Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2025. Vol. 4. No. 2; e-ISSN: 2808-4519

The Effect of SMOTE Application on Support Vector Machine Performance in Sentiment Classification on Imbalanced Datasets

Dini Andriyani[1]*, Ahmad Faqih[2], Sandy Eka Permana[3]

[1,2] *Informatics Engineering, STMIK IKMI Cirebon*
[3] *Informatics Management, STMIK IKMI Cirebon*
andriyanidini433@gmail.com[1*]

Abstract

This research explores the effect of applying Synthetic Minority Oversampling Technique (SMOTE) on the performance of Support Vector Machine (SVM) algorithm in sentiment classification on imbalanced datasets. Public review data was collected from social media platform X (formerly Twitter) regarding the Free Lunch Program, with a total of 2,368 reviews automatically labeled using the BERT model into three categories: positive, negative, and neutral. Sentiment imbalance in the dataset was addressed by applying SMOTE to generate synthetic data on minority classes. The research method follows the stages of Knowledge Discovery in Databases (KDD), including data selection, preprocessing, labeling, transformation using TF-IDF, SVM model training, and performance evaluation. The experimental results show that the application of SMOTE successfully improves the accuracy of the SVM model by 12.48%, from 71.41% to 83.89%. Other evaluation metrics, such as precision, recall, and F1-score, also showed significant improvement from 0.69, 0.71, and 0.68 to 0.84, respectively. These findings confirm that SMOTE is effective in overcoming data imbalance, resulting in a more accurate and reliable sentiment classification model. This research contributes to the application of sentiment analysis in data-driven public policy evaluation.

Keywords: *Support Vector Machine (SVM), Social Media X, SMOTE, Free Lunch Program*

1. Introduction

Advances in informatics have influenced various aspects of life, such as technology, business, and education, with the application of data processing for more accurate decision-making through machine learning and artificial intelligence. It also supports digital learning and online collaboration and improves people's well-being. However, a major challenge in informatics is analyzing public opinion, especially regarding social policies such as the Free Lunch Program. Social media, such as X (Twitter), is often used to collect opinions that become datasets. The main problem is sentiment class imbalance, where positive sentiments are more dominant, while negative or neutral sentiments are underrepresented, which may cause bias in the classification model. In addition, social media data is often unstructured, making analysis difficult. The Support Vector Machine (SVM) method is effective for overcoming classification problems with high-dimensional data[1]. SVM can handle non-linear classification using kernels, which map the data to a higher dimensional space. Previous research shows the use of RBF kernels in SVM can achieve 93% accuracy in the classification of hate speech on social media[2]. Although SVM can handle class imbalance, additional techniques such as SMOTE are needed to balance the class distribution. SMOTE generates synthetic samples of minority classes, improves the representation of those classes, and reduces bias, so the model can better recognize minority classes[3]. Before classification, the sentiments in the dataset are converted into a numerical format using TF-IDF. This method gives weight to important words based on their frequency of occurrence in documents and the number of documents containing the word, so that the model can prioritize more relevant words[4]. This research aims to improve the accuracy of sentiment classification related to the Free Lunch Program by combining SVM and SMOTE methods to overcome class imbalance in the dataset. The public opinion dataset from platform X will be processed with TF-IDF and balanced using SMOTE[5]. The SVM model will be trained to classify the sentiment as positive, negative, or neutral, then tested and evaluated using accuracy, confusion matrix, and classification report (precision, recall, F1-score). If successful, this research can help policymakers understand public opinion more accurately and contribute to the development of machine learning-based sentiment analysis, especially in the context of social policy.

