Exploring The Impact of Feature Engineering and Data Organization on Sentiment Analysis of Twitter Data Using Machine Learning Algorithms

Chujie Deng

School Of Computer Sciences, Universiti Sains Malaysia, Penang, 11800, Malaysia

dengchujie@student.usm.my

Abstract. As one of the most popular micro-blogging platforms, Twitter generates millions of tweets daily, making manual sentiment analysis of such large volumes impractical. Consequently, leveraging machine learning algorithms for efficient sentiment analysis has become a critical challenge. This paper explores the performance of four machine learning algorithms—Logistic Regression (LR), Gaussian Naive Bayes (GNB), Decision Tree (DT), and Gradient Boosting Machine (GBM)—across four datasets of varying sizes. The models were trained using four feature extractors which include unigrams, bigrams, a combination of unigrams and bigrams, and the pre-trained Global Vectors (GloVe) word embedding model, with feature dimensions of 100, 200, and 300. The study reveals the impact of dataset size and feature combinations on the performance of these algorithms, identifying the most effective feature extraction methods. These findings provide valuable insights into the relationship between data scale, feature representation, and algorithmic performance, offering innovative perspectives for future research in sentiment analysis based on tweets.

Keywords: Twitter Sentiment Analysis; Machine Learning; Feature Engineering.

1. Introduction

Twitter, as one of the most popular micro-blogging platforms, has millions of active users posting tweets daily [1]. These tweets cover a wide range of topics, including daily life, product reviews, current events, and technological innovations [2]. Sentiment analysis of this vast amount of text can provide valuable insights for businesses, governments, and individuals [2]. However, with over a million new tweets generated daily, manually analyzing sentiment is impractical [3]. As a result, leveraging machine learning algorithms to automatically classify tweets as positive or negative has become one of the primary applications in natural language processing (NLP).

Despite its potential, analyzing sentiment on Twitter presents unique challenges due to the platform's limit, which encourages users to shorten their text by using slang, abbreviations, and emoticons [4]. Moreover, due to Twitter's casual and socialization-oriented property, tweets are often informal, with a high prevalence of spelling errors and polysemous words [4,5]. Compounding this difficulty is the fact that tweets cover a broad range of topics, unlike more structured text formats like news articles or blogs. These characteristics make sentiment analysis on Twitter particularly challenging, often leading to lower performance compared to other text sources.

To address these challenges, previous researchers have explored various approaches. The three most common methods in sentiment analysis are lexicon-based approaches, machine learning-based approaches, and hybrid models that combine the two [6-9]. In particular, supervised machine learning algorithms such as Stochastic Gradient Descent (SGD), Random Forest (RF), SailAil Sentiment Analyzer (SASA), Multi-Layer Perceptron (MLP), Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), and Support Vector Machine (SVM) have been extensively studied for their applications in sentiment analysis [10]. While these studies focus on the difference of performance of machine learning algorithms in sentiment analysis tasks, relatively little attention has been paid to how factors such as dataset size, feature extractors, and feature representation methods affect algorithm performance.

This study aims to fill that gap by systematically investigating the impact of feature engineering on model performance using a controlled variable approach. The experiment uses the same dataset as Go et al., consisting of 1.6 million balanced samples [11]. The dataset split into four datasets of varying sizes by using stratified sampling. And then feature extraction methods include unigram, bigram, unigram + bigram, and pre-trained Stanford Global Vectors (GloVe) 6B word embedding model were used to generate features with dimensions of 100, 200, and 300 [12]. These features are applied to four machine learning algorithms: Logistic Regression (LR), Gaussian Naive Bayes (GNB), Decision Tree (DT), and Gradient Boosting Machine (GBM). Additionally, a five-fold cross-validation is used to ensure the robustness of the results by minimizing errors caused by dataset splitting, which increased the generalizability of the findings.

