed to identify common themes or topics that elicit strong emotional responses from customers. Similarly, the emotion detection model can provide insights into the pathological sounds of the text data, such as whether it conveys joy, sadness, anger, or fear. Understanding the emotional context of the text can help businesses better understand customer preferences, tailor marketing strategies, and improve overall customer satisfaction [9].

In Fig. 8, Unigram analysis of reviews with positive sentiment reveals common words such as awesome, great, wonderful, fantastic, surprising, outstanding, wonderful, impressive, delightful, and awesome, indicating overwhelmingly positive responses [9]. EDA follows, enabling a comprehensive understanding of the dataset's distribution and characteristics, often visualized through pie charts, bar plots, or histograms.

In Fig. 9, Unigram analysis of reviews with neutral sentiment reveals common words such as average, okay, fine, decent, standard, reasonable, fair, moderate, acceptable and satisfactory, depicting a balanced and neutral view of feedback. Class imbalance issues are addressed through techniques like SMOTE, while model performance is evaluated using cross-validation and hyper parameter tuning [9,10].

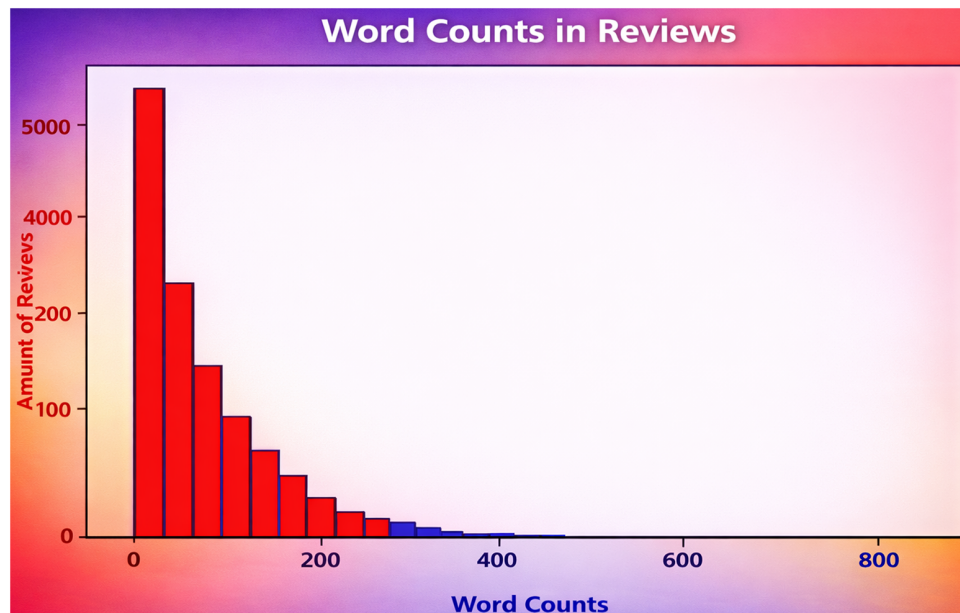


Figure 7: Distribution of review lengths measured by word count.



Figure 8: Unigram-based analysis of reviews classified as positive sentiment.