2. Research Methods

The research method used in this research is Knowledge Discovery in Databases (KDD), which is the process of finding knowledge and useful patterns in large amounts of data[6]. This process involves various stages of data analysis, from data collection to interpretation of results, and the purpose of this technique is to gain knowledge or understanding that can be used in the decision-making process. Here are the stages of the research method:

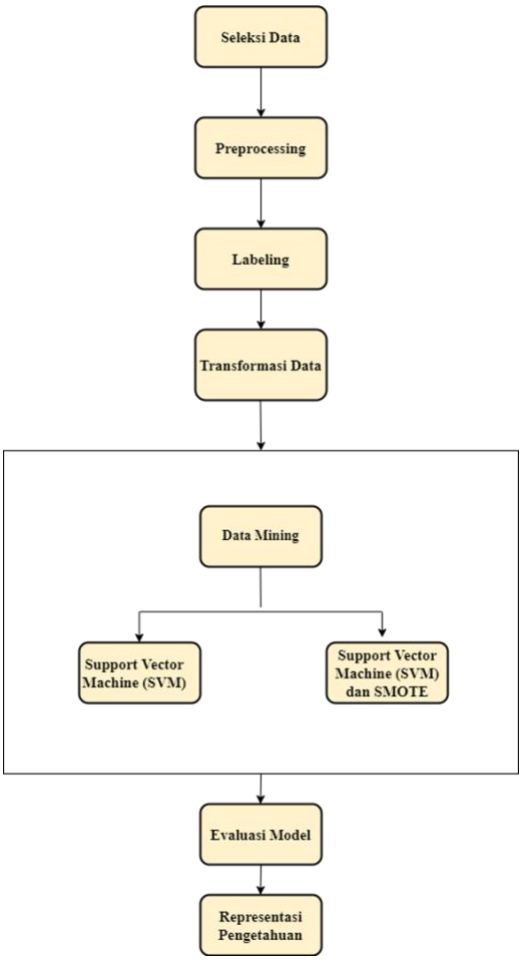

Fig. 1: Stages of The Research Method

2.1. Data Selection

Data selection[7] is the initial stage of this research process. Data selection is done by scraping comments from social media X using the keyword "Free Lunch Program". Then the comments were collected as many as 2,368. There are 15 columns from the scrapped data which include conversation_id_str, created_at, favorite_count. full_text, id_str, image_url, in_reply_to_screen_name, lang, location, quote_count, reply_count, retweet_count, tweet_url, user_id_str and username. Here is an example of twitter scrapping results:

Table 1: Scraped Sentiment Analysis Dataset

.....	*favorite_count*	*full_text*	*id_str*	…….
	0	Makan siang gratis tapi yang gratis cuma beef teriyaki-nya karena pakai sisaan trimming daging sate xixixixixi https://t.co/Wu1Aanar9w	1,84897E+18	
	0	dut makan siang gratis mana dut laper	1,84897E+18	
	0	@KemensetnegRI yang saya pahami di universitas kehidupan Tidak ada makan siang gratis	1,84897E+18	

2.2. Preprocessing

At this stage, the data that has been selected goes through a series of cleaning processes so that it is ready to be analyzed. In this preprocessing stage, it is divided into two, namely data preprocessing in the form of deleting duplicate data from the original data and text preprocessing from cleansing data to stemming process to produce clean data[8].

2.2.1. Data Preprocessing

The stage carried out is the removal of the same or duplicate data from the original data, the initial data preprocessing results as many as 2,368 tweets to 2,341 and the column used is the full_text column.

	full_text
0	Makan siang gratis tapi yang gratis cuma beef ...
1	dut makan siang gratis mana dut laper
2	@KemensetnegRI yang saya pahami di universitas...
3	@detikcom dari 3.005 T sebanyak 1000T akan dik...
4	@unmagnetism Makan siang gratis dulu un https:...
...	...
2363	Gizi seimbang untuk anak bangsa #IndonesiaEmas...
2364	Makanan bergizi itu penting #IndonesiaEmas #Pr...
2365	@roseuro Program makan siang bergizi gratis ko...
2366	@Arie_Kriting Pasti ini nyindir program makan ...
2367	Makan siang gratis membantu anak-anak belajar ...