The results of this experiment show that, regardless of the feature extraction method, the majority of models achieve peak performance when trained on a dataset of 100,000 samples. Additionally, higher dimensional feature representations generally lead to better performance. This trend is more pronounced in GBM and LR. This effect in GNB is relatively less noticeable. However, the performance of models trained by using DT could hardly show this and even have an adverse effect. The experiment also reveals that models using bigram and GloVe word embedding model as the extractor perform worse, while the performance of the unigram + bigram combination is similar with using unigram alone. Therefore, future training efforts should prioritize using GBM or LR with unigram or unigram + bigram feature extraction on datasets of around 100,000 samples, with higher-dimensional features where possible, to achieve optimal performance.

2. Methodology

2.1. Data Collection

This experiment utilizes the publicly available Sentiment140 dataset, which consists of six features: target, ids, date, flag, user, and text, with a total of 1.6 million samples. The dataset was collected via the Twitter API by extracting tweets containing specific emoticons. Based on the emoticons present, the tweets were automatically categorized into two sentiment classes: positive and negative. Go et al assumed that any tweet containing positive emoticons, like “:)” was positive, while tweets with negative emotions, like ":(" were classified as negative [11]. Tweets containing both positive and negative emotions were excluded from the dataset [11].

This dataset was selected due to its large size, allowing for easy partitioning into subsets of varying sizes to investigate the impact of dataset size on model performance. Additionally, the dataset is balanced, with 800,000 positive and 800,000 negative samples, which helps avoid the potential performance bias caused by class imbalance.

2.2. Data Preprocessing

During the preprocessing stage, since the dataset was initially categorized based on emoticons, all emoticons present in the text were replaced with their corresponding textual meanings. This step was taken to prevent the algorithms from learning based on emoticons, which could bias the model's performance due to the nature of the data collection process.

Additionally, Twitter text often contains repetitive elements such as URLs and user mentions. These were replaced with placeholders using regular expressions to maintain consistency. All characters were converted to lowercase, and punctuation and special characters were removed to reduce the complexity of the feature space. This process resulted in a text corpus consisting of words separated only by spaces, minimizing the number of features for later extraction.

Given Twitter’s informal and conversational nature, many tweets contain spelling errors, varied tenses, and filler words. While these variants often express the same meaning, they could increase the sparsity of features if treated as different words. To address this, the regular expressions were applied to reduce any repeated characters that appeared more than three times to the first two letters (e.g.,

"cooooool" → "cool"). Additionally, stopwords were removed using a predefined stopword list, and this work applied the SnowballStemmer to convert words to their base tense, standardizing the text.

Finally, the target and text columns were selected as the label and feature columns, respectively. After removing duplicates and null values, the dataset was split into four subsets of increasing size—D1 (1,000 samples), D2 (10,000 samples), D3 (100,000 samples), and D4 (the full dataset with 1,480,144 samples)—to investigate the effect of dataset size on model performance. Table 1 showed that the average tweet length across all subsets was approximately 40 words, with the longest tweets ranging between 100 and 200 words. Based on this, feature representations of 100, 200, and 300 dimensions were selected to explore the impact of varying feature dimensions on model performance.

Table 1. The statistical analysis of the text data in four datasets.

Data Name	Max Length	Min Length	Mean Length	25%	50%	75%
D1	118	3	42.46400	25.00	39.00	57.25
D2	122	3	42.68870	26.00	39.50	58.00
D3	128	2	42.76727	26.00	40.00	58.00
D4	174	1	42.84299	26.00	40.00	58.00

2.3. Feature Extraction

To explore different feature extraction methods and dimensional representations, traditional Term Frequency-Inverse Document Frequency (TF-IDF) was used to extract unigram, bigram, and a combined unigram + bigram feature set. In addition, the pre-trained GloVe 6B model from Stanford was employed to generate word embedding feature representations. This resulted in a total of four feature combinations, each of which was further represented in three different dimensionalities: 100, 200, and 300 dimensions.

For the combined feature set, the features were concatenated horizontally, which led to its dimensionality being twice as large as the other feature sets. To address this, Singular Value Decomposition (SVD) was applied to reduce the dimensionality while retaining as much information as possible.

