# A Comparative Analysis of Sentiment Classification Models for Improved Performance Optimization

**Varun Iyer**

*Received December 12, 2024*
*Accepted April 15, 2024*
*Electronic access May 15, 2024*

Since its inception, the domain of Natural Language Processing has placed a significant onus on AI/ML engineers to formulate and optimise machine learning models for sentiment analysis. This research aims to contribute a perspective to the question of the accuracy of machine learning models - both simple and complex - in ascertaining sentiment, and to elucidate methods for optimizing their efficacy, with a particular focus on the role of preprocessing techniques and vectorizers. In pursuit of this objective, this study intends to identify the strengths and weaknesses of diverse sentiment analysis models, by conducting a comparative analysis of their performance based on Accuracy, Precision, Recall, and F1 Scores. The research methodology entails the use of a vast range of models, including Decision Tree, KNN, Logistic Regression, Naive Bayes, Perceptron, Random Forest, SVM, LSTM, and Bi-LSTM. The study further employs an array of pre-processing techniques, including Contraction Handling, Stopwords, Lemmatization, and Negation Handling, as well as feature extraction methods such as Bag of Words and TFIDF. Finally, the research highlights the Logistic Regression classification model to be the best-performing model across all metrics.

**Keywords:** Natural Language Processing, Sentiment Analysis, Machine Learning Models, Accuracy, Pre-processing Techniques, Feature Selection, Dimensionality Reduction, Comparative Analysis, Performance Metrics, Model Optimization

## Introduction

The domain of Natural Language Processing has placed a significant onus on AI/ML engineers to formulate and optimize machine learning models for sentiment analysis since its inception. With the unprecedented proliferation of social media and the internet, sentiment analysis has gained unparalleled momentum in terms of its pivotal relevance for businesses, governments, and individuals alike, seeking to glean insights into public opinion, sentiment trends, and customer feedback. Several studies have thus been conducted, to comparatively analyse, comment and explicate the variances and underlying rationales for these variances in the accuracy of sentiment analysis models [1]. Nevertheless, despite numerous models proposed and tested, a wholly efficient and accurate model is yet to be unequivocally proposed. Hence, the author of this paper perceives that unveiling the true underlying principles and mechanics of these models, along with the preprocessing techniques and feature selection methods, would constitute a significant primary step in our arduous journey towards the attainment of a sentiment analyser that is optimally efficient, to the best of our computational abilities.

The first objective/aim of this comparative study is an attempt to uncover the strengths and weaknesses of AI/ML models by comparing their performance in terms of accuracy, speed, and efficiency, with accuracy being given the greatest priority. An effective measure that the author of this paper has employed in discovering the models' true identity, is in creating a matrix of the most popular incorrectly predicted words by the model. This allows us to better understand where and why the model finds certain intricacies in language puzzling. In doing so, we hope to come upon some sense of a consensus on which models work the best for each scenario, although finding its specifics is possibly outside the scope of this study and a means for certain rewarding research.

Another aspect of the issue that the author finds particularly relevant as a topic for further consideration and study is analysing the choice and use of preprocessing techniques that are employed on text, with common methods including data cleaning, feature selection, and dimensionality reduction In doing so, this paper aims to devise/suggest improvements in current preprocessing techniques to make features better suited for natural language processing in machine learning models, either changing the way they are read by the model or perhaps even removing them completely.

The paper's structure includes an abstract that provides a summary of the study's objectives and reasoning, an introduction that highlights the paper's importance and outlines its primary aims, a literature review that examines existing academic literature on comparative sentiment analysis studies and other papers related to the study's methodology and further objectives. Additionally, a methodology section describes the methods/techniques used in the study, including models, prepro-

cessing techniques, and datasets, while the discussion section explores the valuable information gleaned from the study. Furthermore, a discussion section interprets and rationalizes the results, suggesting ways existing methods could be improved, followed by a conclusion that answers the research question, identifies current limitations, and outlines potential areas for future study.

## Literature Review

In this research paper, we aim to present a comprehensive analysis of existing studies in the field of sentiment analysis. The primary objective of this literature review is to offer a holistic understanding of sentiment analysis models and preprocessing techniques, to optimize their performance.

Each paper under review presents unique insights and findings derived from diverse datasets, experimental setups, and evaluation metrics. Through the synthesis and comparison of these studies, our review aims to shed light on the current state-of-the-art sentiment analysis models and techniques.

### Model Performance

In a comparative study conducted by Liu, Siqi[2], it was found that simpler models, such as Logistic Regression and Support Vector Machines, exhibited greater efficacy in predicting sentiments within Yelp reviews compared to more complex models, including Gradient Boosting, LSTM, and BERT.

Al-Otaibi, Shaha, and Al-Rasheed, Amal[3], arranged sentiment analysis models in descending order of accuracy. The ranking included the hybrid heterogeneous SVM, Attentional-graph Neural Network, Recurrent Neural Networks, Corpus-based approach with SVM, TF-IDF with CNN, SVM, Lexicon-based with TF-IDF, Corpus-based using KNN, and CNN models.

Pang et al.[4] conducted research that examined the effects of preprocessing techniques and feature selection on model performance. Notably, their findings demonstrated that better performance, particularly for SVMs, was achieved by considering feature presence rather than feature frequency. This observation contrasts interestingly with the work of McCallum and Nigam (1998)[5], who studied Naive Bayes topic classification. These contrasting results indicate a distinction between sentiment and topic categorization, warranting further investigation. Furthermore, Pang et al. discovered that POS tagging had minimal impact on the outcome, with marginal improvements for Naive Bayes systems and decreased accuracy for SVMs.