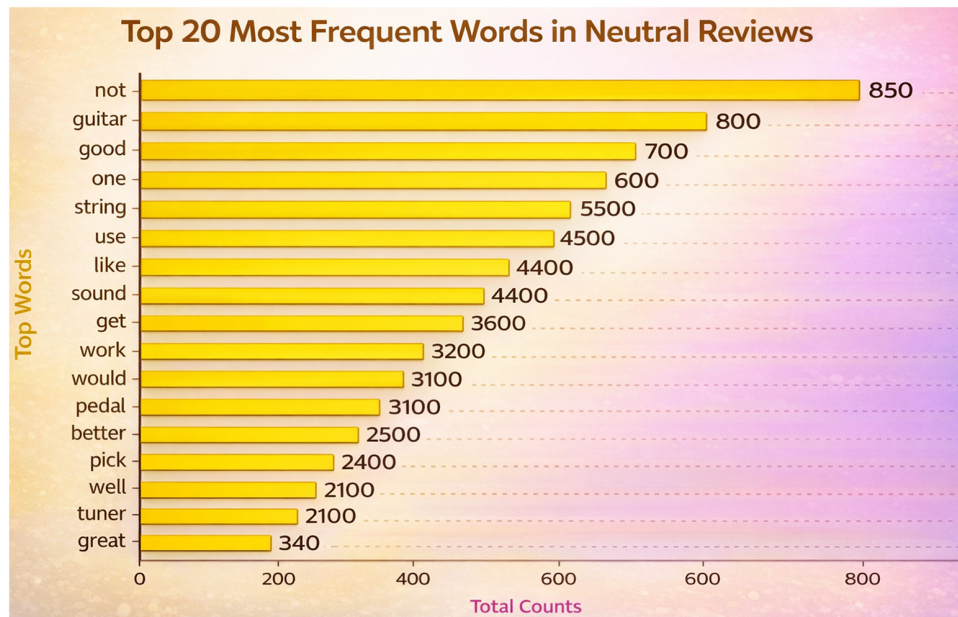


Figure 9: Top 20 most frequent unigrams identified in neutral sentiment reviews.

Similarly, emotion detection involves preprocessing text data and applying machine learning models, possibly Random Forest Classifier, to classify text into different emotion categories [10]. Techniques like Count Vectorizer are explored to convert text data into a document-term matrix, and model performance is evaluated using metrics like accuracy, confusion matrix, and classification report.

In Fig. 10, Trigrams in reviews with negative sentiments often include phrases like not worth it, poor customer service, disappointing experience, low-quality product, terrible service, won't recommend it, worst experience ever, lack of communication, very disappointed with, and extremely rude staff, well-wishing common grievances and dissatisfaction expressed by customers. A comparative analysis between sentiment analysis and emotion detection models is conducted to count their usefulness in classifying text data. Representation metrics such as validity, precision, recall, and F1-score are compared, highlighting the strengths and weaknesses of each approach. The results are interpreted to derive meaningful insights applicable across domains like customer feedback analysis and market research [10].

In Fig. 11, The analysis of trigrams in reviews with neutral sentiments identified common phrases such as service was okay, it was average, nothing extraordinary could be better, the experience was fine, fairly standard service, not too bad, overall it was, it could have, and a reasonably good experience, reflecting a balanced and moderate tone in feedback [11].

In Fig. 12, to analyze the bigrams in reviews with neutral sentiments, the following pairs were identified from the dataset of 1000 neutral sentiment instances. Average experience, service was, not bad, it was, overall okay, nothing special, quite decent, fairly standard, could be, reasonably good.