In contrast, the word embeddings generated by the pre-trained GloVe 6B model are inherently three-dimensional. Since the four selected machine learning algorithms cannot process three-dimensional input, the feature set was flattened from its original format of (sample size, sequence length, embedding dimension) into a two-dimensional format of (sample size, sequence length * embedding dimension). To ensure that the final dimensionality of the features matched that of the other extraction methods, Gaussian Random Projection was used to reduce the dimensionality. While this method may not retain as much information as SVD, it offers a significantly faster computation speed on large datasets, making it more suitable for this context.

2.4. Model Selection

To comprehensively evaluate the impact of different datasets on model performance, a diverse set of machine learning algorithms was selected, each representing a distinct class of models. The chosen algorithms include: (1) LR, which represents models designed to handle linear relationships. (2) DT, a model that excels at capturing non-linear relationships. (3) GNB, chosen as a representative of the Bayesian models. (4) GBM, representing ensemble learning algorithms known for their robustness and performance.

This selection ensures a broad analysis of how various algorithmic approaches perform across different datasets and feature representations.

2.5. Evaluation Metrics

Accuracy, F1-score, and AUC-ROC were selected as the primary metrics to evaluate the performance of each model. These metrics provide a comprehensive assessment of the models' overall correctness and robustness. Accuracy offers insight into the model’s ability to correctly classify data

points, while F1-score balances precision and recall, particularly useful for imbalanced datasets. AUC-ROC evaluates the model's ability to distinguish between classes across different threshold settings, providing a deeper understanding of its classification performance. This combination of metrics ensures a thorough evaluation of model effectiveness from multiple perspectives.

3. Results

3.1. Experimental Setup

In the experimental setup, a five-fold cross-validation method was used to train the models. The average values of the performance metrics from the five training iterations were taken as the model's overall performance under each feature combination. This approach helps mitigate errors caused by dataset splitting and enhances the reliability of the experimental results. All experiments were conducted on a machine equipped with a Ryzen 7900X CPU and 64GB of RAM.

3.2. Results Analysis

The experiment evaluated the performance of four traditional machine learning algorithms (LR, DT, GNB, and GBM) on four datasets of varying sizes (1,000, 10,000, 100,000, and 1,480,144 samples). To achieve this, four different feature extractors were employed: unigram, bigram, combined (unigram + bigram), and the pre-trained Stanford GloVe 6B model. Each method was used to generate feature vectors in three dimensions: 100, 200, and 300. In total, the study involved training 192 different model configurations as Fig. 1 shows.

