IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 25, Issue 2, Ser. I (Mar. – Apr. 2023), PP 32-38
www.iosrjournals.org

Sentiment Analysis for Online Larning in Twitter Dataset Using Support Vector Machine

Yudha Alif Auliya[1], Muhammad Firman Satriya[2], Achmad Maududie[3]
[1,2,3](Faculty of Computer Science , University of Jember, indoneisa)

Abstract:
Background*:Technological advances bring about all kinds of changes. One example is in the field of education, by utilizing technology, it allows online learning activities to occur. Since the announcement of the policy by the government to limit activities outside the home due to the surge in positive cases of the Covid virus, it has resulted in many schools, madrasahs, universities and Islamic boarding schools being closed. This causes activities to be carried out online. The policy has reaped a lot of public opinions expressed through social media, especially Twitter. To interpret these opinions, an analytical method called sentiment analysis is needed.*
Materials and Methods*: Therefore, sentiment analysis is one of the solutions to overcome the problem in the automatic grouping of opinions. Sentiment analysis has the purpose of knowing the opinion or tendency of an opinion towards a problem or object.. Commonly used algorithms in sentiment analysis research are Support Vector Machine and Naive Bayes. In performing text classification, the two algorithms have different performance and accuracy levels. Therefore, it is necessary to make a comparison to determine the performance of each algorithm in classifying text.*
Results*: This study used several methods to analyze sentiment on online learning objects. In this study, two different architectures were used: aive bayes and support vector machine (SVM). The results obtained show that SVM has a better performance with an accuracy level of 0.77, while Naive Bayes has an accuracy level of 0.67. This is strengthened by applying k-fold cross validation with a value of k=5, which also shows an average accuracy level of 0.70 for Support Vector Machine and 0.64 for Naive Bayes).*
Conclusion:*From the analysis results that have been carried out using the Support Vector Machine (SVM), the overall performance value is superior to the model built using the Naive Bayes algorithm, both in terms of accuracy, precision, recall, and f1-score. The average accuracy value obtained from testing using k-fold cross-validation is 66% for SVM, while for Naive Bayes, it is 55.6%.*
Key Word*:Sentiment Analysis; Naïve bayes; Support Vector Machine; K-fold Cross Validation .*

Date of Submission: 05-03-2023 Date of Acceptance: 18-03-2023

I. Introduction

Technological advances bring about all kinds of changes. One example is in the field of education, by utilizing technology, it allows online learning activities to occur. Since the announcement of the policy by the government to limit activities outside the home due to the surge in positive cases of the Covid virus, it has resulted in many schools, madrasahs, universities and Islamic boarding schools being closed. This causes activities to be carried out online. The policy has reaped a lot of public opinions expressed through social media, especially Twitter. To interpret these opinions, an analytical method called sentiment analysis is needed. Therefore, sentiment analysis is one of the solutions to overcome the problem in the automatic grouping of opinions. Sentiment analysis has the purpose of knowing the opinion or tendency of an opinion towards a problem or object [1]. Commonly used algorithms in sentiment analysis research are Support Vector Machine and Naive Bayes. In performing text classification, the two algorithms have different performance and accuracy levels. Therefore, it is necessary to make a comparison to determine the performance of each algorithm in classifying text.

Support Vector Machine (SVM) is one of the supervised learning methods that is usually used to solve classification or regression problems. SVM is also able to solve linear and non-linear problems. The basic principle of SVM is to find the best hyperplane to separate the two classes by maximizing the margin / distance between the support vectors [2]. Multinomial Naive Bayes is a variation of the Naive Bayes algorithm which is usually used to classify categories in documents. Naive Bayes algorithm is an algorithm used to perform classification based on Bayes theorem that utilizes probability calculations. Naive Bayes is popular for its ease and simplicity, but this algorithm is able to provide a good level of accuracy. In addition, the Naive Bayes algorithm can also provide speed in processing large amounts of data. Naive Bayes assumes the presence or

absence of a feature in determining a class is independent, meaning that a feature in a class is not related to the existence of other features of the same class[3].

