Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

Twitter Sentiment Analysis of Kanjuruhan Disaster using Word2Vec and Support Vector Machine

Fariz Muhammad Rizky[1*], Jondri[2], Kemas Muslim Lhaksmana[3]

School of Computing, Telkom University, Bandung, Indonesia
Email: [1]farizmr@students.telkomuniversity.ac.id, [2]jondri@telkomuniversity.ac.id, [3]kemasmuslim@telkomuniversity.ac.id
Correspondence Author Email: farizmr@students.telkomuniversity.ac.id
Submitted: **11/06/2023**; Accepted: **27/06/2023**; Published**: 29/06/2023**

Abstract−The Kanjuruhan disaster on 1 October 2022, gained the peoples attention. People share their thoughts on social media. Through their posts they generated variety of perspectives. Sentiment analysis is possible to use on a dataset of people's posts. Sentiment analysis applied in this disaster and people's social media posts because it provides valuable insights into the public's emotional response and overall sentiment towards the event. By analyzing the sentiments expressed in these posts, we can gauge the level of concern, sympathy, anger, or any other emotions prevalent among the affected. This research applies the supervised learning Support Vector Machine (SVM) method with feature expansion using Word2Vec and TF-IDF as weighting. SVM models for sentiment analysis will generated a model to classify text or social media posts into sentiment categories. By training on labeled data, SVM learns to distinguish between these sentiment classes using extracted features from the text. Later, the model's performance will be analyzed. Three SVM kernels—rbf, linear, and polynomial—are applied. Three split data techniques and two different types of training data are used to train each kernel. Training data with oversampling and training data without oversampling are the two types of training data. The best result gained from using rbf kernel, split ratio 70:30, and oversampling. From it, oversampling trained model have relatively stable in every split rasio and kernel without having significant difference.

Keywords: Football; Kanjuruhan Disaster; Machine Learning; Twitter; Sentiment Analysis

1. INTRODUCTION

On 1 October 2022, a tragic stampede disaster occurred in Liga 1 match between Arema and Persebaya in Kanjuruhan Stadium, Malang, Indonesia. The disaster annihilated 135 people and harmed at least 500 people [1]. Clips and pictures along with public opinion of the disaster filled any social media timeline. There were many perspectives that people narrate through their account, including criticizing the police failure, empathize the victim, protesting the negliance of Indonesia FA (PSSI), and mentioning the anarchist actions of the supporters. To understand and evaluate emotions, perspectives, and opinions that people or communities have expressed toward a particular issue, circumstance, or item, sentiment analysis is necessary. It offers insightful information about public opinion in social media and effectively address issues. The amount of user-generated content in the digital era has led to the emergence of sentiment analysis, which aims to computationally analyze and interpret attitudes expressed in textual data. It has become a crucial area of study in understanding human behavior and opinions within the vast quantities of data available.

In this research, the data was collected from Twitter using 6 keywords related to the Kanjuruhan disaster. The time period for data collection was from 2 October 2022 until 30 December 2022. Duplicate tweets and spam content were removed, and the remaining tweets were manually labeled by a group of 5 individuals. Preprocessing steps were performed to clean the data, including case folding, punctuation and space removal, tokenization, stop word removal, and stemming. The clean data was then weighted using TF-IDF and expanded with Word2Vec based on the Indonesian Wikipedia corpus. The data was split in half for testing, and the training data was duplicated for oversampling. Support Vector Machine (SVM) classification was performed on each training data, and the test data was used for evaluation to analyze the results of the training.

Sentiment analysis is the technique of systematically finding, extracting, and categorizing subjective information, opinions and emotions conveyed in textual data. It includes a variety of sub tasks such as sentiment classification, emotion detection, opinion extraction, and aspect-based sentiment analysis. Natural language processing (NLP), machine learning, and deep learning techniques are used in sentiment analysis to extract valuable insights from massive volumes of text, allowing organizations to make data-driven decisions, improve customer satisfaction, and gain a competitive advantage.

Many research that concerned sentiment analysis has often been done before. Various methods have been used to analyze sentiment that obtained from social media. Research by Hermanto, Kuntoro, *et al*. [2] about sentiment on Google Play Store review using Naïve Bayes and Support Vector Machine (SVM) with SMOTE oversampling has concluded that SVM's performance is higher than Naïve-Bayes.

Marcio Guia *et al*. [3] compared classification result of SVM, Decision Tree, and Random Forest on 400 reviews of cellphone on Amazon with TF-IDF weighting. Of the three classifier, they claimed SVM linear kernel have highest matrix score (accuracy, precision, recall, and F1-score).