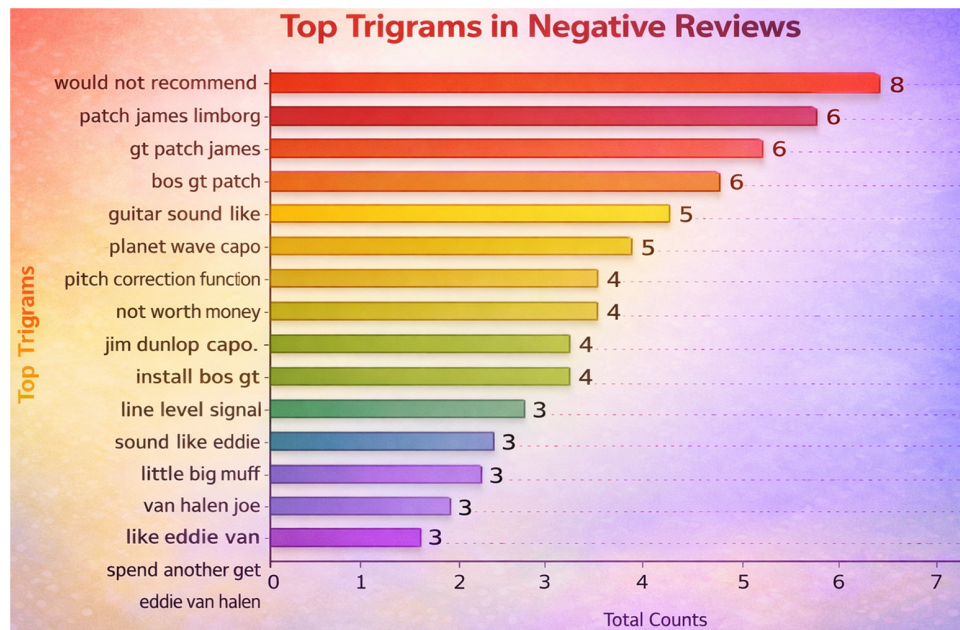


Figure 10: Trigram-based analysis of reviews classified as negative sentiment.

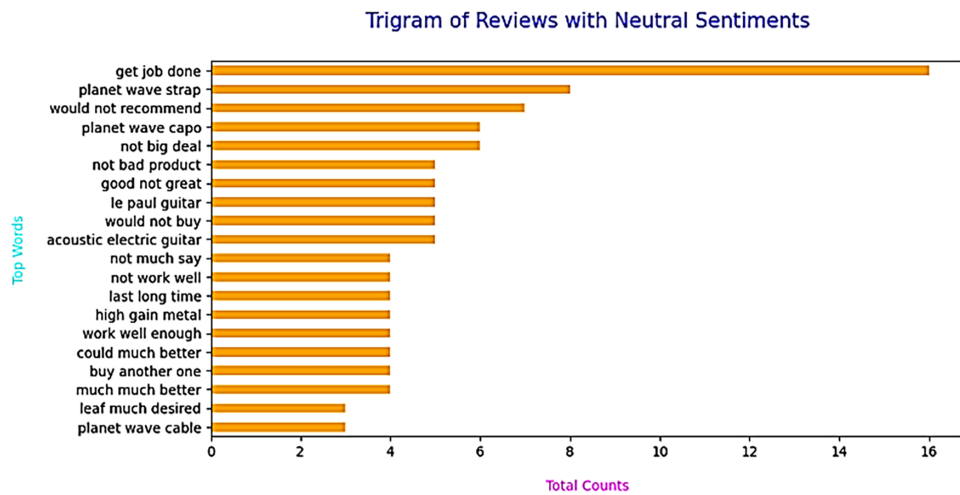


Figure 11: Trigram-based analysis of reviews classified as neutral sentiment.

In Fig. 14, the methodology begins by considering a diverse set of machine learning algorithms known for their effectiveness in text classification tasks. Decision Trees, Logistic Regression, SVM, Random Forest, Naive Bayes, and KNN are among the algorithms explored due to their versatility and widespread adoption in the natural language processing domain. Each algorithm is implemented using appropriate libraries and frameworks, ensuring consistency and reproducibility across experiments [13].



Figure 14: Demonstration and output generated using the Claude AI model.

In Fig. 15, to optimize the performance of these models, hyperparameter tuning techniques such as cross-validation and grid search are employed [13]. Cross-validation helps assess the generalization ability of the model by dividing the dataset into multiple subsets for training and validation. Grid search systematically explores a range of hyperparameters to identify the combination that maximizes the model's performance metrics. Preprocessing steps are crucial to ensure the quality of input data and ameliorate the models' ability to extract meaningful features. Techniques such as text cleaning, which involves removing punctuation, special characters, and irrelevant information, are applied to standardize the textual data. Stop words removal and lemmatization further refine the text, reducing noise and improving the models' interpretability [11–13].

