OPEN ACCESS
J DATA SCI APPL, VOL. 2, NO. 2, PP.085-094, JULY 2019

E-ISSN 2614-7408
DOI: 10.34818/JDSA.2019.2.36

JOURNAL OF DATA SCIENCE AND ITS APPLICATIONS

Sentiment Analysis of Movie Reviews using Naïve Bayes Method with Gini Index Feature Selection

Riko Bintang Purnomoputra[1], Adiwijaya[2], Untari Novia Wisesty[3]

School of Computing, Telkom University
Jalan Telekomunikasi No.1 Bandung, Indonesia

[1] rikostars@student.telkomuniversity.ac.id
[2] adiwijaya@telkomuniversity.ac.id
[3] untarinw@telkomuniversity.ac.id

Received on 20-08-2019, revised on 27-08-2019, accepted on 27-08-2019

Abstract

Sentiment analysis can be used to classify movie reviews whether the movie is good or bad. Unstructured data and a lot of data attributes become a problem in the classification process because it requires much time and computational capabilities. Feature selection in the classification process is needed to overcome computational time. In this paper, we use the Gini Index and multinomial Naive Bayes to reduce the dimension of features and document classification, respectively. Multinomial Naïve Bayes (MNNB) is a popular classifier used for document classification. This study aims to use the Gini Index Text (GIT) for text feature selection with MNNB classifier to classify movie review into positive and negative classes. The data that is used is IMDB dataset that contains reviews in English sentences, the data were divided into two parts, training data is 90% and data testing is 10%. The test results prove that the Gini index as a selection feature can increase accuracy where accuracy without feature selection is 56% and with feature selection of 59.54% with an increase of 3.54%.

Keywords: sentiment analysis, movie review, multinomial naïve bayes (MNNB), gini index text (GIT)

I. INTRODUCTION

With current technological advances, many sites inform about movies that are currently or will be aired, such as IMDB, Rotten Tomatoes, and Metacritic. To determine whether a movie is good or bad, it is necessary to look at the reviews of viewers of the movie so that the movie can attract attention to watch. Some movie reviewers pour their reviews expressing their opinions based on the persona, and there were differences of opinion in the movie review. Some reviews may appear clearly included in positive or negative reviews, but there are still reviews that are not clearly categorized.

One technique for classifying opinions is sentiment analysis or known as opinion mining. With this technique, it can be determined whether the review is positive or negative. There are two approaches to classify them, namely using machine learning and using lexicon-based method. Both approaches classify text into positive and negative classes depending on the polarity of sentences. Lexicon approach generally engages with dictionaries of opinion words or known as sentiment dictionaries to define the sentiment orientation as positive or negative. Meanwhile, machine learning approach uses manual data classification from the dataset and trains the classifier from the sample or called training data, which will later be tested in testing data [1]. The research in [2] provides results of various methods for sentiment analysis, and it can be inferred that the machine learning approach produces the highest accuracy. Nevertheless, the large number of data attributes can be a problem in the classification since it slows down the process and leads to misclassification. In [3], feature selection can improve the accuracy of MNNB performance. However, MNNB is more sensitive to the feature selection method. Therefore, some feature selection may not be able to improve the classification performance.

In [14], the researchers suggested a new complete Gini Index Text (GIT), which is an enhanced Gini Index feature selection that works better with K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) text classification. The results of this study prove the performance of GIT can reduce irrelevant features and still maintain representative features so that it can improve classification performance. However, there is not much research that uses GIT, so it is difficult to see the performance of the GIT for classifier, whether GIT can improve performance with other classification techniques such as Multinomial Naïve Bayes (MNNB).

Based on the problem, the research focuses on the Multinomial Naïve Bayes classification technique using the Gini Index Text feature selection for classifying movie reviews and conducting tests to determine the performance of the model based on the test scenario. The limitation of the problem in this study is the review data used is sample data derived from polarity data on research-based Kaggle [13], which has been added from the IMDB website totaling 4,000 labeled reviews consisting of 2,000 positive reviews and 2,000 negative reviews. The results of this study are the performance of the MNNB with GIT feature selection by calculating the accuracy of classification.