According to Al-Saqqa and Aswajan [4], Word2Vec is one of commonly used feature extraction modeling. They surveyed 12 literatures about use of Word2Vec and they concluded Word2Vec are acceptable to be used in data classification. W2V skip-gram model along with SVM resulted accuracy score up to 0,7 [5]–[9].

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

Rizki on sentiment of Covid-19 vaccine[10] and Nooraeni, *eat al.* [11] on sentiment of KPK Law Bill, both using dataset from Twitter and SVM rbf kernel as classifier. They both satisfied with the matrix values.

This research differs from previous studies as it focuses specifically on sentiment analysis of tweets related to the Kanjuruhan incident [12], utilizing targeted keywords and hashtags. It introduces a novel approach by combining feature expansion techniques using Word2Vec and TF-IDF weighting to improve the accuracy of sentiment classification. The main objective is to achieve accurate sentiment analysis by training an SVM model with these features, aiming to gain insights into public emotions and opinions surrounding the incident.

2. RESEARCH METHODOLOGY

2.1 Stages of study

A flow is used to conduct research in this section. The flow diagram is shown in **Error! Reference source not found.**.

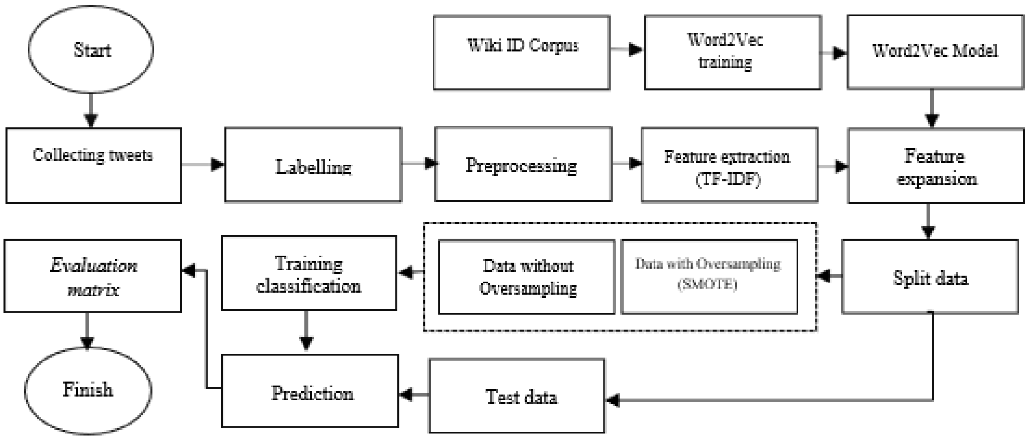

Figure 1. Stages of study flowchart

Data was taken from Twitter with 6 keywords, including "Kanjuruhan", "#UsutTuntas", "#UsutTuntasTragediKanjuruhan", "Tragedi Kanjuruhan", "Aremania", and "TGIPF" and in the time period 2 October 2022 until 30 December 2022. Attributes that taken is tweet id, tweet text, username, and date. A number with the following criteria – tweets with the same id and repeated tweet content (spam; majority came from news channel promotions) – were deleted.

Tweets were then manually labeled by a group of 5 and the preprocessing stage was carried out to clean the data by deleting duplicate tweets, case folding, cleaning tweets from excess punctuation and spaces, tokenization. Stop word removal, and stemming.

The clean data then weighted with TF-IDF. The feature expansion was implemented by adding the Word2Vec model based on the Indonesian Wikipedia corpus (January 2023). Then, data is split in half with the ratio determined for the test scenario. The training data for each scenario is then doubled for oversampling purpose for each scenario, so right now, there are 2 test data for each scenario, without oversampling and oversampled. Training classification with Support Vector Machine (SVM) is performed on each training data. Test data is used to cross-check or predict training results. Evaluation matrix is then generated from the test results to analyze the result of training.

2.2 Sentiment Analysis

Sentiment analysis is a field that discusses the analysis of opinions, expressions and attitudes in the form of text. The object of analysis can be products, services, organizations, individuals, activities and phenomena [13]. This field of study is believed to have first introduced in 2003 by Nasukawa and Yi [13].

