Hindawi
Advances in Human-Computer Interaction
Volume 2021, Article ID 5561204, 7 pages
https://doi.org/10.1155/2021/5561204

Research Article

Statistical Analysis of Public Sentiment on the Ghanaian Government: A Machine Learning Approach

John Andoh, Louis Asiedu, Anani Lotsi, and Charlotte Chapman-Wardy

Department of Statistics & Actuarial Science, School of Physical and Mathematical Sciences, University of Ghana, Legon, Accra, Ghana

Correspondence should be addressed to Louis Asiedu; lasiedu@ug.edu.gh

Received 5 January 2021; Revised 27 January 2021; Accepted 18 February 2021; Published 3 March 2021

Academic Editor: Francesco Bellotti

Gathering public opinions on the Internet and Internet-based applications like Twitter has become popular in recent times, as it provides decision-makers with uncensored public views on products, government policies, and programs. Through natural language processing and machine learning techniques, unstructured data forms from these sources can be analyzed using traditional statistical learning. The challenge encountered in machine learning method-based sentiment classification still remains the abundant amount of data available, which makes it difficult to train the learning algorithms in feasible time. This eventually degrades the classification accuracy of the algorithms. From this assertion, the effect of training data sizes in classification tasks cannot be overemphasized. This study statistically assessed the performance of Naive Bayes, support vector machine (SVM), and random forest algorithms on sentiment text classification task. The research also investigated the optimal conditions such as varying data sizes, trees, and kernel types under which each of the respective algorithms performed best. The study collected Twitter data from Ghanaian users which contained sentiments about the Ghanaian Government. The data was preprocessed, manually labeled by the researcher, and then trained using the aforementioned algorithms. These algorithms are three of the most popular learning algorithms which have had lots of success in diverse fields. The Naive Bayes classifier was adjudged the best algorithm for the task as it outperformed the other two machine learning algorithms with an accuracy of 99%, F1 score of 86.51%, and Matthews correlation coefficient of 0.9906. The algorithm also performed well with increasing data sizes. The Naive Bayes classifier is recommended as viable for sentiment text classification, especially for text classification systems which work with Big Data.

1. Introduction

The explosion of blogging, microblogging, social media, and review sites has armed data analysts with valuable information on users' preferences. Information is now shared all over the world at ever-increasing speeds, volume, and diversity. This connectivity leaves "data prints" which we can use to describe almost everything in our world today. Consequently, one type of data that has become increasingly important in recent times is the opinions and preferences of Internet users regarding products, subjects, and views. This type of data aggregates in e-commerce sites, blogs, social media, and other online platforms. Traditional methods of collecting data on product feedback from customers such as interviews and polling are gradually being phased out by considering the reviews of users on such online platforms. Through the use of machine learning techniques, data analysts can extract and classify this wealth of information to make informed inferences. This process of making the computer understand human language in texts is largely called natural language processing (NLP). NLP techniques can also be used to perform sentiment analysis to summarize opinions from online platforms.

The conjoining of news with social networking and blogging has made Twitter a hotbed for the discussion of events in real time. Twitter currently serves as a medium for the discourse of a wide variety of societal issues such as sports, governance, advocacy, religion, and especially politics. Public views expressed in the form of text on these societal issues are called sentiment texts. For instance, before, during, and after the 2016 US presidential elections, Twitter proved itself as the major election news destination. A record 40 million tweets were posted regarding the elections and its "immediacy and speed" was unmatched by any other traditional news network [1].

There is an active Ghanaian presence on Twitter and other social media platforms who update their statuses regarding happenings in their social circles and conversations on politics. Assuming opinions shared on Twitter mirror public perception as it is unbiased and unrestricted, a sentiment analysis task trained on data from Twitter would yield interesting results for policy analysts and political parties.

As one of the pioneering works in this field, the paper [2] classified reviews by sentiments using Naive Bayes, max entropy, and SVM and analyzed the difficulty under each classification task for sentiment analysis. They sought to recognize whether sentiment classification was a special topic-based categorization, which was a technique for text classification or special sentiment categorization methods needed to be developed to address the novel challenges sentiment analysis tasks presented. Even though all three methods outperformed human classification, they could not reach accuracies achieved by the same methods for topic categorization. The classification task in sentiment analysis becomes challenging if the texts are rhetoric and sarcastic. Features exclusive to sentiment analysis will be needed to accommodate such words.

