THE 1ST JAMBI INTERNATIONAL CONFERENCE ON ENGINEERING, SCIENCE AND TECHNOLOGY (#1 JICEST)

Performance Evaluation of Various Machine Learning Approaches for Sentiment Analysis of Lampung Robusta Coffee

Yodhi Yuniarthe[1,2], Admi Syarif*[1], Sumaryo[3], Warsito[4], and M. I. Ardiansyah[1]

*Corresponding author
*ORCHID IDs: https://orcid.org/0000-0003-3316-0388

[1)] *PhD student, Department of Computer Science-*
Faculty of Mathematics and Natural Sciences, Lampung University, Indonesia, 35145
[2)] *Department of Informatics Faculty of Computer,*
Indonesia Mitra University, Indonesia
[3)] *Department of Agribusiness-Faculty Agriculture,*
Lampung University, Lampung, Indonesia
[4)] *Department of Physics-Faculty of Mathematics and Natural Sciences,*
Lampung University, Indonesia

*email: admi.syarif@fmipa.unila.ac.id

Abstract

Coffee is known as the major commodities between countries. There are two world-famous varieties of coffees, arabica and robusta. Robusta coffee is a drink that comes from the seeds of the coffee plant and is processed into powder. One of the centers of robusta coffee production in Indonesia is Lampung. The characteristics of Lampung Robusta Coffee is a fragrant aroma and bitter taste. The distinctive taste of this Coffee triggers people to taste the coffee. This paper aims to explore the sentiment analysis regarding the Lampung Robusta Coffee on YouTube. This research utilizes Artificial Intelligence (AI) approaches including the Support Vector Machine (SVM), Naïve Bayes, and K-nearest neighbour (KNN) algorithms. We also consider the balanced and unbalanced datasets and adopt data balancing approach, the Synthetic Minority Over-sampling Technique (SMOTE). The experimental results show that the SVM algorithm with data balancing provides the best accuracy. Thus, this research would helps the public make decisions regarding the future marketing and development of coffee products.

Keywords: *Sentiment Analysis, Support Vector Machine, Naïve Bayes, K-Nearest Neighbour, Lampung Robusta Coffee*

1. INTRODUCTION

Coffee is a drink that has a special place in the culture and daily lives of many people worldwide. The Robusta coffee variety from the Coffee Canephora plant has become essential to the global coffee industry. In Indonesia, one of the areas famous for robust coffee production is Lampung [1]. The quality and taste of Lampung Robusta Coffee have attracted the attention of many coffee lovers and stakeholders in the coffee industry [2]. In today's digital era, people's opinions and views about coffee products and brands can easily be found on various social media platforms, websites, and online forums [3]. Sentiment analysis of Lampung Robusta Coffee is essential to understand how people respond to and evaluate this product [4]. This can provide valuable insights for coffee producers, the government, and other stakeholders to improve the quality and acceptance of Lampung Robusta Coffee in the market [2,3].

In this research, the use of Machine Learning technology has become an effective tool in exploring public opinion and sentiment on a large scale [5]. Using Machine Learning-based sentiment analysis methods can automate the collection and evaluation of general views of Lampung Robusta coffee with high accuracy [7,8]. This application attempts to understand consumers' preferences and feelings towards this coffee deeply. This research analyzes sentiment toward Lampung Robusta coffee using a machine-learning approach [8]. Collecting data from YouTube social media, product reviews, and forum discussions, then applying Machine Learning algorithms to classify sentiment into positive and negative [9].

This research uses the SVM method with TF-IDF as feature extraction. This is different from previous research, which applied data balancing with SMOTE. SVM has the advantage of finding the best hyperplane to separate two classes in feature space and using a Structural Risk Minimization (SRM) strategy for more optimal results [10]. The TF-IDF method weights the relationship between words (terms) and documents. The weighting is implemented after the dataset has gone through the pre-processing stage. This stage aims to find out the value of the extracted root words. This important crucial is converted into a vector representing the word so the system recognizes it [11]. SMOTE is a statistical method that balances data between minority and majority classes [12]. SMOTE uses the K-nearest neighbour algorithm to create synthetic data based on existing data.