There are several previous studies which state that the Support Vector Machine has better performance, such as in a study entitled "Comparison of Naive Bayes Algorithm and Support Vector Machine for Sentiment Analysis of Film Reviews" [4] which compares SVM and Naive Bayes using confusion matrix and curves. ROC (Receiver Operating Characteristics) as a reference for model evaluation. This study shows that SVM's performance is superior to Naive Bayes with an accuracy value of 90% and AUC of 0.982. However, other studies also show that Naive Bayes performance has a better performance in classifying, such as in a study entitled "Comparison of Naïve Bayes Classifier and Support Vector Machine for Article Title Classification" [5]. In this study, applying word level n-gram and TF-IDF as feature extraction in making the classification model. The results of this study show that Naive Bayes' performance is better with an f1-score of 0.78.

Another research that also compares the Support Vector Machine with Naive Bayes is a study entitled "Comparison of the Naive Bayes Method and the Support Vector Machine on Twitter Sentiment Analysis" [6][7][8]. In this study, sentiment analysis was carried out using TF-IDF as a method for word weighting. Testing is done by using a confusion matrix. The Naive Bayes algorithm produces better performance than the SVM algorithm with an accuracy value of 73.65. Meanwhile, Naive Bayes produces an accuracy value of 70.20%. Based on the explanation of the problem, to find out which algorithm is better in classifying, this research will focus on implementing and comparing the Support Vector Machine and Naive Bayes. This study aims to determine the performance of each algorithm in classifying text using data sourced from Twitter.

II. Material And Methods

The stages of research used in this study can be seen in the following Figure 1:

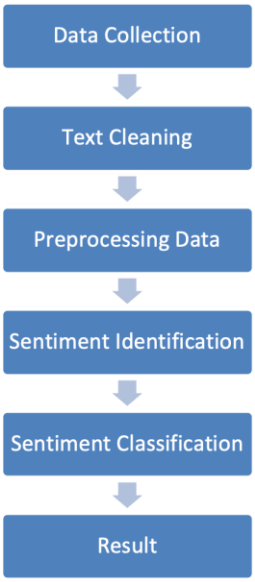

Figure 1. Research stages

1. Data Collection

The data will be used in the form of tweets collected using a crawling technique on Twitter social media. The data collection process is carried out by utilizing the Twitter API. The API key consists of 4 keys: consumer key, consumer secret, access token, and access token secret. Data collection is carried out using predetermined keywords. These keywords include online learning, studying at home, online learning, online lectures, online classes, online schools, online exams, and online exams. Tweets are crawled at different times. Data is collected by crawling from Twitter. The data used are 19680 which have negative, positive, and neutral labels. Data is labeled using the help of the TextBlob library which implements lexicon based.

2. Text Cleaning

At this stage, duplicate tweets, usernames, URLs, @mentions, #taggars, Retweets, numbers, punctuation marks, symbols, and words that only consist of 1 character will be deleted. The goal is to clean tweets of unnecessary words because they will affect the model created. Utilize the re—sub-function in the Regular Expression library.

a. Preprocessing Data

The primary purpose of data preprocessing is to get data that is clean and ready for use. The process is carried out by eliminating or changing data that does not fit into a form that is easier for the system to process.

At this stage, it consists of 4 processes: case folding, tokenization, stopwords removal/filtering, and stemming. After preprocessing, the data will be saved in a file with the extension .csv.

• Case Folding.

Case folding is to uniform all letters into all lowercase/capital letters. In this study, tweets will be uniformed into all lowercase letters. The main purpose of this process is to uniform characters to prevent the computer from recognizing different features while the data read is the same word. The implemented function is .lower(). The case folding process can be seen in Figure 2.

Figure 2. Case Folding Process

• Tokenization.

Tokenization is dividing a sentence into several words/tokens. So that the words that makeup sentences are produced in the tweet data. At this stage, the library used is TweetTokenizer, then tweets will be tokenized using the .tokenize() function. Tokenization Process can be seen in Figure 3.