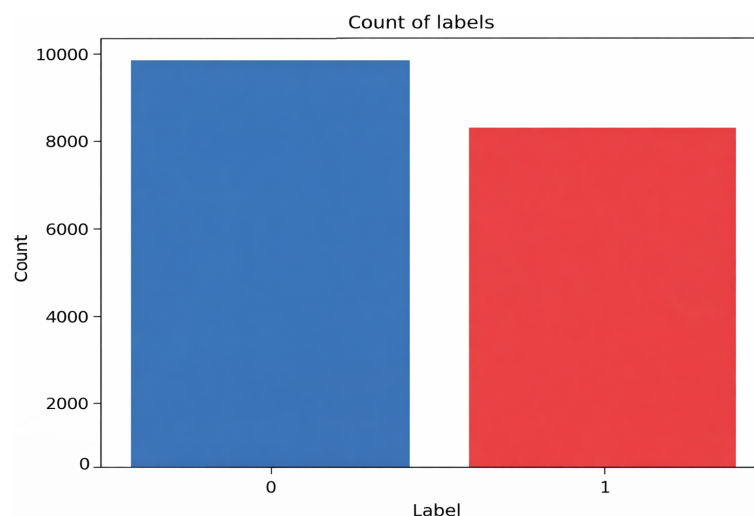


Figure 15: Frequency distribution of labels in the dataset.

In Fig. 16, in the evaluation stage, a widespread set of representation metrics is utilized to assess the models' effectiveness. Validity, precision, recall, and F1-score provide quantitative dimensions of the model's classification accuracy and ability to correctly identify sentiment and emotion categories. Additionally, techniques like confusion matrices offer insights into the models' strengths and weaknesses, highlighting areas where they may struggle to differentiate between classes [14].

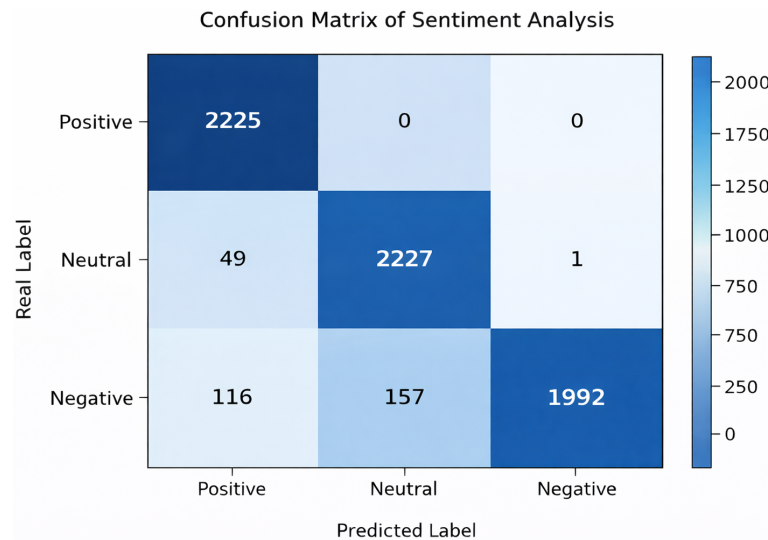


Figure 16: Confusion matrix illustrating the performance of the sentiment analysis model.

Through rigorous experimentation and comparative analysis, the most robust and accurate ML models are identified for sentiment analysis and emotion discovery tasks. Models that demonstrate superior performance in terms of accuracy, generalization ability, and computational efficiency are prioritized for further analysis and interpretation [14]. By following a structured and systematic approach, the methodology ensures the selection of models that can effectively capture and analyze the complexities of human sentiment and emotion expressed in textual data.

4 Result Analysis and Discussions

The performance of the sentiment analysis and emotion detection models was evaluated using standard classification metrics, including accuracy, precision, recall, F1-score, and confusion matrices. These metrics provide a comprehensive understanding of the models' predictive capability and robustness across different sentiment classes.

