Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2024. Vol. 3. No. 3; e-ISSN: 2808-4519

Optimization of Classification Algorithm with GridSearchCV and Hyperparameter Tuning for Sentiment Analysis of the Nusantara Capital City

Achmad Baroqah Pohan[1]*, Irmawati[2], Aliyah Kurniasih[3]

[1,2]Universitas Bina Sarana Informatika, Indonesia
[3]Universitas Ary Ginanjar, Indonesia
achmad.abq@bsi.ac.id[1], irmawati.iat@bsi.ac.id[2], aliyah.kurniasih@esqbs.ac.id[3]*

Abstract

The relocation of the country's capital has implications for social, economic, geographical and political aspects. This raises the public's views which are expressed in *Twitter* tweets that contain public sentiment. Sentiment analysis has a significant influence on understanding public views. Therefore, further research is needed related to the sentiment of analysis of the IKN. In this study, a model that can predict public sentiment is created by developing a model at the training stage using the *GridSearchCV* algorithm and *hyperparameter tuning* with several classification algorithms. The best model was produced with the *Support Vector Machine* classification algorithm that was able to outperform, compared to probability-based and tree-based classification algorithms with an accuracy of 69.95%.

Keywords: *Hyperparameter Tunning, Sentiment, Classification Algorithm*

1. Introduction

The Nusantara Capital City (IKN) of the Archipelago is a city that is projected to be the center of government for a country. The construction of the IKN for the relocation of the state capital from Jakarta to Kalimantan Island will have implications not only on the socio-economic and geographical aspects of the region, but with the realization of the political relocation of the state capital will create a new constitutional dynamic [1]. This constantly makes all layers of people comment through *Twitter*, the views of the community can have elements of sentiment that are positive, negative or neutral.

Sentiment analysis is the process of extracting and looking for patterns from textual data to obtain information contained in the text [2]. Machine learning modeling related to public sentiment analysis is still an interesting research topic. Sentiment analysis is carried out because it has significant relevance in understanding public views and from understanding the public can provide valuable insights for improvement and development [3].

Machine learning algorithms are still the main players in determining the results of model performance. Algorithms that are still trending to use such as the *Support Vector Machine*, where the SVM model's calculation of the weight of each word in sentiment greatly determines the accuracy produced [4]. Then *Naïve Bayes'* algorithm works by assuming the independence of variables so that only the variances associated with variables in a label are the focus in determining the classification and none take into account the entire covariance matrix [5]. Similarly, the *Logistic Regression* algorithm is a statistical analysis method used to predict the possible outcome of binary dependent variables (two categories) based on one or more independent variables. This method models the relationship between the predictor variable and the response variable through a logistic function, which results in a probability between 0 and 1 [6].

The *Decision Tree* algorithm is a predictive analysis method that uses a tree-like structure to make data-driven decisions. This structure consists of nodes, branches, and leaves, where each node represents an attribute, each branch represents the value of that attribute, and each leaf represents a class or result. *Decision tree* are easy to implement and their visualization makes it easy to understand the decision flow [7], [8]. There is also the *Random Forest* algorithm, which is an ensemble learning method used for classification, regression, and various other tasks by building a large number of decision trees during the training process. This method works by combining the results of various decision trees to make more accurate predictions and reduce the risk of overfitting that often occurs in single decision tree models [9].

In this study, classification algorithms such as *Logistic Regression, Naïve Bayes Classifier, Support Vector Machine, Decision Tree,* and *Random Forest* are applied by making tuning on *hyperparameters* for model training with the *GridSearchCV* algorithm process in improving the overall model evaluation performance results.

2. Research Method

2.1. Dataset

The data used in this study is *Twitter* tweet data in Indonesian which is only related to the Capital City of the Archipelago (IKN). Data was obtained through crawling with the keyword *ikn* language *id* (meaning data searched in Indonesian) in the crawling time period, which is March 1, 2024 to March 18, 2024. The crawling results obtained data of 8,616 instances and 12 features.