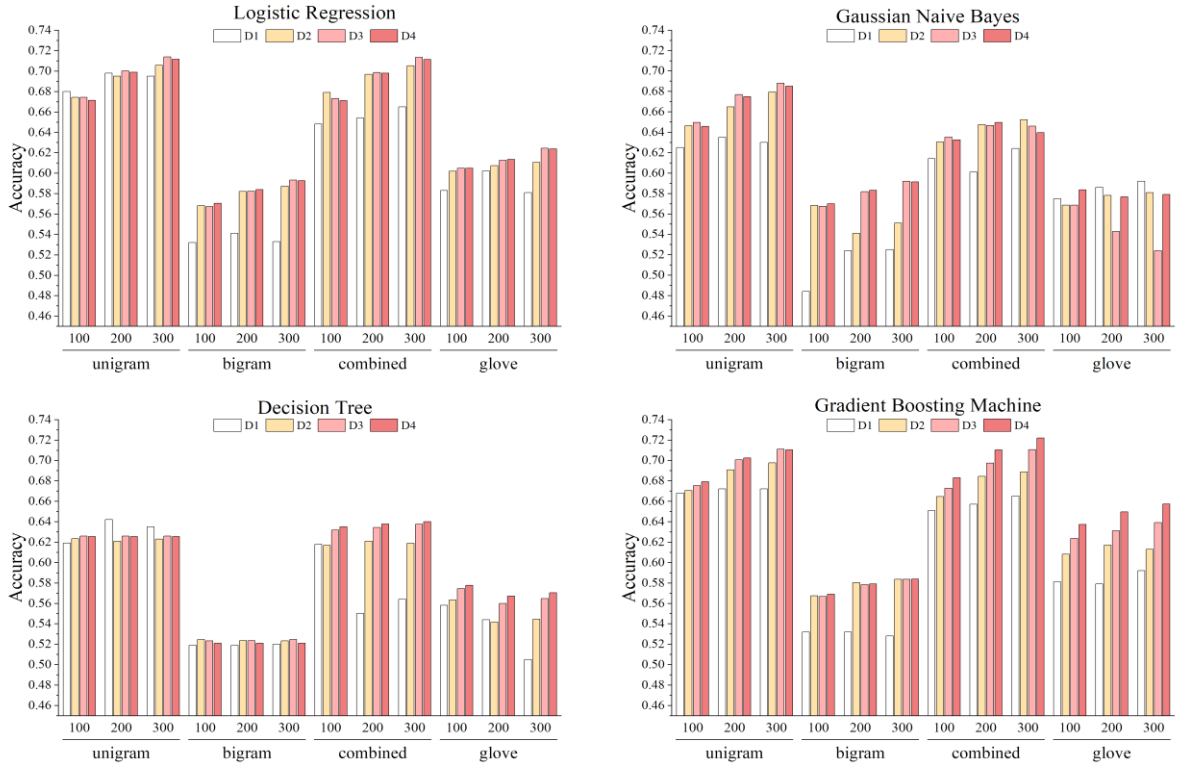

Fig. 1 The accuracy of four algorithms trained in different data sizes with different feature combinations (Figure Credit: Original).

Table 2 demonstrates the best model performance and feature combinations for each of the four algorithms across the datasets. It indicates that the GBM achieved the highest performance when applied to the largest dataset (D4) using the unigram + bigram feature extractor with a 300-dimensional feature set. Similarly, the LR model achieved a very close performance on the D3 dataset size when using the unigram feature set at 300 dimensions.

Table 2. The best performance model in each algorithm.

Model	Data Size	Extractor	Dimension	Accuracy	F1 score	AUC-ROC
LR	D3	unigram	300	0.71391	0.71335	0.71397
GNB	D3	unigram	300	0.68813	0.70655	0.68887
DT	D1	unigram	200	0.64200	0.62047	0.64150
GBM	D4	combined	300	0.72190	0.72398	0.72206

Although the accuracy of the LR and GBM models is generally similar, each algorithm excels in different dataset ranges. LR demonstrates better performance on smaller datasets, such as D1 and D2, where it outperforms GBM in most cases. Conversely, GBM shows stronger performance on the larger datasets, D3 and D4, where its ability to handle more complex data becomes more evident.