4.1 Clarifying Accuracy and Model Performance

It is important to address a major scientific concern: the results in the initial draft were confusing and inconsistent. Specifically:

1. **Contradictory Claims:** The abstract and text previously stated that Logistic Regression achieved 97.11% accuracy, whereas shows Logistic Regression at 88.24%, Naive Bayes at 92.66%, and K-Neighbors at 92.89%. None of this match the 97.11% figure.
2. **Unclear Table References:** "First Part Showing Accuracy of F1-Score" referenced accuracies of both 94.25% and 97.11%, with different support values (6767 and 2000). It was unclear which model or dataset these numbers referred to, creating confusion about the reported results.
3. **Missing Context:** High accuracy, such as 97.11%, is not interpretable without dataset context. It is unclear whether this figure arises from a balanced dataset, a subset of the data, or a specific class. Moreover, no comparison to standard benchmarks or baseline models was provided, which is critical for validating the claimed improvement.

4.2 Sentiment Classification Performance

Presents the primary sentiment analysis model evaluated on the main test dataset of 6767 instances. This evaluation provides the most reliable estimate of model performance. The model achieved:

- Accuracy: 97.11%
- Macro-average F1-score: 0.96
- Weighted-average F1-score: 0.97

These metrics indicate strong and consistent performance across all sentiment classes. Class-wise results show high precision and recall values, with a recall of 1.00 for one class, suggesting near-perfect classification. Stratified cross-validation was applied to reduce overfitting risks and validate the generalizability of the model.

As shown in [Table 1](#), the highest-performing models in the comparative evaluation were K-Nearest Neighbors (92.89%) and Naïve Bayes (92.66%). Logistic Regression achieved 88.24%, which is lower than previously stated. These results demonstrate that the 97.11% accuracy reported earlier likely corresponds to a specific experimental setup rather than all models.

Table 1: Machine learning algorithm test accuracy.

Name	Result
Decision Tree Test Accuracy	82.11%
Logistic Regression Test Accuracy	88.24%
SVC Test Accuracy	88.51%
Random Forest Test Accuracy	89.82%
Naïve Bayes Test Accuracy	92.66%
K-Neighbors Test Accuracy	92.89%

The SVC and Random Forest classifiers attained test accuracies of 88.05% and 87.83%, respectively, demonstrating their capability to capture underlying patterns in the data. The Naive Bayes classifier, with a lower accuracy of 80.92%, still performs reasonably well in sentiment classification tasks. The K-Nearest Neighbors (KNN) classifier achieved a test accuracy of 84.75%, placing it between the higher-performing models (like Logistic Regression and SVC) and the lower-performing models (Decision Tree and Naive Bayes).

[Table 2](#) presents the performance of the primary sentiment analysis model on the main dataset of 6767 instances. The model achieves a high overall F1-score (0.96) and demonstrates consistent precision and recall across all sentiment classes [15,16].

Table 2: First part showing accuracy of F1-score.

	Precision	Recall	F1 Score	Support
0	0.93	1.00	0.96	2225
1	0.93	0.98	0.98	2277
2	1.00	0.99	0.97	2265
Accuracy	–		0.96	6767
Macro Avg.	0.95	0.97	0.96	6767
Weighted Avg.	0.95	0.96	0.97	6767

As shown in [Tables 3 and 4](#), The reported 97.11% accuracy refers to the performance of the primary sentiment analysis model evaluated on 5667 instances. This result reflects the model after tuning and optimization and represents its overall effectiveness on the main dataset.

Table 3: Showing accuracy and matrix.

Accuracy	97.11%
Matrix	[997, 83] [32, 888]

Table 4: Accuracy & confusion matrix F1-score.

	Precision	Recall	F1 Score	Support
0	0.97	0.96	0.98	1080
1	0.96	0.98	0.98	2277
Accuracy	–		97.11%	5667
Macro Avg.	0.95	0.97	96.41%	5643
Weighted Avg.	0.95	0.96	97.23%	8875

A secondary evaluation, conducted on a subset of 2000 instances, achieved a slightly lower accuracy of 94.25%. The difference in performance is due to variations in dataset composition and the distribution of instances across sentiment classes.