2.2. Exploratory Data Analysis

Exploratory data analysis is a stage where what is done is with the aim of understanding data, recognizing a problem from data, and gaining insights from data. Based on the results of the check, there are 7 *features* with data types of objects (strings) including *features* created_at, id_str, full_text, lang, user_id_str, username, and tweet_url, then there are 5 *features* with data types of intergers including quote_count, reply_count, retweet_count, favorite_count, conversation_id_str. The results of the subsequent check did not find any *missing value* data, but there were 85 *duplicates* data as a whole and 178 *duplicates* data in feature full_text.

Table 1 is a description of each *feature* along with examples of the data produced. In the text data itself from the full_text attribute, it can be seen from one example of the data row presented that the narrative of the tweet in the data contains URLs, capital letters, punctuation, and even numbers and slang words in other text data rows that cannot be presented in its entirety.

Table 1: Features Description

No	Features	Description	Data
1	Created_at	UTC standard format date and time	Sun Mar 17 18:55:41 +0000 2024
2	id_str	Unique ID of the tweet in the form of a string	1769437468242440400
3	full_text	Full text of a tweet	Bandara VVIP IKN nantinya tidak bisa dipakai untuk rakyat, melainkan khusus untuk penerbangan kepresidenan dan tamu-tamu penting negara. RAKYAT TERUTAMA WARGA KALIMANTAN ITU BUKAN TAMU/ORANG PENTING !!! Waras lo ? https://t.co/HLyU6mWKLn
4	quote_count	The number of times the tweet was quoted by other users	5
5	reply_count	The number of replies or how many people responded directly to the tweet	13
6	retweet_count	The number of retweets from the tweet	16
7	favorite_count	The number of 'likes' received by the tweet	52
8	lang	The language indicated by the tweet	in
9	user_id_str	The unique ID of the user who created the tweet in the form of a string	1038483432391634944
10	conversation_id_str	The unique ID of the conversation associated with the tweet in the form of a string	1769437468242440400
11	username	Username of the tweet creator	RomitsuT
12	tweet_url	Tweet URL or direct link	https://twitter.com/RomitsuT/status/1769437468242440400

2.3. Data Preprocessing

2.3.1. Convert Datetime

In this process, the crated_at *feature* will be converted from the UTC standard format to the ISO 8601 standard format (created_at_new), i.e. the data will be in the form of a date in the year-month-date format, followed by the time portion in the hour-minute-second format, and the time zone offset from UTC to +07:00, which means that the time is 7 hours ahead of UTC. Furthermore, the *feature* created_at_new is further divided into *feature* date (date_created), time (time_created), year, month, and day. The purpose of this process is to gain insight into what day the tweet occurred the most in that time span. Table 2 is the data from the first 5 rows of the conversion results.

Table 2: Convert Datetime Results

created_at	…	created_at_new	data_created	time_created	year	month	day
Mon Mar 18 00:11:17 +0000 2024	…	2024-03-18 07:11:17+07:00	2024-03-18	07:11:17	2024	March	Monday
Mon Mar 18 00:10:57 +0000 2024	…	2024-03-18 07:10:57+07:00	2024-03-18	07:10:57	2024	March	Monday
Mon Mar 18 00:10:18 +0000 2024	…	2024-03-18 07:10:18+07:00	2024-03-18	07:10:18	2024	March	Monday
Mon Mar 18 00:10:13 +0000 2024	…	2024-03-18 07:10:13+07:00	2024-03-18	07:10:13	2024	March	Monday
Mon Mar 18 00:06:20 +0000 2024	…	2024-03-18 07:06:20+07:00	2024-03-18	07:06:20	2024	March	Monday

Based on the results of the time conversion, it can be known the distribution of the amount of data by day, the most tweets occurred on Friday with 2,102 *instances*, followed by Tuesday 1,832 *instances*, Monday 1,738 *instances*, Sunday 1,449 *instances*, Saturday 781 *instances*, Thursday 377 *instances*, and Wednesday 337 *instances*. The tweet is data generated from crawling for a period of 19 days starting from February 29, 2024 to March 18, 2024.