In Section 2, we discuss some research that related to this study, the Gini-Index Text feature selection, along with calculating the GIT score, dataset, and preprocessing that used in this study, the MNNB Classifier Theory and measuring performance used. In Section 3, we present our proposed system. In Section 4, through experimental results, we compare and discuss the classification performances based on the test scenario. In Section 5, we draw conclusions and contemplate future studies.

II. LITERATURE REVIEW

There are many studies on sentiment analysis on topics such as social media comments, products, politics, and more. There are so many techniques for classifying text. Many researchers are trying to do a combination of techniques to achieve better performance. Research [5] compared to the various feature selection for the Naïve Bayes algorithm. The goal of the study is to determine the selection features that are accurate and stable using Newsgroups and Reuters-21578 data. In the study, preprocessing data are not displayed so that data processing is not explained. The highest accuracy results obtained with the Chi feature of 81% for Reuters-21578 data and 84% for Newsgroups and for stable results obtained using the Term Frequency (TF) and Domain Frequency (DF) techniques but are inefficient compared to other methods. Research [8] compares several features selection, namely Gini Index, Weight Formula (NG), and Word Frequency Mutual Information (MIDF) with KNN and Naïve Bayes classification techniques. The purpose of this study is to compare and validate new weight features based on the Gini Index on the performance of classification techniques. The results of this study, the Gini Index gets a very good performance that is 75% with KNN and 95% with Naïve Bayes. Research [9] compares the Multinomial Naïve Bayes (MNNB) and SVM classification techniques with 18 different features selection. Unfortunately, it does not explain in detail data preprocessing that had been done. The study shows that the Gini Index, Weighted Log-Likelihood Ratio (WLLR), and Cross-Entropy for Text (CET) on MNNB give identical results. Based on the results of this study concluded that SVM requires fewer features to achieve maximum results, while Naïve Bayes requires more features to achieve optimal results and is more sensitive to the feature selection method.

In [11], researchers compared the Gini Index Text with various feature selection on IMDB dataset data, Spanish dataset, and Portuguese dataset data extracted in advance with n-grams and using MNNB, SVM, and Weighted SVM (WSVM) classification techniques. From the conclusion, it was found that the selected features of the Gini Index, Domain Frequency, CET, and CHI got the best performance, 90% with SVM, and 92% with MNNB. The SVM classification technique is a fast algorithm and can work with high dimensions while MNNB works better on short documents or to classify sentences, WSVM works better than SVM if it does not use the feature selection. In research [14], K-Nearest Neighbor (KNN) method and Support Vector Machine (SVM) with Gini Index Text work better for text classification, producing an average performance of 97% in Reuters-21578 data. Moreover, GIT can eliminate many irrelevant and redundant subset of features and retain many representative features that will improve overall classification performance.

A. *Gini Index Text*

The Gini Index is used to separate attributes. This technique is generally used in decision trees and successfully improves the precision of classifications. There have been many studies to improve the Gini Index method, one of which is the Gini Index Text (GIT), which was introduced in [14]. GIT was made to work on documents with very many features. For each feature in the film, the review will be calculated based on GIT A, GIT B, and GIT C in the positive and the negative class. This method can reduce the features of a subset of features while also retain many representative features. GIT is defined as:

$$GiniTextA(w, c_i) = \quad P(c_i|w)^{\,2} \tag{1}$$

$$GiniTextB(w, c_i) = \ \left|\frac{P(c_i|w)^{\,2}}{log_2\, P(w)}\right| \tag{2}$$

$$GiniTextC(w, c_i) = \ \frac{P(c_i|w)^{\,2}}{|log_2\, P(w|c_i)^{\,2}|} \tag{3}$$

With:
$P(w|c_i)$ is the probability of movie reviews feature w for class c_i.

$P(c_i|w)$ is the probability for class c_i in the appearance of the movie reviews feature w.

$P(w)$ is the probability of movie reviews feature w with the total number of words in the movie review document.