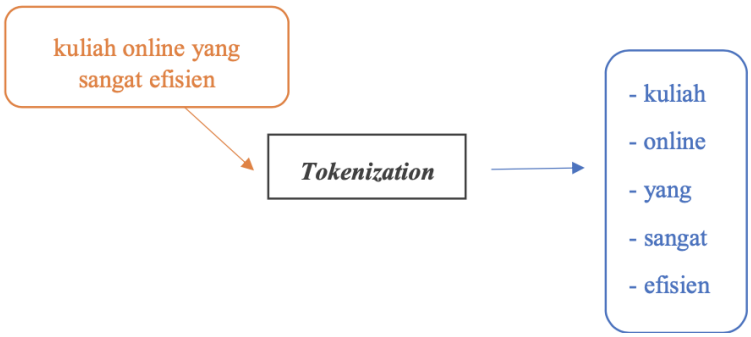

Figure 3. Tokenization Process

• Stopwords Removal / Filtering.

Stopwords Removal removes common words that usually appear in large numbers (and, so, in, to, and so on). Previously prepared non-descriptive words (stopwords), which would be grouped so that they became a stopword list. Stopwords aim to avoid shifting the meaning of the original sentence. While the addition of new words aims. Complete the list of stopwords that are not in the library. Furthermore, each token will be checked against the stopwords list, and if it includes one of the words on the list, the token will be deleted. These words need to be removed because they will affect the accuracy of the classification model. Stopwardproces can be seen in Figure 4.

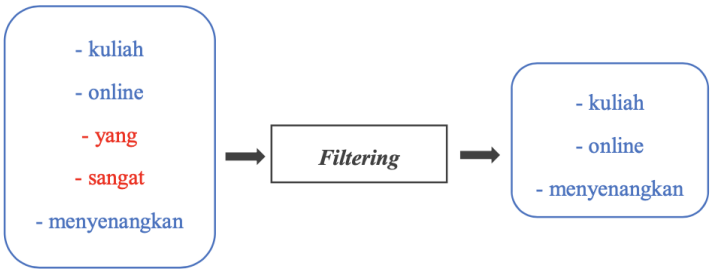

Figure 4. Stopwards Removal

• Stemming.

Stemming is the process of reducing words to their basic form. The stemming process utilizes the help of the Literature library because the text used is Indonesian. The stemming algorithm in the Sastrawi library itself is built based on the Nazief and Adriani stemming algorithms and Enhanced Confix-Stripping. The steaming process will be carried out using the factory.create_stemmer() and .stem() functions. The stemming results will be saved into a file with the extension .csv as the final result of the preprocessing stage. Stemming proces can be seen in Figure 5

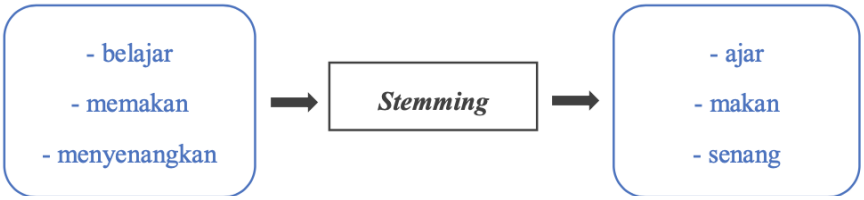

Figure 5. Stemming Proces

b. TF-IDF Transformation

TF-IDF (Term Frequency - Inverse Document Frequency) is a way to assign value/weight to a word (term) in a document [9]. The TF-IDF measures how relevant a word is to a document in a document set. This method combines two concepts to perform word weighting, namely frequency term and frequency document inverse. The frequency term counts the number of occurrences of the word in the document while the inverse document frequency shows how common or frequent the word is in the document set.(10)

III. Results

The results of the implementation of 5-fold cross validation will then be analyzed in terms of comparing the performance of the two algorithms used. The analysis intends to determine the level of accuracy of the algorithm in classifying text with data sourced from Twitter. The next step is to draw conclusions from the stages of research that have been carried out. the testing mechanism using fold cross validation can be seen in Figure 6.

