International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 17 • Issue 2 pp. 91-98 March 2026-Sept 2026 www.csjournals.com

Analysis of Positive Public Opinion on Twitter Using Sentiment

Mamta[1], Dr. Vishal Verma[2], Sandeep Rathi[3]
[1]MCA Student, [2]Professor, [3]Asst. Prof.
Deptt. of Comp. Sc. and Application, Chaudhary Ranbir Singh University, Jind, Haryana, India

Abstract

Twitter is a stream of large volume of short informal text and is a valuable resource for sentiment analysis which reflects public opinion at scale. This paper describes a binary classification system based on machine learning that detects positive sentiment in tweets. In other words, it reduces the traditional positive/negative/neutral formulation to a positive vs. not-positive decision. Three supervised classifiers Logistic Regression, Linear Support Vector Machine (SVM) and Multinomial Naive Bayes are trained and evaluated on the Sentiment140 dataset (1,600,000 tweets, distantly supervised using emoticons). Tweets are preprocessed by regular expression based cleaning, removal of stop-words and Porter stemming and then converted to TF-IDF feature vectors. Logistic Regression has the highest accuracy (80.14%, F1-score 0.8045) on an 80:20 stratified train–test split, followed by Linear SVM (79.54%, F1-score 0.7977) and Multinomial Naive Bayes (77.43%, F1-score 0.7637). Results agree that discriminative linear classifiers outperform the generative Naive Bayes model on high-dimensional, sparse TF-IDF representations of Twitter text, and the proposed binary formulation offers an efficient, practically deployable approach to large-scale social media sentiment monitoring.

keywords : Sentiment Analysis, Twitter, Natural Language Processing, Machine Learning, Logistic Regression, Support Vector Machine, Naive Bayes, TF-IDF, Binary Classification, Distant Supervision.

I. INTRODUCTION

There is vast amounts of unstructured, textual data generated every second on social media platforms such as Twitter. It is a microblogging service where users share (or ‘tweet’) their opinion regarding products, events, organizations and public figures [1]. This continuous stream of organic, real-time expression makes Twitter a uniquely valuable data source for understanding public opinion at scale, and has spurred considerable research interest in automated sentiment analysis of tweet text.

Sentiment analysis is a sub-field of Natural Language Processing (NLP) that automates the classification of emotional tone in a body of text [2]. Sentiment analysis has traditionally been applied to long-form content such as product reviews, but is increasingly being applied to short, informal microblogging text. Tweets are very short and need to be very economical with language, leading to the widespread use of slang, abbreviations, emoticons and hashtags that pose challenges to classical NLP pipelines built for formal text.

This paper describes a binary sentiment classification system for the specific task of positive sentiment detection from tweets. Instead of attempting a full multiclass classification (positive, negative, neutral), the study simplifies the decision boundary to positive rather then not positive. This formulation reduces model complexity and computational overhead, while still being directly applicable to applications such as brand monitoring and customer satisfaction tracking, where the primary question is whether sentiment is positive or not. Three supervised machine learning algorithms, namely Logistic Regression, Linear SVM and Multinomial Naive Bayes are trained and systematically compared using an identical pre-processing and feature-extraction pipeline.

II. RELATED WORK

Go, Bhayani and Huang [1] proposed distant supervision for Twitter sentiment classification, where emoticons are used as noisy labels to automatically construct a training corpus of 1.6 million tweets without human annotation. On a manually annotated test set, Naive Bayes, Maximum Entropy and SVM classifiers trained on this corpus attain over 80% accuracy. Their preprocessing pipeline (username normalisation, URL removal and repeated-letter reduction) is now a standard reference for subsequent Twitter NLP research and the dataset they produced (Sentiment140) is used directly in this study.