The authors in [3] evaluated the effectiveness of statistical keyword extraction methods in conjunction with ensemble learning algorithms. They compared base learning algorithms (Naive Bayes, support vector machines, logistic regression, and random forest) with five widely used ensemble methods (AdaBoost, bagging, dagging, random subspace, and majority voting). Their study revealed that the bagging and random subspace ensembles of random forest yield promising results. They also found that the use of keyword-based representation of text documents in conjunction with ensemble learning enhances the predictive performance and scalability of text classification schemes.

In another comparative study of various classification techniques for sentiment analysis, the authors in [4] pointed out that, in selecting a particular algorithm, one will need to consider the type of specific input required. This implies that, in order to achieve higher accuracies, it is important to know which algorithm will be appropriate given the available input data. Their study identified Naive Bayes, max entropy, boosted trees, and random forest classifiers as the most widely used algorithms in sentiment analysis. They concluded by noting that each classifier had its advantages and disadvantages and could all be assessed on the basis of accuracy, resources (computing power), data input, time for training, etc. The random forest classifier achieved the highest accuracy and exhibited improvements over time. It is however costly in terms of resources as it uses longer training times and requires high computing power. It will be interesting to investigate how well the random forest classifier will perform when dealing with short text classification in Twitter data which will be considered in this study.

One of the earliest uses of sentiment analysis using Twitter as corpus was in [5]. Their work highlights the importance of preprocessing techniques as they are necessary to achieve higher accuracies. They employed "emoticons" as noisy labels to achieve maximum accuracies of 82.7% for Naive Bayes when using unigram and bigram features, 82.7% for max entropy using unigram and bigram features, and 82.9% for SVM when using only unigram features. They concluded by highlighting the shortcomings of sentiment analysis at the time, which included the handling of neutral tweets, internalization (so as to be able to use them for multiple languages), and the utilization of emoticon data.

The authors in [6] also asserted that the challenge encountered in machine learning method-based sentiment classification is the abundant amount of data available. They explained that this amount makes it difficult to train the learning algorithms in a feasible time and degrades the classification accuracy of the built model. They recommended feature selection as essential in developing robust and efficient classification models whilst reducing the training time.

The effect of training data sizes in classification tasks has been of interest to researchers obviously because of its purported influence on accuracy. The authors in [7], for instance, measured the effect of training data sizes on classification using SVM and Naive Bayes. They concluded that the effect was not significant. The authors in [8] also found that the complexity of the features can affect accuracy and that some classifiers could even work better with less data. This study among others will investigate training sizes effect.

Generally, the study seeks to identify the most suitable machine learning processes for collecting, analyzing, and predicting public sentiments from Twitter. The study does this specifically by analyzing tweets on sentiments about political discourse in the country, analyze the various conditions under which the algorithms work well with the tweet data, and statistically evaluate the performance of the study algorithms.

2. Materials and Methods

2.1. Data Acquisition and Authorization. Twitter returns a collection of tweets that match a specified query. The standard search Application Program Interface (API) accessed from the Twitter developer page is free but developers do not have access to the entire database of tweets. Only tweets from the last 30 days can be accessed with this standard search API.

We secured assess to the standard API for some period to extract tweets related to the subject for this study. This was done through creation of a developer account which was eventually approved.

2.1.1. The Tweets. The ease in obtaining data from tweets through the Twitter API for developers was one key motivation for performing this sentiment analysis. *R* packages capable of accessing the Twitter API which were used for this study include "twitteR" and "rtweet." After obtaining authorization, the tweets were collected using the keywords, "NPP," (the ruling party) and "nanakuffoaddo" (President of Ghana) and making it specific to Ghana by tagging the geolocation for Ghana. 3,000 tweets were collected over a two-month period (January 2020 to March 2020).

Figure 1 shows a word cloud diagram of tweets used in the study.

The word cloud diagram like the bar plot explores frequent words; however, the word cloud is often desirable because it represents the words with their relative frequencies aesthetically.