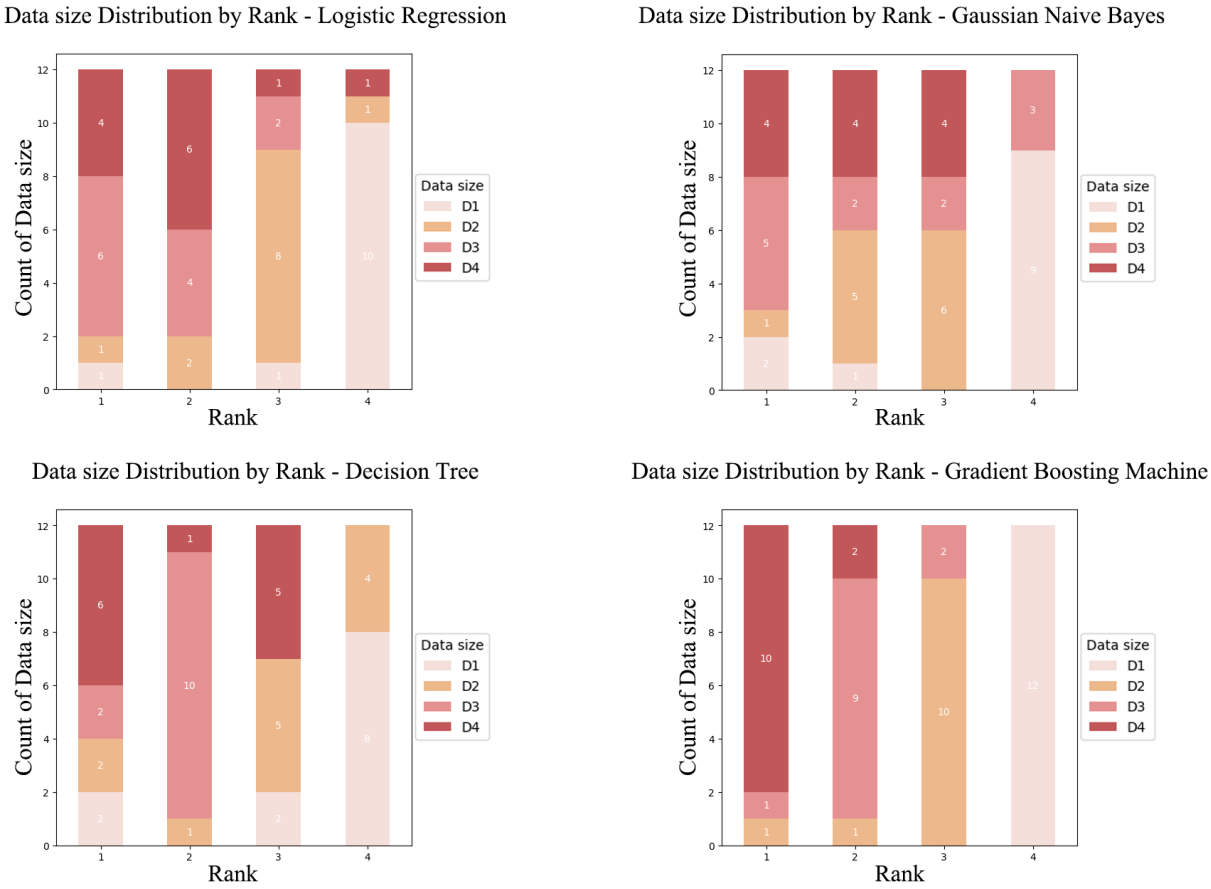

Fig. 2 Distribution of dataset sizes in the accuracy ranking of models trained using the same feature extractor and dimension (Figure Credit: Original).

The Fig. 2 indicate that, under the same feature combinations, the proportion of larger datasets increases as the accuracy ranking improves. The top two rankings are primarily occupied by the larger datasets, D3 and D4. This trend suggests that the accuracy of the models for these four algorithms generally increases with the dataset size. This phenomenon is more pronounced in more complex algorithms. In GBM model, larger datasets occupy the top rankings most frequently, followed by DT and LR while GNB algorithm shows this trend to the least extent.

Table 3 illustrates that, among the four algorithms, GNB and GBM are the most sensitive to dataset size, as evidenced by their average accuracy differences of 4.67292% and 4.63350%, respectively. Additionally, there is a noticeable variation between the maximum and minimum accuracy differences across all four algorithms. Using the relative difference formula:

$$Rleative\ Difference = \frac{maximum - minimum}{minimum} \quad (1)$$

It is observed that the relative differences for LR and DT are 10.27340 and 18.50444, while those for GNB and GBM are much lower, at 4.67239 and 5.36837. This indicates that the sensitivity to dataset size also varies significantly depending on the algorithms and feature combinations used.

Table 3. Impact of dataset size on accuracy for each algorithm with the same feature combination (difference between the highest and lowest accuracy in the same group).

Model	Minimum of accuracy difference (%)	Maximum of accuracy difference (%)	Mean of accuracy difference (%)
LR	0.53400	6.02000	3.12800
GNB	1.51400	8.58800	4.67292
DT	0.45000	8.77700	2.89467
GBM	1.10914	7.06341	4.63350