2.3.2. Removing Duplicate Data and Missing Values

In this step, it is to delete *duplicate* data which is as many as 85 *instances*, after which there is an indication of *duplicate* data in *feature* full_text from the previous 178 *instances* to as many as 93 *instances*. Furthermore, *duplicate* data was also deleted based on the 90 full_text *feature*, so that the data dimensions became 8,438 *instances* and 18 *features*.

2.4. Text Preprocessing

1. Text Normalization
   At this stage, the tweet text data on feature full_text that will be used as input to the model is improved in the form of removing mentions, removing hashtags, removing RTs, removing URLs that are replaced with single spaces, converting non-ASCII characters to ASCII, removing certain patterns from text strings and then recombining the words generated using single spaces, lowercase, remove punctuation, remove single character, remove double spaces in the middle of sentences, remove double spaces at the beginning and end of sentences, and change English words to Indonesian.
2. Changing the Slang
   Furthermore, correct the word commonly called slang by using a corpus in the form of a txt document containing slang and standard words and then the tweet text data will be corrected based on the data on the corpus.
3. Stopword Removal
   Stopword removal is the process of removing common words in the tweet data based on the stopword data in the NLTK (Natural Language ToolKit) corpus.
4. Converting Number to Strings
   This process converts the numbers in the tweet data into words in Indonesian.
5. Lemmatization
   This process converts the words in the tweet data into root words based on the WordNetLemmatizer corpus data in NLTK (Natural Language ToolKit).

2.5. Labeling

In *machine learning* modeling, especially in the sentiment analysis task in this study, where a data from crawling *Twitter* tweets does not have a label, while *training* in *machine learning* modeling for sentiment analysis tasks requires labeled data. At this stage, the labeling process is carried out using the *polarity score* sentiment technique based on *positive* and *negative* corpus lexicon [10]. Table 3 is the first 10 rows of the results of the text data before and after *text preprocessing*, then *feature* polarity_score along with the labeling of sentiment (*feature* polarity) from the text data that has been cleaned based on the generation of *feature* polarity_score.

Table 3: Text Preprocessing and Labeling Results

No	full_text	text_preprocessed	polarity_score	polarity
1	@rasa2086 Dukung IKN	dukung ikn	4	positive
2	Bukan Kampus, Kepala Otorita Tegaskan Stanford Hanya Bangun Pusat Riset di IKN. Mio Mirza JATENG IS RED Nahan Pagiii Mawar Fajri https://t.co/eVkNXbFK92	kampus kepala otorita tegaskan stanford bangun pusat riset ikn mio mirza jateng be redaksi nahan pagiii mawar fajri	0	neutral
3	@tempodotco alhamdulillah untung SUMATERA tercinta ngk ikut2an IKN. selamat menikmati ya kalimantan.	alhamdulillah untung sumatera tercinta ngk ikut2an ikn selamat menikmati iya kalimantan	13	positive
4	Luar Biasa! Investasi Mengalir di IKN Sudah Tembus Rp 49,6 Triliun. Barca Joao Felix Mio Mirza JATENG IS RED hyeri Mawar Fajar https://t.co/UepfC1Uttc	investasi mengalir ikn tembus rp empat ratus sembilan puluh enam triliun barca joao felix mio mirza jateng be redaksi hyeri mawar fajar	3	positive
5	@tempodotco Warga kalimantan timur akan menjadi yg paling dirugikan dengan belum disahkannya RUU masyarakat adat terkait IKN. Masih pingsan semua orang2 Kaltim itu rupanya !!! MASIH BELUM PAHAM JUGA KALIAN !!! Malang kali nasib kalian, pingsanlah terus !!!	warga kalimantan timur dirugikan disahkannya ruu masyarakat adat terkait ikn pingsan orang2 kaltim paham malang kali nasib pingsanlah	-11	negative
6	@aiiptu Dukung pembangunan IKN	dukung pembangunan ikn	4	positive
7	@MurtadhaOne1 Bagimana sy bs percaya anda ini soal IKN? Dari suaranya sangat jelas itu keributan antara warga dngan perusahaan sawit.	bagimana percaya ikn suaranya keributan warga dngan perusahaan sawit	0	neutral
8	Kenapa beliau berbicara seperti teman sekelas https://t.co/8jI33BXtOR	beliau berbicara teman sekelas	-3	negative
9	Sebenernya kasihan. Kalian terusir dari tanah kalian sendiri. Tapi kan kalian dulu mendukung IKN. https://t.co/WEu5TcubAg	sebenernya kasihan terusir tanah mendukung ikn	1	positive
10	@DokterTifa Knp sih kamu d jawa sana sirik kl di luar jawa d bangun? Dr tirta yg sok tau krn pindah agama lgsg bisa koar koar? Hello.. Nanti juga cari makan d IKN., pret	sih jawa sirik jawa bangun dr tirta sok pindah agama langsung teriak teriak hello cari makan ikn pret	-12	negative

