SINERGI Vol. 27, No. 2, June 2023: 145-152
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2023.2.001

Sentiment Analysis from Twitter about Covid-19 Vaccination in Indonesia using Naïve Bayes and XGboost Classifier Algorithm

Alvin Irwanto, Leonard Goeirmanto
Department of Informatics, Faculty of Computer Science, Universitas Mercu Buana, Indonesia

Abstract
The pandemic that hit the world has greatly impacted our life. But after some time, it seems that it will be going to end because the vaccine has already been made. In response to this, some people expressed their opinions about this vaccination on social media, for example, in the form of tweets on Twitter. The authors use those opinions or tweets as sentiment analysis material to determine the assessment of this vaccination. The tweet data in this study was obtained through data crawling using the Twitter API with the Python programming language. The variables used in this case are public tweets and their sentiments. This sentiment analysis process uses the Classification method with the Naive Bayes Classifier and will be compared with the XGBoost Classifier algorithm. The results of this study indicate that people are more likely to respond positively to this vaccination. In this case, the Naive Bayes Classifier got better performance with 0.95 from ROC - AUC Score and 134 ms in runtime compared to the XGBoost Classifier algorithm with 0.882 in ROC - AUC Score and 1 minute and 59 seconds in runtime.

Keywords:
Covid-19 vaccination;
Naïve Bayes Classifier;
Sentiment Analysis
Twitter;
XGboost Classifier;

Article History:
Received: February 22, 2022
Revised: July 8, 2022
Accepted: December 12, 2022
Published: June 2, 2023

Corresponding Author:
Leonard Goeirmanto
Department of Informatics,
Faculty of Computer Science,
Universitas Mercu Buana,
Indonesia
Email:
leonard@mercubuana.ac.id

INTRODUCTION

The Covid-19 pandemic that happens has destroyed almost all sectors of human life. For example, in work and college, which previously had to come to a place, they are now required to stay at home, which is done online. Various efforts have been made to prevent the spread, ranging from wearing masks, and physical distancing, to frequent hand washing. However, vaccination is the most effective solution at this time [1]. The primary objective of vaccination is to achieve herd immunity, which implies that people are pushed to get vaccinations for at least 70% of the population to be immune. Although it cannot prevent 100% of the spread of Covid-19, it can at least reduce the effects of its spread. In Indonesia, various types of vaccines are used, ranging from Sinovac, AstraZeneca, Pfizer, and Moderna, to the most recent, Zifivax.

However, this vaccination program has not been able to run optimally because some people still have doubts about this vaccine. There are various opinions on social media regarding this Covid-19 vaccination, for example, on Twitter. There are still some people on Twitter, in particular, who are hesitant about this vaccination. Moreover, coupled with cases of death after vaccination, some people are increasingly afraid and doubt the use of vaccines [2].

This study aims to analyze the public's response to Covid-19 vaccination by classifying it into positive and negative responses. It is hoped that the results of this sentiment can be used as information and evaluation material for related parties in seeing public opinion about the Covid-19 vaccination. A more positive response can mean that the socialization has been conveyed well and can be understood by the people in Indonesia. Otherwise, a more negative response could mean the opposite, which means there is still miscommunication in the community regarding this vaccination.

Furthermore, this study also compares two algorithms to figure out who has the best performance to perform the data. The authors use two algorithms as a model for extracting and mining the value from the data. Those algorithms are Naïve Bayes Classifier and XGBoost Classifier. The authors choose those algorithms because both algorithms are already known to have high performance in applying classification cases such as in research and machine learning competitions. The other reasons will be explained in the model implementation section. In the end, the authors will discuss which algorithms have the best performance for handling this case.

MATERIAL AND METHODS

Related works

Sentiment analysis has attracted the attention of many data mining researchers. Sentiment analysis is primarily used to express an individual's opinion [3]. In conclusion, the current cutting-edge divided classes into positive and negative categories. Much research has used various methodologies to accomplish sentiment analysis [4]. According to Abdullah and Hadzikadic, sentiment analysis for text uses two primary methodologies: symbolic or lexicon-based and machine learning approaches [5][6]. In this study, the authors focus on machine learning approaches as a methodology.