Previous research [13] explained sentiment analysis of 2019 presidential candidates on YouTube social media using a lexicon-based method with 200 data, resulting in an accuracy of 74.00% using SVM. Research [14] regarding Sentiment Analysis of the School Zoning System on YouTube using K-Nearest Neighbour with 200 data resulted in an accuracy of 62.625%. Research [15] explains sentiment analysis about Ambulance patrol volunteers on YouTube with 600 data results in an accuracy of 66.67% on Naïve Bayes and 64.98% on Decision Tree. Further research [16] Sentiment Analysis of User Reviews of COVID-19 information using an application with 6000 data. The SVM results have an accuracy of 76.5%, followed by NBC with 72.3% and KNN with an accuracy of 59.1%. Unlike previous research, this implements data balancing using SMOTE with SVM, Naïve Bayes, and KNN methods. Based on the accuracy results in this study of 86.69%, it shows that implementing data balancing with SMOTE provides better accuracy results.

This research will analyze and classify sentiment data (positive or negative) on YouTube regarding Public Opinion towards Lampung Robusta Coffee using the SVM, Naive Bayes, and KNN methods in sentiment analysis. Next, we will compare the performance levels of the three methods with the imbalanced and balanced datasets in this research. The results of this research will provide valuable insight into public perceptions of Lampung Robusta Coffee. They can assist in decision-making regarding the future marketing and development of this coffee product. The structure of this research is as follows: section 2 explains related work regarding sentiment analysis, section 3 explains the stages of the method and describes training the model in detail, section 4 presents the results and discussion using the SVM classification model, the results of tests and experiments that have been implementation, in Section 5 contains conclusions from the results of this research.

2. RELATED WORKS

Several studies have been conducted regarding sentiment analysis using Machine Learning methods. Table 1 shows various sentiment analyses using SVM, Naive Bayes, and KNN methods. Based on the information in Table 1, previous research used an imbalanced dataset; this research is different; it implemented data balancing using SMOTE. The testing is implemented with two datasets, the imbalanced and balanced datasets. Based on the results of this research, it shows that the balanced dataset is better than the imbalanced dataset. Research [13] with automatic labelling shows a decrease in accuracy. In contrast to this research, it uses manual labelling to obtain more objective sentiment labels.

Table 1. Previous Approaches to Sentiment Analysis

No	Authors	Data	Metode	Results (Accuracy)
1	Gaol et al., 2019 [13]	Amount of data: 200 Source: YouTube	SVM Lexicon Based	SVM: 74.00% SVM + LB: 73.68%
2	Anggaini & Tursina, 2019 [14]	Amount of data: 160 Source: YouTube	KNN	KNN: 62.625%
3	Wahid & Saputri, 2022 [15]	Amount of data: 600 Source: YouTube	NB DT	NB: 66.67% DT: 64.98%
4	Salma & Silfianti, 2021 [16]	Amount of data: 6000 Source: Playstore	SVM, NB, KNN	SVM: 76.50% NB: 72.30% KNN: 59.10%
5	Rahat et al.,[17]	Amount of data: 10000 Source: Twitter	SVM, NB	SVM: 82.48% NB: 76.56%
6	This Research	Amount of data: 203 Source: YouTube	SVM, NB, KNN	SVM+Imbalanced: 79.35% SVM+Balanced: 86.89% NB+Imbalanced: 70.73% Balanced: 80.85% KNN+Imbalanced: 73.91% KNN+Balanced:79.41%

3. METHODS

Figure 1 shows the flow of research in this paper. The first process is to collect data using Twitter. The next stage was labeling the data, done manually by giving a sentiment value, and then the data pre-processing was the text cleaning stage. In this study, word weighting implements TF-IDF, then data distribution is implemented using holdout validation. The next step uses the sampling method to balance the data. The final process in this study is the performance evaluation of the algorithm using the Confusion Matrix and comparing the three algorithms.