After the text preprocessing and labeling process, duplicate data was generated as many as 771 instances based on clean text data (feature text_preprocessed). Furthermore, the duplicate data is deleted so that the data dimensions become 7,667 instances and 18 features. Then a re-check was carried out regarding the missing value data and indicated that there was data that was missing value of 1 instance in feature text_preprocessed and deleted, then the data dimensions became 7,666 instances and 18 features.

2.6. Data Transformation

Data transformation is the process of changing data from string categorical to numerical categorical, this process is carried out on feature polarity as a sentiment feature for the target prediction model from machine learning classification. Table 4 below is the before and after process of data transformation on features polarity.

Table 4: Data Transformation Results

Before	After
negative	0
neutral	1
positive	2

2.7. Data Visualization

Figure 1 is the amount of data for each label based on days, this number is the data that has gone through the data and text preprocessing stage that is ready to be used for the machine learning model creation process. Based on this image, the distribution of the most tweet data on Friday was 1,974 instances, followed by Tuesday 1,665 instances, Monday 1,378 instances, Sunday 1,255 instances, Saturday 726 instances, Thursday 344 instances, and Wednesday 324 instances.

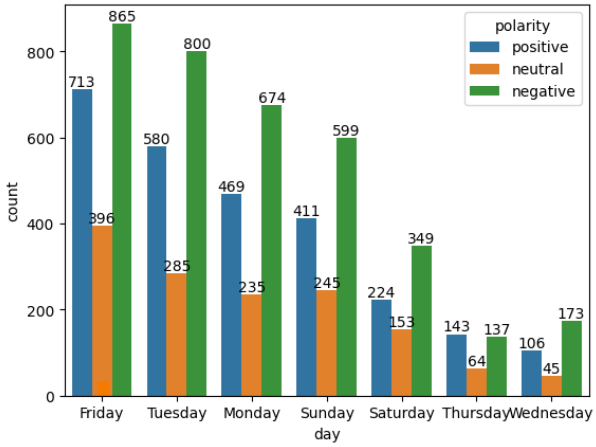

Fig. 1: Number of Label Data in Days

In Table 5, the amount of data based on the day is presented, namely in the data of the crawling results compared to the amount of data after going through the data process and text preprocessing, in the difference column is the column about the number of data deleted. It can be seen that there was a decrease in the amount of data from before and after due to indications of 949 instances of duplicate data and 1 instance of missing value data carried out in the previous process. So there is a difference in the overall amount of data from 8,616 instances to 7,666 instances, and the amount of data deleted is 950 instances.

Table 5: Comparison of the Amount of Data Before and After Preprocessing

Day	Before	After	Difference
Friday	2.102	1.974	128
Tuesday	1.832	1.665	167
Monday	1.738	1.378	360
Sunday	1.449	1.255	194
Saturday	781	726	55
Thursday	377	344	33
Wednesday	337	324	13
Sum	**8.616**	**7.666**	**950**

Figure 2 is the percentage of the amount of data based on labels, namely negative labels of 46.9% for 3,597 instances, positive labels of 34.5% for 2,646 instances, and neutral labels of 18.6% for 1,423 instances. Based on the distribution of the amount of data, it is indicated that there is an imbalances class data, but in this study, the process of handling imbalance classes is not carried out because the process does not always have a significant influence on machine learning modeling on text data [11].

