International Journal of Information System & Technology
Akreditasi No. 158/E/KPT/2021 | Vol. 6, No. 4, (2022), pp. 419-430

Comparison of the Multinomial Naive Bayes Algorithm and Decision Tree with the Application of AdaBoost in Sentiment Analysis Reviews PeduliLindungi Application

Cecep Muhamad Sidik Ramdani[1], Andi Nur Rachman[2], Rizki Setiawan[3]*
Department of Information System Universitas Siliwangi Tasikmalaya, Indonesia
Email: [1]cecepmuhamad@unsil.ac.id, [2]andy.rachman@unsil.ac.id, [3]rizkisetiawan@unsil.student.ac.id

Abstract

One solution that the Indonesian government has implemented in controlling and tracking COVID-19 cases is using the PeduliLindungi application. User review data on the PeduliLindungi application is available on the Google Play Store, the data can be analyzed to determine the trend of public sentiment towards the PeduliLindungi application using sentiment analysis techniques. One of the methods used for sentiment analysis is machine learning, but in the machine learning method there is a problem, namely the relatively low level of accuracy. In this study, there are 2 machine learning algorithms that are used and compared, namely the Multinomial Naïve Bayes (MNB) and Decision Tree (DT) algorithms combined with the AdaBoost (AB) method to improve the accuracy of the PeduliLindungi application review data classification accuracy. In the experiment conducted, the tendency of public sentiment towards the PeduliLindungi application was 67% positive and 33% negative from a total of 8305 data. Multinomial Naïve Bayes before being combined with AdaBoost produces an average accuracy value of 83,7%, while Decision Tree produces an average accuracy value of 82,8%. After being combined, MNB+AB produces an average accuracy value of 88,8%, while the DT+AB method produces an average accuracy value of 84,1%. The use of AdaBoost can improve the accuracy of the Multinomial Naive Bayes algorithm and Decision Tree for the PeduliLindungi application review data classification process.

Keywords: *Sentiment Analysis, PeduliLindungi, Multinomial Naive Bayes, Decision Tree, AdaBoost.*

1. Introduction

One of the solutions implemented by the Indonesian government for controlling and tracking COVID-19 cases is to use the PeduliLindungi application. The PeduliLindungi application is an application developed to assist government agencies in carrying out the tracking process to stop the spread of COVID-19. It's been almost a year since the PeduliLindungi application has been used by the community in carrying out activities, however, there are still many problems stemming from the application. The problems that exist include there are still many vaccination data that have not appeared in the application. Then there are also cases of data theft that occurred through the PeduliLindungi application where the data is important personal data of the community.

On the Google Play Store service, the PeduliLindungi application has a high total download of more than 50 million downloads. User reviews of the PeduliLindungi application on the Google Play Store can be analyzed with a special technique, namely sentiment analysis, so that it can be classified regarding whether public reviews of the PeduliLindungi application are positive or negative.

Methods that can be used for the sentiment analysis process include machine learning-based, lexicon-based and hybrid approach methods [1]. For methods based on machine learning, it is divided into supervised learning and unsupervised learning [2]. The classification process in sentiment analysis is basically a problem with text classification,

ISSN : 2580-7250

therefore the supervised learning method can be directly used and implemented such as using the Support Vector Machine (SVM) or Naïve Bayes Classifier (NBC) method [3].

One of the algorithms that can be used is Multinomial Nave Bayes which is another variation of Nave Bayes, Multinomial Nave Bayes is a Probabilistic Reasoning method that aims to classify data in certain classes. By using the Multinomial Naïve Bayes method, the classification process can be carried out well and has a fairly high average level of accuracy [4].

Another algorithm that can be used to perform sentiment analysis is the Decision Tree. Decision Tree is a simple representation for classification, the process in the Decision Tree is to convert data in the form of tables into a tree or tree and then change the shape of the tree into rules or rules [5].

The problem that exists in the machine learning method according to Sommerville is that it has a relatively low level of accuracy [6]. In research related to sentiment analysis on user reviews of the COVID-19 information application, namely PeduliLindungi using the Naïve Bayes Classifier, SVM and KNN methods, the accuracy value for SVM is 76,5%, followed by NBC 72,3%, and KNN has an accuracy of 59, 1%, so other methods are needed to carry out sentiment analysis and further classification to improve accuracy results [7].