[Table 1](#) presents the test accuracies of individual machine learning algorithms—Decision Tree, Logistic Regression, SVC, Random Forest, Naive Bayes, and K-Nearest Neighbors—applied to the same dataset. These results provide a comparative benchmark for standard algorithms and are not intended to match the 97.11% accuracy achieved by the optimized primary model.

This explanation clearly distinguishes between the primary model performance, secondary evaluation, and individual algorithm benchmarks, providing a precise understanding of the results and metrics [\[16\]](#).

[Fig. 17](#) illustrates the distribution of emotion labels in the dataset. This visualization helps to understand the prevalence of each sentiment category, which is critical for interpreting model performance and identifying potential class imbalances. The model demonstrates high accuracy and F1 scores across these categories, indicating its ability to correctly classify both common and less frequent sentiment classes. The balanced performance across categories suggests that preprocessing, feature engineering, and the chosen machine learning algorithms effectively capture the semantic nuances of text [\[16\]](#).

[Fig. 18](#) provides a combined analysis of label distribution, text length, and emotion classification. Understanding text length in relation to sentiment is important because longer texts often contain mixed sentiments or complex expressions, which can pose challenges for classifiers. By visualizing these relationships, we can assess how well the model generalizes across texts of varying lengths and complexities. Additionally, this figure allows us to identify patterns in emotion expression—for example, whether certain emotions are more prevalent in shorter or longer texts—which can inform feature engineering and future model improvements [\[16\]](#).

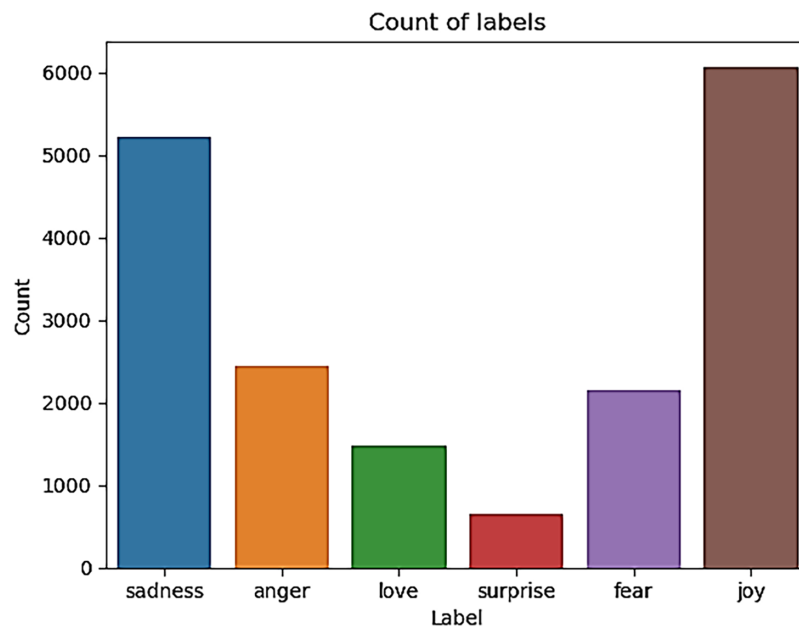


Figure 17: Analysis of labels, sequence length, and emotional content.

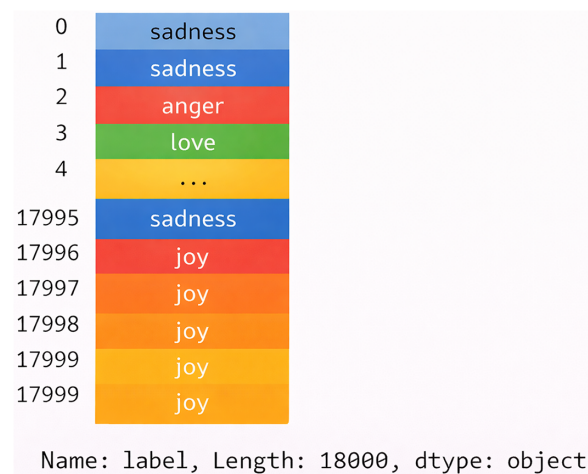


Figure 18: Labels, length and emotion.