Gini Index Text A will produce a high score if the word w only appears in one class. Yet if w is evenly distributed among classes, it has a low score. Gini Index Text B has the $P(w)$ is normalized with logarithms and absolute values to increase its deviation so that it can calculate specific features and general features more efficiently. On Gini Index Text C, the $P(w)$ is normalized as $P(w|c_i)^{\,2}$ with logarithm and absolute value to increase its deviation. Then, if a feature w is specific and large class, $P(w|c_i)^{\,2}$ has very small value and it can be excluded from the comparison process with threshold.

B. *Multinomial Naïve Bayes*

The Naïve Bayes Classifier is a simple probabilistic classifier that applies the Bayes theorem with a high assumption of independence. Bayes' theorem is a theorem used in statistics to calculate the probability of a hypothesis [12]. Generally, there are three distribution models Bernoulli, Multinomial, and Poisson. These three models are used as a classifier, namely Bernoulli Naïve Bayes, Multinomial Naïve Bayes, and Poisson Naïve Bayes and not as an independent document.

Multinomial Naïve Bayes (MNNB) calculates the number of words in a document so that it assumes the independence of the appearance of words in the document. This assumption shows that the likelihood that each word event in a document is free does not take into account word order and word context in documents [12]. MNNB will be used to classify movie review documents into positive and negative classes MNNB defined as:

$$argmax\ P(C_j)\prod_i P(w_i|C_j) \tag{4}$$

$$P(w_i, c_k) \ = \frac{1 + \ Count(w, c_k)}{|V| + \ Count(c_k)} \tag{5}$$

With:

$Count(w, c_k)$ is the sum of movie review feature w that appear in a c_k classes (positive and negative)

$Count(c_k)$ is the sum of all movie review feature w in the c_k classes (positive and negative)

$|V|$ is all the vocabulary that appears in movie review documents

C. *Measuring Performance*

Confusion Matrix is a method for measuring performance for classification in machine learning. Confusion matrix taking into account four terms that are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) which respectively denote the number of positive reviews classified positive, the number of negative reviews classified as negative, the number of negative reviews classified positive and the number of positive reviews classified as negative.

TABLE I
CONFUSION MATRIX

Predicted	**Actual Class**	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Using a confusion matrix will be very useful for calculating the performance of classification results such as accuracy of precision, recall, and F1-score. The results of the testing data prediction are validated by using accuracy testing to measure the classification performance of the model based on the confusion matrix.

Accuracy is a general measurement method that is often used to see the level of success of conducted experiments. Accuracy is calculated based on the correctness of the classification results of all documents, the number of true predictions divided by the total number of true and false predictions defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (6)$$

III. RESEARCH METHOD

The system design used for this research is as follows. First, the obtained data is processed to reduce the number of movie review features with preprocessing namely tokenization, case folding, stopword removal, and lemmatization. Subsequently, the data is divided into training and testing. For training data, features are selected based on the Gini Index Text (GIT) score. After features are selected, Multinomial Naïve Bayes (MNNB) is trained on selected features. Testing data is classified with MNNB that have been trained, and then the accuracy of the classification results is calculated. The flowchart of the proposed is illustrated in figure 1.