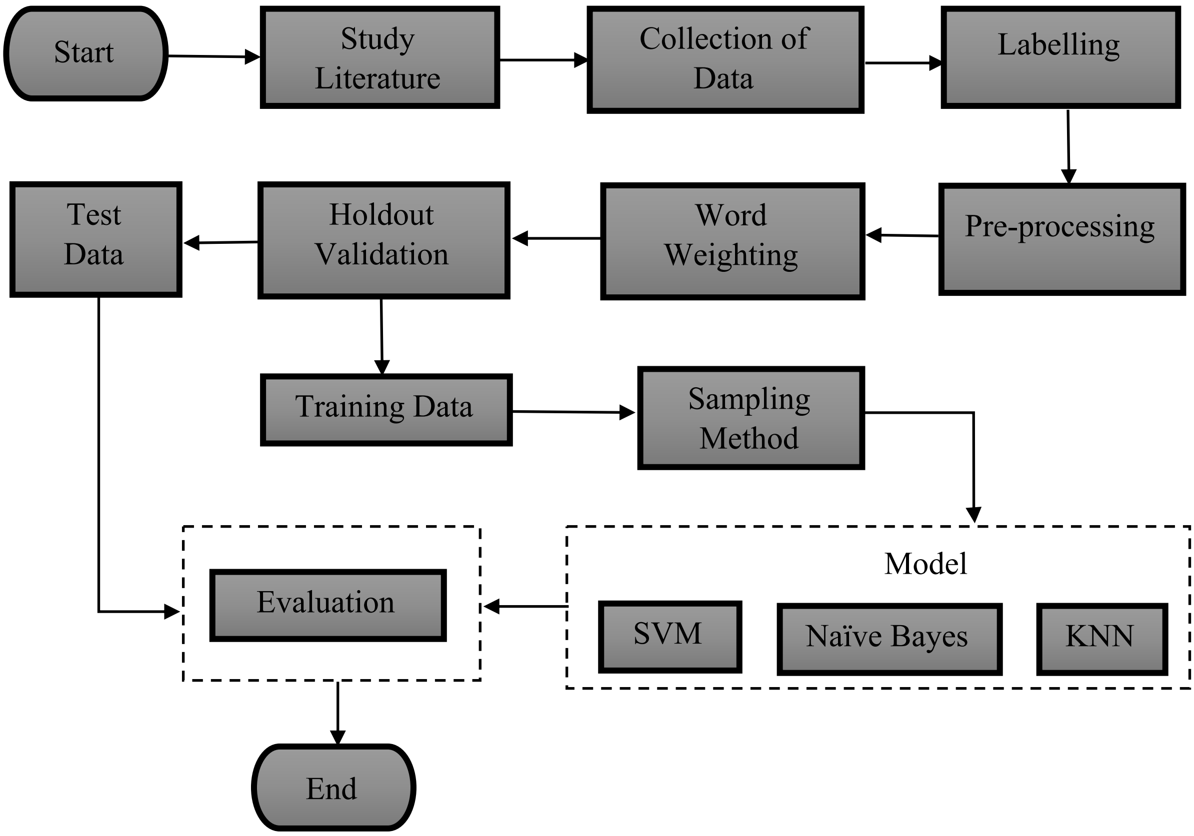

Figure 1. Flow in Research

3.1. Collection of Data

Data collection uses YouTube for sentiment analysis on public opinion towards Lampung Robusta Coffee. The data collected is in Indonesian text, using the Python 3 programming language as a development tool [18]. The YouTube API is used to obtain comments using the video URL regarding the discussion of Lampung Robusta Coffee. Comment data is collected and stored in table form with text attributes from August 2023 to September 2023 [19]. The data obtained was 942, and then filtering and labelling were implemented manually so that the data used was 203.