The word cloud from our data suggests "Ghana" is the most popular word found in tweets regarding the governance, probably the central theme of most Ghanaian public sentiments. Some other relevant keywords are also shown in Figure 1.

The tweets extracted had various useful attributes like the screen name of user, tweet text, time stamp, geolocation, and number of "retweets" and "likes." For this study, only the texts were extracted. The following are samples of the tweets in their raw forms:

(i) "Npp has a vision for Ghana."

(ii) "No reasonable Ghanaian will vote for NDC or NPP again!"

(iii) "When its NPP its a different Narative. When its the NDC, then yeah the NDC Is corrupt."

(iv) "@CheEsquire All hail the NPP government."

(v) "While we're busy with NDC vs NPP - Ghana is losing."

These tweets presented were processed in stages to remove unwanted characters like numbers, punctuation, special characters, and stopwords in order to reduce noise and prepare them for the classifiers. Each of the tweets above conveys some form of sentiment, which shall be classified as positive, negative, or neutral.

2.1.2. Annotation of Tweets. The tweets were manually annotated by one researcher and cross-referenced by another, before being considered for sentiment classification. All the tweets which were classified differently were removed. Tweets that were regarded as positive towards the government were classified as positive sentiments (e.g., "Npp has a vision for Ghana") while those regarded as negative towards the government were regarded as negative sentiments (e.g., "No reasonable Ghanaian will vote for NDC or NPP again!"). Tweets that had no sentiment, or could neither be classified as positive nor negative, were regarded as neutral sentiments.

Out of the 3,000 tweets, 990 tweets were prepared for sentiment classification after the cleaning and annotation phases. Figure 2 shows the prior distribution of the sentiment texts.

From Figure 2, 14% of the tweets had positive connotations, about 33% of the tweets were negative, and 53% were neutral.

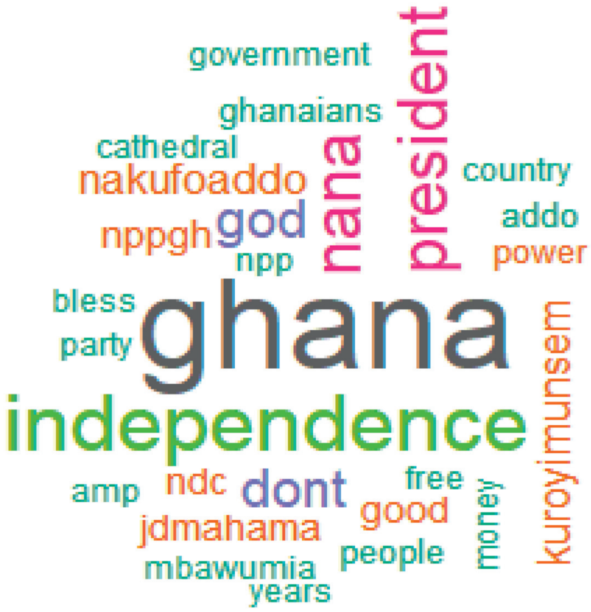

Figure 1: Word cloud of tweets.

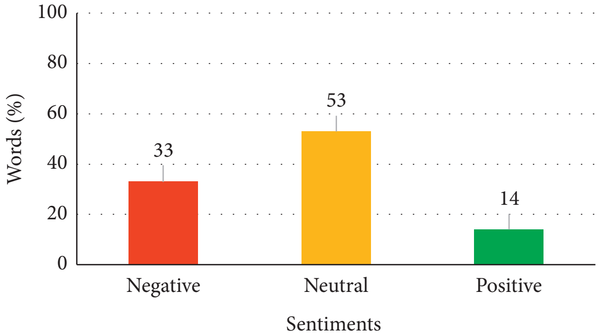

Figure 2: The prior distribution of the sentiment texts or tweets.

2.2. Random Forest Model. The random forest model classifier is actually a bagging method of various classifiers or decision trees. The idea is to average the results of various decision trees in order to reduce overall variance. Each tree is independent and identically distributed (i.i.d.) and the expectation of a number of trees is the same as that of the individual trees. The random forest is the collection of these individual trees and the results of a classification represent the majority votes of the trees.