### Preprocessing Techniques

Liu, Siqi, reaffirmed the findings of Saif, Hassan, et al.[6], demonstrating that model performance decreases after removing stopwords from the dataset. Interestingly, Liu (following the recommendations laid out in Saif, Hassan, et al.) attempted to eliminate infrequent terms instead of stopwords, resulting in a significant improvement in overfitting and the overall F1 score. This technique also led to a substantial reduction of over 85% in the feature space, from 2,130,891 to 264,977, resulting in faster training and testing times. Consequently, the elimination of infrequent terms may be a superior preprocessing technique compared to stopword removal, particularly in commercial applications where time efficiency is crucial.

Liu also investigated the effects of normalization on sentiment analysis models. While a slight improvement in accuracy was observed, normalization had a more proportionate effect on reducing overfitting and feature space, thereby aiding in overall runtime reduction. Hence, normalization may not significantly enhance accuracy but remains vital for commercial applications of sentiment analysis.

Magliani et al.[7] examined various preprocessing techniques' effects on model accuracy. Although these techniques enhanced accuracy overall, the usage of a dictionary during the preprocessing stage had no discernible impact on the results while increasing preprocessing time. Consequently, the present study refrains from employing a dictionary, influenced by the results of this research.

### Feature Extraction Methods

Khoo et al[8]. compared lexicon-based feature extraction methods with vectorization methods, such as the bag-of-words classifier. Their study demonstrated that lexicon-based models exhibited significantly lower accuracy compared to classifier models. Lexicon-based models achieved accuracy rates of 75% to 77%, with no significant differences among them, whereas machine learning models utilizing the bag-of-words approach easily surpassed 80% accuracy. Although lexicons may provide baseline scores of 75% without trainings, making them suitable for commercial applications where accuracy can be slightly compromised, their limited accuracy is a significant reason why lexicon-based classifiers were not explored in this study.

Gonçalves et al[9]. and Avinash, M. & Elango, Sivasankar[10], corroborated their findings, indicating that the accuracy of feature extraction techniques heavily depends on the data's nature. To optimize models for commercial and research applications, understanding which classifiers perform best in specific situations is essential. Consequently, this study employed multiple vectorizers to compare model performances when utilizing different extraction formats.

Through an extensive literature review, we have examined the implications and findings of previous research on sentiment analysis. These studies have shed light on the performance of sentiment analysis models, the impact of preprocessing techniques, and the effectiveness of various feature extraction methods. Bearing these studies in mind, we hope to better calibrate

our study to follow the guidelines and stray away from clearly established errors.

## Results

### Model Performance

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|---|---|---|---|---|
| Decision Tree | 69.54 | 71.53 | 69.54 | 68.82 |
| Logistic Regression | 77.61 | 77.67 | 77.61 | 77.6 |
| Linear SVM | 75.21 | 75.24 | 75.21 | 75.2 |
| Perceptron | 70.95 | 70.98 | 70.95 | 70.94 |
| KNN | 68.74 | 68.97 | 68.74 | 68.64 |
| Multinomial Naive Bayes | 76.52 | 76.56 | 76.52 | 76.51 |

**Table 1** Performance with the TF-IDF Vectorizer

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|---|---|---|---|---|
| Decision Tree | 68.73 | 70.72 | 68.73 | 67.96 |
| Logistic Regression | 77.81 | 77.85 | 77.81 | 77.8 |
| Linear SVM | 75.64 | 75.66 | 75.64 | 75.64 |
| Perceptron | 69.88 | 69.91 | 69.88 | 69.87 |
| KNN | 63.39 | 65.34 | 63.39 | 62.2 |
| Multinomial Naive Bayes | 76.21 | 76.26 | 76.21 | 76.2 |

**Table 2** Performance Metrics with the Bag-of-Words Vectorizer

| Model | Precision (%) | Recall (%) | F1 Score (%) |
|---|---|---|---|
| LSTM | 73 | 73 | 73 |
| Bi-LSTM | 74 | 74 | 74 |