2341 rows × 1 columns

Fig. 2: Preprocessing Data Result

2.2.2. Text Preprocessing

The steps taken are as follows:

1. Cleansing
   The initial process is data cleansing. The stages carried out are cleaning the text from components that are not relevant for analysis such as emojis, punctuation marks, URLs, or symbols[9].
2. Case Folding
   The second process is case folding. The stages carried out are changing all text to lowercase letters to be consistent and avoid word differences only because of capitalization[10].
3. Tokenization
   The third process is tokenization. Tokenization is the process of breaking the text into small units called tokens, which can be words, sentences, or other elements, according to the needs of the analysis. The purpose of this process is to simplify text analysis by dividing raw text into parts that are easier to process and analyze using natural language processing (NLP) algorithms.
4. Stopword Removal
   The fourth process is stopword_removal. Stopword removal aims to remove common words that are considered less informative in text analysis, such as "and," "or," "is," and similar words. These words are called stopwords and generally do not contribute significantly to understanding the context or content of the document[11].
5. Normalization
   The fifth process is normalization. Normalization aims to make text data uniform in a consistent format. This process involves steps such as correcting spelling, converting uppercase letters to lowercase, removing special characters or punctuation marks, and equalizing word variations with the same meaning.
6. Stemming
   The last process is stemming. Stemming aims to simplify words into their basic or root form (stem). This process removes suffixes or affixes on words, so that various variations of words that have similar meanings can be combined into one basic form.

Table 2: Preprocessing Result

Full_text	*cleansing*	*case_folding*	*tokenization*	*stopword_removal*	normalisasi	*stemming*
Makan siang gratis tapi yang gratis Cuma beef teriyaki-nya karena pakai sisaan trimming daging sate xixixixixi https://t.co/Wu1Aanar9w	Makan siang gratis tapi yang gratis Cuma beef teriyakinya karena pakai sisaan trimming daging sate xixixixixi	makan siang gratis tapi yang gratis Cuma beef teriyakinya karena pakai sisaan trimming daging sate xixixixixi	[‘makan’, ‘siang’, ‘gratis’, ‘tapi’, ‘yang’, ‘gratis’, ‘Cuma’, ‘beef’, ‘teriyakinya’, ‘karena’, ‘pakai’, ‘sisaan’, ‘trimming’, ‘daging’, ‘sate’, ‘xixixixixi’]	[‘makan’, ‘siang’, ‘gratis’, ‘gratis’, ‘beef’, ‘teriyakinya’, ‘pakai’, ‘sisaan’, ‘trimming’, ‘daging’, ‘sate’, ‘xixixixixi’]	makan siang gratis gratis beef teriyakinya pakai sisaan trimming daging sate xixixixixi	makan siang gratis gratis beef teriyakinya pakai sisa trimming daging sate xixixixixi
……	…….	…….	……..	……..	……	……

2.3. Labeling

After preprocessing, the clean data is labeled with sentiment (positive, negative, or neutral) using IndoBERT, a BERT architecture-based model adapted for the Indonesian language. IndoBERT excels at understanding Indonesian context, making it ideal for NLP tasks such as text classification and sentiment analysis. The process involved fine-tuning IndoBERT with previously labeled data, allowing the model to recognize important patterns and accurately apply labels to new data. Here are the results:

Table 3: Labeling Result

index	stemming	label
0	makan siang gratis gratis beef teriyakinya pakai sisa trimming	POSITIF

index	stemming	label
	daging sate xixixixixi	
1	dut makan siang gratis dut laper	POSITIF
2	kemensetnegri paham universitas hidup makan siang gratis	NEGATIF

2.4. Data Transformation

Data transformation is the process of modifying raw data to make it more suitable for analysis or modeling, with the aim of improving the quality, consistency, and informative value of the data so that it is more useful for the model or analysis to be applied. The data transformation method, namely in the form of TF-IDF (Term Frequency-Inverse Document Frequency) is a way to evaluate how important a word is in a particular document compared to the number of words in the same set of documents. This research implements the Support Vector Machine (SVM) algorithm with a linear kernel, combined with TF-IDF feature extraction. The validity of the results is tested using a confusion matrix. The results show that the combination of TF-IDF feature extraction and SVM is able to achieve an accuracy of 99.34%, which indicates an excellent classification[12].