Agba et al. [3] performed SVM-based sentiment classification on Twitter data to evaluate the reputation of skincare brands, finding a dominance of positive sentiment (69%) and confirming the use of binary or near-binary classification in brand-monitoring scenarios. Rasool et al. [4] investigated sentiment in apparel brands with a combination of Naive Bayes and lexicon dictionary approaches. They emphasized the need for automation due to the volume and velocity of Twitter data. Zakaria et al. [5] applied sentiment analysis in the automotive domain and found more than 85% positive sentiment for two major brands using traditional machine learning classifiers.

Lin [6] designed a Naive Bayes classifier using a custom language model which achieved better accuracy than the lexicon-based baselines. Bae and Lee [7] investigated the effect of positive sentiment of influential users on audience reaction by analysing over three million tweets and proposed a positive-negative influence measure. The pioneering work of Pang and Lee [8] proved the viability of opinion mining as a separate research area, and Pang, Lee and Vaithyanathan [9] were among the first to apply In this paper, we use Naive Bayes, Maximum Entropy and SVM to classify review text sentiment. [10] proposed the use of emoticons to reduce the dependency on manually labeled training data, which was directly used in the distant-supervision method proposed by [1], and hence adopted in this work.

However, despite this large body of work, a review of the literature shows significant gaps remain. Many studies use static and outdated datasets, most try full multiclass classification, which adds complexity but not always practical utility, relatively few studies focus solely on binary positive/not-positive classification, and many existing studies test a single algorithm in

isolation rather than offering a controlled, like-for-like comparison across multiple models on the same pipeline. This paper fills these gaps directly.

III. METHODOLOGY

A. Dataset: In this work, we use the Sentiment140 data set [1] that consists of 1,600,000 tweets retrieved from Twitter API with distant supervision labels: tweets with positive emoticons are labelled positive (4), tweets with negative emoticons are labelled negative (0); the emoticons were removed from the text. In this study the labels 0 (not positive) and 4 (positive) were binarized. It is perfectly balanced in class with 800,000 tweets in each class, and has no missing values in any of its six original fields (target, id, date, flag, user, text).

B. Text Pre-processing :The raw tweet text was pre-processed by a sequential pipeline consisting of the following six steps: (i) Removal of all the non-alphabetic characters from the text by using the regular expression [^a-zA-Z]; (ii) Lowercasing of the text; (iii) Tokenisation based on whitespaces; (iv) Stop-word removal using the NLTK English stop-word corpus; (v) Stemming using the Porter Stemmer to transform the words into their root form; (vi) Re-assembling the stemmed words in a single cleaned string per tweet. The pipeline was first tested on a representative sample tweet, and then applied to the entire tweet corpus.

C. Extracting Features: The text of the cleaned tweets was converted to numerical feature vectors using Term Frequency-Inverse Document Frequency (TF-IDF) weighting: TF-IDF(t,d,D) = TF(t,d) × log(|D| / |{d ∈ D : t ∈ d}|), where t is a term, d is a document (tweet), and D is the corpus. This weighting gives greater credit for the terms that appear in many of the tweets, but are rare in the corpus, and less credit for common terms, which contain little of the sentiment. When I used TfidfVectorizer from scikit-learn for the entire corpus, the resulting feature vector matrix was 1,600,000 × 684,358 and had 18,986,995 non-zero elements.

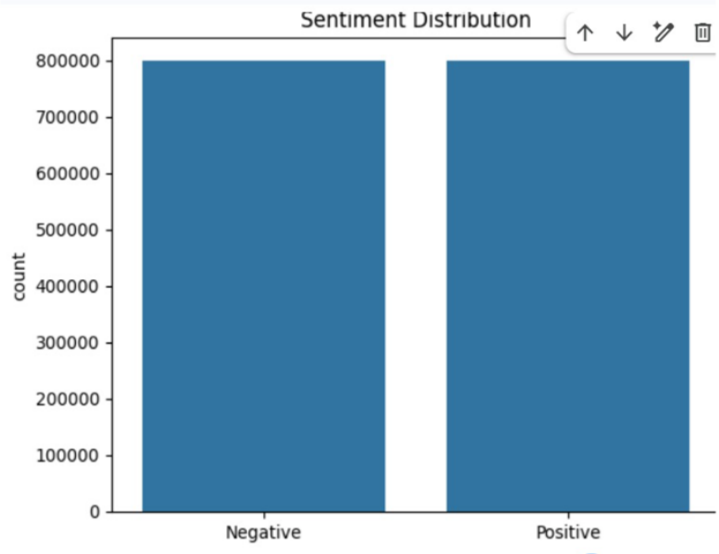

Fig: Sentiment distribution across the dataset (800,000 positive, 800,000 negative).