**Table 3** Performance of RNNs

## Performance Analysis Based on Text Length Variations

### For Bag-of-Words Vectorized Models

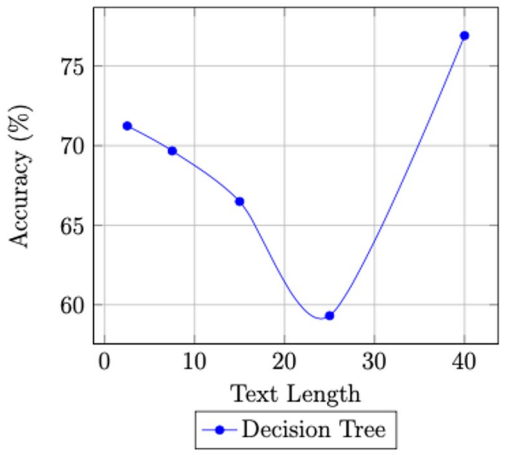


**Fig. 1** Performance of Decision Tree Across Different Text Lengths

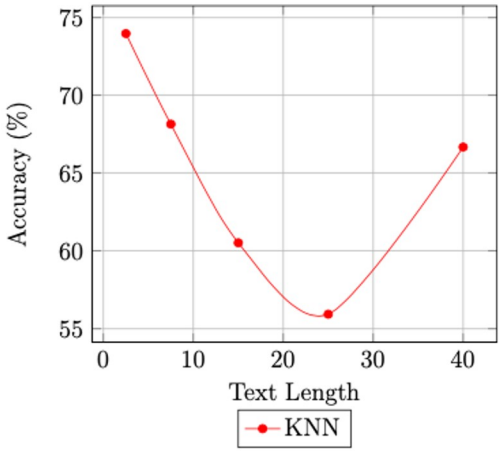


**Fig. 2** Performance of KNN Across Different Text Lengths

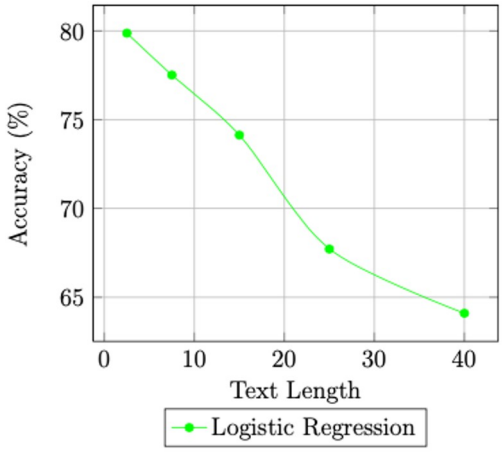


**Fig. 3** Performance of Logistic Regression Across Different Text Lengths

### For TF-IDF Vectorized Models

## Discussion

The tabulated data herein delineates the mean outcomes derived from multiple iterations, concomitant with parameter refinement, conducted in the course of this inquiry. The ensuing analyses and discussions are predicated upon these empirical results.

Before delving into the specifics of our models, it is noteworthy to observe a consistent trend in the empirical data. These models, despite their structural diversity, exhibit similar misclassifications when utilizing identical vectorization methods. This observation underscores the significant influence of the vectorization process on the efficacy of sentiment analysis models, potentially overshadowing the model's inherent structural intricacies. It emphasizes the importance of thoughtfully se-

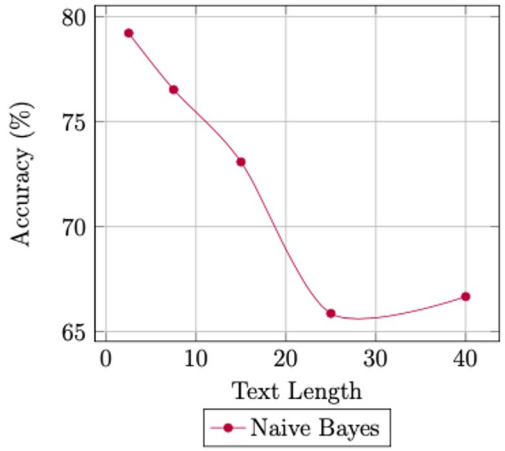


**Fig. 4** Performance of Multinomial Naive Bayes Across Different Text Lengths

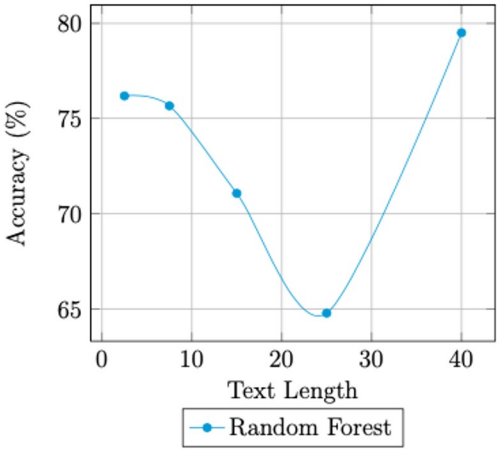


**Fig. 6** Performance of Random Forest Across Different Text Lengths

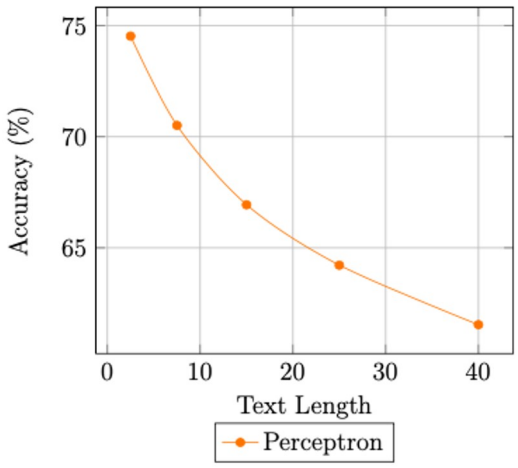


**Fig. 5** Performance of Perceptron Across Different Text Lengths

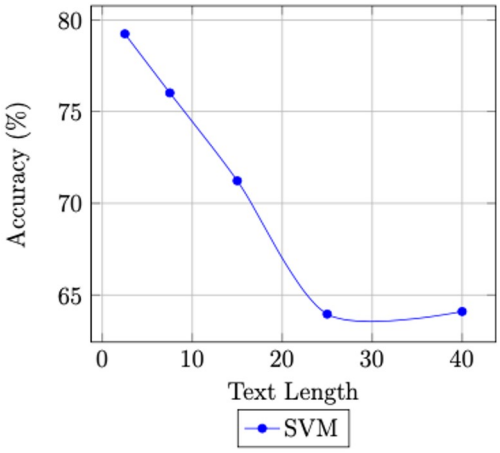


**Fig. 7** Performance of Support Vector Machine Across Different Text Lengths

lecting and fine-tuning the vectorizer, taking into account its configuration parameters and their alignment with the dataset, as a pivotal factor in enhancing sentiment analysis accuracy.

It is of particular interest to observe that both the Bag of Words (BoW) and TFIDF models yielded strikingly similar misclassification statistics. This intriguing alignment in their performance metrics suggests underlying commonalities in their respective approaches to sentiment analysis.