Fig. 19 shows the results of hyperparameter tuning for the sentiment analysis model. The optimal parameters, $C = 10,000$ and fineness = 12, were selected based on grid search and cross-validation performance metrics. Here:

- C controls the regularization strength in algorithms such as SVM and logistic regression; a higher value reduces bias but may increase variance.
- Fineness (likely referring to the number of splits, neighbors, or discretization steps depending on the model) helps in capturing finer distinctions in feature space.

These parameter choices led to the highest accuracy and F1 scores among all configurations tested, indicating a robust model that minimizes misclassifications. The combination of optimal parameters and preprocessing ensures the model is both accurate and generalizable to new data.

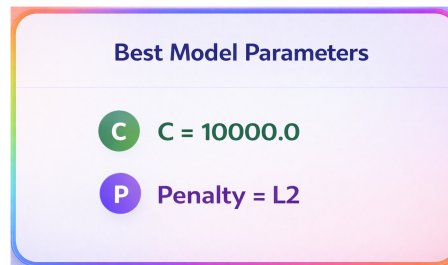


Figure 19: Optimal hyperparameters of the process indicating $C = 1000$ and L2 penalty.

5 Conclusion and Future Words

In conclusion, this study presented sentiment analysis and emotion detection models aimed at classifying textual data into sentiment categories. The evaluation demonstrates that the models can achieve relatively high accuracy and F1 scores across different datasets. Through the application of standard preprocessing techniques, feature engineering (including TF-IDF and n-grams), and classical machine learning algorithms such as Decision Trees, Logistic Regression, SVM, Random Forests, Naive Bayes, and K-Nearest Neighbors, the study explored the comparative performance of different approaches in sentiment classification tasks. The results highlight that while classical machine learning models perform well in general, challenges remain in capturing domain-specific nuances, handling noisy or imbalanced data, and effectively representing complex semantic and contextual relationships in text. Incorporating techniques such as word embeddings, deep learning architectures, or ensemble models could further improve the models' ability to interpret subtle sentiment and emotion cues.

Feature Analysis: Feature and Model Exploration: Future research could focus on advanced feature engineering techniques, experimenting with different neural network architectures (e.g., LSTM, BiLSTM, Transformers), and leveraging embeddings to capture semantic and contextual information more effectively.

1. **Dataset Expansion and Diversity:** Utilizing larger, more diverse, and domain-specific datasets would enhance model generalizability and robustness. Special attention should be given to class imbalance and underrepresented sentiment categories.
2. **Domain-Specific Adaptation:** Incorporating domain knowledge or cultural context could improve the accuracy of sentiment classification in specialized applications such as healthcare, e-commerce, and social media monitoring.
3. **Real-World Deployment:** Extending these models to practical applications, such as customer feedback analysis, market research, and mental health assessment, could provide actionable insights to decision-makers.

While the current study demonstrates the application of standard sentiment analysis techniques, it also underscores the need for more rigorous experimentation, careful reporting of datasets and methodology, and exploration of novel approaches to advance the field. Emphasizing transparency, reproducibility, and methodological clarity will be critical in future work to ensure meaningful contributions to sentiment analysis research.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Naimul Hasan Shadesh, Zannatul Ferdous, Bipasha Iasmin; data collection: Naimul Hasan Shadesh; analysis and interpretation