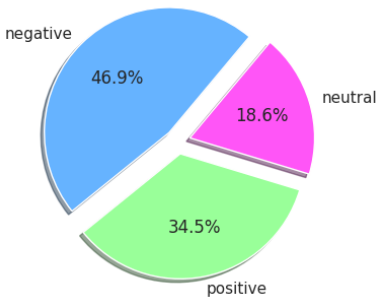

Fig. 2: Data Label Percentage

2.8. Training and Testing Data Split Ratio

Data division is the process of dividing data by determining which features will be the input of the model (X) and which features will be the target of the model (Y), then the data on X and Y will be re-divided with a percentage of 80% for train data and 20% for test data. In

this study, the input model is feature text_preprocessed, namely text data that has been preprocessed, and the target model is feature polarity as the sentiment of the text data. Then the results of data split on the data train were produced as many as 6,132 instances, and test data as many as 1,534 instances. The data train is used for the machine learning model training process and the test data is used for the machine learning model evaluation process. The amount of data from the train data and test data based on sentiment labels is presented in Table 6.

Table 6: Results of Data Split

Data	Label	Sum
Train	Negative [0]	2.868
	Neutral [1]	2.127
	Positive [2]	1.137
Test	Negative [0]	729
	Neutral [1]	519
	Positive [2]	286

2.9. TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) is one of the extraction features to convert text data into numeric which works by calculating the weight of words in a document or data line based on the frequency of words appearing in a document or data line and inverse document frequency, where documents or data lines that often contain certain words will have a higher weight while words that appear in the entire document (data) or appear too rarely in the document will have more weight low. Formula (1) to calculate the TF-IDF value is the result of $tf_{v,d}$ and $tf_{v,d}$, where $tf_{v,d}$ is the sum of the frequencies of the word t (for example) in a document or data row d (for example), and $tf_{v,d}$ is the frequency inverse of the entire document. TF increases with the number of word occurrences in a document or data row and IDF increases with word step across a document collection or entire data row [11].

$$x_{v,d} \rightarrow tf_{v,d} - idf_{v,d} = \left(1 + \log_{10} tc_{v,d}\right) \times \left(\log_{10} \frac{M}{df_v}\right) \quad (1)$$

In this study, the model input data (X) on the data train and data test was carried out by TfidfVectorizer using max_features=1000, but the data train was carried out with the fit_transform process and in the data test only the transform process was carried out.

2.10. Training Model with Hyperparameter Tuning

Hyperparameter tuning is the process of adjusting hyperparameters and hyperparameter values that are set before the training process [12]. Hyperparameter tuning is used with the GridSearchCV algorithm where this algorithm works by looking for a certain subset of the hyperparameter space of a machine learning model algorithm so that it can directly affect the performance results of the training model [12].

In this study, machine learning algorithms such as Naïve Bayes Classifier, Support Vector Machine, Decision Tree, and Random Forest were used to train the model by tuning on the hyperparameters of each algorithm. This training uses GridSearchCV with model algorithms, variables that contain tuning of each algorithm's hyperparameters, and 10-fold cross validation [13].

Table 7 is the hyperparameters and the variation of values used in each machine learning algorithm used. The best hyperparameter value was produced along with the score of the training model, namely the best model training score on the Logictic Regression algorithm of 71.38% with the best hyperparameter penalty L2 and the value of C=10, then followed by the next best model training score on the SVM algorithm 70.79%, Complement NB 67.06%, Random Forest 66.54%, Multinomial NB 64.94%, Bernoulli NB 63.37%, Gaussian NB is 59.72%, and the lowest result on the Decision Tree algorithm is 56.10%.