To improve the accuracy of the test results and also reduce bias, a method must be added, one method that can be used is the ensemble method using AdaBoost (Adaptive Boosting) [8]. The use of AdaBoost to increase the accuracy of the classification algorithm has been proven, as in a study related to sentiment analysis of restaurant reviews using the Naïve Bayes algorithm which initially only had an accuracy value of 70% increased to 99,5% after using AdaBoost [9].

Based on the description above, sentiment analysis will be carried out using the Multinomial Naïve Bayes algorithm and Decision Tree by applying AdaBoost boosting to increase accuracy and compare the results of the two algorithms before and after the implementation of AdaBoost.

2. Research Methodology

In research [10] related to the sentiment analysis of moving the new capital by comparing the Bernoulli Nave Bayes and Multinomial Nave Bayes classification methods, the Multinomial Nave Bayes performance value is higher than Bernoulli Nave Bayes, namely 93,45% for Multinomial Nave Bayes and 90,19% for Bernoulli Naive Bayes.

Judging from the results of previous studies, the application of Decision Tree has a high accuracy value compared to other algorithms. In research related to sentiment analysis of the PSBB effect on twitter, Decision Tree has the highest accuracy value compared to KNN and Naïve Bayes with an accuracy value of 83,3% for Decision Tree, 80,80% for KNN and 80,03% for Nave Bayes [11].

Reviews of the PeduliLindungi application have also been analyzed and investigated, one of which is in research [12] using the SVM algorithm. The results of the research on sentiment analysis of the PeduliLindungi application that has been carried out have obtained an accuracy model of 70,46%. In another study related to sentiment analysis on user reviews of the COVID-19 information application, namely PeduliLindungi using the Naïve Bayes Classifier, SVM and KNN methods, the accuracy value for SVM was 76,5%, followed by NBC 72,3%, and KNN had an accuracy of 59,1%, so we need other methods in conducting sentiment analysis and further classification to improve the accuracy of the results [7].

On research related to the analysis of user reviews for the PeduliLindungi application on google play using the support vector machine and naïve bayes algorithm based on particle swarm optimization, NB+PSO produces an accuracy value of 69,00% and AUC value = 0,859. SVM+PSO produces an accuracy value of 93,0% and AUC value = 0,977.

It can be concluded in this study that the SVM+PSO algorithm has a higher accuracy than the NB+PSO algorithm [13].

Other research related to sentiment analysis on PeduliLindungi application using TextBlob and VADER Library conclude that the VADER Library is specifically aimed at analyzing sentiment on social media. This shows that the VADER tweet library is dominated by positive, neutral, and negative sentiments, and the difference in sentiment in the two data collection periods is not too large [14].

AdaBoost is one method to increase the accuracy value in the classification process, as evidenced in research [9] related to restaurant review sentiment analysis using the Naïve Bayes algorithm which initially only had an accuracy value of 70% increased to 99,5% after using AdaBoost.

2.1. Originality

In this study, the data used came from user reviews on the Google Play Store on the PeduliLindungi application. The labeling process carried out on reviews uses vaderSentiment. vaderSentiment is one of the automatic labeling, for the provisions of the score used in this study is if the review has a score > 0 then the review is labeled positive, if the review has a score < 0 then the review is labeled negative and if there is data that does not enter in the existing conditions, the data will be deleted. The classification methods compared in this study are Multinomial Naïve Bayes and Decision Tree by applying AdaBoost boosting with the aim of increasing the accuracy of the two methods used.

2.2. System Design

The proposed system design consists of 6 stages: (1) Literature Study, (2) Data Collecting, (3) Data Processing, (4) Feature Extraction, (5) Text Classification, (6) Result Analysis. These stages are shown in Figure 1.

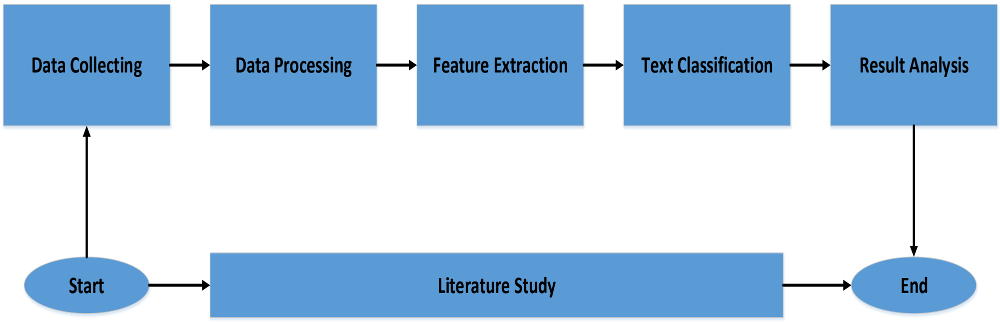


Figure 1. The system design of proposed research

a) Literature Study

At this stage, learn all things related to sentiment analysis and the application of using the python programming language from various library sources in the form of books, journals, final projects, digital newspapers and library search results on the internet. Literature studies are also carried out during the process of data collecting, data processing, feature extraction, text classification and analysis of results as a guide in conducting research.