Given an ensemble of classifiers $h_1((\mathbf{x})), h_2((\mathbf{x})), \ldots, h_k((\mathbf{x}))$ with training set drawn randomly from the random vector $\mathbf{X}, \mathbf{Y}$, we define the margin function as

$$\mathrm{mg}((\mathbf{X},\mathbf{Y})) = \mathrm{av}_k \mathbf{I}(h_k(\mathbf{X}) = \mathbf{Y}) - \max_{j \neq Y} \mathrm{av}_k \mathbf{I}(h_k(\mathbf{X}) = j), \tag{1}$$

where $\mathbf{I}(\cdot)$ is the indicator function.

This margin measures the extent to which the average votes at $\mathbf{X}, \mathbf{Y}$ for the actual class exceeds the average vote for any other class. The larger the margin, the more confidence we have in the classification [9]. The performance of the algorithm will be assessed by varying m, the number of variables considered at split, within $\left\{\sqrt{p}/2, \sqrt{p}\right\}$ and the number of trees.

2.3. Naive Bayes Model. The Naive Bayes algorithm is based on Bayes' Theorem, a probabilistic method used for calculating likelihoods of events based on conditional probabilities. The probability of a document d being assigned to a category or class c is given by

$$P(c \mid d)\alpha P(c) \prod_{1\leq k\leq n_d} P\left(t_k|c\right), \tag{2}$$

where $P(t_k|c)$ is the conditional probability of a term t_k in d of a certain class c [10]. In line with the objectives of the study, the Naive Bayes model was tested on various data with different feature space sizes and number of observations. It is our expectation to find the optimal conditions so as to get the best accuracy out of the model.

2.4. Support Vector Machine Model. If we have N training data of pairs, $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_\mathbf{n}, y_n)$, with $\mathbf{x}_i \in \mathbb{R}^k$ and $y_i \in \{\pm 1\}, i = 1, 2, \ldots, n$, we can define the hyperplane as

$$x\colon\ f(x) = \mathbf{x}^\mathbf{T}\beta + \beta_0 = 0, \tag{3}$$

where β is a unit vector with $\|\beta\| = 1$. The classification is then determined by

$$G(x) = \operatorname{sign}\left[\mathbf{x}^\mathbf{T}\beta + \beta_0\right]. \tag{4}$$

If the classes are easily separable, the function $f(x) = \mathbf{x}^\mathbf{T}\beta + \beta_0$ with $y_i f(x_i) > 0, \forall i$. The hyperplane with the biggest margin between points of different classes is now reduced to the optimization problem

$$\begin{aligned} &\max_{\beta,\beta_0,\|\beta\|=1} M \\ &\text{subject to}\quad y_i\left(\mathbf{x}_\mathbf{i}^\mathbf{T}\beta + \beta_0\right) \geq M, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{5}$$

Kernels are used to modify dimensionality of the data to find the flat affine dimensional subspace hyperplane to correctly determine the accurate support vector classifier. The structure of the data determines the kind of kernel to use, whether to use a linear, polynomial, radial basis function, or sigmoid. In this study, the results present the optimal conditions for this algorithm on sentiment text classification based on kernel type.

2.5. Performance Metrics of Machine Learning Models. Comparison of performances of various machine learning models is very important and does not need to be just superficial. For instance, just comparing "Accuracy" amongst different models may be inadequate and statistically insufficient especially when the accuracies are close.

In this study, we compare performances of the study algorithms using Cohen's kappa statistic (measure of reliability), F1 score (which strikes a balance between precision and recall), sensitivity, specificity, classification accuracy, and Matthews Correlation Coefficient (MCC). According to [11], the MCC is a more reliable statistical rate which produces a high score only if the prediction obtained good results in all categories of the confusion matrix categories. For more details about the adopted performance metrics, please refer to [11].

3. Results and Discussion

3.1. Model Training. The random forest, Naive Bayes, and SVM classifiers are trained using manually annotated tweets. The data was divided into three (150, 300, and 540 tweets) to investigate the algorithms' performances with varying data sizes. 70% of each dataset was used for training the algorithms and the remaining 30% for testing.

The dataframe of the document term matrix created from the corpus which has been split into 3 has these dimensions:

From Table 1, the observations are the individual tweets whereas the variables are obtained from 99.5% of the most common words/items in the tweets.