2.5 Data Mining

This process is part of the KDD stage that aims to convert raw data into valuable information. The Data Mining process is divided into two approaches, namely modeling with Support Vector Machine (SVM) and modeling with Support Vector Machine (SVM) equipped with SMOTE as an additional method to improve accuracy.

```
# Inisialisasi model SVM
model = SVC(kernel='linear')

# Latih model dengan data latih
model.fit(X_train, y_train)

# Lakukan prediksi pada data uji
y_pred = model.predict(X_test)

print(model)
```

```
SVC(kernel='linear')
```

Fig. 3: Modeling Script Using Support Vector Machine

The code above implements the Support Vector Machine (SVM) model for classification using a linear kernel. First, the SVM model is initialized with a linear kernel (kernel='linear') to linearly separate the data by finding the optimal hyperplane. Then, the model is trained using the training data (X_train, y_train), so that the SVM can learn the patterns that separate the data according to their labels. After training, the model is used to predict the test data label (X_test), with the prediction results stored in y_pred. These results can later be compared with the original labels to evaluate the accuracy of the model. Once the data is processed, a Support Vector Machine (SVM) model is used to perform sentiment classification. SVM is an algorithm that works by separating data from different categories through optimal margins. The advantage of SVM is its ability to work well on unbalanced datasets. The model will be trained to predict whether a comment has a positive, negative, or neutral sentiment, using the features that have been generated from the pre-processing stage. Research[13] entitled "Analysis of Public Sentiment Regarding PPKM on SVM-Based Twitter Using Python" discusses sentiment related to the implementation of PPKM in Java and Bali during the COVID-19 pandemic using the Support Vector Machine (SVM) algorithm to classify sentiment from comments on Twitter. The research stages include problem identification, data collection from Twitter, data preprocessing (such as text cleaning, tokenization, and stemming), manual labeling, division of training and test data, and term weighting using TF-IDF. The classification process was performed using SVM with two kernels, namely linear and RBF kernels. The evaluation results showed that the linear kernel achieved 86% accuracy, while the RBF kernel had 84% accuracy, with the findings indicating that PPKM is effective in suppressing the spread of COVID-19 based on the sentiments analyzed.

TF-IDF DAN SMOTE

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from imblearn.over_sampling import SMOTE

# Inisialisasi TF-IDF Vectorizer dan transformasikan data teks menjadi vektor numerik
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(data['stemming'])  # Hasil TF-IDF sebagai fitur
y = data['label']  # Label

# Terapkan SMOTE untuk menangani data tidak seimbang
smote = SMOTE(random_state=42)
X_smote, y_smote = smote.fit_resample(X, y)