b) Data Collecting

Data collecting was carried out using data collection techniques in the PeduliLindungi application review on the Google Play Store used in this study. The data retrieval process uses the python programming language with the Google Play Scraper library so that it can easily obtain PeduliLindungi application review data on the Google Play Store.

c) Data Processing

This stage is the process of processing the data obtained from the data retrieval stage with the aim of controlling the size of the word list and is expected to improve performance in the classification process carried out. At this stage there is a text preprocessing process consisting of data cleaning, case folding, tokenization, spell checking, stop removal, stemming and translate.

Data labeling and polarity stages are also carried out at this stage, for the labeling stage using the vaderSentiment library provided that if the data has a score > 0 then the label is positive and if the score is < 0 then the label is negative. Then at the polarity stage if the data has a positive label then the polarity value is 1 and if the data has a negative label then the data polarity value is -1.

d) Feature Extraction

At this stage, two processes are carried out, namely CountVectorizer and TF-IDF. CountVectorizer is a feature to convert text into vector representation and also to count the number of words in a document. TF-IDF is an approach to weighting the number of words by a measure of how often they appear in a document. This approach is used to overcome the problem where the number of raw words leading to the feature is too burdensome for words that occur frequently, it can cause suboptimal in some classification algorithms. For the application in this research, use the Tfidfvectorizer library with the python programming language.

e) Text Classification

At this stage the training data process and sentiment prediction are carried out, the data that has been processed at the feature extraction stage will be defined into X and Y variables. Then the K-Folds Cross Validation process is carried out to test the data that has been previously processed, where the k value is used. in the K-Folds Cross Validation process is 5 so that the test will be carried out in 5 repetitions and the tested data is divided into 5 parts.

f) Result Analysis

The last process is the analysis of the results, namely the process of evaluating the model and predicting the data that has been classified using the model used. At this stage, a comparison of the performance of the Multinomial Naïve Bayes method, Decision Tree, Multinomial Naïve Bayes with AdaBoost and Decision Tree with AdaBoost is carried out.

The results of the evaluation carried out are accuracy, precision, recall, F1-Score and confusion matrix values. So from these results it can be analyzed that the results of the four methods have different accuracy results or not and whether the application of AdaBoost can increase the accuracy value of the method used.

3. Results and Discussion

3.1. Data Collecting

At the Data Collecting stage, the data collection method is implemented using the Python programming language, with the help of the Google Play Scraper library. The process of using the library requires several parameters, including the package name of the application to be used, language, country, sort method used, the amount of data to be retrieved and filters for ratings on reviews. The data collection experiment was carried out in one trial. With a lot of data taken is 10000 data and the data that has been obtained is saved in the form of a CSV file. The data collection process can be seen in Figure 2.

```
In [11]: from google_play_scraper import Sort, reviews
         result, continuation_token = reviews(
             'com.telkom.tracencare',
             lang='id',
             country='id',
             sort=Sort.MOST_RELEVANT,
             count=10000,
             filter_score_with=None
         )
         df_crawl = pd.DataFrame(np.array(result),columns=['review'])
         df_crawl = df_crawl.join(pd.DataFrame(df_crawl.pop('review').tolist()))
         df_crawl.head(10)
```

Figure 2. Data Collecting Process

3.2. Data Processing

3.2.1. Text Preprocessing

The stages of text preprocessing have 7 stages carried out, these stages are described as follows:

a) Data Cleaning

The data cleaning stage is carried out for cleaning review sentences by removing punctuation from sentences that are not useful in the classification process, including also eliminating numbers, symbols and strange characters in the review text.

Table 1. Data cleaning

Reviews Data	Data Cleaning
Update mulu dipake ngelag.. Gaenak ama antrian di blakang pas di bioskop.. Alhasil di suruh mundur dulu deh.. ðŸ’©	Update mulu dipake ngelag Gaenak ama antrian di blakang pas di bioskop Alhasil di suruh mundur dulu deh

b) Case Folding

The case folding process is used to convert review data into lowercase letters using the lowercase function.