Trial 1	Test	Train	Train	Train	Train
Trial 2	Train	Test	Train	Train	Train
Trial 3	Train	Train	Test	Train	Train
Trial 4	Train	Train	Train	Test	Train
Trial 5	Train	Train	Train	Train	Test

Figure 6. Testing Mechanism Using Fold Cross Validation

1. Training and testing SVM

Dataset is divided into training data and test data with a ratio of 80:20. Next, training and testing is carried out with the data. The result using SVM is the following

Table1. Cross Validation Results SVM

		Actual		
		Negative	***Positive***	***Netral***
Prediction	***Negative***	993	136	119
	Positive	214	996	141
	Netral	169	124	1044

Table 1 shows the results of the classification using 3936 test data, successfully classified 3033 data accurately, with details of 993 data labeled negative, 996 data labeled positive, and 1044 data labeled neutral. The application of 5-fold cross validation aims to determine the level of accuracy, precision, recall, and f1-score with 5-fold rotating data. The results of the process can be seen in the following Table 2.

Table2. Cross Validation Results Naïve Bayes

		Actual		
		Negative	***Positive***	***Netral***
Prediction	***Negative***	971	191	86
	Positive	354	889	108
	Netral	325	236	776

The accuracy results in Figure 2 show that the accuracy value generated using the SVM algorithm is higher than Naive Bayes both in the 1st iteration to the 5th iteration. The highest accuracy value was obtained in the 4th iteration, namely 0.76 for SVM and 0.69 for Naive Bayes.

2. Crross Validation Results

The application of 5-fold cross validation aims to determine the level of accuracy, precision, recall, and f1-score with 5-fold rotating data. The results of the process can be seen in the following figure eight until ten :

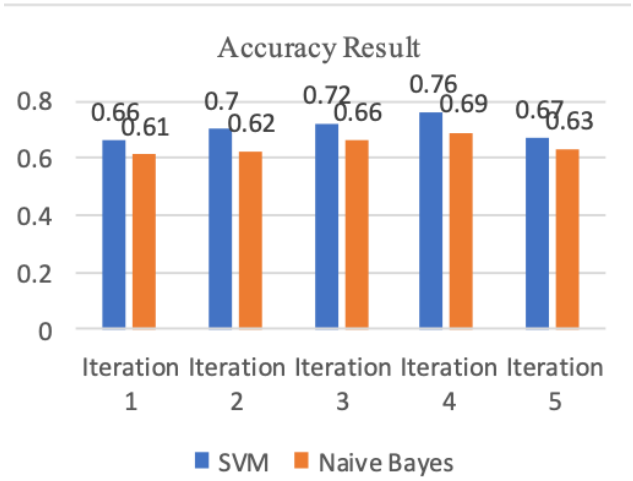

Figure 7. Accuracy Result

The accuracy results in Figure 7 show that the accuracy value generated using the SVM algorithm is higher than Naive Bayes both in the 1st iteration to the 5th iteration. The highest accuracy value was obtained in the 4th iteration, namely 0.76 for SVM and 0.69 for Naive Bayes