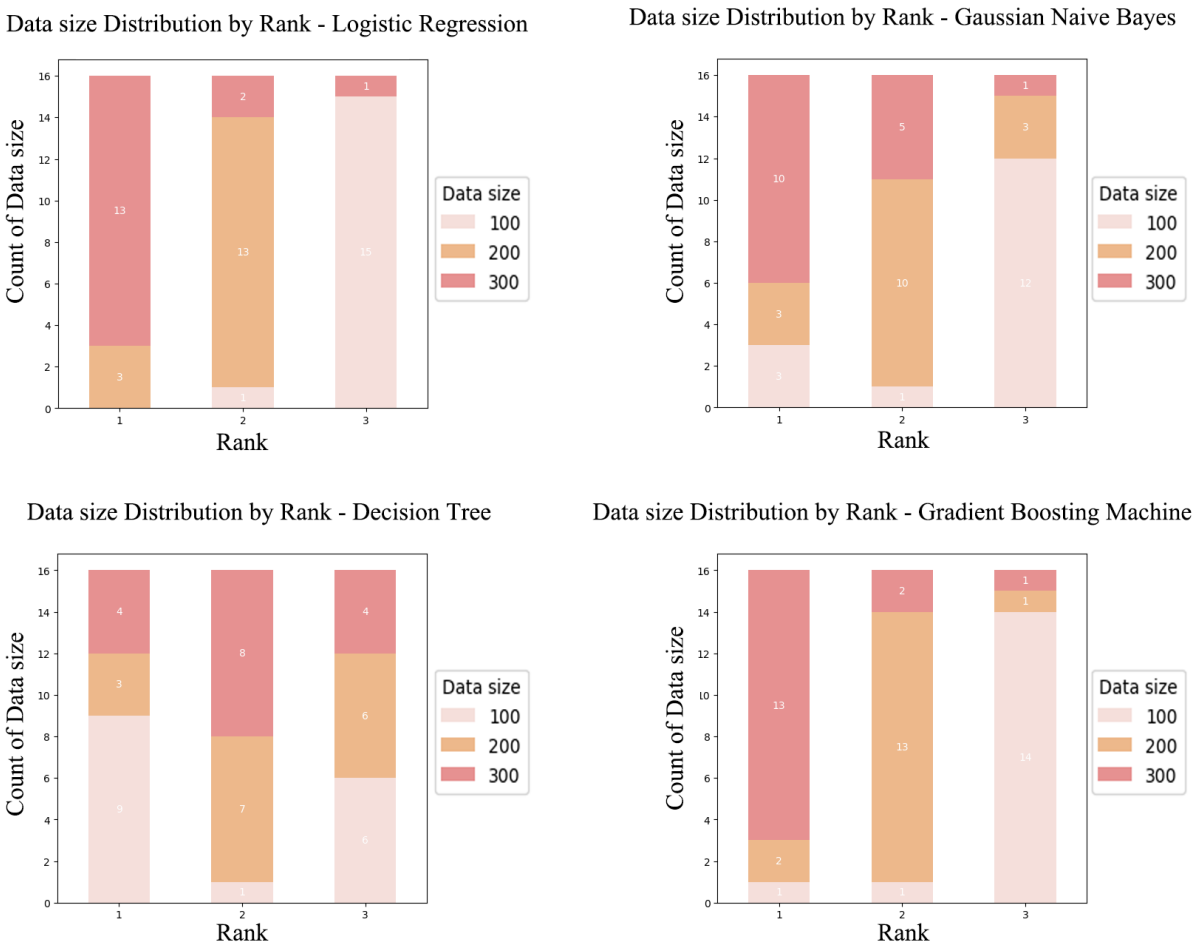

Fig. 3 Distribution of dimensions in the accuracy ranking of models trained using the same feature extractor and data size (Figure Credit: Original).

Fig. 3 reveals a pattern similar to the previous analysis of the impact of dataset size: as accuracy ranking improves, the proportion of high-dimensional feature representations increases. It is also more pronounced in more complex algorithms. However, the DT model shows better performance with lower-dimensional features. It can be observed that for all four algorithms, models using bigram or pre-trained GloVe 6B model as feature extractors tend to fall within the lower accuracy range.

On the other hand, Fig. 4 illustrates that model using unigram as the feature extractor consistently achieve higher accuracy, followed by those using the combined feature extractor. However, given the lower accuracy performance of models using bigram alone, it can be inferred that the primary contribution in the combined extractor comes from the unigram features. Simply combining unigram and bigram features does not seem to allow the model to capture more informative features effectively.