print(smote)
```

```
SMOTE(random_state=42)
```

Fig. 4: Modeling Script Using Support Vector Machine

This code performs text data preprocessing and addresses class imbalance by using the Synthetic Minority Oversampling Technique (SMOTE). First, the TfidfVectorizer is initialized to convert the raw text in the normalization column into a numerical vector representation based on TF-IDF (Term Frequency-Inverse Document Frequency), which is stored as a feature in the X variable. The classification label

for each text is stored in the y variable of the label column. After the texts are converted into numerical vectors, SMOTE is applied to handle the class imbalance in the dataset by synthetically generating additional samples for the minority class, so that the proportion of the minority class becomes equal to that of the majority class. The balanced data is stored in X_SMOTE and y_SMOTE, ready to be used in the model training stage.

2.6. Model Evaluation

Model evaluation aims to measure the performance of the model in predicting the data, using metrics such as accuracy (proportion of correct predictions), precision (accuracy in predicting a particular class), recall (ability of the model to find positive tweets), and F1-score (harmonic mean between precision and recall) to assess the balance between the two.

2.7. Knowledge Representation

The accuracy comparison between standard SVM and SVM with SMOTE helps understand the effect of handling class imbalance on model performance. Standard SVM tends to ignore minority classes, while SMOTE adds synthetic samples to improve the detection of such classes, thus improving accuracy. This evaluation shows how effective SMOTE is in helping the model recognize underrepresented classes.

3. Result and Discussion

This process begins by calculating the distribution of each class, which is determining the amount of data included in each sentiment category. Based on the dataset from the scraping results taken, namely 2,368 and after the preprocessing process by removing duplicate data, the total dataset becomes 2,341. Therefore, from 2,341 data, data with positive sentiment labels amounted to 1,289, data with negative sentiment labels amounted to 804 and data with neutral sentiment labels amounted to 248. This indicates a significant class imbalance. In accordance with research[14] there is an imbalance in sentiment results, where words with positive sentiment have a much higher value in documents that are actually classified as negative, and vice versa.

Table 4: Sum of Sentiment Result Distribution

Sentiment	Number
Positif	1.289
Negatif	804
Netral	248
Total	2.341

Based on the table above, it can be seen that class imbalance can degrade the performance of machine learning models, as the models tend to prioritize the majority class and ignore the minority class, leading to less accurate predictions on fewer classes. The analysis of Support Vector Machine (SVM) performance in classifying sentiment on unbalanced datasets, before and after the application of Synthetic Minority Oversampling Technique (SMOTE), is a significant topic in the field of machine learning and sentiment analysis. Unbalanced datasets, where the number of samples in each class differs significantly, may cause the model to overlook minority classes, reducing the accuracy and reliability of predictions. In sentiment analysis, SVM serves to separate text data based on positive, negative, or neutral sentiments. However, in unbalanced datasets, SVM may have difficulty recognizing classes with fewer samples. Therefore, SMOTE is used to overcome this problem by generating synthetic samples on minority classes. SMOTE helps the model learn better about the patterns in the minority class, thus improving classification performance. In research[15] sentiment analysis requires a reliable classification method. In this study, SVM was used as a classification method with the application of SMOTE to overcome data imbalance in three sentiment categories. Omnibus Law is generally reported neutrally by CNNIndonesia.com. The one-vs-all method showed better classification results than the one-vs-one method. The application of SMOTE provides a slight improvement in results because the data imbalance is not too extreme. Modeling using the one-vs-all method with SMOTE on a data distribution of 90% for training and 10% for testing resulted in an average macro F1-score of 60.33%. Performance evaluation in this study is done by comparing metrics such as accuracy, confusion matrix, and Classification report which includes precision, recall, and F1-score before and after the application of SMOTE. The following table compares the SVM model evaluation without SMOTE and the model evaluation with SMOTE:

Table 5: Comparison of Model Evaluation Results

	Support Vectoc Machine and SMOTE	Support Vector Machine without SMOTE	Difference
Accuracy	83,89%	71,41%	12,48%
Precision	0,84	0,69	0,15
Recall	0,84	0,71	0,13
F1-score	0,84	0,68	0,16