3.2. Labelling Data

Labelling of the dataset was done manually by three labellers [20]. In the labelling process, two main labellers are responsible for determining the sentiment value of each tweet contained in the dataset. The label supporters will also choose the final judgment if the two main labellers have different judgment sentiments. In this study, Polaris is used as a measure of sentiment. Positive polarity is used for data containing positive sentiments, while negative polarity is used for data containing negative sentiments. 145 data, or 71.4%, have a positive sentiment, and 58 data, or 28.6%, have a negative sentiment. Table 2 shows the results of the manual labelling.

Table 2. Manual Labelling Results

Comment	Sentiment
Inget waktu di Lampung..metik sendiri di ongsreng sendiri..di tumbuk sendiri..waduh nikmat banget..	Positive
Robusta lampung sekarang rasanya pahit kaya kopi biasa	Negative

3.3. Pre-processing

Data pre-processing is implemented after the data collection stage to clean and prepare the data before further analysis. Several processes are carried out at this stage: cleaning, case folding, tokenizing, normalization, stopword removal, and stemming [21]. The following is a more detailed explanation of the functions in dataset processing. Cleaning is the process of removing noise or interference from raw text data. At this stage, we will process all research text data in several steps, such as deleting usernames, URLs, hashtag signs (#), mention signs (@), numbers, punctuation marks, HTML characters, and other symbols. Case folding is a process in text processing that involves changing capital letters to lowercase letters or vice versa. The main goal is to equalize the representation of words with different capitalization letters, thus facilitating the analysis and processing of text in natural language processing (Natural Language Processing). Tokenizing is breaking text into smaller units called "tokens." Normalization is the stage of changing words, not by Enhanced Spelling (EYD) or the standard language used. The normalization process ensures that the text's words follow spelling standards. Stopwords are a collection of words that appear frequently in a text, but removing them will not change the meaning of the text. In the stopword removal stage, stopwords are identified and removed from the text. The purpose of stemming is to minimize the variation of terms with the same root word, thus improving the quality of data and information found in information searches.

3.4. Word Weighting

In this research , we adopt the TF-IDF (Term Frequency-Inverse Document Frequency) method. The TF-IDF method determines the level of importance of a word or group of words in a dataset. This weighting process involves two main components: Term Frequency (TF) and Inverse Document Frequency (IDF) [22]. Term Frequency (TF) measures how often certain words appear in a document. The more often the word appears, the higher the weight. Inverse Document Frequency (IDF) measures the word's uniqueness level in the entire dataset. Using the TF-IDF method, words frequently appearing in a document but rarely appearing in other documents will have a high weight [23]. Conversely, words that appear frequently in the entire dataset will have a low weight due to their lack of uniqueness. The result of this word weighting will be a numerical representation of each document in the dataset [24]. These representations can then be used as input for the machine learning algorithms defined in this study. The algorithm will use word weights to identify patterns and perform further classification or analysis of the data.

3.5. Holdout Validation

Figure 2 shows the dataset distribution in this study into two parts, namely training data and testing data. The process of data distribution is done through holdout validation. The division of data training and data testing into five scenarios: Scenario 1 is data training by 50% and data testing by 50%; Scenario 2 is data training by 55% and data testing by 45%; Scenario 3 is data training by 60% and data testing by 40%, scenario four training data by 65% and data testing by 35%, and scenario 5 data training by 70% and data testing by 30%. Table 3 shows the results of the distribution of holdout validation with 5 data distribution scenarios.