Sentiment analysis is a subfield of Natural Language Processing (NLP) and Machine Learning (ML) that focuses in extracting opinions, emotions, and assumptions from text. Texts can come from a variety of sources, including social media, product reviews, and furthermore. The primary goal of sentiment analysis is to discover how the author of the text felt about a specific topic. In this case, sentiment analysis attempts to categorize texts into positive or negative categories[13].

The techniques used in sentiment analysis differ. A lexicon-based method is one popular approach, in which text is analyzed based on a predefined set of words (lexicon) that have a sentiment score assigned to them. Machine learning techniques such as logistic regression, Naïve Bayes, Support Vector Machines (SVM), and deep learning such as Convolutional Neural Networks (CNN) are used as alternative techniques. These methods involve model training of a dataset that already labeled with sentiment label (positive or negative) and then applying the trained model to classify unlabeled text sentiment[13], [14].

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

2.3 Support Vector Machine

Support Vector Machine (SVM) was first introduced by Vapnik in 1992. Vapnik explained that SVM is a classification method for nonlinear and linear problems by finding the separator function (hyperplane) based on the distance between the vectors of each class. This vector is called a "support vector", in that way, SVM is able to perform uniformity (generalization) on new input data [15], [16].

Support Vector Machines (SVM) are supervised learning models that good at classification tasks by mapping data points. The goal of SVM is to find an optimal hyperplane that separates data points from different classes as much as possible. SVM has been used successfully in sentiment analysis to classify textual data into positive, negative, or neutral sentiment categories. SVM's versatility allows them to handle both binary and multi-class sentiment classification problems, making them suitable for a wide range of applications[15].

Support Vector Machine has several kernels that can be used. The kernel in SVM is divided into two, Linear and Nonlinear. Nonlinear kernels are further divided into several types, such as Polynomial, Radial Basis Function (RBF), and Sigmoid [16].

Support Vector Machine is a method for machine learning that is used for classification and regression[15]. SVM can be used to classify text into positive, negative, or neutral sentiment categories in the context of sentiment analysis.

2.4 Word2Vec Model

Word2Vec is a method used in machine learning that turns words in text into vectors of numbers that are able to be utilized for text analysis or other natural language processing (NLP) purposes. The embedding concept is implemented by Word2Vec, which enables words in text to be represented as numerical vectors that may be used in computations [4].

In Word2Vec, CBOW (Continuous Bag of Words) and Skip-Gram are the two primary algorithms. The CBOW algorithm is used to predict missing words in the context of given phrases. In contrast, the Skip-Gram method is used to predict the words that will appear in the given word's context [4], [5].

The CBOW approach will attempt to fill in the missing context by using each word as input. While the Skip-Gram approach makes assumptions about the words that exist in a word's context [4].

Word2Vec produces vectors with many different uses. First, we can locate "similar" words in a text by using this vector in mathematical operations like the cosine of similarity. Second, these vectors can be applied to tasks like sentiment analysis and text classification [4], [5], [7].

The model that used in this research is the Indonesian Wikipedia dump of January 2023.

2.5 Feature Extraction (TF-IDF)

TF-IDF (Term Frequency—Inverse Document Frequency) is a popular feature extraction method commonly used in natural language processing and information retrieval tasks. It calculates the weight or importance of a word in a corpus by considering its frequency within individual documents (sentences) as well as its frequency across the entire corpus[17].

The TF component of TF-IDF calculates the frequency of a word within a specific document. It represents the local importance of a word within that document. The TF value increases as the word appears more frequently in the document. For example, if a word appears five times in a document, its TF value for that document will be higher compared to a word that appears only once[17].

The TF-IDF formula is displayed in formulas 1 and 2[17].

$$IDF_t = log\left(\frac{N}{df_t}\right) \quad (1)$$

$$W_{d,t} = tf_{d,t} \times IDF_t \quad (2)$$

Where *d* is the *d*-document, *t* is the word to look for in *d*, *W* is the weight of the *d*-th document to *t*. *IDF* is the Inverse Document Frequency, *N* is the total document, *df* is the number of documents with the searched word.

2.6 SMOTE (Oversampling)

SMOTE is an approach for handling imbalance data through oversampling. This method combines/synthesizes the closest data which is part of the minority class to create a new sample [2].

Several oversampling techniques have been proposed and employed in sentiment analysis to balance class distributions and improve the performance of sentiment classifiers. Synthetic Minority Over-sampling Technique (SMOTE) is a popular oversampling method that creates synthetic instances by interpolating between existing minority class instances. SMOTE generates synthetic samples in feature space, preserving the underlying structure of the minority class. Variations of SMOTE, such as Borderline-SMOTE and ADASYN, have been proposed to address specific challenges and improve classification results.