We shall adopt a baseline of 33% to compare the accuracy of the algorithms. This baseline is obtained by dividing the tweets in three. For any model, a performance above 33% implies it is better than a random selection of the label by the respective classifier.

3.2. Results of the Random Forest Algorithm. The random forest algorithm was run on the three randomly shuffled sets of data. Dataset 1 contained 150 tweets; Dataset 2, 300 tweets; and Dataset 3, 540 tweets. As stated earlier, 70: 30 ratio was adopted to split each dataset for training and testing, respectively. The datasets were also trained considering the set $m = \left\{\sqrt{p}/2, \sqrt{p}\right\}$ (p is the number variables considered) and the set of trees (500, 1000).

The plot in Figure 3 also shows the Out-of-Bag (OOB) estimate of error for all the classes in the model. The black line shows the OOB estimate for the model as a whole while the green, red, and blue represent negative, neutral, and positive classes, respectively.

The best performances of the random forest algorithm in terms of Out-of-Bag estimates of error are given in Table 2.

Using the best performance of the random forest model ($m = \sqrt{p}$ and 1000 trees), we get the following confusion matrix and some performance statistics shown in Tables 3 and 4, respectively.

From Table 4, using $m = \sqrt{p}$ and 1000 trees for the random forest result in an overall accuracy of 52.22% with a runtime of 10.22 seconds.

The random forest model was now tested on the training data to further investigate the model. Table 5 compares the kappa and accuracy.

To optimize random forest models, we vary the number of trees and number of variables at split (m). The R package "randomForest" uses a default of 500 trees and $m = \sqrt{p}$, where p is the number of variables considered. From Table 5, the model has a kappa statistic of 0.3 (fair reliability) and an accuracy of 53.33% when used to classify the test data. This is slightly above the baseline of 33%.

Table 1: Dimensions of datasets.

Dataframe (df)	Observations	Variables (p)
df1	150	845
df2	300	466
df3	540	399

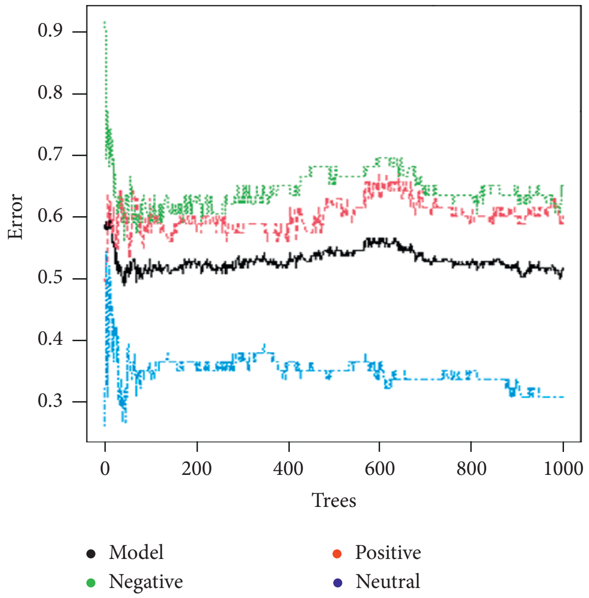

Figure 3: Random forest plot of OOB error estimates of classes.

Table 2: Random forest training.

Dataset	m	Trees	OOB estimate (%)	Kappa	Accuracy
150	$\sqrt{p}$	500	64.76	0.067	0.3778
300	$\sqrt{p}$	1000	53.81	0.2924	0.5222
540	$\sqrt{p}$	1000	46.32	0.2414	0.4875

Table 3: Confusion matrix of the random forest model.

			Actual (%)	
		Negative	Neutral	Positive
Predicted (%)	Negative	54.55	22.73	22.73
	Neutral	15.38	84.62	0.00
	Positive	23.64	32.73	43.64

Table 4: Performance statistics of the random forest model.

Runtime (sec.)	95% accu. CI	Accuracy		Sensitivity			Specificity	
			Neg	Neu	Pos	Neg	Neu	Pos
10.22	(0.143, 0.6287)	0.5222	0.4444	0.3235	0.8276	0.8413	0.9643	0.4918

Table 5: Prediction comparison with testing on training data.