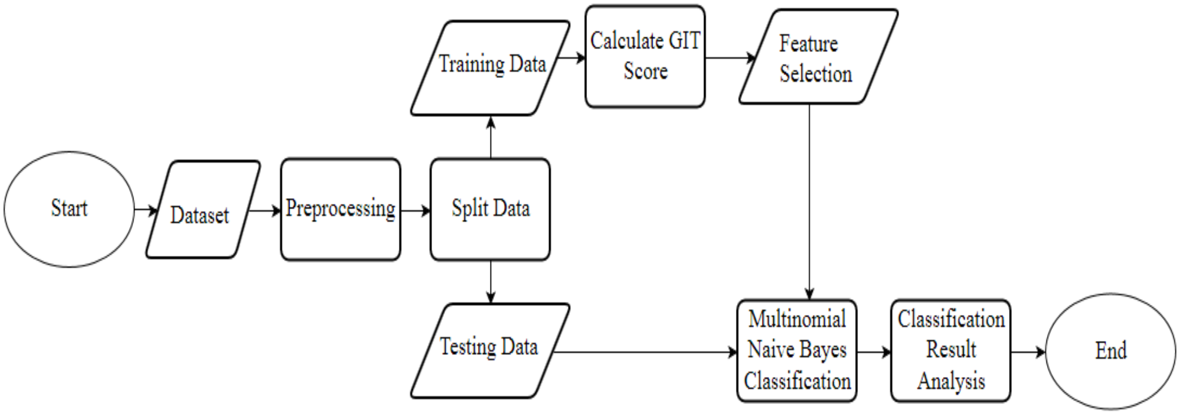

Fig. 1 System Design Flowchart

A. Dataset

The beginning of the system design is data retrieval. This study uses the dataset in research [13] with the title Sentiment Polarity Dataset Version 2.0 on Kaggle data that has been updated at https://www.kaggle.com/nltkdata/movie-review, with total files containing 32,967 positive review files and 31,752 negative review files. The data are divided into three parts: 2,000 files, 3,000 files, and 4,000 files. Each section has 50% positive review files and 50% negative review files and is divided into 90% training data and 10% testing data.

B. Preprocessing

At this process, the data is processed, and word that has no value is removed to reduce the number of features to increase the effectiveness and efficiency of the classification process. The following are the stages of pre-processing.

- Tokenization
- Data Cleaning
- Case Folding
- Stopwords Removal
- Lemmatization

C. Calculates the Gini Index Text Score

The features of the training data will be calculated using the Gini Index Text (GIT) formula GIT A, GIT B, and GIT C. Each feature will have a score that will be used to select features with the highest score. Here is the example of ten positive features in 4,000 data e features of the training data will be calculated using the Gini Index Text (GIT) formula GIT A, GIT B, and GIT C. Each feature will have a score that will be used to select features with the highest score. Here is the example of 10 positive features in 4,000 data:

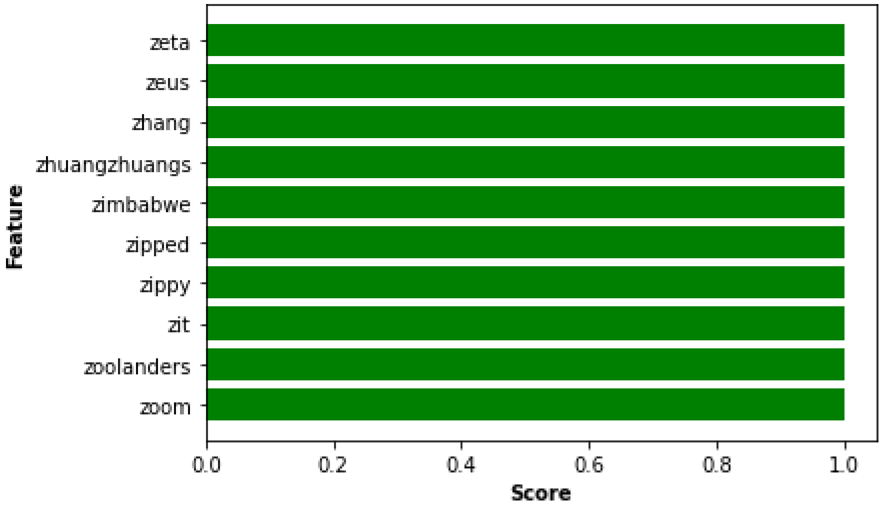

Fig. 2 Ten Best Positive Features in 4,000 Data using GIT A

Fig. 2 shows 10 movie review features with the highest score in the positive class. Based on the calculation of features scores with GIT A get a maximum score of 1. This proves that these features only appear in one class, thus achieve the highest score.