Table 2. Case Folding

Data Cleaning	Case Folding
Kenapa setelah update data hilang semua Kolom pengisian NIK juga kurang Jadi tidak bisa terdeteksi Tolonglah diperbaiki skrg apa pake peduli lindungi Tapi kenapa malah bikin susahhhhhhh	kenapa setelah update data hilang semua kolom pengisian nik juga kurang jadi tidak bisa terdeteksi tolonglah diperbaiki skrg apa pake peduli lindungi tapi kenapa malah bikin susahhhhhhh

c) Tokenization

Tokenization is a step carried out in research to separate a review sentence into per word. The word generated in this process is independent and the word is useful as an entity for the preparation of the matrix in the next process, namely in calculating the presence of words in a document. This stage uses the nltk library with the function used is word_tokenize.

Table 3. Tokenization

Case Folding	Tokenization
yang salah input nik puskesmas yang dibuat pusing kita nya di wa dan email ga ada balasan	['yang', 'salah', 'input', 'nik', 'puskesmas', 'yang', 'dibuat', 'pusing', 'kita', 'nya', 'di', 'wa', 'dan', 'email', 'ga', 'ada', 'balasan']

d) Spell Checking

The spell checking stage serves to correct typos or abbreviations into words that are in accordance with the KBBI. In its application, this process uses a dictionary for the checking process.

Table 4. Spell checking

Tokenization	Spell Checking
['ini', 'gmn', 'sih', 'abis', 'diperbarui', 'ko', 'malah', 'jadi', 'eror', 'gabisa', 'buka', 'sertifikat', 'gabisa', 'buka', 'status', 'vaksinasi']	['ini', 'bagaimana', 'sih', 'habis', 'diperbarui', 'ko', 'bahkan', 'jadi', 'eror', 'tidak', 'bisa', 'buka', 'sertifikat', 'tidak', 'bisa', 'buka', 'status', 'vaksinasi']

e) Stop Removal

The stop removal stage is used to remove words that are too common and less important which the frequency of occurrence of words is too frequent in the review data. At this stage using the nltk library with the function used is stopwords with language parameters used, namely Indonesian and English.

Table 5. Stop removal

Spell Checking	Stop Removal
['sangat', 'bagus', 'tapi', 'kenapa', 'saya', 'mau', 'cek', 'sertifikat', 'vaksin', 'tidak', 'bisa', 'min']	['bagus', 'cek', 'sertifikat', 'vaksin', 'min']

f) Stemming

The stemming stage in this study serves to make changes to the words contained in the review data into their basic form. The python library used at this stage is Sastrawi.Stemmer.StemmerFactory, the library is used to carry out the process of stemming Indonesian words.

Table 6. Stemming

Stop Removal	Stemming
['aplikasi', 'pemerintah', 'bernama', 'pedulilindungi', 'bagus', 'manfaatnya', 'sulit', 'kebanyakan', 'teman', 'mengeluhkan', 'sertifikat', 'vaksin', 'aplikasi', 'terimakasih', 'bantuannya']	[['aplikasi'], ['perintah'], ['nama'], ['pedulilindungi'], ['bagus'], ['manfaat'], ['sulit'], ['banyak'], ['teman'], ['keluh'], ['sertifikat'], ['vaksin'], ['aplikasi'], ['terimakasih'], ['bantu']]

g) Translate

The data labeling process in this study uses the vaderSentiment library which does not yet support Indonesian, so the data must first be translated into English with the aim of increasing the accuracy of the sentiment data labeling process.

Table 7. Translate

Stemming	Translate
[['bagus'], ['banget'], ['sertifikat'], ['lengkap'], ['perlu'], ['jalan'], ['negeri']]	[['good'], ['really'], ['certificate'], ['complete'], ['need'], ['road'], ['country']]

3.2.2. Data Labeling And Polarity

a) Data Labeling

At the data labeling stage in this study, the vaderSentiment library is used to determine the sentiment score of each PeduliLindungi application user review data. For sentiment labels used are Positive and Negative with a score value used as a labeling reference, if the review has a score > 0 then the review is labeled positive, if the review has a score < 0 then the review is labeled negative and if there is data that does not fit into the provisions exists, the data will be deleted.