One plausible explanation for the congruence in misclassifications between Bag-of-Words (BOW) and TF-IDF models lies deeply entrenched in their shared reliance on word frequency as a primary determinant of sentiment. Both methodologies, while divergent in nuances, gravitate towards assessing textual content based on the prevalence of specific words within the corpus. TF-IDF, accounting for term rarity and discriminatory power, and BOW, simplifying texts into word occurrences, converge in this reliance on word frequency. However, this mutual emphasis on word frequency manifests as a foundational limitation, especially when confronted with complex contextual understanding.

The models' reliance on frequently occurring terms creates challenges in interpreting nuanced sentiment implications embedded within the usage of commonly encountered words. They cannot discern relationships between words based on their order or position in a sentence, thereby hindering their comprehension of the contextual flow and meaning of the text. This absence of sequential analysis impedes the models' ability to grasp grammar intricacies and sentence structures, leading to misinterpretations and inadequate sentiment analysis.

Another critical aspect contributing to misclassifications is the models' incapacity to retain memory of previously stated

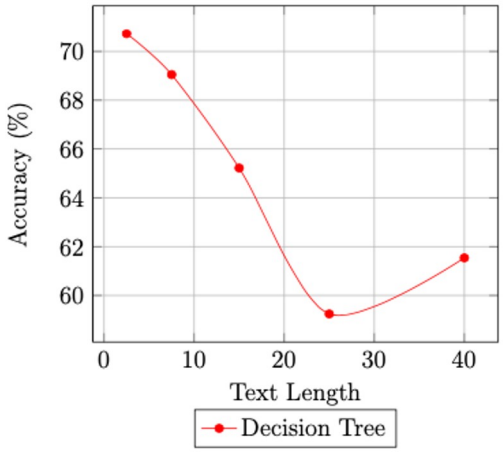


**Fig. 8** Performance of Decision Tree Across Different Text Lengths

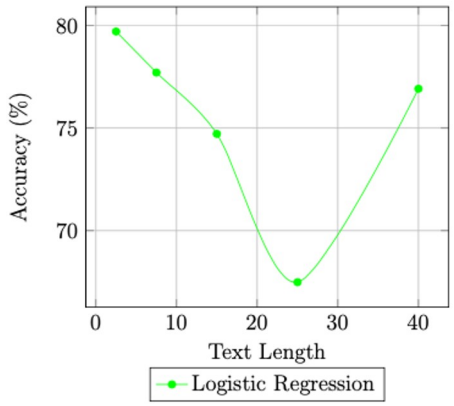


**Fig. 9** Performance of Logistic Regression Across Different Text Lengths

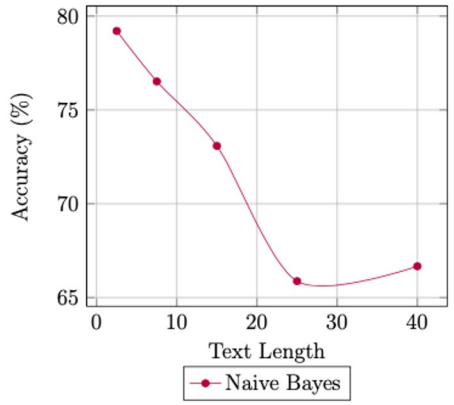


**Fig. 10** Performance of Multinomial Naive Bayes Across Different Text Lengths

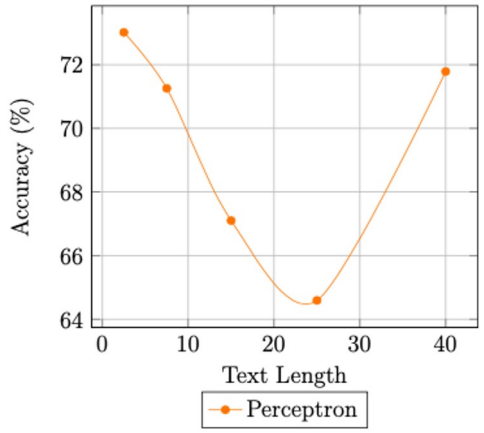


**Fig. 11** Performance of Perceptron Across Different Text Lengths

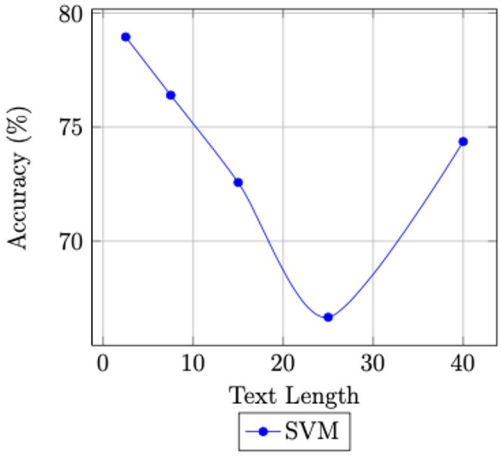


**Fig. 12** Performance of Support Vector Machine Across Different Text Lengths

words and their contextual meanings throughout the text. This absence of memory prevents a holistic interpretation of the text, disregarding the cumulative impact of earlier phrases on subsequent sentiment expressions, thereby hindering accurate sentiment analysis.

Furthermore, both BOW and TF-IDF models exhibit insensitivity to the register and style of delivery, rendering them incapable of handling slang, dialects, or understanding sarcasm. The models' lack of comprehension regarding the essence of the conversation, including discourse coherence and thematic shifts, poses additional challenges. Their inability to capture the evolving nature of conversations and the underlying essence further contributes to comparable misclassification statistics between these models.