Many people express their thoughts and opinions freely on social media platforms such as Twitter. Therefore, many researchers and scientists study the topic of sentiment analysis that came from Twitter data. This research can help and figure out many things, depending on what the research is for. For example, it helps e-commerce businesses focus on improving service and company quality, leading to improved traffic, sales, and profitability.

It is also used in sentiment analysis from tweet data during the Republic of Indonesia's presidential candidates in the 2019 - 2024 period [7]. In that research, they used the Naive Bayes method and also compared it with other methods such as Support Vector Machine and K-Nearest Neighbors algorithm. Their experiments showed that the Naïve Bayes method has 80.90% accuracy compared to KNN (75.58%) and SVM (63.99%).

The successful application of SMOTE for handling imbalanced data is already used in observing imbalanced data text to predict users selling products to females daily [4]. For this study, they intend to use SMOTE, Tomek, and SMOTE-Tomek to analyze the imbalanced data text in females daily. This research used Support Vector Machine (SVM) and Logistic Regression (LR). The training model uses 10-Fold Cross validation with performance evaluation, which uses Precision, Recall, and G-mean. The results of this study revealed that the effects of SMOTE, Tomek, and SMOTE-Tomek on Precision-Recall in individuals selling products (majority class) are not significantly different. It also reduced Precision-Recall. But there is a significant improvement in people selling products (minority class).

Naïve Bayes is an algorithm in machine learning based on the Bayes theorem, usually used in classification, especially for text classification [8]. In addition, the Naïve Bayes model is widely used for huge data sets [9]. The Naïve Bayes method is based on conditional probability and the maximum likelihood of occurrence [10].

XGBoost, eXtreme Gradient Boosting, is an implementation of gradient boosted decision trees designed for execution speed and model performance [11]. It's a powerful algorithm capable of dealing with a wide range of data anomalies. Building a model using XGBoost is simple, but enhancing the model with XGBoost is more challenging. Multiple parameters are used in this method. Parameter adjustment is required to enhance the model. For this research, the authors used n_estimators for the parameter tuning using the value of 500. This is the best value that authors get after several tests.

Data Collection

The data collection method used in this research is by collecting tweet data or so-called Crawling data from Twitter. The data is taken from May until October 2021. The data was taken at that time because the mass vaccination was already running in Indonesia and May. There was a case where it was reported that someone had died after being vaccinated, which certainly decreased the public's trust in this vaccination and doubts about it. The crawling data ended in October because the authors think that at that time. The Covid-19 vaccination is almost evenly distributed and has become a must for all citizens, so opinion tweets about this vaccination are starting to decrease.

The dataset in this research contains two columns, Comment and Value, with 10208 rows. To get the data, the authors use the Python library, namely Tweepy, which can collect user tweets based on certain keywords. To be able to collect this tweet data, a certain token and the key is required, including access token, access token secret, consumer key, and consumer key secret, which is obtained via the Twitter API. In this research, the authors use hashtags as keywords to search for tweets about this vaccination. The hashtags that the authors use are ”#vaksinasi”, “#vaksin”, “#vaksinasicovid19”, “#vaksincovid19”, “#sinovac”, ”#astrazeneca”, “#pfizer”, and “#zifivax”. The authors here only crawl the tweet or comment without determining or checking who wrote it. The sample data is shown in Table 1.

Labeling Data

Labeling the data is important for the machine learning process. The data labeling process is done manually, and the sentence will be determined to have a positive or negative meaning. This study only used two sentiments, namely positive and negative, denoted by the numbers 0 (negative) and 1 (positive).