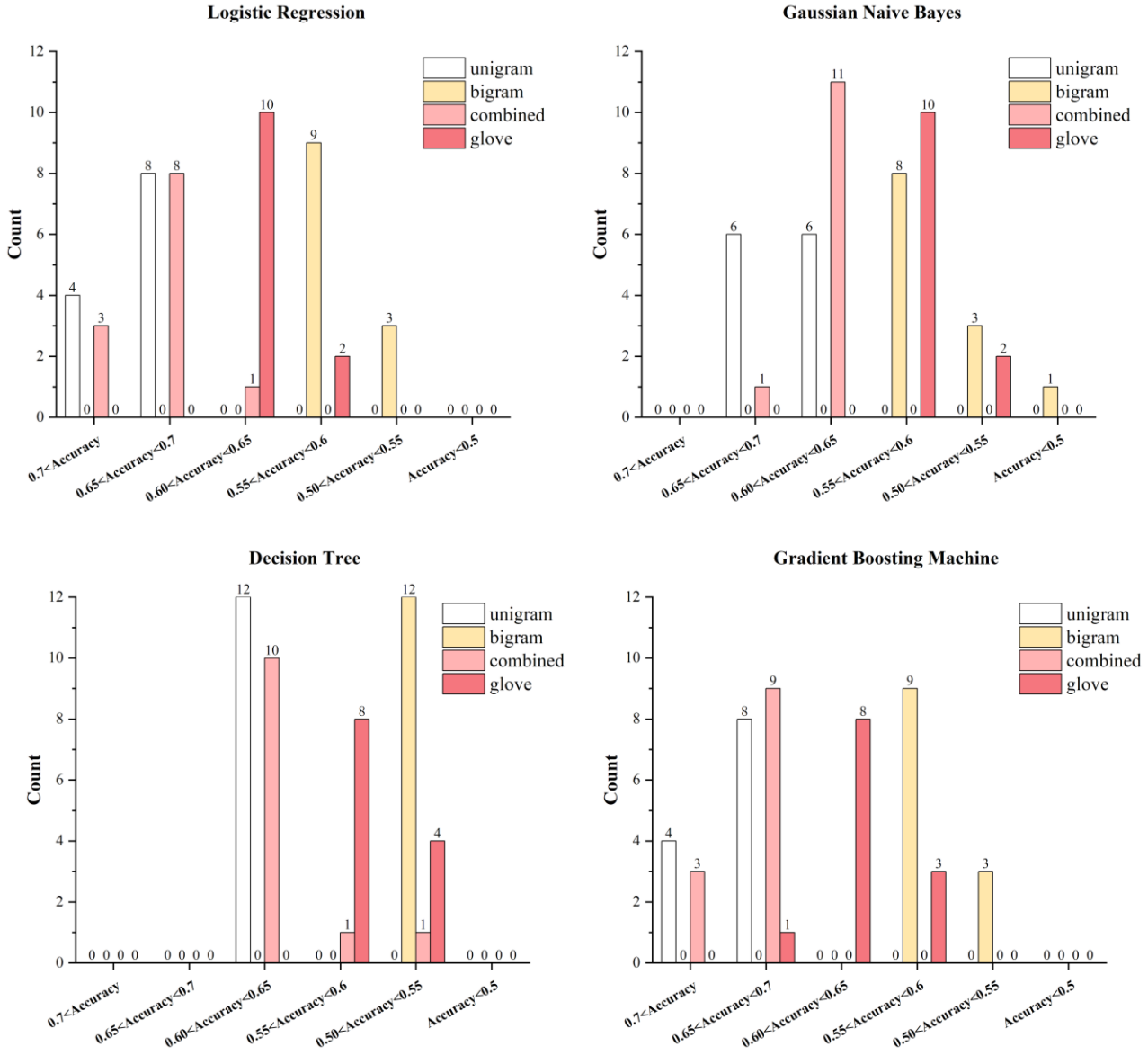

Fig. 4 The distribution of feature extractors in different accuracy ranges for each algorithm (Figure Credit: Original).

4. Conclusion

The results indicate that for sentiment analysis tasks based on Twitter text, the LR and GBM algorithms are well-suited. When combined with a unigram feature extractor to obtain high-dimensional features, these algorithms perform effectively, particularly when trained on larger datasets, leading to significant improvements in model performance.