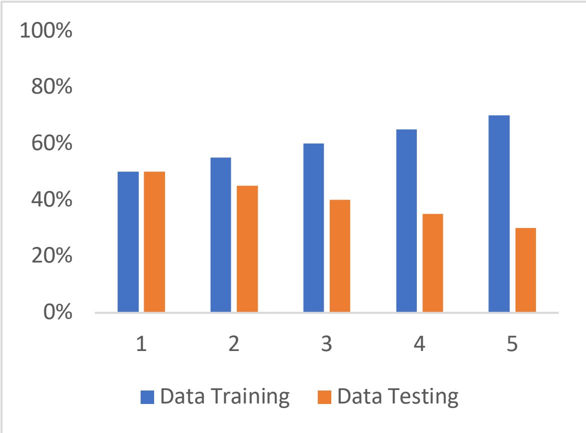

Figure 2. Holdout Validation Scenario

Table 3. Imbalanced Dataset Scenario

	Scenario 1		**Scenario 2**		**Scenario 3**		**Scenario 4**		**Scenario 5**	
Sentiment	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
	50%	**50%**	**55%**	**45%**	**60%**	**40%**	**65%**	**35%**	**70%**	**30%**
Positive	72	73	79	66	86	59	94	51	101	41
Negative	29	29	32	26	35	23	37	21	44	17
Total	101	102	111	92	121	82	131	72	142	61

3.6. Sampling Method

For the data, we used 203 data from YouTube, to take this data we used severed keyword like kopi Lampung, Robusta Lampung. The data in this study has an unequal number of positive and negative sentiments. The data in the research dataset shows more positive sentiment. The number of positive sentiments is 145, and the number of negative sentiments is 58. Based on this sentiment comparison, the researchers balanced the data using the Synthetic Minority Over-sampling Technique (SMOTE) method. The Synthetic Minority Over-sampling Technique (SMOTE) method is applied to overcoming class imbalances. This technique synthesizes a new sample from the minority class to balance the dataset by re-sampling the minority class sample. The library used to balance data is imbalanced-learn [25]. The data to be balanced is the training data between positive and negative, while the testing data is still adjusted according to the distribution of scenarios. Table 3 shows the results of dividing the data by adding the Synthetic Minority Over-sampling Technique (SMOTE) to the balanced dataset.

Table 4. Balanced Dataset Scenario

	Scenario 1		**Scenario 2**		**Scenario 3**		**Scenario 4**		**Scenario 5**	
Sentiment	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
	50%	**50%**	**55%**	**45%**	**60%**	**40%**	**65%**	**35%**	**70%**	**30%**
Positive	72	73	79	66	86	59	94	51	101	44
Negative	72	29	79	26	86	23	94	21	101	17
Total	144	102	158	92	172	82	188	72	202	61

3.7. Sentiment Classification

This study's sentiment classification stages use three algorithms: Support Vector Machine (SVM), Naïve Bayes, and K-nearest neighbour. The data used for sentiment classification is tweet data processed in the previous dataset processing stage. The dataset was divided into training and test data in the early stages. The training data helps train the classification algorithm, while the test data helps evaluate the performance of the modified algorithm. The classification results in the form of positive and negative sentiment predictions were evaluated using evaluation metrics such as the Confusion Matrix, Accuracy, Precision, Recall, and F1-Score. The explanation for the classification of sentiment is as follows.

3.7.1. Support Vector Machine (SVM)

SVM, a machine learning approach, operates within a hypothesis space reliant on linear functions in a high-dimensional feature space [26]. It undergoes training via optimization-based learning algorithms, and its accuracy hinges significantly on the choice of kernel function and training parameters. SVM can be categorized into two main types: Linear SVM and Non-linear SVM. In the case of Linear SVM, it separates data by establishing a hyperplane with a soft margin that separates different classes. At the same time, Non-linear SVM employs the kernel trick to map data into a higher-dimensional space for better separation [27]. The core idea in SVM revolves around identifying the optimal separator within the hyperplane. This entails searching for the optimal value of the function f(x) along the margin of the hyperplane.