The positive label seen from the contents of the tweet contains positive, supportive sentences and statements of agreement. The negative label is a class with data containing negative meanings, ridicule, and contradictory sentences. Consistency is important here, so the machine is not confused and makes a bad accuracy. For example, if it already considers profanity negative, don't consider the other half of the dataset positive if it contains profanity.

The authors label the easiest examples first. The obvious positive/negative examples should be labeled as soon as possible, and the hardest ones should be left to the end when you already comprehend the problem better. For example, if the tweet is sarcastic or ironic, the authors delve deeper into this case and look back at the tweet to avoid misunderstanding the meaning of the tweet

Table 1. Sample Dataset

No.	Comment
1	Serem nih efek #AstraZeneca 😢 https://t.co/8xyNHRNB0H
2	Efek pasca vaksin #AstraZeneca ini malah lebih berat dibanding pas positif Covid dulu, hehe 😅
3	Jadi seperti itulah cerita mengenai #vaksin #COVID19. Jangan mudah terpengaruh oleh isu - isu negatif, yang bisa ja… https://t.co/9BetZGU6o3
4	#AstraZeneca bener banget ...... anw, manfaat nya masih lebih tinggi dari mudarat nya kan https://t.co/y5zamaRj0F
5	Vaksin Sinovac Kemampuannya Melampaui Uji Klinis https://t.co/E8mf9mvzSY via @holopiscom #Vaksin #Covid19… https://t.co/hAZDTqGU3Z

.

Preprocessing Data

There are seven steps in preprocessing data: Case folding, Remove redundant and unrelated tweet, Tokenizing, Filtering, Stemming, TF-IDF and SMOTE. Each step will be explained in the following.

Case folding itself is divided into several steps: removing the link, hashtag, and username that usually come after the tweet, changing the sentences into lowercase, and removing punctuation, number, and special characters that will not be used in later stages. Characters other than letters are removed and are considered delimiters.

Removing redundant and unrelated tweets is done after the case folding because many tweets have the same sentence but have a different link at the end. If it runs this process first, it will end with many duplicate data. The data also contains many tweets that only contains unrelated tweet, such as tweet that does not use Bahasa Indonesia, talk about another topic, and only contains one character that does not even have any meaning. So, in this process, the authors also removed it from the dataset.

To remove those redundant and unrelated Tweets, the authors use two methods: manually and using Python. For the redundant data, it used Python to search the tweets that contain the same sentences. After finding those tweets, the next process is to delete them and just use one of the tweets. The second method is done manually by searching the tweet unrelated to the topic or without meaning. After those processes, the dataset is reduced from 10208 rows to 4176.

Tokenizing is the process of breaking sentences into separate words called tokens [12]. This process separates words in the dataset based on readable spaces [13][14]. The sentence from the tweet will be broken down into word for words in an array. These tokens help better identify the context or the model's development. By evaluating the sequence of words, tokenization helps interpret the text's meaning. This is how we teach machines about words.

Stopword itself is a common word that usually appears in large numbers and is considered meaningless. Examples of stopwords in Indonesian are “yang”, “di”, “dari”, etc. It must be removed because it will take up space in the database or taking up valuable processing time. So, the authors remove them for this by storing a list of words considered stop words. Table 2 shows the data after this process.

Table 2. Data After Filtering

No.	Comment	Value
1	['serem', 'nih', 'efek']	0
2	['efek', 'pasca', 'vaksin', 'berat', 'dibanding', 'positif', 'covid']	0
3	'cerita', 'mengenai', 'jangan', 'mudah', 'terpengaruh', 'oleh', 'isu', 'isu', 'negatif']	1

Stemming itself is the process of returning data to its root word. This stage will eliminate the suffix and prefix in the token/word (reduce inflected) so that a word with a suffix or prefix will return to its basic form. The example in Bahasa Indonesia is:

Penerima → Terima
Ditetapkan → Tetap
Melawan → Lawan

TF-IDF is a statistical measure of a word's relevance to a document in a collection of documents. This is accomplished by multiplying two metrics: the number of times a word appears in a document and the word's inverse document frequency over a collection of documents. TF itself stands for Term's Frequency and IDF is Inverse Document Frequency. This process converts the dataset into a matrix of TF-IDF features. This is one of the most important processes because Machine Learning algorithms cannot work with raw data directly.