Table 8. Data Labeling

Reviews Data	Score	Label
kok masih load tidak berhenti terus tadi mau batal malah masih load di tiap hari	-0,296	NEGATIVE
aplikasi yang mudah digunakan dan lengkap sangat bermanfaat	0,6705	POSITIVE

From the data labeling process, it produces a different amount of data from the initial amount of data. The data which initially amounted to 10000 data, after being processed changed to 8305 data.

b) Polarity

The polarity stage serves to provide a polarity value for each label in the review data. The provision for assigning a polarity value in this study is that if the review data is labeled positive it will be assigned a polarity value of 1 and if the review data is labeled negative it will be assigned a polarity with a value of -1.

Table 9. Polarity

Reviews Data	Label	Polarity
kok masih load tidak berhenti terus tadi mau batal malah masih load di tiap hari	NEGATIVE	-1
aplikasi yang mudah digunakan dan lengkap sangat bermanfaat sebagai data saver	POSITIVE	1

In experiments conducted with 8305 data, it was found that the tendency of public sentiment towards the PeduliLindungi application was 67% positive and 33% negative.

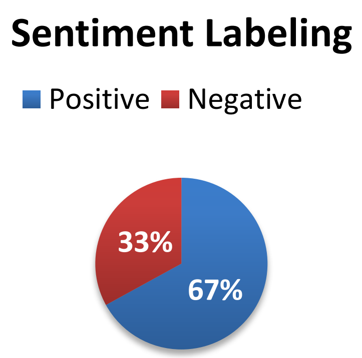


Figure 3. Sentiment Labeling

3.3. Feature Extraction

In the feature extraction stage, two processes are carried out, namely CountVectorizer and TF-IDF. CountVectorizer is used to convert text into vector representation and also to count the number of words in a document. Meanwhile, TF-IDF is used to give weight to the number of words with a measure of how often they appear in a document.

The CountVectorizer process in this study uses the sklearn library with the method used is CountVectorizer.

```
In [65]: #countvectorizer
bow_transformer = CountVectorizer(ngram_range=(1,3),strip_accents='unicode', max_features=1000)
print((dataset.content.shape))
rdf = bow_transformer.fit_transform(dataset.content)
print(rdf.toarray())
print('Shape of Sparse Matrix: ', rdf.shape)
print('Amount of Non Zero occurances: ', rdf.nnz)
filename = 'count_vectorizer.pkl'
pickle.dump(bow_transformer, open(filename, 'wb'))
print(rdf)
```

Figure 4. CountVectorizer

There is a weakness for CountVectorizer, which is that it only gives a score based on its occurrence in certain documents without taking into account the high frequency of occurrence of words in other documents. To overcome the weakness of the CountVectorizer process in this study, the TF-IDF process is applied to give weight to the number of words with a measure of how often the word appears in a document. The implementation of TF-IDF uses the sklearn library with the function used is TfidfTransformer.

3.4. Text Classification

The next step in this research is text classification, the process at this stage is training data and sentiment prediction. The training data process is carried out using the sklearn library with the KFold method. The parameter used for n_splits on KFold is 5, where n_splits is the value of k so that the test is repeated 5 times and also all existing data is divided into 5 parts in the tests carried out.

In the text classification process, an evaluation process is also carried out using a confusion matrix. The implementation of the confusion matrix in python uses the sklearn.metrics library with the function used is confusion_matrix. Then in this process, measurement of accuracy, precision, recall and f1-score values is also carried out in each classification process carried out. The classification modeling stages are Multinomial Naive Bayes, Classification using Decision Tree, Classification using Multinomial Naive Bayes with AdaBoost and Classification using Decision Tree with AdaBoost.

The results of the classification process using Multinomial Naïve Bayes for the experiments carried out can be seen in Table 10.

Table 10. Result classification process using Multinomial Naive

N-Fold	Accuracy	Precision	Recall	F1-Score
1	0,833232	0,857469	0,794167	0,809525
2	0,854906	0,87214	0,808388	0,827703
3	0,839855	0,852123	0,772364	0,795582
4	0,847682	0,859744	0,772209	0,798278
5	0,810957	0,822398	0,698804	0,72436
Average	**0,8373264**	**0,8527748**	**0,7691864**	**0,7910896**

The results of the classification process using the Decision Tree for the experiments carried out can be seen in Table 11.