In machine learning usages, oversampling approaches are essential for managing the class imbalance problem. Because of its success in generating synthetic minority samples, the Synthetic Minority Over-sampling

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

Technique (SMOTE) has attracted a lot of interest among various oversampling methods. SMOTE balances the class distribution by interpolating new instances between existing minority class samples[18].

2.7 Evaluation

A confusion matrix can be used to measure performance evaluation. The accuracy value and f1-score are used in this study. Because it can interpret the average of the precision and recall values, the F1-score is employed[19]. Confusion matrix can be seen as in Table 1**.**

Table 1. Confusion Matrix

		Actual value	
		Positive (+)	**Negative (-)**
Prediction result	**Positive (+)**	*True positive* (TP)	*False positive* (FP)
	Negative (-)	*False negative* (FN)	*True negative* (TN)

The evaluation metrics used are determined by the specific goals of the sentiment analysis task. Accuracy is a simple metric that provides an overall measure of correctness, but it can be misleading when dealing with imbalanced datasets. Precision and recall provide useful information about the model's performance for positive and negative sentiments separately, such allowing researchers to focus on the desired sentiment class. The F1-score combines Precision and Recall to provide a balanced evaluation of the model's effectiveness[19].
The formulas for obtaining accuracy, precision, recall, and f1 score can be seen in formulas 3, 4, 5, and 6 [19].

Accuracy is a metric used to measure the overall performance of a classification model. It represents the ratio of correct predictions (TP and TN) to the total number of predictions (TP, TN, FP, and FN). Accuracy provides an overall assessment of how well the model is performing but may not be the most informative metric, especially if the classes are imbalanced.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (3)$$

Precision is a metric that measures the proportion of positive predictions that are actually correct.

$$Precision = \frac{TP}{(TP+FP)} \quad (4)$$

Recall, also known as sensitivity or true positive rate, is a metric that measures the proportion of actual positive instances correctly identified by the model.

$$Recall = \frac{TP}{(TP+FN)} \quad (5)$$

The F1 score is a metric that combines both precision and recall into a single value. It provides a balance between the two measures and is calculated as the harmonic mean of precision and recall.

$$F1\ Score = \frac{2\times(Recall\times Precision)}{(Recall+Precision)} \quad (6)$$

3. RESULTS AND DISCUSSION

3.1 Collecting and Dataset Labelling

Table 2. Sample data of labeled tweets

Tweet	**Label**
Sabar tunggu hasil investigasi TGIPF, Jika Tim ini bekerja professional tentu semuanya akan terbuka. @ZainudinAmali @mohmafudmd @iriawan84	Positive
Ngurusin m****k cepat, keadilan pending #UsutTuntasTragediKanjuruhan	Negative

The scraping technique is used to gather the dataset from Twitter. 6 keywords were used to find tweets related to the topic for tweets between 2 October 2022 to 30 December 2022 and using Indonesian language. The keywords are "#UsutTuntas", "#UsutTuntasTragediKanjuruhan", "Tragedi Kanjuruhan", "Aremania", "Kanjuruhan," and "TGIPF. A total of 72,502 tweets were obtained from this process. 39,891 tweets were left after duplicate tweets and those marked as spam were removed from initial result.

Then, in order to reduce the labeler's workload, the tweets have been reduced to 13,408 by taking 250 tweets per day. To ensure consistency and mitigate potential biases, a majority voting system was implemented. This means that for each tweet, the final label assigned was determined by the majority agreement among the five labelers. In cases where there was a tie, a consensus was reached through discussion or an additional vote. The

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

tweets are given the labels "positive" and "negative." There were 7,297 positive tweets and 6,016 negative tweets, according to the labeling results.

3.2 Preprocessing

At this stage, the data from the labeled tweets will be cleaned up and used as a classification. Duplicate removal, case folding, data cleaning, tokenization, stop word removal, and stemming are the stages of the preprocessing procedure[20].

Let's imagine the stage of feature preparation in more detail. It begins with several preprocessing steps aimed at refining the text data before feature extraction.

First, duplicate removal is performed, where tweets with identical IDs are identified and only one instance of each tweet is retained, while the duplicates are discarded. This ensures that each tweet is represented uniquely in the dataset.

Next, case folding is applied, which involves converting all uppercase characters in the text to their lowercase counterparts. For example, a sentence like "Bubarkan Arema yang mana?" is transformed to "bubarkan arema yang mana?". This step helps to ensure consistency and avoid treating words with different casing as distinct entities during subsequent processing.