The dataset used in this research is imbalanced, with 3810 data having positive and 366 negative labels. This could be a problem because most machine learning algorithms are designated to work best with balanced data that the target classes have similar prior probabilities [15].

If the data is imbalanced, the machine will make predictions more inclined towards the majority class. To handle that problem, the authors used SMOTE for this research. SMOTE is used to oversample the minority class by producing "synthetic" examples rather than over-sampling with replacements [16]. SMOTE works by choosing examples in the feature space that are close together, drawing a line in the feature space between the examples, and generating a new sample at a position along that line. To be more specific, a random case from the minority class is selected initially. Then, initialize the k of the closest neighbours found (usually k=5). After that, A randomly chosen neighbor is picked, and a synthetic example is generated at a randomly chosen point in the feature space between the two examples. After applying that technique, the data class are equal.

SMOTE itself is well known for handling imbalanced data cases [17]. It can be seen from several studies that have used this method, for example, in research conducted by Jonathan, Putra and Ruldeviyani [18] and also research by Lu, Cheung and Tang [19], where the application of SMOTE is very optimal for the results. It should also be noted that in applying this method, the given dataset should be clean and clear from noise. In this research, the dataset is already through several preprocessing and data cleaning processes, so the resulting data is ready to be applied with the SMOTE method. Figure 1 shows how the label comparison after applying this method.

Model Implementation

The implementation of the model is done by using Python as the programming language and Jupyter Notebook as the software to run the program. Those algorithms were chosen because both algorithms are already known to have high performance in classification.

Moreover, the Naïve Bayes Classifier is already being used in many kinds of research and has been proven to run text classification cases well. Since a Naive Bayes text classifier is based on the Bayes' Theorem, encoding such probabilities is very helpful, which enables us to compute the conditional probability of occurrence of two events depending on the probabilities of occurrence of each individual event. For instance, the probability of the phrases "user" and "interface" appearing in texts under the category of "design website" is higher than the probability of those words appearing under the categories of "economy and business."

On the other side, the XGBoost algorithm has recently dominated applied machine learning and Kaggle competitions for structured or tabular data. The beauty of the XGBoost algorithm is its scalability, which allows for rapid learning via parallel and distributed computing while still utilizing memory efficiently. The multiple basic decision tree models used in this model are then combined and enhanced to provide a single combined outcome. Even more specific conditions are needed for the dataset before using an XGBoost model. For example, the labels must be 1/0 in classification problems, which is appropriate in this instance.

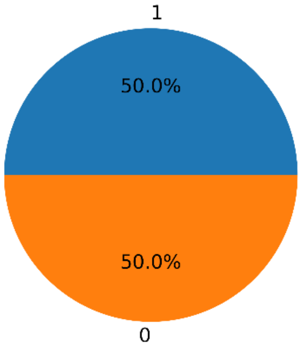

Figure 1. Comparison between the labels after applying SMOTE

In the end, the authors want to know which is the best for handling a case like this. Hopefully, this research can be used as a reference for future research.

Evaluation and Validation

Evaluating a model is crucial in creating a successful machine learning model. The counts of test records correctly and incorrectly predicted by a classification model are used to evaluate the model's performance. F1-Score is come from the weighted harmonic mean of precision and recall [20]. Because the minority class only counts for a small percentage of the data, excluding all of the minority class samples has minimal impact on overall accuracy. It is also important to note that the F1 score is a well-balanced combination of precision and recall. As a result, it is a more acceptable statistic for evaluating minority class categorization [21]. In addition, the authors also used the confusion matrix and ROC AUC score and plot to make sure how the model works.