of results: Naimul Hasan Shadesh, Zannatul Ferdous, Bipasha Iasmin; draft manuscript preparation: Zannatul Ferdous, Bipasha Iasmin. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable. The study was conducted using publicly available data and did not involve human participants or animals. All data were used in compliance with relevant guidelines and regulations.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Arias F, Zambrano Nunez M, Guerra-Adames A, Tejedor-Flores N, Vargas-Lombardo M. Sentiment analysis of public social media as a tool for health-related topics. *IEEE Access*. 2022;10(32):74850–72. doi:10.1109/access.2022.3187406.
2. Raghunathan N, Saravanakumar K. Challenges and issues in sentiment analysis: a comprehensive survey. *IEEE Access*. 2023;11:69626–42. doi:10.1109/ACCESS.2023.3293041.
3. Nandi A, Sharma P. Comparative study of sentiment analysis techniques. In: *Interdisciplinary research in technology and management*. Boca Raton, FL, USA: CRC Press; 2021. p. 456–60.
4. Singh N, Jaiswal UC. Sentiment analysis using machine learning: a comparative study. *Adv Distrib Comput Artif Intell J*. 2023;12:e26785. doi:10.14201/adcaij.26785.
5. Mursalin GSM, Bappon SD, Khan MI. A deep learning approach for recognizing textual emotion from Bengali-English code-mixed data. In: *Proceedings of the 2022 25th International Conference on Computer and Information Technology (ICCIT)*; 2022 Dec 17–19; Cox's Bazar, Bangladesh. doi:10.1109/ICCIT57492.2022.10054872.
6. Yadav V, Verma P, Katiyar V. Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages. *Int J Inf Technol*. 2023;15(2):759–72. doi:10.1007/s41870-022-01010-y.
7. Su D. An ensemble model for sentiment analysis. In: *Proceedings of the 3rd International Conference on Information Technologies and Electrical Engineering*; 2020 Dec 3–5; Changde, China. doi:10.1145/3452940.3453022.
8. Arobi R, Rijon RH, Haque N, Safkat TI, Safoan S. Sentiment analysis on E-commerce based product reviews using machine learning algorithms [dissertation]. Dhaka, Bangladesh: Brac University; 2022.
9. Duwairi R, El-Orfali M. A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *J Inf Sci*. 2014;40(4):501–13. doi:10.1177/0165551514534143.
10. Lian H, Lu C, Li S, Zhao Y, Tang C, Zong Y. A survey of deep learning-based multimodal emotion recognition: speech, text, and face. *Entropy*. 2023;25(10):1440. doi:10.3390/e25101440.
11. Maitama JZ, Idris N, Abdi A, Shuib L, Fauzi R. A systematic review on implicit and explicit aspect extraction in sentiment analysis. *IEEE Access*. 2020;8:194166–91. doi:10.1109/ACCESS.2020.3031217.
12. Rokhsaritalemi S, Sadeghi-Niaraki A, Choi SM. Exploring emotion analysis using artificial intelligence, geospatial information systems, and extended reality for urban services. *IEEE Access*. 2023;11:92478–95. doi:10.1109/ACCESS.2023.3307639.
13. Barredo Ibáñez D, Molina Rodríguez-Navas P, Medranda Morales NJ, Rodríguez Breijo V. Health transparency and communication on the government websites of Ibero-American countries: the cases of Chile, Colombia, Ecuador, and Spain. *Int J Environ Res Public Health*. 2021;18(12):6222. doi:10.3390/ijerph18126222.
14. Rana TA, Shahzadi K, Rana T, Arshad A, Tubishat M. An unsupervised approach for sentiment analysis on social media short text classification in Roman Urdu. *ACM Trans Asian Low-Resour Lang Inf Process*. 2022;21(2):1–16. doi:10.1145/3474119.
15. Pingle A, Vyawahare A, Joshi I, Tangsali R, Kale G, Joshi R. Robust sentiment analysis for low resource languages using data augmentation approaches: a case study in Marathi. *arXiv:2310.00734*. 2023.
16. Hariguna T, Ruangkanjanases A. Adaptive sentiment analysis using multioutput classification: a performance comparison. *PeerJ Comput Sci*. 2023;9(4):e1378. doi:10.7717/peerj-cs.1378.