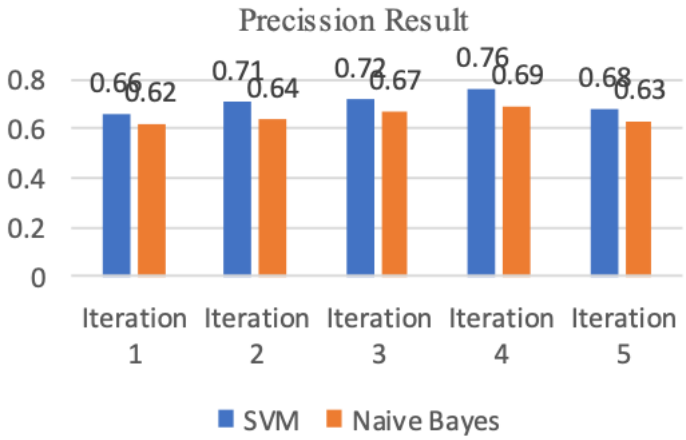

Figure 8. Precission Result

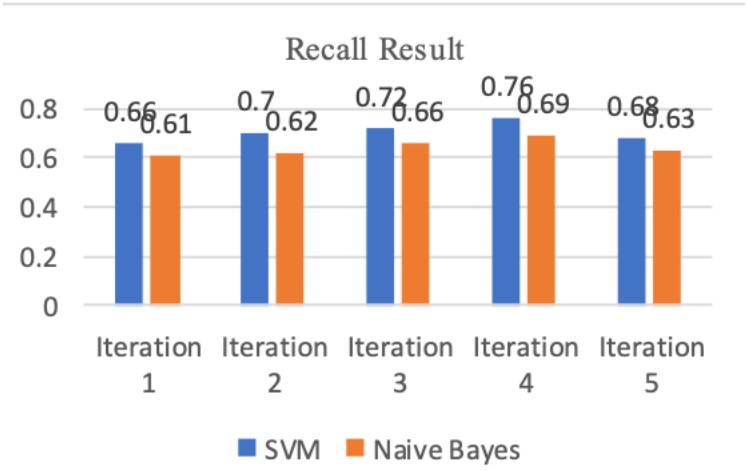

Figure 9. Recall Result

The precision and recall values generated from the two algorithms are represented in Figure 8 and Figure 9. The SVM algorithm produces better precision and recall values than Naive Bayes in each iteration. The highest precision and recall values were obtained in the 4th iteration for both SVM and Naive Bayes. While the lowest precision and recall values are in the 1st iteration.

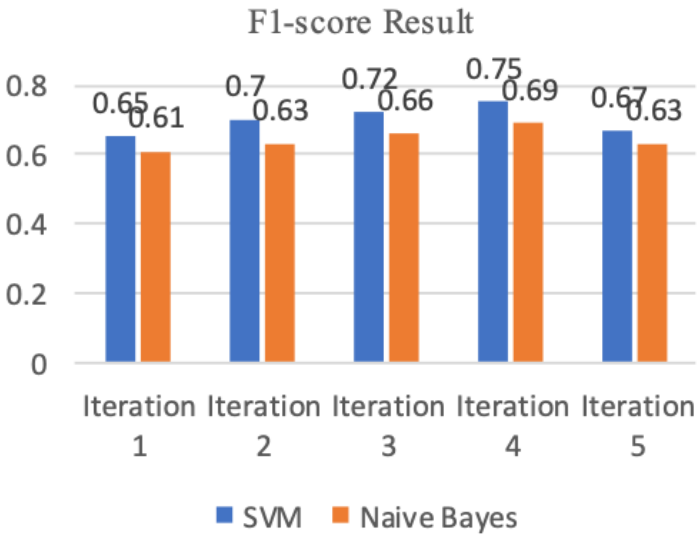

Figure 10. F1-Score Result

The results of the f1-score in Figure 10 show that the f1-score value generated by the SVM algorithm shows better performance in the 1st to 5th iterations. The highest f1-score value is obtained in the 4th iteration while the lowest f1-score is obtained in the 1st iteration for both SVM and Naive Bayes.

3. Analysis

From the results of the application of the SVM and Naive Bayes algorithms for text classification, it shows that the SVM algorithm is able to perform text classification better than Naive Bayes. The SVM algorithm is able to accurately classify test data as much as 3033 data from 3936 test data. While the Naive Bayes algorithm can accurately classify test data as many as 2636 data from 3936 test data. This is validated by applying 5-fold cross validation which performs a 5-fold training and testing process with continuously rotating train and test data. This process produces an accuracy level of the SVM algorithm with an average value of 0.70. Meanwhile, Naive Bayes produces an average accuracy value of 0.64.

IV. Discussion

From the research stages that have been carried out, it can be concluded that to classify text in sentiment analysis using data sourced from Twitter, the SVM algorithm has a better level of accuracy, precision, recall, and f1-score. The average performance values produced by SVM are accuracy 0.70, precision 0.70, recall

0.70, and f1-score 0.70. While the average values generated by Naive Bayes are accuracy 0.64, precision 0.65, recall 0.64, and f1-score 0.64. The performance results on 5-fold cross validation did not show significant changes in the values of accuracy, precision, recall, and f1-score in each iteration. So it can be stated that to classify text in sentiment analysis using data sourced from Twitter with online learning topics, the SVM algorithm has better performance.