The model validation for this research used two kinds of validation. First is the K-fold Cross Validation technique. This method is mostly used to create model predictions and measure a predictive model's performance when applied in practice. Furthermore, K-fold cross validation is utilized to reduce bias in the data [22]. The technique includes only one parameter called k, which specifies the number of groups into which a given data sample should be divided [23]. When a precise value for k is specified, it can be substituted for k in the model reference. For this research, it chose the value of k=10, meaning the data will be divided into ten groups. The first group will become the validation data, and the rest will be training data. After the process, the second group will become the validation data and the rest will be the train data, which happens over and over again until all fold get part as validation data.

The second technique that is used for validation is splitting with a ratio of 80% data train and 20% data test [24]. This is actually commonly used for the validation model. This technique is used because the authors want to test the model with data that is not in the training or validation data. This is like a simulation of the application of this model to real-world data. With this technique, the author can also provide more evaluation metrics such as Confusion Matrix and ROC-AUC Curve to better understand the model.

RESULTS AND DISCUSSION

The main purpose of this research is to know the public opinion in Indonesia about the Covid-19 vaccination and create a machine learning model that can classify the tweets into two classes, which are positive and negative. In addition, this research also figures out which algorithm has the best performance between the Naïve Bayes and XGBoost Classifier to handle this kind of case.

In our data, most of the tweets are positive sentiments. It probably means that people in Indonesia are already well-socialized about this vaccination. The tweets show that many people in Indonesia already known about the important of this vaccination to end this pandemic. The government and the related parties are also can socialize the procurement of this vaccination well by continuing to invite people to vaccinate.

In the machine learning model process, the next step is to create the model after applying the whole preprocessing data. The models used in this research are the XGBoost classifier and Naïve Bayes Classifier, where the process is done using Python as s tool. To increase the model's performance, the authors use the hyper parameter for each model. For Naïve Bayes Classifier, the hyperparameter used is alpha, a Laplace smoothing parameter. The default value for alpha is 1, but in this case, the authors found that the optimum value of alpha is 0.6. For XGBoost Classifier, the hyperparameter that the authors used is n_estimators. The function of n_estimators indicate the number of times the modeling cycle should be repeated. The value selection must be careful because if we put too low a value can cause underfitting, and if too high can cause overfitting. The typical values if between 100 to 1000. For this case, the optimum value for n_estimator is 500. After creating the model, the next step is the evaluation process. It is done by using K-fold cross validation with the value of k=10. Table 3 shows how the performance of both algorithms.

From Table 3, it can see that F1-Score for XGBoost classifier is higher than Naïve Bayes Classifier with a difference of only about 2%, where the XGBoost Classifier obtain 0.968. In contrast, the Naïve Bayes Classifier obtains slightly less, which are 0.945. However, looking at the runtime, we can see that the Naïve Bayes Classifier can have a shorter training time than the XGBoost Classifier. The same thing happens in the testing process, the Naïve Bayes Classifier can predict the data faster than the XGBoost Classifier. Because the model's efficiency is evaluated by the running time [25], the Naïve Bayes Classifier runs more efficiently than the XGBoost Classifier.

The second technique is the splitting 80:20 ratio. The preparation is same like the Cross Validation. The difference is the process after the preprocessing data.

In this technique, the authors separate data into 2 data, first, the data used for data modeling and second, for data test. The data test here is used as a simulation when the model meets the never met data. It is come from approximately 20% from the whole dataset. The test data is used as a data that the model will predict.

All test data is not used in the training model process. This is for make sure that the model that train here has not met the data for validation before. The model will predict the test data without the label to get the performance. Ultimately, the predicted label will be compared with the actual label. After running the model, the result is listed in Table 4.

The result shows that the XGBoost Classifier have a bigger F1 score than the Naive Bayes. However, let's look closely to the confusion matrix in Figure 2. It is shown that the Naive Bayes Classifier performs better to predict the negative label, while XGBoost has more problems predicting it. The XGBoost have a better F1-Score because it predicts more accurately in the positive label, while in the negative label, it performs the opposite. These results also align with the ROC - AUC score in Table 4.