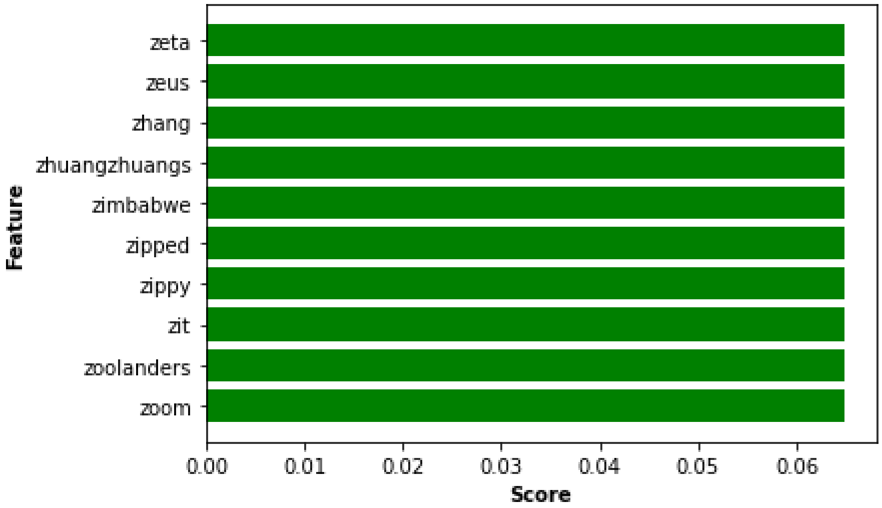

Fig. 3 Ten Best Positive Features in 4,000 Data using GIT B

Fig. 3 shows 10 movie review features with the highest score in the positive class. Based on the calculation of features scores with GIT B get a maximum score of 0.065. This proves that these features only appear in one class and are divided by $log_2 P(w)$ so that they can achieve better deviations.

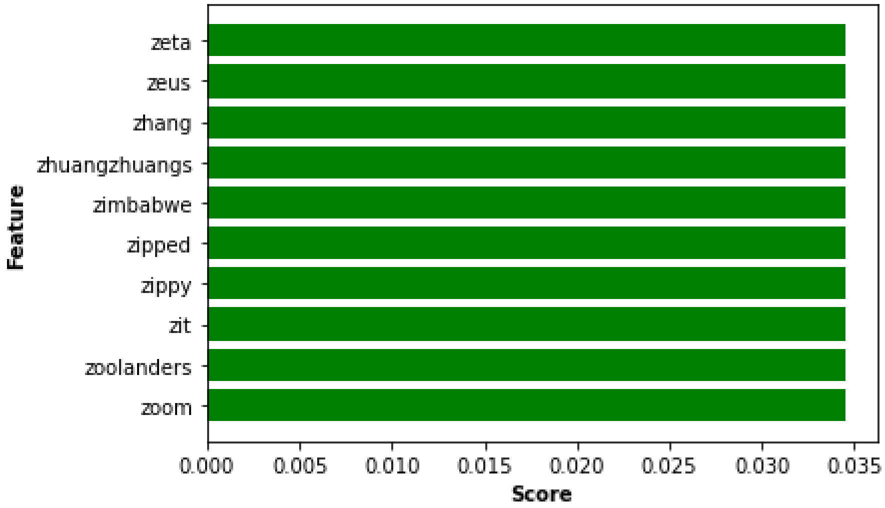

Fig. 4 Ten Best Positive Features in 4,000 Data using GIT C

Fig. 4 shows ten movie review features with the highest score in the positive class. Based on the calculation of features scores with GIT C get a maximum score of 0.034, this proves that these features only appear in one class and are divided by $|log_2 P(w|c_i)^2|$. The goal is almost the same as GIT B to achieve a better deviation.

Based on the Fig. 2, Fig. 3, and Fig. 4 above points out that GIT A gets a score of 1, GIT B gets a score of 0.065, and GIT C gets a score of 0.034 for each of the ten best features. After getting score for each feature, the next step is feature selection, which is to select the features with a high GIT score.

D. Multinomial Naïve Bayes Classification

In the Multinomial Naïve Bayes classification process, the features that have been selected based on the best Gini Index Text score of k features are used by the Multinomial Naïve Bayes for the model training process. After that MNNB who had been trained will predict positive or negative labels from the test data. Testing will be conducted periodically by adding features with the highest score every iteration to be able to see the performance of the selected features. The next process is calculate the accuracy of the classification, prediction, which will be explained in the next chapter.

IV. RESULTS AND DISCUSSION

In the testing phase, two testing scenarios are performed. The first is tested using the Multinomial Naïve Bayes method only without using feature selection. The second is testing using the Multinomial Naïve Bayes by selecting the Gini Index Text feature, meaning that this test will reduce features based on the Gini Index Text score.