V. Conclusion

From the research stages that have been carried out, it can be concluded that to classify text in sentiment analysis using data sourced from Twitter, the SVM algorithm has a better level of accuracy, precision, recall, and f1-score. The average performance values produced by SVM are accuracy 0.70, precision 0.70, recall 0.70, and f1-score 0.70. While the average values generated by Naive Bayes are accuracy 0.64, precision 0.65, recall 0.64, and f1-score 0.64. The performance results on 5-fold cross validation did not show significant changes in the values of accuracy, precision, recall, and f1-score in each iteration. So it can be stated that to classify text in sentiment analysis using data sourced from Twitter with online learning topics, theFSVM algorithm has better performance.

References

[1]. FahrurRozi, S. Hadi Pramono, and E. Achmad Dahlan, "Implementasi Opinion Mining (Analisis Sentimen) Untuk Ekstraksi Data Opini Publik Pada Perguruan Tinggi," J. EECCIS, vol. 6, no. 1, pp. 37–43, 2012.

[2]. Y. Tan and J. Wang, "A Support Vector Machine with a Hybrid Kernel and Minimal Vapnik-Chervonenkis Dimension," IEEE Trans. Knowl. Data Eng., vol. 16, no. 4, pp. 385–395, 2004, doi: 10.1109/TKDE.2004.1269664.

[3]. P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," Mach. Learn., vol. 29, no. 2–3, pp. 103–130, 1997.

[4]. F. Borges *et al*., "An Unsupervised Method based on Support Vector Machines and Higher-Order Statistics for Mechanical Faults Detection," in *IEEE Latin America Transactions*, vol. 18, no. 06, pp. 1093-1101, Jun 2020, doi: 10.1109/TLA.2020.9099687.

[5]. Mahajan, A., & Ganpati, A. (2014). Performance evaluation of rule based classification algorithms. *International Journal of Advanced Research in Computer Engineering & Technology*, *3*(10), 3546-3550.

[6]. M. Bernardini, L. Romeo, P. Misericordia and E. Frontoni, "Discovering the Type 2 Diabetes in Electronic Health Records Using the Sparse Balanced Support Vector Machine," in *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 1, pp. 235-246, Jan. 2020, doi: 10.1109/JBHI.2019.2899218

[7]. Aribowo, A., Basiron, H., Yusof, N., & Khomsah, S. (2021). Cross-domain sentiment analysis model on Indonesian YouTube comment. *International Journal of Advances in Intelligent Informatics*, *7*(1), 12-25. doi:https://doi.org/10.26555/ijain.v7i1.554

[8]. Kusumaningrum, R., Nisa, I., Nawangsari, R., & Wibowo, A. (2021). Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning. *International Journal of Advances in Intelligent Informatics*, *7*(3), 292-303. doi:https://doi.org/10.26555/ijain.v7i3.737

[9]. Winarno, E., Hadikurniawati, W., Septiarini, A., & Hamdani, H. (2022). Analysis of color features performance using support vector machine with multi-kernel for batik classification. *International Journal of Advances in Intelligent Informatics*, *8*(2), 151-164. doi:https://doi.org/10.26555/ijain.v8i2.821

[10]. Hartono, H., Sitompul, O., Tulus, T., & Nababan, E. (2018). Biased support vector machine and weighted-smote in handling class imbalance problem. *International Journal of Advances in Intelligent Informatics*, *4*(1), 21-27. doi:https://doi.org/10.26555/ijain.v4i1.146

[11]. A. Eid, S. Mghabghab, J. Costantine, M. Awad and Y. Tawk, "Support Vector Machines for Scheduled Harvesting of Wi-Fi Signals," in *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 11, pp. 2277-2281, Nov. 2019, doi: 10.1109/LAWP.2019.2943250.

[12]. N. S. MohdNafis and S. Awang, "An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification," in *IEEE Access*, vol. 9, pp. 52177-52192, 2021, doi: 10.1109/ACCESS.2021.3069001.

Yudha Alif Auliya. "Sentiment Analysis for Online Larning in Twitter Dataset Using Support Vector Machine." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 25(2), 2023, pp. 32-38.