Following case folding, data cleaning is carried out. This step involves removing punctuation marks, numbers, and emojis from the text. For instance, the resulting sentence from the previous step, "bubarkan arema yang mana?", would be further processed to remove any punctuation marks, numbers, and emojis, resulting in "bubarkan arema yang mana".

Tokenization is then performed on the cleaned text. Tokenization involves breaking down the sentences into arrays of individual words. Using the example sentence, the tokenized representation would be ["bubarkan"], ["arema"], ["yang"], ["mana"] each representing a separate word in the sentence.

The next step is stop word removal, where common words that don't carry much semantic meaning are removed from the tokenized sentences. This step helps eliminate noise and reduce the dimensionality of the data. A combination of a pre-existing stop word list, such as Sastrawi, and a self-made corpus is used to identify and remove these common words. After stop word removal, the sentence would be represented as ["bubarkan"], ["arema"], ["mana"] with the common words like "yang" being removed.

Stemming is the subsequent process, where the remaining words are transformed into their base or root form. This is done to consolidate words with similar meanings, which may appear in different inflected forms. In our example, the words "bubarkan" and "arema" would be stemmed to their respective base forms, resulting in ["bubar"], ["arema"], ["mana"].

By following these preprocessing steps, the text data is transformed and prepared for the next feature extraction phase, such as TF-IDF, where the importance and weights of the words in the tweets will be calculated. These processed and transformed features can then be utilized for various downstream tasks, such as sentiment analysis, topic modeling, or text classification.

3.3. Feature Extraction & Expansion

The feature extraction stage in this process involves utilizing TF-IDF as the weighting technique to extract features from the dataset. TF-IDF assigns weights to the terms (words) present in the tweets based on their frequency within each tweet and their frequency across the entire dataset. This results in a numerical representation, typically a float value, for each word in the dataset.

The TF-IDF values reflect the importance of each word in characterizing the content of the tweets. Words that are more frequent between individual tweets but less common across the entire dataset tend to have higher TF-IDF weights. On the other hand, words that are common across many tweets receive lower TF-IDF weights.

Once the TF-IDF feature extraction is completed, the results are expanded by incorporating feature expansion using a Wikipedia ID corpus Word2Vec model. Word2Vec is a popular technique for word embedding that captures semantic relationships between words by representing them as dense vectors. The Wikipedia ID corpus Word2Vec model enhances the feature space by providing additional context and meaning to the words.

The feature expansion process maps the TF-IDF weights of the words to the corresponding Word2Vec vectors from the Wikipedia ID corpus model. This expands the feature representation of the tweets by incorporating the semantic similarities and relationships captured in the Word2Vec vectors. The result is a more enriched and comprehensive feature for the classification training phase.

Wikipedia Indonesia corpus will be trained to produce a model using Word2Vec. Vectors that resulted by trained Wikipedia Indonesia model will be used to generate similarity score between words in the dataset in feature expansion phase.

3.4 Data split

The data of expanded features is now split into two categories: training data and test data. The test data will be utilized to check the accuracy of the model training outcomes, whereas the training data forms the basis for model training. In Table 3 show how this study used ration, 80:20, 70:30, and 60:40.

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

Table 3. Ratio split of data

Ratio	Amount of train data	Amount of test data	Total
80:20	10.649	2.663	
70:30	9.318	3.994	13.312
60:40	7.987	5.325	

80:20, 70:30, or 60:40 are the commons ratio for splitting data in the term of machine learning. It balances training and testing data, having sufficiently large training data and reasonably sized testing/evaluation data. It ensures that the model has access to enough data for learning while also providing considerable amounts for evaluation. The greater portion of the split is allocated to training data, which allows the model to learn patterns in the data. More training data increases the model's chances of capturing underlying patterns and generalizing well to new data.

3.5 Oversampling

The imbalance data is oversampled. After the data has been split according to the ratio stated in 3.1. The test data is oversampled using SMOTE, which results in the same class frequency. Table 4 displays a graph of the quantity of test data before SMOTE and after.