	Kappa	95% accu. CI	Accuracy
Model on testing data	0.3085	(0.4251, 0.6393)	0.5333
Model on training data	0.8642	(0.8623, 0.9446)	0.9095

Table 6: Confusion matrix of the Naive Bayes model.

		Actual (%)		
		Negative	Neutral	Positive
Predicted (%)	Negative	98.33	1.67	0.00
	Neutral	0.00	100.00	0.00
	Positive	0.003	0.00	100.00

Table 7: Performance statistics of the random forest model.

Runtime (sec.)	95% accu. CI	Accuracy	Sensitivity			Specificity		
			Neg	Neu	Pos	Neg	Neu	Pos
0.09	(0.9657, 0.9998)	0.9938	1.0000	0.9783	1.0000	0.9901	1.0000	1.0000

The high variation between the accuracy of the algorithm when used to classify the train data and the test data shows evidence of overfitting. In general, the performance of the random forest model is not too appreciable. The huge variance from the 95% confidence bounds and the high OOB estimates of error rates are also not ideal. From the tests, we can conclude that the best random forest models were achieved with 1000 trees and $m = \sqrt{p}$.

3.3. Results of the Naive Bayes Algorithm. The Naive Bayes model was run similarly on the three randomly shuffled datasets (150, 300, and 540 tweets). The confusion matrix of the best Naive Bayes algorithm and some performance metrics are shown in Tables 6 and 7, respectively.

It is evident from Tables 7 that the overall accuracy of the Naive Bayes algorithm is 99.38% which is highly appreciable. The results from Tables 6 and 7 show that the Naive Bayes model performs remarkably well. It is worthy of note that the performance of the algorithm increases with increasing dataset size. The runtime for the Naive Bayes algorithm (shown in Table 7 as 0.09 seconds) is relatively better than the runtime of the random forest algorithm (shown in Table 4 as 10.22 seconds).

3.4. Results of the Support Vector Machine (SVM) Algorithm. An SVM classifier was also used on the same datasets (150, 300, and 540 tweets). SVM can be extended to solve multiclass categories problems, not just binary, as has been discussed in the methodology. The appropriateness of the various kernel methods for this task is also explored with the most suitable ones reported in Table 8.

The SVM model had its best performance when ran on Dataset 2, with an accuracy of 56.67% and kappa statistic value of 0.3506 (fair reliability). The confusion matrix and other performance statistics of the SVM algorithm are shown in Tables 9 and 10, respectively.

In training the SVM model, the various kernel types were run on all three datasets and the best performing kernels are reported in Table 8. The RBF kernel worked best on Dataset 1 with 150 observations and 845 variables but was not suitable for Datasets 2 and 3. The linear kernel outperformed all the other kernels in Datasets 2 and 3 and consequently had the best accuracy.

Table 8: Support vector machine training.

Dataset	Kernel type	No. of support vectors	Kappa	Accuracy
150	RBF	105	0.1	0.4
300	Linear	91	0.3506	0.5667
540	Linear	317	0.3115	0.5375

Table 9: Confusion matrix of the SVM model.

		Actual (%)		
		Negative	Neutral	Positive
Predicted (%)	Negative	45.00	35.00	20.00
	Neutral	24.00	72.00	4.00
	Positive	26.67	20.00	53.33

It is evident from Table 10 that the SVM model had the best performing accuracy of 56.67% with 95% CI of 45.8%–67.08% when used to classify the study data. The runtime averaged around 0.11 seconds. The SVM model performs slightly above the 33% baseline and the huge variability in the confidence interval of the accuracy is an evidence of low precision.

3.5. Comparison of Models. Table 11 shows the results of the best performing models under the three different algorithms.

From Table 11, the Naive Bayes model outperformed the random forest and SVM, recording the highest kappa statistic value of 0.9906 (near perfect reliability), F1 score value of 86.51%, MCC of 0.9906, accuracy of 99.38%, and the lowest runtime (0.09 seconds). The SVM model had a slight edge in performance over the random forest algorithm.