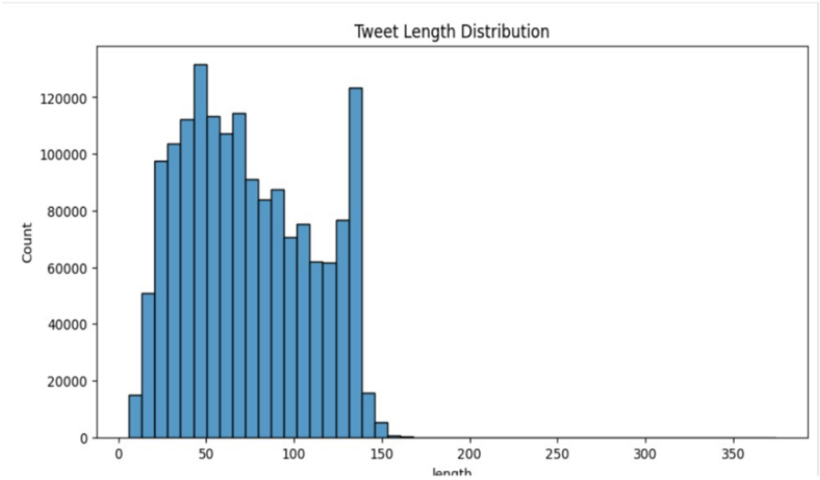

Fig: Distribution of tweet character length across the corpus.

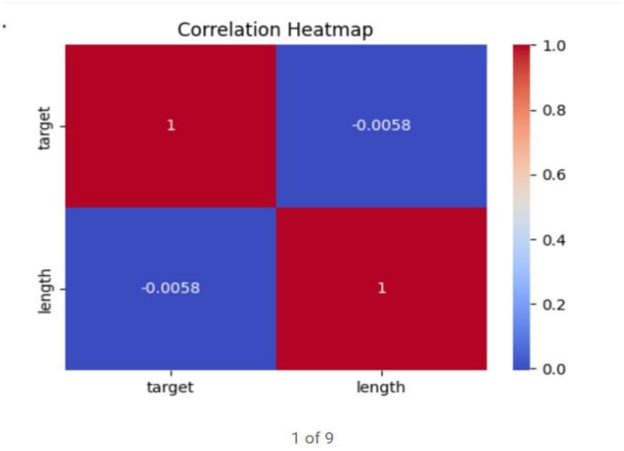

Fig: Correlation heatmap between tweet length and sentiment label.

D. Classification Models: Three supervised classifiers were trained on the same features (TF-IDF), which are based on the "probabilistic", "discriminative-linear" and "margin-based" paradigms, respectively: • Logistic Regression [11]: a discriminative linear classifier based on a sigmoid function for modeling the probability of class membership, given a linear combination of the input features. SVM for classification: Learned with max_iter=400. Linear SVM: a margin based classifier that is trying to find the best separating hyperplane between the classes. Implementations with the LinearSVC (default regularisation), and the Multinomial Naive Bayes [12] (a generative probabilistic classifier following Bayes' theorem, assuming features are conditionally independent, given the class label). The Laplace smoothing has been applied as default.

E. Experimental design: The TF-IDF feature matrix and binarised labels were split into a training set (80%) and a testing set (20%) using stratified sampling with random_state = 2 to ensure the class balance in the original data. This gave us a total of 1,280,000 training tweets and 320,000 test tweets. All three classifiers were trained with the same training set, and tested using the same test set that was held out. The metrics module of scikit-learn [13] was used to calculate accuracy, precision, recall and F1 score.