Table 4. Amount of test data before and after oversample

Ratio	Before		After	
	Positive	Negative	Positive	Negative
80:20	7.486	3.163	7.486	7.486
70:30	6.562	2.756	6.562	6.562
60:40	5.638	2.349	5.638	5.638

Before SMOTE, the test data likely had a significant class imbalance, with the majority class having a positive class frequency compared to the negative class. This imbalance may affect the training model’s ability to accurately classify instances from the negative class. However, after applying SMOTE, in Table 4 shows that the class frequencies have been adjusted to achieve a balanced frequency distribution. The samples generated by SMOTE have effectively increased the quantity of instances belonging to the minority class, aligning it with the majority class. The oversampling process using SMOTE ensures that the model appears to an equal quantity of instances from each class during the training phase by balancing the class distribution. This helps to avoid bias toward the majority class and allows for a more accurate assessment of the model's performance on both classes.

3.6 Classification

Implementation of the Support Vector Machine (SVM) classification will use the rbf kernel, linear, and polynomial based on determined test ratios.

3.6.1 RBF kernel

On data with RBF kernel training, the test results are as in Table 5.

Table 5. RBF kernel classification result

	Non-oversampling		Oversampling	
Ratio	***Accuracy***	***Macro F1 Score***	***Accuracy***	***Macro F1 Score***
80:20	0,87	0,84	0,87	0,84
70:30	0,87	0,84	0,88	0,86
60:40	0,86	0,83	0,87	0,85

According to Table 5, when using the RBF kernel, the best results are obtained when using a 70:30 ratio with oversampling. Even so, the obtained result, both oversampling or without oversampling, relative stable. The table demonstrates the performance of the RBF kernel. The results indicate that the most favorable outcome is obtained when using a 70:30 ratio combined with oversampling. Nonetheless, it is noteworthy that both oversampling and non-oversampling approaches yield relatively small difference, suggesting that the RBF kernel is robust and less sensitive to the choice of training and testing ratios.

3.6.2 Linear Kernel

On data with linear kernel training, the test results are as in Table 6.

Table 6. Linear kernel classification result

	Non-oversampling		Oversampling	
Ratio	***Accuracy***	***Macro F1 Score***	***Accuracy***	***Macro F1 Score***
80:20	0,87	0,83	0,86	0,84

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

70:30	0,87	0,84	0,86	0,83
60:40	0,86	0,82	0,86	0,84

According to Table 6, when using a linear kernel, the best results are obtained when using a ratio of 80:20 and 70:30 without oversampling. The accuracy results are not too different from the others. The experimental results from Table 6 demonstrate that a linear kernel performs exceptionally well when used with a ratio of 80:20 or 70:30 for training and testing, without the application of oversampling techniques. The obtained accuracy results in these configurations are quite similar to the other approaches, indicating that the linear kernel consistently delivers competitive performance.

3.6.3 Polynomial kernel

On data with polynomial kernel training, the test results are as in Table 7.

Table 7. Polynomial kernel classification result

	Non-oversampling		Oversampling	
Ratio	***Accuracy***	***Macro F1 Score***	***Accuracy***	***Macro F1 Score***
80:20	0,82	0,74	0,84	0,81
70:30	0,80	0,73	0,84	0,81
60:40	0,79	0,70	0,83	0,79

According to Table 7, when using polynomial kernels, the best results are obtained when using ratios of 80:20 and 70:30 with oversampling. The accuracy results are not significantly different from the others. The findings suggest that when employing polynomial kernels, the best outcomes are achieved when using ratios of 80:20 and 70:30, along with oversampling. However, the accuracy results in these cases do not differ significantly from the other methods. This implies that polynomial kernels can still yield satisfactory accuracy even without oversampling, and the improvement gained through oversampling is relatively modest.