The random forest classifier had a relatively low performance with an F1 score of 52.18%, MCC of 0.3404, and an accuracy of 53.33%. The relatively high computational time of 10.22 seconds is as a result of the numerous averaging of trees which further makes the classifier unattractive.

Table 10: Performance statistics of the SVM model.

Runtime (sec.)	95% accu. CI	Accuracy	Sensitivity			Specificity		
			Neg	Neu	Pos	Neg	Neu	Pos
0.11	(0.458, 0.6708)	0.5667	0.3333	0.5294	0.8276	0.8254	0.8750	0.6557

Table 11: Comparing performance results of the three classifiers.

	Kappa	F1 score	MCC	Accuracy	Runtime (sec.)
Random forest	0.3085	0.5218	0.3404	0.5333	10.22
Naive Bayes	0.9906	0.8651	0.9906	0.9938	0.09
SVM	0.3506	0.5473	0.3638	0.5667	0.11

4. Conclusion and Recommendations

As stated earlier, the study considered about 990 tweets collected from Ghanaian Twitter users from January to February of 2020. The tweets were collected using keywords that identify with the government in order to gather public sentiment. Prior analysis of the data showed that 14% of the tweets had positive connotations, about 33% of the tweets were negative, and 53% were neutral. This indicates some sort of public disapproval of the government. Most of the tweets also were centered around keywords like "free" from the free SHS policy the government implemented and "cathedral" from the plan of the government to build a national cathedral.

From Table 11, the SVM classifier performed slightly better (F1 score of 0.5473, MCC of 0.3638, and accuracy of 0.5667) than the random forest classifier. Literature on investigating and optimizing the various conditions like kernel types of the SVM, tree sizes, and variable split sizes for random forest and other ensemble methods could be explored in quest to improve their performance for sentiment text classification tasks.

The results of this study as shown in Table 11 also revealed that the Naive Bayes classifier has the highest Cohen's kappa statistic value of 0.99 (near perfect reliability of the algorithm), F1 score of 86.51%, Matthews Correlation Coefficient of 0.9906, and classification accuracy of 99.38%. The algorithm also recorded the lowest run time/computational time of 0.09 seconds. This makes the Naive Bayes algorithm relatively the best classifier for the sentiment text classification task.

The findings of the study are consistent with existing literature which suggests Naive Bayes models perform well with high-dimensional feature spaces and with little data. The study therefore recommends the Naive Bayes model as a viable algorithm for text classification.

This study brings to light the potential benefits of harvesting social media data from Twitter, for instance, and making analysis on them. The study provides an avenue for monitoring product performance on the markets, public sentiment, and track progress of policies, having many countless applications. Future studies can consider the creation of a web tool for performing sentiment analysis using the Naive Bayes classifier on tweets in real time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] M. Isaac and S. Ember, "For election day influence, twitter ruled social media," *The New York Times*, vol. 8, p. 11, 2016.

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Computational Linguistics*, vol. 35, pp. 311-312, 2009.

[3] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.

[4] A. Gupte, S. Joshi, P. Gadgul, A. Kadam, and A. Gupte, "Comparative study of classification algorithms used in sentiment analysis," *International Journal of Computer Science and Information Technologies*, vol. 5, pp. 6261–6264, 2014.

[5] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, Stanford University, Stanford, CA, USA, 2009.

[6] A. Onan and S. Korukoğlu, "A feature selection model based on genetic rank aggregation for text sentiment classification," *Journal of Information Science*, vol. 43, no. 1, pp. 25–38, 2017.

[7] O. Abdelwahab, M. Bahgat, C. J. Lowrance, and A. Elmaghraby, "Effect of training set size on svm and naive bayes for twitter sentiment analysis," in *Proceedings of the 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 46–51, IEEE, Abu Dhabi, UAE, December 2015.

[8] A. Hadjarian, J. Zhang, and S. Cheng, "An empirical analysis of the training and feature set size in text categorization for e-discovery," in *Proceedings of the International Conference on Artificial Intelligence and Law*, Rome Italy, June, 2013.

[9] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] C. D. Manning, P. Raghavan, and H. Schütze, "Probabilistic information retrieval," in *Introduction to Information Retrieval*, pp. 220–235, Cambridge University Press, Cambridge, UK, 2009.

[11] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, pp. 1–13, 2020.