F. Data exploration analysis: We conducted an exploratory analysis of the data first, to assess the quality of the data and to provide a characterisation of the corpus, before training the models. There were no missing values found for the 6 original fields of the data set. Tweet character length was observed to have a mean of around 74 characters, median of 69 characters, and up to 374 characters. The distribution is skewed to the right as is typical of short-form social media text. The length of the tweet and the sentiment label showed a very small linear correlation. This indicates that lexicon content (TFIDF) is the most significant discriminating signal for the downstream classifiers, while the length of the tweet is not an informative parameter to predict polarity.

IV. RESULTS

Model	Accuracy	F1-Score
Logistic Regression	80.14%	0.8045
Linear SVM	79.54%	0.7977
Naive Bayes	77.43%	0.7637

Logistic Regression obtained best accuracy (80.14%) and F1-score (0.8045) with the balanced precision and recall on both classes (precision 0.81/0.79, recall 0.79/0.82 for not-positive/positive respectively). Linear SVM performed competitively, trailing Logistic Regression by approximately 0.6 percentage points — consistent with the close theoretical relationship between the two linear, discriminative algorithms, which differ primarily in loss function (log-loss versus hinge loss). Of the three, Multinomial Naive Bayes performed the worst (accuracy of 77.43% and an F1-score of 0.7637), which reflects the limiting effect of its conditional feature-independence assumption on natural language data, where word co-occurrence patterns carry meaningful sentiment information that Naive Bayes cannot directly exploit. The ranking Logistic Regression > Linear SVM > Naive Bayes observed is consistent with relative performance trends reported by Go et al. [1] on the same underlying dataset. The relatively narrow margin across all three models (2.71 percentage points in accuracy)) suggests that, given a fixed TF-IDF feature representation and preprocessing pipeline, the specific choice of linear or probabilistic classification algorithm has a comparatively modest effect relative to the quality of the underlying feature representation itself.

Model	Precision (0/1)	Recall (0/1)
Log. Reg.	0.81 / 0.79	0.79 / 0.82
SVM	0.80 / 0.79	0.79 / 0.80
Naive Bayes	0.78 / 0.77	0.76 / 0.79

Table=PER-CLASS PRECISION AND RECALL (0 = NOT POSITIVE, 1 = POSITIVE)

The positive class has a consistently higher recall than the negative (or not-positive) class, across the three models, suggesting a mild common tendency to prefer the positive label when faced with classification uncertainty. This is probably due to the fact that the two distributions of the words learned from the distant supervision training labels are slightly asymmetric, which is a candidate area for calibration of the threshold in deployment scenarios where the cost of a false positive and false negative are not the same.

V. DISCUSSION The results of the experiments give rise to several conclusions. Firstly, the discriminative linear models (Logistic Regression, SVM) can be shown to always outperform the generative model of the Naive Bayes class-conditional feature distributions on this task, which highlights the advantage of explicitly modelling the decision boundary over estimating the class-conditional feature distributions under the independence assumption that holds for natural language. Second, the dataset is perfectly class-balanced, both overall and when stratified in the same manner as used in this study, making accuracy a reliable and non-misleading measure for this study when compared to many real-world imbalanced sentiment studies. Third, the accuracy limits around 80% for all three classical models are in line with the known limitations of bag-of-words / TF-IDF representations that do not capture word order, negation scope or meaning in context, which encourage the use of contextual embeddings or transformer-based models in future work. For its part, Naive Bayes is the least accurate of the three models but also the most computationally efficient, so it's a worthwhile compromise for applications with limited resources or those deployed on a real-time basis.