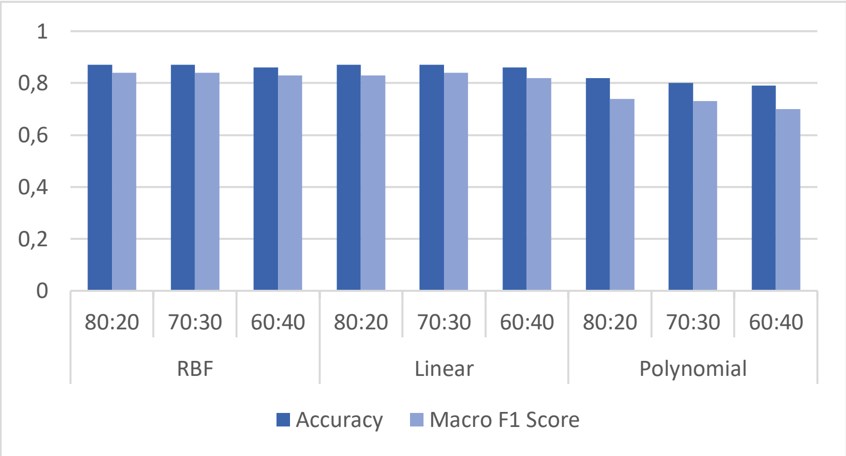

Figure 1. Comparison of non-oversampling training performance

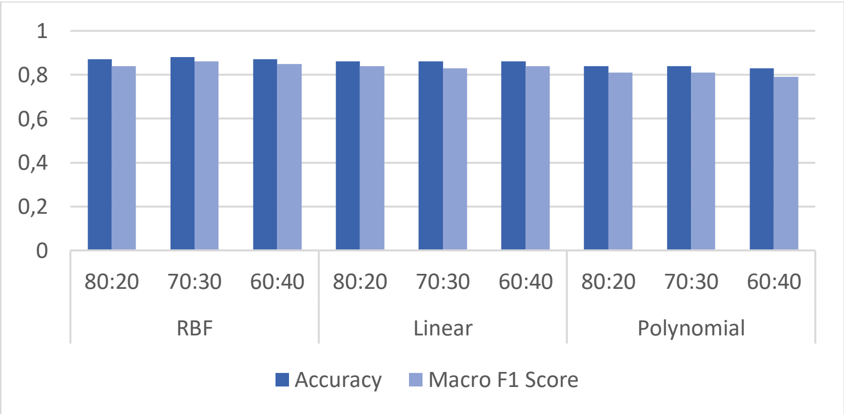

Figure 2. Comparison of oversampling training performance

Figure 1 and Figure 2 provide a side-by-side comparison of different methods used in the Support Vector Machine (SVM) model training stage. These figures offer insights into the performance and effectiveness of each method, allowing for a comprehensive evaluation of their impact on the dataset.

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

4. CONCLUSION

The testing phase revealed that the inclusion of oversampling did not have a significant impact on the dataset's accuracy. Therefore, for the train model, the RBF kernel, a split ratio of 70:30, and oversampling were used to obtain the highest accuracy results. According to Table 4, it can be observed that the non-oversampled data initially displayed an imbalance between the positive and negative classes. However, during the classification phase, all kernels resulted a sufficiently high F1 score. This result, portraited by F1 score, indicates that the labeled tweets in the dataset are satisfactory, as the models were able to effectively capture and classify the sentiment expressed in the tweets despite the initial class imbalance. In this study, oversampled data and RBF kernel has biggest average of Macro F1 score. According to these findings, the accuracy while using oversampling is fairly constant across all split ratios and kernels, with little noticeable variation. The results obtained from the experiments indicate that the classification accuracy remains stable even without the utilization of oversampling techniques. This suggests that the model is robust enough to handle the class imbalance present in the dataset. However, a notable observation is that when tested with a polynomial kernel, the range between the highest and lowest accuracy values is 0.05, which is larger compared to prior tests where the difference was typically around 0.01. This difference in the range of accuracy scores between the polynomial kernel and the other kernels implies that the polynomial kernel may be more sensitive to the class imbalance within the dataset. The larger variation in accuracy could be attributed to the polynomial kernel's ability to capture more complex patterns and interactions among features, potentially leading to varying performance across different samples. For further study, we expected to add more class, especially neutral class, in the training data, furthermore apply of various preprocessing method are could be tested to see if there any effect to the classification.

REFERENCES

[1] V. Febrianto, "Death count in Kanjuruhan tragedy climbs to 135," ANTARA News, Oct. 24, 2022. Accessed: Jan. 20, 2023. [Online]. Available: https://en.antaranews.com/news/256465/death-count-in-kanjuruhan-tragedy-climbs-to-135

[2] Hermanto, A. Y. Kuntoro, T. Asra, E. B. Pratama, L. Effendi, and R. Ocanitra, "Gojek and Grab User Sentiment Analysis on Google Play Using Naive Bayes Algorithm And Support Vector Machine Based Smote Technique," J Phys Conf Ser, vol. 1641, p. 012102, 2020, doi: 10.1088/1742-6596/1641/1/012102.

[3] M. Guia, R. Silva, and J. Bernardino, "Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis," Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2019, doi: 10.5220/0008364105250531.

[4] S. Al-Saqqa and A. Awajan, "The Use of Word2vec Model in Sentiment Analysis," Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control, pp. 39–43, 2019, doi: 10.1145/3388218.3388229.