Both vectorization methodologies have been subjected to meticulous examination in this inquiry. This evaluation is cen-

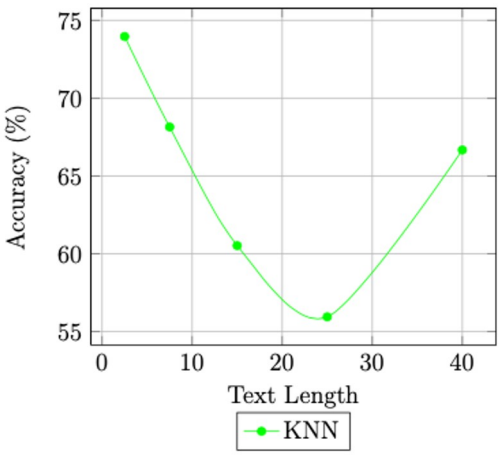


**Fig. 13** Performance of KNN Across Different Text Lengths

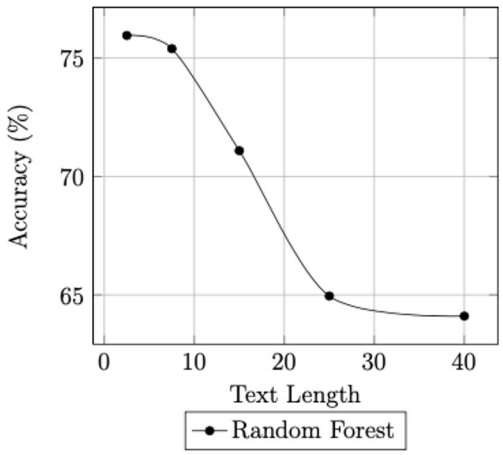


**Fig. 14** Performance of Random Forest Across Different Text Lengths

tred on comprehending the intricacies of the vectorization process and, by extension, assessing the effectiveness of the resulting models in accommodating subtle linguistic nuances and contextual intricacies, particularly in comparison to more intricate methodologies.

A notable limitation manifests in the vectorizers' treatment of contractions, especially those that are misspelt. This discernible inadequacy significantly hampers their capacity to discern sentiment in domains characterized by informal or unedited text, potentially resulting in significant misclassifications.

Another noteworthy challenge encountered by these frequency-based vectorizers pertains to the differentiation of homonyms, as supported by our empirical findings evidenced in the phrases 'two' and 'too'. This observation underscores the need for heightened contextual discernment within the vectorizers' architectural framework, emphasizing that the recurring misclassification of homonyms constitutes an inherent trait across all frequency-based vectorizers, regardless of the models' architectural structure.

The interpretation of work-related terminology emerges as an additional area of complexity in this study. Terms such as 'get', 'go', and 'work' are susceptible to misinterpretation by these models, attesting to the vectorizers' proclivities towards frequency-based analytical paradigms, often to the exclusion of a comprehensive consideration of contextual semantics.

A related concern arises in the vectorizers' noticeable difficulty with temporal expressions. This points to a significant challenge to vectorizers' contextual analysis, potentially leading to consequential inaccuracies in sentiment classification. This limitation becomes particularly pronounced in contexts where the timing of sentiment expression holds substantial importance.

Negations in linguistic expressions, encapsulated by the exemplar 'cannot', emerge as a recurrently encountered challenge for these frequency-based vectorizers, and their subsequent models, which demonstrate a propensity for ineffective recognition of the implicit sentiment reversal concomitant to negatory expressions. This deficiency necessitates focused ameliorative measures, potentially involving nuanced handling of negations or the adoption of specific vectorization methodologies, to engender heightened precision in sentiment analysis, particularly in contexts wherein negations bear a significant contextual impact on the corpus' meaning.

The numerical expanse in textual data poses a formidable challenge to most models subjected to our scrutiny in this study. The models in this investigation demonstrate a proclivity towards treating individual numbers as discrete units, thereby impeding their capacity to discern contextual nuances and accurately construe numerical information within the text. To surmount this constraint, it is imperative to incorporate specialised methodologies that consider the contextual environment in which numbers are situated. This augmentation holds the potential to enhance the models' adeptness in processing numerical data with greater fidelity and precision.

Emotional expressions, especially those of polarizing valence such as 'sad' and 'good', were significantly misclassified in this investigation. This posits that a fundamental reliance on frequency-based analytical paradigms may not comprehensively encapsulate the emotional subtleties latent within such lexical constructs. This perceived shortcoming underscores the urgent need for refining the vectorizers'/models' responsiveness to emotional nuance, warranting potential recourse to differential weighting schemes for such expressions.

Polysemy, as epitomized by the frequently misclassified term 'back', introduces a distinctive challenge within the ambit of frequency-vectorized models. This phenomenon attests to a heightened sensitivity towards terms engendering multiple and divergent interpretations or connotations. This characteristic accentuates the urgent requisite for an enriched level of con-

textual discernment, thereby advocating for the employment of differentiation techniques, particularly pertinent to polysemous lexical entries, within the vectorizer's architectural framework.

As demonstrated in the results section of this paper, we measured the varying accuracy models when reacting to different text lengths. It is important to note that the maximum length of text in the corpus was 222 characters, and the minimum length was 1 character.

When graphing the accuracy of models across text lengths, we found varying shapes. Dealing with bag-of-words vectorized models, distinct patterns emerged in the performance of Decision Tree, Random Forest, and KNN models across varying word counts. Notably, these models showcased a consistent dip in performance as the word count increased, aligning with the challenges of handling longer text segments. However, a compelling deviation surfaced when examining a corpus length of 30-50 words: an upsurge in performance was observed. This unexpected enhancement in model accuracy suggests a distinctive characteristic of tweets within this word count range, exhibiting more discernible splits conducive to effective node classification by Decision Trees. This phenomenon highlights the models' proficiency in handling shorter texts, leveraging simpler and more effective feature splits based on a smaller, yet more distinct set of features. Meanwhile, Logistic Regression consistently demonstrated decreased performance as word count increased, indicating its struggle to effectively process longer texts within the Bag-of-Words representation.