VI. LIMITATIONS AND FUTURE WORK :

There are some limitations to this study. The Sentiment140 Corpus dates back to 2009 and the language on Twitter has undergone significant change over the years. The binary formulation does not include neutral sentiment, and is not applicable in cases where full multiclass classification is required. Sarcasm and irony are not explicitly modelled, as a well documented source of misclassification in sentiment analysis. Although TF-IDF is effective, it is unable to convey the meaning of a text in the context or sequentially, and this study is limited to English-language tweets.

Future work includes fine-tuning pre-trained transformer models like BERT that have shown state-of-the-art performance on most NLP benchmarks and grasp the meaning of words in context, in order to better achieve accuracy and robustness in the presence of sarcasm. Other possible directions include real-time pipelines for sentiment monitoring linked with the Twitter API, specialized sarcasm-detection modules, and extending the task of multilingual sentiment analysis (e.g., using

mBERT or XLM-RoBERTa) and using dense word embeddings such as Word2Vec or GloVe instead of TF-IDF to capture more semantic similarity among related sentiment terms.

VII. CONCLUSION :

In this paper, the authors proposed a binary machine learning model to classify tweets as positive for sentiment classification, and evaluated its performance in the Sentiment140 dataset using three classifiers with the same preprocessing and feature extraction processes: Logistic Regression, Linear SVM, and Multinomial Naive Bayes. Logistical Regression did the best (80.14%, 0.8045), followed closely by Linear SVM (79.54%, 0.7977) and Naive Bayes, which ranked last (77.43%, 0.7637). The results validate that discriminative linear classifiers are more appropriate than generative independence-assuming classifiers for high-dimensional, sparse representations of text, and that the proposed binary classification formulation provides an efficient approach to large-scale Twitter sentiment monitoring that is practically deployable. Future research combining transformer-based contextual models will further enhance classification accuracy and robustness in informal and/or sarcastic language.

REFERENCES

[1] A. Go, R. Bhayani, and L. Huang, “Twitter Sentiment Classification using Distant Supervision,” CS224N Project Report, Stanford University, 2009.

[2] Natural Language Processing (NLP), GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/nlp/introduction-to-natural-language-processing-nlp/.

[3] M. W. Agba, et al., “Memanfaatkan Analisis Sentimen Twitter untuk Mengelola Reputasi Merek: Studi Kasus Skincare Merek Menggunakan Support Vector Machine”, vol. 4, no. 3, 2025, doi: 10.38035/jim.v4i3.1196. A. Rasool, R. Tao, K. Marjan and T. Naveed, “Twitter Sentiment Analysis: A Case Study for Apparel Brands,” J. Phys.: Conf. Series 357, 022064 (2012). Ser., 2015, doi: 10.1088/1742-6596/1176/2/022015.

[5] M. Z. Zakaria, T. B. Kurniawan, Misinem, “Twitter Sentiment Analysis on Automotive Companies”, 2022, doi: 10.61453/jods.v2022no06.

[6] A. Lin, Improving Twitter Sentiment Analysis using Naive Bayes and Custom Language Model, Computation and Language, Nov. 2017.

[7] Y. Bae and H. C. Lee, Sentiment Analysis of Twitter Audiences: Measuring the Positive or negative Influence of Popular Twitterers, Korea University, 2012, doi: 10.1002/asi.22768. For more information, please refer to Found.

[8] Opinion Mining and Sentiment Analysis” by B. Pang and L. Lee. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008. “Thumbs Up?”

[9] B. Pang, L. Lee and S. Vaithyanathan, Proc. of Sentiment Classification using Machine Learning Techniques. EMNLP, 2002, pp. 79–86

[10] J. Read, “Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification,” in Proc. ACL Student Research Workshop, 2005.

11]GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/.

[12]GeeksforGeeks, Naive Bayes Classifiers, [Online] 20162018. Available: https://www.geeksforgeeks.org/machine-learning/naive-bayes-classifiers/.

[13] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011

[14] Learning in Python, SKLearn: scikit-learn, Machine Learning in Python, 2014. Learn. Res., vol. 12, pp. 2825–2830, 2011.