Table 11. Result classification process using Decision Tree

N-Fold	Accuracy	Precision	Recall	F1-Score
1	0,860325	0,853345	0,849832	0,851496
2	0,847682	0,83319	0,831318	0,832235
3	0,835641	0,810473	0,814986	0,812637

N-Fold	Accuracy	Precision	Recall	F1-Score
4	0,82059	0,78772	0,792706	0,790106
5	0,780252	0,733129	0,720716	0,726201
Average	**0,828898**	**0,8035714**	**0,8019116**	**0,802535**

The results of the classification process using Multinomial Naïve Bayes with AdaBoost for the experiments carried out can be seen in Table 12 below.

Table 12. Result classification process using Multinomial Naïve Bayes with AdaBoost

N-Fold	Accuracy	Precision	Recall	F1-Score
1	0,894641	0,895773	0,879726	0,886375
2	0,901264	0,894086	0,887597	0,890672
3	0,888019	0,876755	0,862461	0,868992
4	0,888621	0,876583	0,855505	0,8649
5	0,867549	0,85547	0,812259	0,829457
Average	**0,8880188**	**0,8797334**	**0,8595096**	**0,8680792**

The results of the classification process using Decision Tree with AdaBoost for the experiments carried out can be seen in Table 13 below.

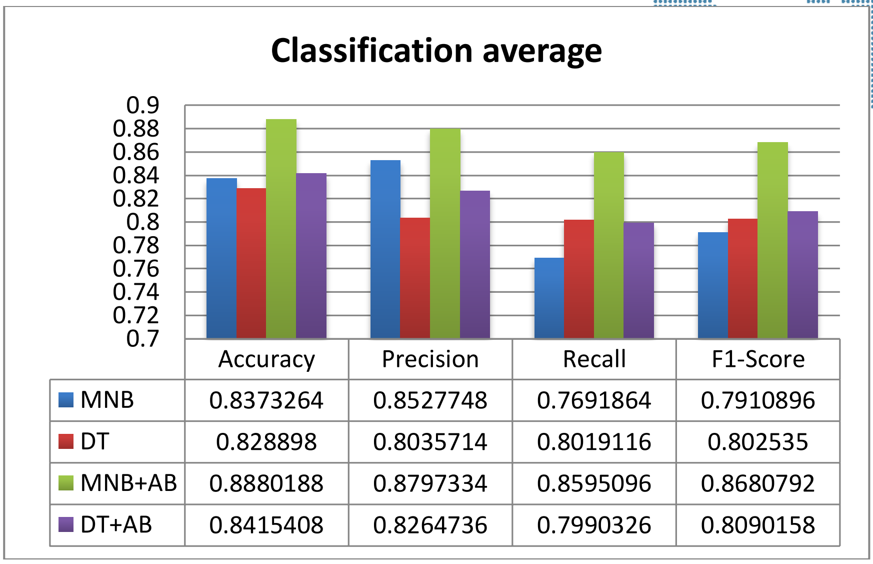
Table 13. Result classification process using Decision Tree with AdaBoost

N-Fold	Accuracy	Precision	Recall	F1-Score
1	0,861529	0,856078	0,848719	0,852022
2	0,853702	0,840693	0,835952	0,838209
3	0,842865	0,833176	0,794954	0,809399
4	0,851294	0,834005	0,80479	0,816887
5	0,798314	0,768416	0,710748	0,728562
Average	**0,8415408**	**0,8264736**	**0,7990326**	**0,8090158**

3.5. Result Analysis

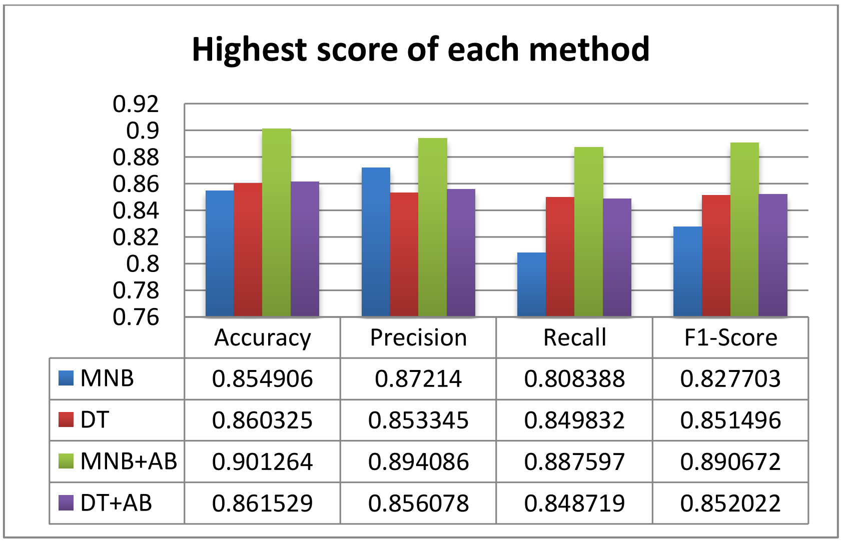
From the evaluation process that has been carried out on the four classification processes, the highest average accuracy value obtained by the Multinomial Naïve Bayes method with AdaBoost is 88,8%. Meanwhile, the average value of the Decision Tree method with AdaBoost is 84,15%. Average value is obtained from repeated testing using KFold, with the value of n_splits is 5. The graph for the average value of the classification is shown in Figure 5.