In our exploration utilizing TF-IDF vectorized models, we witnessed a stark reversal in performance when compared to bag-of-words models. Notably, Decision Trees, Random Forests, and Naive Bayes consistently exhibited a decline in performance as the length of the corpus increased. This consistent trend suggested these models struggled with the complexities introduced by longer text segments within the TF-IDF representation. However, an intriguing contrast surfaced among Logistic Regression, Perceptron, and KNN models, demonstrating increased performance specifically when presented with a corpus length ranging from 30 to 50 words. This unique behaviour suggested that these models leveraged the TF-IDF representation to better handle longer text segments within this specific word count range, indicating a potential advantage in discerning important features or patterns within such texts.

When comparing our observations with those set out by Jain and Tongia[11] (sentiment analysis on movie reviews), we were surprised to see the stark contrast present in terms of model performance and behaviour. Both studies illuminate the influence of text length on model performance, aligning with the observation that shorter reviews tend to yield higher accuracy with specific models. In congruence with Jain and Tongia's research, our study utilizing Bag-of-Words models showcases a decline in accuracy for Decision Trees and Naive Bayes models as word count increases, resonating with their findings regarding diminishing accuracy in longer texts. However, an interesting departure emerges with TF-IDF models, where our study delineates a reversal in performance characteristics for various models with longer texts, distinct from their observed patterns. While Jain and Tongia highlighted consistent performance in SVM and Logistic Regression across review lengths, our study reveals nuances in Logistic Regression's behaviour within TF-IDF models, demonstrating improved accuracy in longer texts within a specific word count range, raising questions regarding corpus length-specific behaviour for certain models.

The comparison between Term Frequency-Inverse Document Frequency (TF-IDF) models and their Bag of Words (BOW) counterparts unveils notable performance disparities, consistently favouring TF-IDF approaches. The TF-IDF models consistently outperformed their BOW equivalents across various metrics, indicating the superiority of TF-IDF in capturing sentiment nuances within textual data. Several factors contribute to this divergence in performance.

TF-IDF models leverage a more nuanced representation of textual data compared to BOW models. By considering the importance of words not just based on their frequency within a document but also their rarity across the entire corpus, TF-IDF captures more meaningful and discriminative terms. This nuanced approach likely enables TF-IDF models to better discern subtle sentiment variations, leading to their superior performance in sentiment analysis tasks.

Within the realm of TF-IDF models, Logistic Regression emerges as the standout performer, consistently exhibiting the best performance across different word count intervals. The robustness of Logistic Regression in handling sentiment analysis tasks can be attributed to its simplicity, efficient handling of sparse data, and ability to model linear relationships between features and sentiment labels. The logistic regression model's ability to generalize and derive meaningful patterns from TF-IDF representations contributes significantly to its superior performance.

Furthermore, Logistic Regression's remarkable stability and minimal fluctuation when dealing with differing text lengths highlight the model's capability to maintain consistent sentiment analysis results irrespective of variations in the number of words considered. This phenomenon might stem from the model's capacity to assign appropriate weights to words based on their importance in the entire corpus, leading to stable predictions across different textual lengths.

In contrast, models such as Random Forest and Decision Tree, while performing reasonably well, display a distinct pattern in their performance concerning word length. These models excel particularly with shorter word lengths, likely due to their capacity to capture concise and more definitive sentiment patterns within shorter textual contexts. However, their performance tends to decrease as the word length increases, indicating a struggle in handling longer texts or deriving sentiment insights

from lengthier contexts effectively.

The inherent nature of Random Forest and Decision Tree models might cause this decline in performance with increasing word length. These models, reliant on decision trees and ensemble learning, might face challenges in effectively capturing nuanced sentiment variations within longer texts, resulting in reduced accuracy and performance as the text length grows.

## Methodology

This section encompasses the exploration of datasets, preprocessing techniques, feature extraction methods, models, and evaluation metrics. These components were meticulously selected to provide comprehensive insights into the effectiveness of different sentiment analysis approaches.

### Data Selection

Regarding the dataset, we utilized the Sentiment140[12] dataset, which comprises 1.6 million tweets extracted from Twitter using its API. The authors went with the Sentiment140 dataset, compared to possible alternatives such as the Multi-Domain Sentiment Dataset (consisting of Amazon reviews), as they found the heightened existence of slang and real-time/everyday language would provide insights on the real-world performance of models and vectorizers alike. In this study, we focused on two fields: the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive) and the tweet text itself (excluding emoticons). The dataset was automatically collected based on positive and negative emotions, following the approach proposed by Go et al. To enhance the dataset's accuracy, we propose incorporating human annotation, particularly for sarcastic comments, to establish more reliable ground truths. Additionally, incorporating data from other social media platforms such as Reddit could augment the dataset's credibility by providing diverse text forms and sentence structures.

### Preprocessing

For preprocessing, our study employed four main stages. First, we used contraction handling, to rectify any popularly used contractions (for example, "don't"), to simplify the corpus for models. Next, we attempted to handle commonly used slang, allowing for greater clarity, as the models could interpret these words with their traditional meaning. Moreover, we used negation handling, to make the corpus in terms of contradictions and provide the model with the direct meaning of the statement. Finally, lemmatization allowed us to further simplify text into its base forms. While choosing how to preprocess data, we reinforced our hypotheses about the techniques' effectiveness, by comparatively analysing model performance with and without each technique. Only techniques that yielded significantly better results (approx. rises of 4-7% in Accuracy, Precision, Recall and F1 Score) were chosen.