A. Testing without Selection Features

In this test, the data were classified using the Multinomial Naïve Bayes method to determine the level of accuracy in the prediction classification of movie reviews without using feature selection. The features generated from the pre-processing process are directly classified without being reduced. The data were divided into three parts, using 2,000, 3,000, and 4,000 d. The results of this test are the accuracy of the classification:

TABLE II
MULTINOMIAL NAÏVE BAYES PERFORMANCE RESULTS

Data	**Accuracy**	***Features***
2000	62.5%	6968
3000	54.84%	8825
4000	56%	10336

Based on Table II shows the results of performance using 2,000, 3,000, and 4,000 data with an average accuracy of 57.78%. The best results were obtained using 2,000 data, with an accuracy of 62.5%.

B. Testing with Selection Features

In this test, the data were classified using the Multinomial Naïve Bayes method to determine the level of accuracy in the prediction classification of film reviews with the best k features. The testing phase will be conducted periodically in the amount of 100 positive features and 100 negative features per iteration to get the best results based on the features selected. The data were divided into three parts using 2,000, 3,000, and 4,000 data. The results of this test are the accuracy, and the following F1-scores are the results of the test:

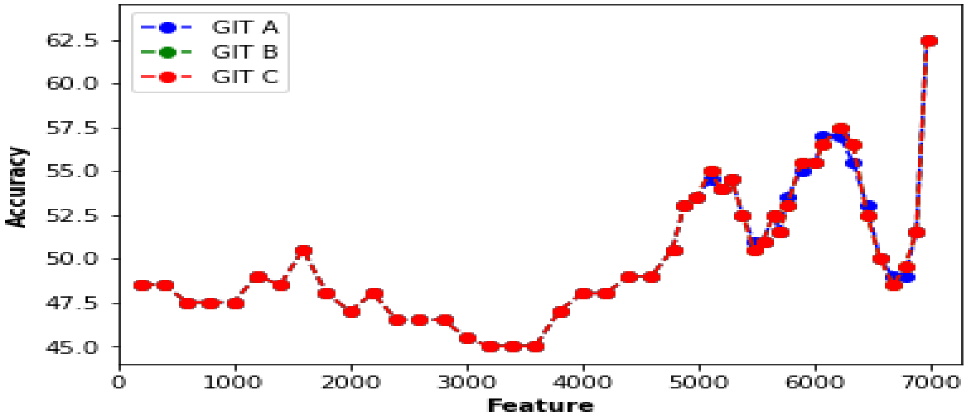

Fig. 5 Accuracy Results Using 2,000 Data

Figure 5 shows the accuracy and F1-score line graphs using 2,000 data on the number of features used with a total of 6,968 features. The highest accuracy obtained by GIT of 62.5% this indicated that GIT A, B, and C did not improve accuracy result.

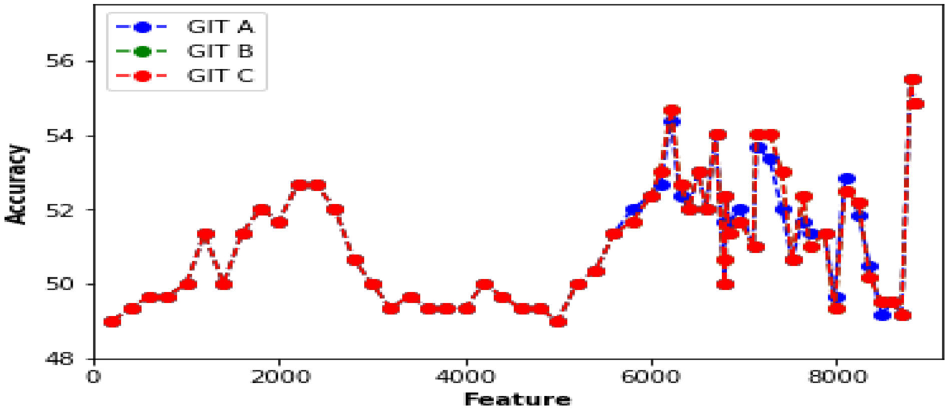

Fig. 6 Accuracy Results Using 3,000 Data

Figure 6 shows the accuracy and F1-score line graphs using 3,000 data on the number of features used with a total of 8,825 features. It can be seen that the performance of GIT A tends to be lower than GIT B and C by using 6,000 to 8,600 features. Feature selection with GIT achieves the highest accuracy on 8,795 features.