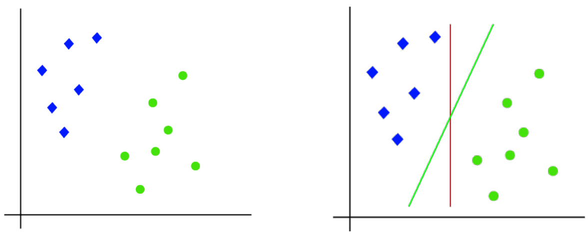

Figure 3. Linear SVM [27]

3.7.2. Naïve Bayes

Naive Bayes classification relies on maximum likelihood estimation to categorize samples into their most probable classes [28]. In this context, if we have an input vector, X, comprising features and a corresponding class label, Y, the notation P(Y|X) represents Naive Bayes. This notation signifies the probability of observing class label Y given the features X, often referred to as the posterior probability for Y. The initial possibility, P(Y), representing the prior probability, is also factored into the classification process. During training, the task involves estimating the posterior probabilities (P(Y|X)) for each combination of X and Y using the information from the training data [29].

3.7.3. K-nearest neighbour (K-NN)

K-nearest neighbour is an algorithm that predicts a label from data based on the nearest neighbours. The K-nearest neighbour algorithm works by determining the value of K and then calculating the distance of the new data with the training data. The three generally used distance metrics are the Euclidean Distance, Manhattan Distance, and Minkowsky Distance. After calculating the distance, look for K-neighbours close to the new data. The simple principle of this method is "Data that will predict whether it belongs to the positive or negative class"[30].

3.8. Evaluation

Evaluation in this study includes a comparison of sentiment analysis using three algorithms, namely Support Vector Machine (SVM), Naïve Bayes, and K-nearest neighbour (KNN), with imbalanced and balanced datasets. Tests on each dataset aim to determine Accuracy, Precision, Recall, and F1-Score value changes. After the classification process using Support Vector Machine (SVM), Naïve Bayes, and K-nearest neighbour (KNN), comparisons were made between the classification results of the three algorithms. This evaluation produces metrics such as the Confusion Matrix and Classification Report, including accuracy, precision, recall, and f1-score. The Confusion Matrix provides information about the predicted classification results from the system compared to the actual data that has been labelled manually. This Confusion Matrix consists of four parts, namely True Positive, False Positive, True Negative, and False Negative, describing how the algorithm can correctly recognize positive and negative sentiments. Classification system performance is generally calculated using data in the Confusion Matrix table, as shown in Table 5 below.

Table 5. Confusion Matrix

Actual	**Prediction**	
	Negative	**Positive**
Negative	TN (True Negative)	FP (False Positive)
Positive	FN (False Negative)	TP (True Positive)

The True Negative (TN) value is the number of negative data detected correctly, while False Negative (FN) is positive data seen as negative. True Positive (TP) values are positive data that is correctly detected, while False Positive (FP) are negative data that is detected as positive. The Confusion Matrix table is used to measure the performance of a classification method by calculating the value of accuracy, precision, recall, and f1-score [9].

a. Accuracy

Accuracy is the prediction of true positive and true negative data from the entire dataset.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \, x \, 100\% \qquad (1)$$

b. Precision

Precision is a measure of the accuracy of the results of a model. Precision is the ratio of the correct positive prediction to the overall positive predicted outcome.

$$Presisi = \frac{TP}{TP + FP} \, x \, 100\% \tag{2}$$

c. Recall

Recall is a measure of the completeness of a model. Recall is a comparison of correct positive predictions compared to data that has positive labels.

$$Recall = \frac{TP}{TP + FN} \, x \, 100\% \tag{3}$$

d. F1-Score

F1-Score is a comparison of the average value resulting from precision and recall.

$$F1\,Score = 2\,x\,\frac{(recall\ x\ precision)}{(recall + precision)} \tag{4}$$