Contraction handling involved expanding contractions using a predefined list to ensure standardized English words. This simplification aimed to enhance corpus comprehension and optimize performance, especially for complex deep learning-based models. Slang handling utilises a comprehensive slang database to translate slang words into their conventional interpretations. For instance, "lol" was converted to "laugh out loud." This preprocessing step aimed to achieve the same goals as contraction handling.

The next preprocessing stage involved lemmatization, which leveraged NLTK's WordNet Lemmatizer based on WordNet[13], a lexical database for English. Lemmatization maps words to their base or root forms, selecting the most common form found in the WordNet database. Unlike stemming, which can result in unintelligible words due to prefix removal, lemmatization considers the actual forms existing in the English language. This choice was made to ensure improved accuracy.

The final preprocessing stage involved negation handling, which inverted the polarity of opinionated words affected by negatory phrases. This step aimed to enhance corpus interpretability for models and future research.

A commonly employed preprocessing technique, when it comes to sentiment analysis endeavours, is Parts of Speech tagging (referred to as POS tagging). Here, words in a corpus are assigned a particular grammatical feature/part of speech (adverb, adjective, verb, etc.), making it slightly easier to understand the role of each word in a sentence. However, we decided not to employ POS tagging based on Pang et al.'s findings that it had no significant impact on accuracy. Additionally, spell-checking methods could have positively influenced accuracy, particularly for social media-based databases. However, practical implementation was challenging due to substantial processing times.

For feature extraction, our study opted for vectorizers, specifically the bag-of-words model and the TF-IDF vectorizer, while avoiding lexical extraction techniques. This choice was motivated by the demonstrated performance limitations of lexical extraction techniques, as highlighted by Khoo et al.

### Vectorizers

In summary, the bag-of-words model represents the corpus as a multiset of words, disregarding grammar and word order but considering multiplicity. We chose this approach due to its simplicity of implementation and its inherent ability to reduce data dimensionality, despite lacking contextual understanding.

In contrast, the TF-IDF vectorizer considers both term frequency (as in the bag-of-words model) and inverse document frequency (the frequency of a word within the corpus). This approach aims to provide a more comprehensive understanding of the data for the models. The choice to employ both vectorizers

allowed us to compare the performance of the same models with different feature extraction techniques and assess the impact on various evaluation metrics.

### Models

Regarding the models used, we employed a diverse range to gain insights into how different factors, such as vectorizers, corpus type, and preprocessing techniques, influenced overall performance.

Firstly, a classification and regression tree variant of the standard decision tree model was utilized with a maximum node depth of 100. Secondly, a standard K-Neighbours Classifier was employed with a specified nearest neighbour limit of 3. Additionally, a standard logistic regression model with a maximum iteration limit of 10,000 was utilized (Note: the model incorporates a random number generator for feature selection, resulting in varied outcomes when run multiple times.).

The subsequent models consisted of a standard multinomial Naive Bayes classifier and a linear perceptron classifier with a maximum iteration limit of 1,000. Following these, a random forest classifier with a maximum node depth of 100 and a C-Support Vector Machine with a regularization parameter of 2 were employed.

The final two models were deep learning-based, namely LSTM and Bi-LSTM. The LSTM model was based on Hochreiter's 1997 proposal[14], with a dimensionality of 176 and a dropout rate (both regular and recurring) of 0.2. The Bi-LSTM model used the same LSTM specifications but employed a bidirectional wrapper, enabling bidirectional data flow.

These particular models were chosen to compare a wide variety of approaches towards sentiment analysis. Linear models were facilitated in the form of Logistic Regression, Perceptron and Naive Bayes classifiers. Moreover, we were interested in how tree-based approaches would fare, leading to the utilisation of Decision Tree and Random Forest algorithms. We employed instance-based learning, in the form of a K-Nearest Neighbors (KNN) algorithm, and experimented with Support Vector Machines as well, due to their unique approach toward learning. Finally, the usage of both LSTM and Bi-LSTM models allowed us to analyse the performance of neural networks in the study as well. Unfortunately, the usage of slightly more advanced neural network architectures, like BERT transformers, fell outside the scope of this study, due to their resource-intensive nature.

### Evaluating Metrics

Finally, to perform the comparative analysis, we employed six key evaluation metrics. Firstly, we utilized a confusion matrix to visualize algorithm performance and assess true positives, false positives, true negatives, and false negatives ratios. Additionally, we employed accuracy, which measures the proportion of correctly classified sentiments out of the total sentiments in the dataset, to determine the average correctness of the models.

Three more metrics were employed: precision, recall, and f1 score. Precision quantifies false positives, while recall represents true positives divided by the total number of positives. The f1 score calculates the weighted harmonic mean of the precision and recall metrics.

Lastly, we applied a k-means clustering algorithm to the misclassified data of each model to identify frequently misclassified words. This analysis aimed to identify patterns or the lack thereof in the models' behaviour, offering insights to optimize factors and improve model performance.

## Conclusion

This comprehensive study navigated the labyrinth of sentiment analysis models and preprocessing techniques, aiming to dissect their efficacy and unearth optimization pathways. Diving into an extensive comparative analysis, the study scrutinized diverse models, ranging from traditional algorithms to deep learning architectures, juxtaposing their performance across various metrics and vectorization methodologies.