Table 3. Model Performance

Algorithm	F1-Score	Runtime
Naïve Bayes Classifier	0.945	134 ms
XGBoost Classifier	0.968	1 min 59 s

Table 4. Model Performance 2

Algorithm	F1-Score	ROC - AUC Score
Naïve Bayes Classifier	0.947	0.950
XGBoost Classifier	0.976	0.882

Figure 3 shows how both of the models work in predicting the data. Classifiers that create curves closer to the top-left corner of the ROC space perform better, whereas those that produce curves closer to the 45-degree diagonal of the ROC space perform less in performance. From the curve, we can see that the Naïve Bayes Classifier model is closer to the top-left corner, which tells us that the Naïve Bayes Classifier model can more distinguish between classes for this case.

Compared to the research before, this study used TF-IDF instead using BoW. Furthermore, this research also uses the SMOTE technique to handle the imbalanced data. The result shows that the F1-Score for the Naïve Bayes Classifier and XGBoost Classifier algorithm in this research have a better performance than the previous one [26] in Table 5. It means that the SMOTE technique is well executed and, in the end, also impact the model that successfully handles the data with good performance.

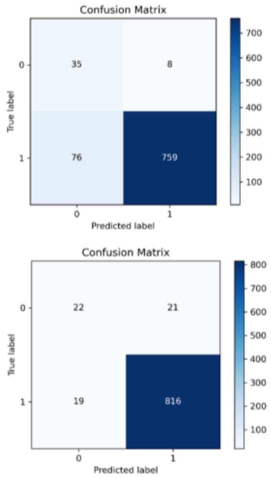

Figure 2. Confusion Matrix of Naïve Bayes Classifier model (On the left), and Confusion Matrix of XGBoost Classifier model (On the right)

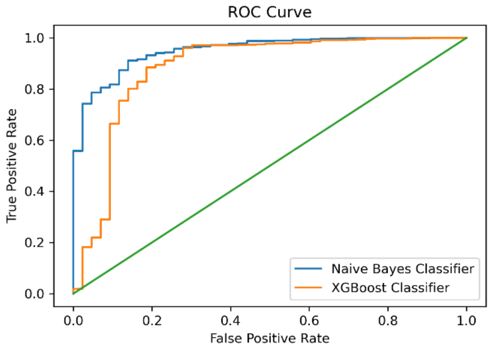

Figure 3. ROC – AUC curve of Naïve Bayes and XGBoost Classifier model

Table 5. Comparison with previous research

Study	Additional technique	Model	F1-Score
A. I. Saad [25]	Bag of Words (BoW)	XGB	0,623
		Naïve Bayes	0,71
This Study	TF-IDF + SMOTE	XGBoost	0.968
		Naïve Bayes	0.945

CONCLUSION

This study collected a dataset of tweets about vaccination in Indonesia from May until October 2021. Twitter Data, or Tweet, cannot be used directly, so we already perform the preprocessing, and because the data is imbalanced, we also use SMOTE method to make it balanced. Our proposed model used two algorithms to figure out who has the better performance of classifying tweets as positive and negative. The model evaluation here uses two methods. Those are 10-fold Cross Validation and Splitting the 80:20 ratio to ensure the model works well. The result shows that more Tweets responded positively to this vaccination program, as seen in Figure 2, where the positive response was 91.2% compared to the negative response, which was only 8.8%. It means that socialization has been conveyed well and can be understood by the people in Indonesia. The people have also understood this vaccination's importance for the common good and themselves.

In addition, this study also reveals that the XGBoost Classifier algorithm has a better F1-Score for predicting the tweets compared to Naïve Bayes Classifier. However, from the confusion matrix and ROC-AUC curve, we can see that the Naïve Bayes Classifier does better in classifying the tweets both for the positive and negative labels. The Naïve Bayes Classifier is also more efficient if we look at the runtime process. So, the authors conclude that in this research, Naïve Bayes Classifier has a better performance than the XGBoost Classifier. As we can see, a higher score does not always indicate a better final result. This research also finds out that the application of the SMOTE method also had a good impact on imbalanced data such as this study. It can be seen from the high F1-Score, which indicates that the model is good at learning between positive and negative labels.