Table 7: Model Training Results with Hyperparameter Tunning

Algorithm	Hyperparameter and Values	Best Hyperparameters	Best Score
Logistic Regression	penalty= L1 dan L2 C=0.1, 1, 10, 100, 1000	penalty=L2 C=10	**71,38%**
Gaussian NB	var_smoothing=(0, -9, num=10)	var_smoothing=0.01	59,72%
Multinomial NB	alpha=0.01, 0.1, 1, 10	alpha=0.01	64,94%
Complement NB	alpha=0.01, 0.1, 1, 10	alpha=0.1	67,06%
Bernoulli NB	alpha=0.01, 0.1, 1, 10	alpha=0.1	63,37%
SVM	Kernel= rbf, linear, poly C=0.1, 1, 10 gamma (RBF)=1, 0.1, 0.01	Kernel=rbf C=10 gamma=0.1	70,79%
Decision Tree	criterion=gini, entropy max_depth= none, 10, 20 min_sample_split=2, 5, 10	criterion=gini max_depth=none min_samples_split=10	56,10%
Random Forest	n_estimators=100, 200, 300 max_depth=none, 10, 20 min_samples_split=2, 5, 10	n_estimators=200 max_depth=none min_samples_split=10	66,54%

3. Result and Discussion

Table 8 is the result of the overall model evaluation on each algorithm used on data that has three labels that are imbalance class. The metrix used in precision, recall and f1-score uses macro weighting. Of the overall performance metrics used, the best model was produced in the Support Vector Machine algorithm with an accuracy value of 69.95%, precision of 67.67%, recall of 54.96% and f1-score of 66.86%. These results prove that SVM on Indonesian text data can create the best hyperplane in separating data on the RBF kernel and a C value of 10 with a gamma of 0.1 compared to other probability-based algorithms such as the Naïve Bayers Classifier, and other tree-based

algorithms such as Decision Tree and Random Forest, although Random Forest works by maximizing each tree compared to Decision Tree, but this study proves that the SVM algorithm is still superior.

Table 8: Model Evaluation Results

Algoritma	Accuracy	Precision	Recall	F1-Score
Logistic Regression	68,90%	65,91%	65,75%	65,78%
Gaussian NB	61,54%	62,43%	63,85%	60,45%
Multinomial NB	63,43%	62,89%	54,67%	55,41%
Complement NB	66,88%	63,71%	63,06%	63,35%
Bernoulli NB	63,17%	61,55%	63,14%	61,13%
SVM	**69,95%**	**67,67%**	**66,27%**	**66,86%**
Decision Tree	56,78%	54,17%	54,96%	54,38%
Random Forest	66,30%	64,56%	61,47%	62,24%

Figure 3 is the result of the confusion matrix with the SVM algorithm, where through the test data of 20% (1,534 instances) the data predicted as negative sentiment (0) resulted in true as negative as 570 instances, but produced false as neutral (1) as many as 97 instances and false as positive (2) as many as 124 instances. Likewise, the data predicted as neutral sentiment (1) produces true as neutral for 151 instances, but produces false as negative (0) for 66 instances and false as positive (2) for 43 instances. Then in the data predicted as positive sentiment (2) resulted in true as positive for 352 instances, but produced false as negative (0) for 93 instances and false as neutral (1) for 38 instances.

So the actual data generated for negative sentiment (0) was 729 instances, neutral sentiment (1) was 286 instances, and positive sentiment (2) was 519 instances. However, the prediction data generated for negative sentiment (0) was 791 instances, neutral sentiment (1) was 260 instances, and positive sentiment (2) was 483 instances. Based on these results, there was a prediction error for negative sentiment of 62 instances, neutral 26 instances, and positive 36 instances.