TABLE III
HIGHEST ACCURACY RESULTS ON 3,000 DATA

Method	Accuracy	Features
MNNB with GIT A, B, C	55.51%	8795
MNNB	54.84%	8825

Table III shows the accuracy results using 8,795 features in 3,000 data, the accuracy of GIT A, B, and C are higher than not using feature selection. This shows that the features selected can improve classification performance. The accuracy result is 55.51%.

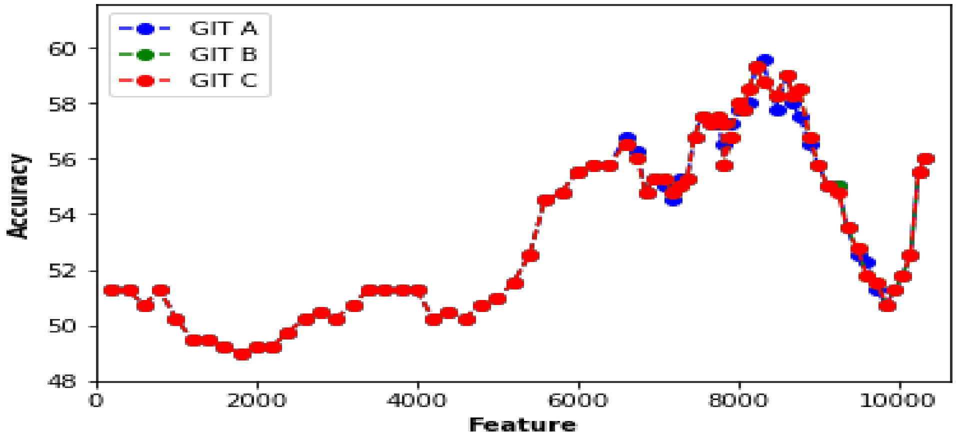

Fig. 7 Accuracy Results Using 4,000 Data

Figure 7 shows the accuracy line graphs using 4,000 data on the number of features used with a total of 10,336 features. The classification results show that accuracy with feature selection is higher than not using feature selection. The performance of GIT A achieves higher accuracy compared to GIT B and C. This shows that the features selected in GIT A have more influence on the classification of data.

TABLE IV
HIGHEST ACCURACY RESULTS ON 4000 DATA

Method	Accuracy	Features
MNNB with GIT A	59.54%	8312
MNNB with GIT B, C	59.29%	8217
MNNB	56%	10336

Table IV shows the accuracy results from 4,000 data. The accuracy of GIT A, B, and C are higher than not using feature selection. This shows that the features selected can improve classification performance with the accuracy result are 59.54% with GIT A and 59.29% with GIT B and C.

From the test results, it can be seen that there is an increase in accuracy by using feature selection. For example, in Table 3 the result is 54.84% without feature selection and 55.51% with 8,795 selected features, an increase of 0.67% with GIT A, B and, C. In Table 5 shows accuracy results of 56% without feature selection and 59.54% with GIT A using 8312 features an increase of 3.54% and 59.29% with GIT B and C an increase of 3.29%.

This test also proves different accuracy results using GIT A, GIT B, and GIT C . For example, Table 4 shows that GIT A gets a higher accuracy compared to GIT B and GIT C at 59.54% using 8,312 features while GIT B and GIT C got 59.29% using 8,217 features with a difference of 0.25%.

Based on the results, Multinomial Naïve Bayes with Gini Index Text can work well. This proves that using feature selection can make accuracy increases, for instance, 4,000 data (which should have 10,336 features), and with only 8,795 features it produces better accuracy so that selecting features can reduce misclassification. The bigger the data, the better the features will be compared to the smaller data so the classification performance can look better. To see the results of deeper feature reduction the authors suggest using the threshold in feature selection so that the performance of GIT A, B and C can be seen clearly.