REFERENCES

[1] C. Zhang, Y. Li, J. Cao and X. Wen, "On the Mass COVID-19 Vaccination Scheduling Problem," *Computers & Operations Research*, vol. 141, pp. 1-16, 2022, doi: 10.1016/j.cor.2022.105704

[2] K. Dharma et al., "COVID-19 Vaccine Hesitancy – Reasons and Solutions to Achieve a Successful Global Vaccination Campaign to Tackle the Ongoing Pandemic," *Human Vaccines & Immunotherapeutic*, vol. 17, no. 10, pp. 3495-3499, 2021, doi: 10.1080/21645515.2021.1926183

[3] L. Nemes and A. Kiss, "Social Media Sentiment Analysis based on COVID-19," *Journal of Information and Telecommunication*, vol. 5, no. 1, pp. 1-15, 2021, doi: 0.1080/24751839.2020.1790793

[4] M. Cindo, D.P. Rini and E. Ermatita, " Sentiment Analysis on Twitter by using Maximum Entropy and Support Vector Machine Method," *SINERGI*, vol. 24, no. 2, pp. 87-94, 2020, doi: 10.22441/sinergi.2020.2.002

[5] M. Abdullah and M. Hadzikadic, "Sentiment Analysis of Twitter Data: Emotions Revealed Regarding Donald Trump during the 2015-16 Primary Debates," *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, Boston, MA, USA, 2017, pp. 760-764, doi: 10.1109/ICTAI.2017.00120

[6] S. H. Jayady, H. Antong, "Theme Identification using Machine Learning Techniques," *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 1, no. 2, pp. 123-134, 2021, doi: 10.51662/jiae.v1i2.21

[7] M. Wongkar and A. Angdresey, "Sentiment Analysis Using Naive Bayes Algorithm of The Data Crawler: Twitter," *2019 Fourth International Conference on Informatics and Computing (ICIC)*, Semarang, Indonesia, 2019, pp. 1-5, doi: 10.1109/ICIC47613.2019.8985884.

[8] A. S. T. Nishadi, "Text Analysis: Naïve Bayes Algorithm using Python JupyterLab," *International Journal of Scientific and Research Publications*, vol. 9, no. 11, pp. 126-135, 2019, doi: 10.29322/IJSRP.9.11.2019.p9515

[9] Y. Jiang et al., "Python-Based Visual Classification Algorithm for Economic Text Big Data," *Discrete Dynamics in Nature and Society*, vol. 2022, pp. 1-8, 2022, doi: 10.1155/2022/4616793

[10] M. Mayo and E. Frank, "Improving Naive Bayes for Regression with Optimized Artificial Surrogate Data," *Applied Artificial Intelligence*, vol. 34, no. 6, pp.484-514, 2020, doi: 10.1080/08839514.2020.1726615

[11] O. Sagi and L. Rokach, " Approximating XGBoost with an Interpretable Decision Tree," *Information Science*, vol. 572, pp. 522-542, 2021, doi: 10.1016/j.ins.2021.05. 055

[12] J. H. Clark et al., "Canine: Pre-training an Efficient Tokenization-Free Encoder for Language Representation," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 73-91, 2022, doi: 10.1162/tacl_a_00448

[13] B. Jonathan, P. H. Putra and Y. Ruldeviyani, "Observation Imbalanced Data Text to Predict Users Selling Products on Female Daily with SMOTE, Tomek, and SMOTE-Tomek," *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, Bali, Indonesia, 2020, pp. 81-85, doi: 10.1109/IAICT50021.2020. 9172033