4. RESULTS AND DISCUSSION

4.1. Results

4.1.1. Support Vector Machine (SVM)

The classification model used is the Support Vector Machine (SVM). SVM classification modelling uses the Sklearn library with Support Vector Classification (SVC) with the dataset divided into 2: training and testing data. The dataset is divided according to the schema to determine the most optimal scenario composition. Then, testing uses several kinds of Support Vector Machine kernels, namely Linear, Radial Basis Function (RBF), Polynomial, and Sigmoid, to find the kernel with the highest level of accuracy. This study uses the Radial Basis Function (RBF) kernel. The Confusion Matrix evaluates SVM classification performance for calculating Accuracy, Precision, Recall, and F1-Score values. The data used are imbalanced datasets and balanced datasets. The results of testing using 5 data sharing scenarios with imbalanced datasets and balanced datasets in the Support Vector Machine algorithm are in Table 7.

Based on Table 7, the Support Vector Machine using data balancing techniques (SMOTE) can provide best accuracy results of 86.89%. Precision is 86.00%, recall is 97.73%, and f1-score is 91.49% in the 5th scenario. Whereas for the imbalanced dataset, the accuracy results with the Support Vector Machine in the 2nd scenario are accuracy is 79.35%, precision is 78.31%, recall is 98.48%, and f1-score is 87.25%. The Confusion Matrix for the best scheme for the imbalanced dataset for the 2nd scenario and the balanced dataset for the 5th scenario can be seen in Table 6 below.

Table 6. Confusion Matrix SVM

Dataset Imbalanced			Dataset Balanced		
Actual	**Prediction**		**Actual**	**Prediction**	
	Negative	**Positive**		**Negative**	**Positive**
Negative	8	18	Negative	10	7
Positive	1	65	Positive	1	43

Table 6 is an imbalanced dataset showing the results of the Confusion Matrix in 2nd scenario with the Support Vector Machine method. A total of 8 data are correctly predicted as negative data (True Negative). 65 data are rightly expected as positive data (True Positive). As many as 18 data are mis predicted as positive data. It should be negative data (False Positive). As many as 1 data are mis predicted as negative data. It should be positive (False Negative). The balanced dataset displays the results of the Confusion Matrix in the 5th scenario with the Support Vector Machine method. A total of 10 data are correctly predicted as negative data (True Negative). 43 data are rightly expected as positive data (True Positive). As many as 7 data were mis predicted as positive data. It should have been negative data (False Positive). As many as 1 data were mis predicted as negative data. It should have been positive data (False Negative).

Table 7. Compare Performance of SVM (Percentage)

Scenario	Dataset Imbalanced				Dataset Balanced			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
1	77.45	76.60	98.63	86.23	78.43	80.72	91.78	85.90
2	**79.35**	**78.31**	**98.48**	**87.25**	84.78	86.11	93.94	89.86
3	78.05	77.33	98.31	86.57	79.27	82.81	89.83	86.18
4	76.39	75.76	98.04	85.47	84.72	83.33	98.04	90.09
5	77.05	76.79	97.73	86.00	**86.89**	**86.00**	**97.73**	**91.49**

4.1.2. Naïve Bayes

The Naïve Bayes classification with the dataset is divided into 2: training data and testing data. The dataset is divided according to the schema to determine the most optimal schema composition. Then, the test uses alpha and binarized parameters to find the parameters with the highest level of accuracy. This study uses the parameters alpha = 1 and binarize = 0.0. The results of testing using 5 data sharing scenarios with imbalanced datasets and balanced datasets in the Naïve Bayes algorithm are in Table 9.

Based on Table 9, the Naïve Bayes using data balancing techniques (SMOTE) can provide best accuracy results of 80.56%. Precision is 81.36%, recall is 94.12%, and F1-score is 87.27% in the 4th scenario. Whereas for the imbalanced dataset, the accuracy results with the Naïve Bayes in the 3rd scenario are 70.73%, precision 72.15%, recall 96.61%, and f1-score 82.61%. The Confusion Matrix for the best scheme for the imbalanced dataset for the 3rd scenario and the balanced dataset for the 4th scenario can be seen in Table 8 below.