V. CONCLUSION

Using the Multinomial Naïve Bayes Method with Gini Index Text Feature Selection can improve the accuracy of classification results. For instance, the 4,000 data has an increase in accuracy of 3.54% with 8,312 features. In 3,000 data, the accuracy increased by 0.67% with 8,795 features. Gini Index Text feature selection produces a different performance as in GIT A has a better performance compared to GIT B and GIT C as seen in 4,000 data accuracy result are 59.54%, while GIT B and GIT C is 59.29% with a difference of 0.25%. GIT B and GIT C produce the same performance in every test. This happens due to the features selected are not different. Feature selection can be seen to be more influential on the classification of data using 4,000 data compared to 3,000 data and 2,000 data. This shows that bigger data resulting in better features. The result of this study can be used to analyze the effect of Gini Index Text feature selection on Multinomial Naïve Bayes to classify documents. We recommend using different approaches on selecting features such as threshold so feature reduction can be seen more clearly. Thus, it can be concluded that the Gini Index Text Feature Selection can improve the performance of the Multinomial Naïve Bayes and can be an alternative solution to classify movie reviews data quite accurately. In the future works to see a more detailed performance from the Gini Index Text feature selection is by benchmarking with other popular feature selection, such as Chi, Mutual Information and TF-IDF and adding more classes in the data.

REFERENCES

[1] Dhaoui, C, Webster C and Tan L, "Social media sentiment analysis: lexicon versus machine learning", Journal of Consumer Marketing, Vol. 34 No. 6, pp. 480-488, 2017.

[2] Vimalkumar, Bhumika, “Analysis of Various Sentiment Classification Techniques”, International Journal of Computer Applications (0975 – 8887) Volume 140 – No.3, 2016.

[3] Prusa, Joseph D., Taghi M. Khoshgoftaar and David J. Dittman. "Impact of feature selection techniques for tweet sentiment classification." The Twenty-Eighth International Flairs Conference. 2015.

[4] Narayanan Vivek,Arora Ishan, Bhatia Arjun, “Fast and accurate sentiment classification using an enhanced Naive Bayes model”, 2013.

[5] Erik Lux “Feature selection for text classification with Naive Bayes” Charles University, Prague, 2012.

[6] W. Shang, H. Huang, H. Zhu, “A novel feature selection algorithm for text categorization”, Expert Systems with Applications, vol. 33, no. 1, pp. 1-5, 2007.

[7] Prasad Tirath, Ahuja Sanheev, “Sentiment Analysis of Movie Reviews: A study on Feature Selection & Classification Algorithms”, National Institute of Technology Raipur, India, 2016.

[8] Jia Xiaoqiang, Sun Jiagya, “AN IMPROVED TEXT CLASSIFICATION METHOD BASED ON GINI INDEX” Journal of Theoretical and Applied Information Technology, Vol. 43 No.2, 30th September, 2012.

[9] Varela Pedro, Martins Andre, Aguiar Pedro, Figueiredo Mario, “An Empirical Study of Feature Selection for Sentiment Analysis” Instituto Superior T´ecnico, Lisboa, Portugal, 2013.

[10] Iqbal Furqan “Sentiment Analysis Using Ensemble Learners and Gini Index”, International Journal of Engineering and Techniques - Volume 4 Issue 2, Mar-Apr 2018.

[11] Varela Pedro,” Sentiment Analysis”, Instituto Superior T´ecnico, Lisboa, Portugal, 2012.

[12] Butar Thio, Fauzi Mochammad, Indriati, “Penentuan Rating Review Film Menggunakan Metode Multinomial Naïve Bayes Classifier dengan Feature Selection berbasis Chi-Square dan Galavotti-Sebastiani-Simi Coefficien”, Universitas Brawiijaya, 2018.

[13] Bo Pang and Lillian Lee, “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts”, Proceedings of the ACL, 2004.

[14] Park, Heum & Kwon, Soonho & Kwon, Hyuk-Chul. “Complete Gini-Index Text (GIT) feature-selection algorithm for text classification”, 2nd International Conference on Software Engineering and Data Mining, SEDM 2010, pp. 366 – 371, 2010

.