The highest accuracy, precision, recall and f1-score values for each method. The highest accuracy value is obtained by the Multinomial Naïve Bayes method with AdaBoost method, which is 90,1%. For the highest value, the accuracy of the Decision Tree method with AdaBoost is 86,15%. The graph of the highest score for each classification method is shown in Figure 7.



	Accuracy	Precision	Recall	F1-Score
MNB	0.8373264	0.8527748	0.7691864	0.7910896
DT	0.828898	0.8035714	0.8019116	0.802535
MNB+AB	0.8880188	0.8797334	0.8595096	0.8680792
DT+AB	0.8415408	0.8264736	0.7990326	0.8090158

Figure 5. Classification average



	Accuracy	Precision	Recall	F1-Score
MNB	0.854906	0.87214	0.808388	0.827703
DT	0.860325	0.853345	0.849832	0.851496
MNB+AB	0.901264	0.894086	0.887597	0.890672
DT+AB	0.861529	0.856078	0.848719	0.852022

Figure 6. Highest score of each method

From the results of the tests carried out, that the MNB+AB method has an average value and the highest score is better than the other methods used in this study. The MNB+AB method has an average accuracy value of 88,8%, then the highest value of MNB+AB in each method is 90,1%. The DT+AB method has an average accuracy value of 84,1%, then the highest value of DT+AB in each method is 86,15%.

In the experiment conducted by the MNB method, which initially had an average accuracy value of 83,7%, after being added with AdaBoost it increased by 5% to 88,8%. The highest value of the MNB method in the experiments carried out was 85,4%, after being combined with AdaBoost the MNB method increased accuracy by 4,6% to 90,1%.

The DT method produces an average accuracy value of 82,8%, after adding AdaBoost it increases to 84,1% with an increase of 1,2%. The highest value for the DT method is 86,03%. After being combined with AdaBoost, the DT method in the experiments carried out has increased accuracy by 0,12% to 86,15%.

Based on the analysis that has been carried out, it can be concluded that the application of AdaBoost can increase the accuracy for the Multinomial Naïve Bayes and Decision Tree methods even though the increase in the Decision Tree method is not too significant.

In this study, the Multinomial Naïve Bayes method with AdaBoost has better accuracy than Decision Tree with AdaBoost.

4. Conclusion

The trend of public sentiment towards the PeduliLindungi application taken from user review data on the Google Play Store in the experiment carried out was positive opinion with a percentage of 67%, while negative opinion had a percentage of 33% of the total 8305 data.

The results of the performance of the Multinomial Naïve Bayes algorithm before the implementation of AdaBoost has an average accuracy value of 83,7% and the highest accuracy value is 85,4%. For the performance results of the Decision Tree algorithm before the implementation of AdaBoost, the average accuracy value is 82,8% and the highest accuracy value is 86,03%. The performance results of the Multinomial Naïve Bayes algorithm after the implementation of AdaBoost have an average accuracy value of 88,8% and the highest accuracy value is 90,1%. For the results of the Decision Tree algorithm performance after the implementation of AdaBoost has an average accuracy value of 84,1% and the highest accuracy value is 86,15%. From the analysis that has been carried out on the results of the classification test, the application of AdaBoost can improve the accuracy of the Multinomial Naive Bayes algorithm and Decision Tree even though the increase in the Decision Tree method is not very significant. In this study, the Multinomial Naive Bayes method with AdaBoost has better accuracy than Decision Tree with AdaBoost.