Notably, both Bag-of-Words and TF-IDF models showcased recurrent misclassifications, predominantly influenced by their reliance on word frequency, constraining their contextual comprehension and impeding nuanced sentiment interpretation. The analysis further highlighted varying model behaviours concerning corpus length, indicating performance fluctuations contingent upon text length.

Addressing the core research query, the study delineated the complex landscape of sentiment analysis, unveiling the critical role of vectorization methodologies in model performance. Overall, the Logistic Regression classification model outperformed the rest across most metrics, although Linear SVM and Multinomial Naive Bayes demonstrated similarly impressive results. Moreover, TF-IDF models exhibited superior performance, emphasizing the significance of nuanced feature representation for sentiment interpretation.

Acknowledging its limitations, the study primarily delved into textual data from a single source—Twitter—limiting the generalizability of findings across diverse platforms. Future endeavours should encompass a broader spectrum of social media platforms and diverse datasets to fortify the study's findings. Additionally, exploring advanced contextual models or hybrid methodologies could unlock new avenues for sentiment analysis refinement.

In conclusion, this study serves as a beacon in the ongoing quest for optimized sentiment analysis models. While unveiling the nuances and challenges inherent in existing methodologies, it paves the way for future advancements, fostering a deeper understanding of sentiment interpretation within the realm of Natural Language Processing.

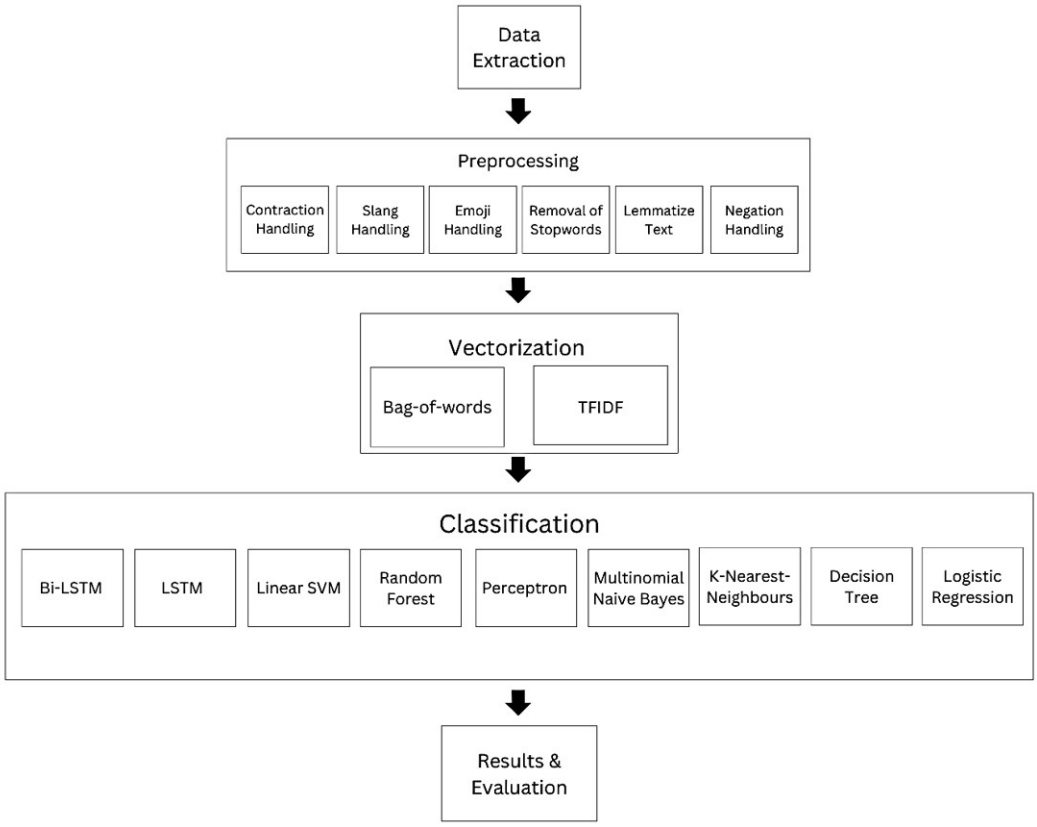


**Fig. 15** Workflow Model

## References


1 S. Ghosh, H. Arnab and R. Abhishek, *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*, p. 174–83.

2 S. Liu, *Sentiment Analysis of Yelp Reviews: A Comparison of Techniques and Models*, arXiv.org (2020).

3 T. Al-Otaibi and A. Al-Rasheed, *Informatica*, **46, 6**, year.

4 B. Pang and L. Lee, *Foundations and Trends® in Information Retrieval*, **2. 1–2**, 1–135.

5 K. McCallum, AAAI Conference on Artificial Intelligence.

6 H. Saif, H. Yulan and H. Alani, *The Semantic Web – ISWC*, p. 508–24.

7 T. Magliani, P. Fornacciari, S. Manicardi and E. Iotti, *A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter*.

8 C. Khoo and S. Johnkhan, *Journal of Information Science*, **44, 4**, 491–511.

9 P. Gonçalves, M. Araújo, F. Benevenuto and M. Cha, Proceedings of the First ACM Conference on Online Social Networks.

10 Madasu and S. Elango, *Multimedia Tools and Applications*, **79**, 9–10, 6313–35.

11 V. Jain, *International Journal of Science Engineering Development Research*, **7, 11**, 270–277.

12 Go, *Twitter Sentiment Classification using Distant Supervision*.

13 S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*.

14 S. Hochreiter and J. Schmidhuber, *Neural Computation*, **9, 8**, 1735–80.