In this study, the accuracy increased by 12.48% after the application of SMOTE. Initially, the accuracy of the SVM method alone was 71.41%, but after the application of SMOTE, it increased to 83.89%. Classification report results in this study show that precision increased from 0.69 to 0.84, which indicates the model is more accurate in prediction without many errors. Recall increased from 0.71 to 0.84, indicating that the model with SMOTE is better at recognizing samples from minority classes. F1-score, which balances precision and recall, also increased from 0.68 to 0.84. The improvement in all these metrics shows that the application of SMOTE helps the model handle data imbalance, resulting in a more accurate and balanced classification.

4. Conclusion

The effect of SMOTE application on Support Vector Machine (SVM) performance in sentiment classification on unbalanced datasets can be seen from the results of data distribution analysis which shows significant class imbalance, with positive sentiment dominating (1,289 data), negative sentiment (804 data), and neutral sentiment being the least (248 data). This imbalance could potentially decrease the accuracy of the model, as SVM tends to prioritize the majority class. However, the application of SMOTE managed to significantly improve the performance of the model. Model accuracy increased by 12.48%, from 71.41% to 83.89%, while other evaluation metrics also showed significant improvements. Precision increased from 0.69 to 0.84, recall from 0.71 to 0.84, and F1-score from 0.68 to 0.84. These improvements show that SMOTE effectively reduces the bias towards majority classes and improves the model's ability to recognize minority classes. However, some challenges remain, such as the neutral class which still requires additional approaches to improve its prediction.

References

[1] S. Rabbani, D. Safitri, N. Rahmadhani, A. A. F. Sani, and M. K. Anam, "Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 153–160, Oct. 2023, doi: 10.57152/malcom.v3i2.897.

[2] O. Nurdiawan, R. Herdiana, and S. Anwar, "Penerapan Algoritma Support Vector Machine dalam Mengukur Kepuasan Pembelajaran Hybrid Learning," *MEANS (Media Inf. Anal. dan Sist.*, vol. 6, no. 2, pp. 130–134, Jan. 2021, doi: 10.54367/means.v6i2.1511.

[3] O. H. Rahman, G. Abdillah, and A. Komarudin, "Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 17–23, Feb. 2021, doi: 10.29207/resti.v5i1.2700.

[4] C. Cahyaningtyas, Y. Nataliani, and I. R. Widiasari, "Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis SMOTE," *AITI (Jurnal Teknol. Informasi)*, vol. 18, no. 2, pp. 173–184, Nov. 2021, doi: 10.24246/aiti.v18i2.173-184.

[5] F. M. Basysyar, G. Dwilestari, and A. I. Purnamasari, "ANALYSIS STUDENT EMOTIONS AND MENTAL HEALTH ON," vol. 10, no. 2, pp. 361–368, 2024, doi: 10.33480/jitk.v10i2.5967.ANALYSIS.

[6] A. Karimah, G. Dwilestari, and Mulyawan, "Analisis Sentimen Komentar Video Mobil Listrik Di Platform Youtube Dengan Metode Naive Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 1, pp. 767–737, 2024, doi: 10.36040/jati.v8i1.8373.

[7] T. M. Permata Aulia, N. Arifin, and R. Mayasari, "Perbandingan Kernel Support Vector Machine (SVM) Dalam Penerapan Analisis Sentimen Vaksinisasi Covid-19," *SINTECH (Science Inf. Technol. J.*, vol. 4, no. 2, pp. 139–145, Oct. 2021, doi: 10.31598/sintechjournal.v4i2.762.

[8] M. I. Fikri, T. S. Sabrila, and Y. Azhar, "Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter," *SMATIKA J.*, vol. 10, no. 02, pp. 71–76, 2020.

[9] V. K. S. Que, A. Iriani, and H. D. Purnomo, "Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 2, pp. 162–170, May 2020, doi: 10.22146/jnteti.v9i2.102.

[10] A. R. Isnain, A. I. Sakti, D. Alita, and N. S. Marga, "Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma SVM," *J. Data Min. dan Sist. Inf.*, vol. 2, no. 1, p. 31, Feb. 2021, doi: 10.33365/jdmsi.v2i1.1021.

[11] H. Saputra, "Analisis Sentimen Pada Vaksin Booster Menggunakan Algoritma Support Vector Machine Multiclass Di Twitter," *J. Teknol. Pint.*, vol. 3, no. 10, pp. 1–26, 2023, [Online]. Available: http://teknologipintar.org/index.php/teknologipintar/article/view/506

[12] D. Atika, Styawati, and A. Ari Aldino, "Term Frequency-Inverse Document Frequency Support Vector Machine Untuk Analisis Sentimen Opini Masyarakat Terhadap Tekanan Mental Pada Media Sosial Twitter," *J. Teknol. dan Sist. Inf.*, vol. 3, no. 4, pp. 86–97, 2022, [Online]. Available: http://jim.teknokrat.ac.id/index.php/JTSI

[13] R. Wati and S. Ernawati, "Analisis Sentimen Persepsi Publik Mengenai PPKM Pada Twitter Berbasis SVM Menggunakan Python," *J. Tek. Inform. UNIKA St. Thomas*, vol. 06, no. 02, pp. 240–247, Nov. 2021, doi: 10.54367/jtiust.v6i2.1465.

[14] O. I. Gifari, M. Adha, I. R. Hendrawan, and F. F. Setlight Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *J. Inf. Technol.*, vol. 2, no. 1, pp. 36–40, 2022.

[15] W. P. Hutami, H. Wijayanto, and I. D. Sulvianti, "Penerapan Support Vector Machine dengan SMOTE Untuk Klasifikasi Sentimen Pemberitaan Omnibus Law Pada Situs CNNIndonesia.com," *Xplore J. Stat.*, vol. 11, no. 1, pp. 26–35, Jan. 2022, doi: 10.29244/xplore.v11i1.852.