Based on the results of the analysis carried out, suggestions that can be given for future research include the sentiment labeling process can use other automatic labeling, can be added selection features such as information gain so that it can produce better accuracy than the method used in this study and can be developed by using another boosting method, so that it can produce a method that is better than the combination of Multinomial Naive Bayes with AdaBoost and Decision Tree with AdaBoost.

Acknowledgments

I would like to express my gratitude to all those who have helped in this research, so that it can be published in international journals.

References

[1] D. Sharma, M. Sabharwal, V. Goyal, and M. Vij, "Sentiment analysis techniques for social media data: A review", Springer Singapore., vol. 1045, **(2020)**.

[2] W. Medhat, A. Hassan, and H. Korashy, "entiment analysis algorithms and applications: A survey," Ain Shams Eng. J., vol. 5, no. 4, **(2014)** , pp. 1093–1113.

[3] B. Liu, "Sentiment analysis: Mining opinions, sentiments, and emotions", **(2015)**.

[4] M. T. H. Bunga, B. S, Djahi, and Y. Y. Nabuasa, "Multinomial Naive Bayes Untuk Klasifikasi Status Kredit Mitra Binaan Di PT . Angkasa Pura I Program Kemitraan", J-Icon., vol. 6, no. 2, pp. 30–34, **(2018)**, pp. 30–34.

[5] S. Rizkia, E. Budi Setiawan, and D. Puspandari, "Analisis Sentimen Kepuasan Pelanggan Terhadap Internet Provider Indihome di Twitter Menggunakan Metode Decision Tree dan Pembobotan TF-IDF", e-Proceeding Eng., vol. 6, no. 2, **(2019)**, pp. 9683–9693.

[6] E. K. Adhitya, R. Satria, and H. Subagyo, "Komparasi Metode Machine Learning dan Metode Non Machine Learning untuk Estimasi Usaha Perangkat Lunak", IlmuKomputer.com J. Softw. Eng., vol. 1, no. 2, **(2015)**, pp. 109–113.

[7] A. Salma and W. Silfianti, "Sentiment Analysis of User Reviews on COVID-19 Information Applications Using Naive Bayes Classifier, Support Vector Machine, and K-Nearest Neighbor", Int. Res. J. Adv. Eng. Sci., **(2021)**.

[8] M. B. Hamzah, "Classification of Movie Review Sentiment Analysis Using Chi-

Square and Multinomial Naïve Bayes with Adaptive Boosting", J. Adv. Inf. Syst. Technol., vol. 3, no. 1, **(2021)**, pp. 67–74.

[9] L. Dini, U. Sekolah, T. M. Informatika, D. Komputer, N. Mandiri, and R. S. Wahono, "Integrasi Metode Information Gain Untuk Seleksi Fitur dan Adaboost Untuk Mengurangi Bias Pada Analisis Sentimen Review Restoran Menggunakan Algoritma Naïve Bayes", J. Intell. Syst., vol. 1, no. 2, (**2015**) , pp. 120–126.

[10] N. S. Wardani, A. Prahutama, and P. Kartikasari, "Analisis Sentimen Pemindahan Ibu Kota Negara Dengan Klasifikasi Naïve Bayes Untuk Model Bernoulli Dan Multinomial", J. Gaussian, vol. 9, no. 3, **(2020)**, pp. 237–246.

[11] M. Syarifuddin, "Analisis Sentimen Opini Publik Terhadap Efek PSBB Pada Twitter Dengan Algoritma Decision Tree-KNN-Naive Bayes", Inti Nusa Mandiri., vol. 15, no. 1, **(2020)**, pp. 87–94.

[12] A. R. Zaidah, "Analisis sentiment berbasis aspek pada aplikasi pedulilindungi menggunakan latent dirichlet allocation dan support vector machine", UIN Sunan Ampel Surabaya., **(2021)**.

[13] A. Mustopa, Hermanto, Anna, E. B. Pratama, A. Hendini, and D. Risdiansyah, "Analysis of user reviews for the pedulilindungi application on google play using the support vector machine and naive bayes algorithm based on particle swarm optimization", 5th Int. Conf. Informatics Comput. ICIC, **(2020)**,.

[14] F. Illia, M. P. Eugenia, and S. A. Rutba, "Sentiment Analysis on PeduliLindungi Application Using TextBlob and VADER Library", Proc. Int. Conf. Data Sci. Off. Stat., vol. 2021, no. 1, **(2021)** , pp. 278–288.