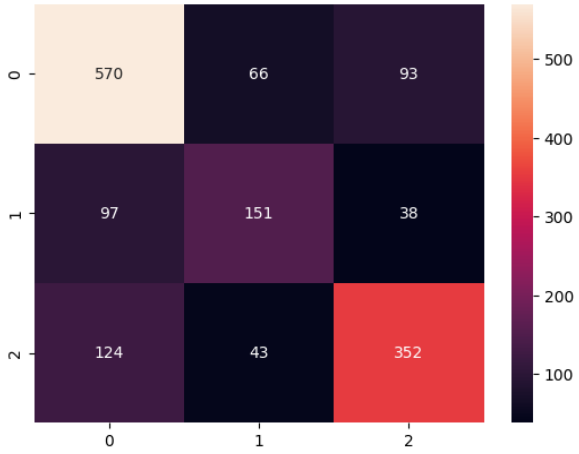

Fig. 3: Confusion Matrix SVM Result

4. Conclusion

Based on the performance metric value generated from the model evaluation, it is proven in this study that tuning the hyperparameter to the classification algorithm can maximize the overall model performance. However, from the results obtained, the best model, namely the SVM classification algorithm with an accuracy of 69.95%, can still be improved. These results can occur due to the existence of imbalance class data, or the use of feature extraction that is not suitable for this data. But overall, when viewed from the loading process and the order of use of methods, it can be said that it is quite good and maximum, which means that where it can be concluded, namely hyperparameter tuning with GridSearchCV and then 10-fold cross validation is able to collaborate well on classification algorithms such as Naïve Bayes Classifier, Support Vector Machine, Decision Tree, and Random Forest.

References

[1] E. Nugrohosudin, "Kedudukan Kepala Otorita Ibu Kota Nusantara," J. Legis., vol. 5, no. 2, pp. 79–90, 2022.
[2] A. K. Santoso, A. Noviriandini, A. Kurniasih, B. D. Wicaksono, and A. Nuryanto, "Klasifikasi Persepsi Pengguna Twitter Terhadap Kasus Covid-19 Menggunakan Metode Logistic Regression," JIK (Jurnal Inform. dan Komputer), vol. 5, no. 2, pp. 234–241, 2021.
[3] A. Supian, B. Tri Revaldo, N. Marhadi, L. Efrizoni, and R. Rahmaddeni, "Perbandingan Kinerja Naïve Bayes Dan Svm Pada Analisis Sentimen Twitter Ibukota Nusantara," J. Ilm. Inform., vol. 12, no. 01, pp. 15–21, 2024.
[4] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," J. Teknol. Inf. dan Ilmu Komput., vol. 8, no. 1, pp. 147–156, 2021.
[5] C. Mulia and A. Kurniasih, "Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Bank Customer Churn Menggunakan Algoritma Naïve bayes dan Logistic Regression," Pros. Semin. Ilm. Nas. Online Mhs. Ilmu Komput. dan Apl., pp. 552–559, 2023.
[6] W. F. Hidayat, T. Asra, and A. Setiadi, "Klasifikasi Penyakit Daun Kentang Menggunakan Model Logistic Regression," Indones. J. Softw. Eng., vol. 8, no. 2, pp. 173–179, 2022.
[7] I. Irmawati, H. Hermanto, E. H. Juningsih, S. Rahmatullah, and F. Aziz, "Prediksi Lama Tinggal Pasien Rawat Inap Di Rumah Sakit Pada Masa Pandemi Covid-19 Menggunakan Metode Ensemble Learning Dan Decission Tree," J. Inform. Kaputama, vol. 5, no. 2, pp. 391–397, 2021.
[8] E. Priyanti, "Penerapan Decision Tree Untuk Klasifikasi Tingkat Pendapatan," IJCIT (Indonesian J. Comput. Inf. Technol., vol. 7, no. 1, pp. 7–12,

2022.
[9] N. Wuryani and S. Agustiani, "Random Forest Classifier untuk Deteksi Penderita COVID-19 berbasis Citra CT Scan," J. Tek. Komput. AMIK BSI, vol. 7, no. 2, pp. 187–193, 2021.
[10] Fajri Koto and Gemala Y. Rahmaningtyas, "InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs," 2017 Int. Conf. Asian Lang. Process., pp. 391–394, 2017.
[11] M. P. Pulungan, A. Purnomo, and A. Kurniasih, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier," J. Teknol. Inf. dan Ilmu Komput., vol. 10, no. 7, pp. 1493–1502, 2023.
[12] Fatihah Rahmadayana and Yuliant Sibaroni, "Sentiment Analysis of Work from Home Activity using SVM with Randomized Search Optimization," J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 5, no. 5, pp. 936–942, 2021.
[13] A. B. Pohan, Irmawati, and A. Kurniasih, "Penerapan Teknik SMOTE dalam Memprediksi Kegagalan Mesin Menggunakan Support Vector Machine dan Logistic Regression," J. Inform. Kaputama, vol. 7, no. 2, pp. 181–187, 2023.