[5] M. Matuq Ashi, M. Ahmed Siddiqui, and F. Nadeem, "Pre-trained Word Embeddings for Arabic Aspect-Based Sentiment Analysis of Airline Tweets," in The International Conference on Advanced Intelligent Systems and Informatics, A. E. Hassanien, M. F. Tolba, K. Shaalan, and A. T. Azar, Eds., Cairo: Springer International Publishing, 2018. doi: 10.1007/978-3-319-99010-1.

[6] M. Sahbuddin and S. Agustian, "View of Support Vector Machine Method with Word2vec for Covid-19 Vaccine Sentiment Classification on Twitter," Journal of Informatics and Telecommunication Engineering, vol. 6, no. 1, 2022, doi: 10.31289/jite.v6i1.7534.

[7] J. Xue, X. Ban, H. Guo, and X. Zhu, "Sentiment Analysis Based on Weibo Comments," in 2018 13th World Congress on Intelligent Control and Automation (WCICA), Changsa, China: IEEE, Jul. 2018, pp. 1166–1171. doi: 10.1109/WCICA.2018.8630471.

[8] Naufal Adi Nugroho and Erwin Budi Setiawan, "Implementation Word2Vec for Feature Expansion in Twitter Sentiment Analysis," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 5, no. 5, pp. 837–842, Oct. 2021, doi: 10.29207/resti.v5i5.3325.

[9] M. A. Fauzi, "Word2Vec model for sentiment analysis of product reviews in Indonesian language," International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 1, p. 525, Feb. 2019, doi: 10.11591/ijece.v9i1.pp525-530.

[10] M. Rizki, "Analisis Sentimen Masyarakat Terhadap Vaksin COVID-19 Menggunakan Metode Support Vector Machine Pada Media Sosial Twitter," Uin-suska.ac.id, 2022, [Online]. Available: http://repository.uin-suska.ac.id/58497/

[11] R. Nooraeni, H. D. Sariyanti, A. F. F. Iskandar, S. F. Munawwaroh, S. Pertiwi, and Y. Ronaldias, "Analisis Sentimen Data Twitter Mengenai Isu RUU KPK Dengan Metode Support Vector Machine (SVM)," Paradigma - Jurnal Komputer dan Informatika, vol. 22, no. 1, pp. 55–60, Mar. 2020, doi: 10.31294/p.v22i1.6869.

[12] "Rekor Kematian Kedua di Dunia, Tragedi Kanjuruhan Lampaui Hillsborough," CNN Indonesia, Oct. 02, 2022. https://www.cnnindonesia.com/olahraga/20221002070354-142-855202/rekor-kematian-kedua-di-dunia-tragedi-kanjuruhan-lampaui-hillsborough (accessed Jan. 20, 2023).

[13] L. Bing, Sentiment Analysis Mining Opinions, Sentiments, and Emotions, 2nd ed. Cambridge University Press, 2020. doi: 10.1017/9781108639286.002.

[14] D. Sharma, M. Sabharwal, V. Goyal, and M. Vij, "Sentiment Analysis Techniques for Social Media Data: A Review," 2020, pp. 75–90. doi: 10.1007/978-981-15-0029-9_7.

[15] Suyanto, Data Mining untuk Klasifikasi dan Klasterisasi Data. Bandung: Informatika Bandung, 2017. Accessed: Jan. 25, 2023. [Online]. Available: https://opac.perpusnas.go.id/DetailOpac.aspx?id=1069411#

[16] B. Santosa, "Tutorial Support Vector Machines," Surabaya, 2015.

[17] S.-W. Kim and J.-M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," Human-centric Computing and Information Sciences, vol. 9, no. 1, p. 30, Dec. 2019, doi: 10.1186/s13673-019-0192-7.

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, June 2023 Page: 219−227
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3612

[18] X. Huang, C.-Z. Zhang, and J. Yuan, "Predicting Extreme Financial Risks on Imbalanced Dataset: A Combined Kernel FCM and Kernel SMOTE Based SVM Classifier," Comput Econ, vol. 56, no. 1, pp. 187–216, Jun. 2020, doi: 10.1007/s10614-020-09975-3.

[19] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, p. 6, Dec. 2020, doi: 10.1186/s12864-019-6413-7.

[20] S. Pradha, M. N. Halgamuge, and N. Tran Quoc Vinh, "Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data," in 2019 11th International Conference on Knowledge and Systems Engineering (KSE), IEEE, Oct. 2019, pp. 1–8. doi: 10.1109/KSE.2019.8919368.