[14] E. Dwianto and M. Sadikin, "Analisis Sentimen Transportasi Online pada Twitter Menggunakan Metode Klasifikasi Naïve Bayes dan Support Vector Machine," *Jurnal Ilmiah Teknik Informatika*, vol. 10, no. 1, p. 94-100, 2021, doi: 10.22441/format.2021. v10.i1.009

[15] A. Indrawati et al., "Analyzing the Impact of Resampling Method for Imbalanced Data Text in Indonesian Scientific Articles Categorization," *Baca: Jurnal Dokumentasi dan Informasi*, vol. 41, no. 2, p. 133, 2020, doi: 10.14203/j.baca.v41i2.702

[16] A. C. Flores, R. I. Icoy, C. F. Peña and K. D. Gorro, "An Evaluation of SVM and Naive Bayes with SMOTE on Sentiment Analysis Data Set," 2018 International Conference on Engineering, *Applied Sciences, and Technology (ICEAST)*, Phuket, Thailand, 2018, pp. 1-4, doi: 10.1109/ICEAST.2018. 8434401

[17] F. Hu and H. Li, "A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRSBoundary-SMOTE", *Mathematical Problems in Engineering*, vol. 2013, ID 694809, pp. 10 pages, 2013, doi: 10.1155/ 2013/694809

[18] B. Jonathan, P. H. Putra and Y. Ruldeviyani, "Observation Imbalanced Data Text to Predict Users Selling Products on Female Daily with SMOTE, Tomek, and SMOTE-Tomek," *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, Bali, Indonesia, 2020, pp. 81-85, doi: 10.1109/IAICT50021.2020. 9172033

[19] Y. Lu, Y. -M. Cheung and Y. Y. Tang, "Bayes Imbalance Impact Index: A Measure of Class Imbalanced Data Set for Classification Problem," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3525-3539, Sept. 2020, doi: 10.1109/ TNNLS.2019.2944962

[20] M. Khalafat, J. S. Alqatawna, R. M. H. Al-Sayyed, M. Eshtay, and T. Kobbaey, "Violence Detection over Online Social Networks: An Arabic Sentiment Analysis Approach," *International Journal of Interactive Mobile Technologies (iJIM),* vol. 15, no. 14, pp. 90-100, 2021, doi: 10.3991/ijim.v15i14.23029

[21] Y. Yang, H. -G. Yeh, W. Zhang, C. J. Lee, E. N. Meese and C. G. Lowe, "Feature Extraction, Selection, and K-Nearest Neighbors Algorithm for Shark Behavior Classification Based on Imbalanced Dataset," in *IEEE Sensors Jou*rnal, vol. 21, no. 5, pp. 6429-6439, 2021, doi: 10.1109/JSEN.2020.3038660

[22] T. Wong and P. Yeh, "Reliable Accuracy Estimates from k-Fold Cross Validation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586-1594, 2020, doi: 10.1109/TKDE.2019.2912815

[23] S. Amalia, I. Deborah and I.N. Yulita, "Comparative analysis of classification Algorithm: Random Forest, SPAARC, and MLP for Airlines Customer Satisfaction," *SINERGI*, vol. 26, no. 2, pp. 213-222, 2022, doi: 10.22441/sinergi.2022.2.010

[24] S. Qaiser, N. Yusoff, F. K. Ahmad, and R. Ali, "Sentiment Analysis of Impact of Technology on Employment from Text on Twitter," *International Journal of Interactive Mobile Technologies (iJIM),* vol. 14, no. 07, pp. 88-103, 2020, doi: 10.3991/ijim.v14i07. 10600

[25] D. Zhang and Y. Gong, "The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure," *IEEE Access*, vol. 8, pp. 220990–221003, 2020, doi: 10.1109/ACCESS.2020. 3042848

[26] A. I. Saad, "Opinion Mining on US Airline Twitter Data Using Machine Learning Techniques," *2020 16th International Computer Engineering Conference (ICENCO)*, Cairo, Egypt, 2020, pp. 59-63, doi: 10.1109/ICENCO49778.2020.9357390