This is because larger datasets provide more samples, helping algorithms capture a wider variety of patterns and reducing the risk of overfitting. Higher-dimensional feature representations more effectively capture complex relationships between words and contextual information, thus improving model performance.

While increasing dataset size and feature dimension generally enhances model performance, this is not the case for DT. This difference can be attributed to the property of DT, which select the most informative split at each node for feature selection. When the feature dimension is high, the model may struggle to effectively identify the truly important features. Additionally, DT is prone to overfitting, particularly in high-dimensional spaces, where they may capture noise instead of relevant information, leading to a decline in model performance.

This suggests that bigger datasets and higher-dimensional features do not always guarantee better performance. According to the experimental data, most of the algorithms reached peak performance with the D3 dataset size. Therefore, for similar tasks in the future, selecting a dataset of similar size (D3) could be a priority for optimal model training.

The models trained using bigram and GloVe as feature extractors showed significantly lower accuracy. This may be attributed to the setting of the max features parameter for bigram, which increases sparsity and negatively impacts model accuracy. Since bigram features are generated by pairing two words, they inherently produce more features compared to unigram. However, when the max features limit is applied, the model retains only the specified number of features. As a result, many valuable word-pair combinations may be discarded, leading to a sparser feature space than with unigram. This increased sparsity likely results in the loss of important information, contributing to the decline in accuracy.

On the other hand, the poor performance of models trained using GloVe word embeddings can likely be attributed to flattening the original three-dimensional feature space into a two-dimensional form. During this process, the structural information captured by the word embeddings may have been lost. Additionally, the subsequent use of Gaussian random projection for dimensionality reduction may have compressed or distorted important features, reducing the model's ability to effectively learn from the data.

References

[1] Krommyda Maria, Rigos Anastasios, Bouklas Kostas, et al. An experimental analysis of data annotation methodologies for emotion detection in short text posted on social media. Informatics. 2021, 8(1): 19-33.

[2] Habib Mohammad W, Zainab N Sultani. Twitter sentiment analysis using different machine learning and feature extraction techniques. Al-Nahrain Journal of Science, 2021, 24(3): 50-54.

[3] Manda Kundan Reddy. Sentiment Analysis of Twitter Data Using Machine Learning and Deep Learning Methods. 2019.

[4] Gupta Bhumika, Negi Monika, Vishwakarma Kanika, et al. Study of Twitter sentiment analysis using machine learning algorithms on Python. International Journal of Computer Applications, 2017, 165(9): 29-34.

[5] Le Bac, Huy Nguyen. Twitter sentiment analysis using machine learning techniques. Advanced Computational Methods for Knowledge Engineering: Proceedings of 3rd International Conference on Computer Science, Applied Mathematics and Applications. Springer International Publishing, 2015: 279-289.

[6] Ahmad Munir, Shabib Aftab. Analyzing the performance of SVM for polarity detection with different datasets. International Journal of Modern Education and Computer Science, 2017, 9(10): 29-36.

[7] Pang Bo, Lillian Lee. Opinion mining and sentiment analysis. Foundations and Trends in information retrieval, 2008, 2(1–2): 1-135.

[8] Saif Hassan, He Yulan, Fernandez Miriam, et al. Contextual semantics for sentiment analysis of Twitter. Information Processing & Management, 2016, 52(1): 5-19.

[9] Ahmad Munir, Aftab Shabib, Ali Iftikhar, et al. Hybrid tools and techniques for sentiment analysis: a review. International Journal of Multidisciplinary Sciences and Engineering, 2017, 8(3): 29-33.

[10] Ahmad Munir, Aftab Shabib, Muhammad Syed Shah, et al. Machine learning techniques for sentiment analysis: A review. International Journal of Multidisciplinary Sciences and Engineering, 2017, 8(3): 27-32.

[11] Go Alec, Richa Bhayani, Lei Huang. Twitter sentiment classification using distant supervision. CS224N project report, Stanford, 2009, 1(12): 2009.

[12] GloVe: Global Vectors for Word Representation. URL: https://nlp.stanford.edu/projects/glove/. Last Accessed 2024/10/18.