Table 8. Confusion Matrix Naive Bayes

Dataset Imbalanced			Dataset Balanced		
Actual	**Prediction**		**Actual**	**Prediction**	
	Negative	**Positive**		**Negative**	**Positive**
Negative	1	22	Negative	10	11
Positive	2	57	Positive	3	48

Table 8 is an imbalanced dataset showing the results of the Confusion Matrix in the 3rd scenario with the Naïve Bayes method. A total of 1 data are correctly predicted as negative data (True Negative). 57 data are rightly expected as positive data (True Positive). As many as 22 data are mis predicted as positive data. It should be negative data (False Positive). As many as 2 data are mis predicted as negative data. It should be positive (False Negative). The balanced dataset displays the results of the Confusion Matrix in the 4th scenario with the Naïve Bayes method. A total of 10 data are correctly predicted as negative data (True Negative). 48 data are rightly expected as positive data (True Positive). As many as 11 data were mis predicted as positive data. It should have been negative data (False Positive). As many as 3 data were mis predicted as negative data. It should have been positive data (False Negative).

Table 9. Compare Performance of Naive Bayes (Percentage)

Scenario	**Dataset Imbalanced**				**Dataset Balanced**			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
1	69.61	71.00	97.26	82.08	80.39	80.46	95.89	87.50
2	70.65	71.91	96.97	82.58	77.17	77.78	95.45	85.71
3	**70.73**	**72.15**	**96.61**	**82.61**	79.27	81.82	91.53	86.40
4	69.44	71.01	96.08	81.67	**80.56**	**81.36**	**94.12**	**87.27**
5	70.49	72.41	95.45	82.35	80.33	80.77	95.45	87.50

4.1.3. K-nearest neighbour (K-NN)

The K-nearest neighbour classification with the dataset is divided into 2: training data and testing data. The dataset is divided according to the schema to determine the most optimal schema composition. Then, the test uses the n neighbours, weight, and p parameters to find the parameter with the highest level of accuracy. In this study, the parameter n neighbours is 5, weight is distance, and p is 1. The results of testing using 5 data sharing scenarios with imbalanced datasets and balanced datasets in the K-nearest neighbour algorithm are in Table 11.

Based on Table 11, the K-nearest neighbour using data balancing techniques (SMOTE) can provide best accuracy results of 79.41%. Precision is 84.21%, recall is 87.69%, and F1-score is 85.91% in the 1st scenario. Whereas for the imbalanced dataset, the accuracy results with the K-nearest neighbour in the 2nd scenario are accuracy is 73.91%, precision is 75.61%, recall is 93.94%, and f1-score is 83.78%. The Confusion Matrix for the best scheme for the imbalanced dataset for the 2nd scenario and the balanced dataset for the 1st scenario can be seen in Table 11 below.

Table 6. Confusion Matrix K-NN

Dataset Imbalanced			**Dataset Balanced**		
Actual	**Prediction**		**Actual**	**Prediction**	
	Negative	**Positive**		**Negative**	**Positive**
Negative	6	20	Negative	17	12
Positive	4	62	Positive	9	64

Table 10 is an imbalanced dataset showing the results of the Confusion Matrix in the 2nd scenario with the K-nearest neighbour method. A total of 6 data are correctly predicted as negative data (True Negative). 62 data are rightly expected as positive data (True Positive). As many as 20 data are mis predicted as positive data. It should be negative data (False Positive). As many as 4 data are mis predicted as negative data. It should be positive (False Negative). The balanced dataset displays the results of the Confusion Matrix in the 1st scenario with the K-nearest neighbour method. A total of 17 data are correctly predicted as negative data (True Negative). 64 data are rightly expected as positive data (True Positive). As many as 12 data were mis predicted as positive data. It should have been negative data (False Positive). As many as 9 data were mis predicted as negative data. It should have been positive data (False Negative).

Table 11. Compare Performance of K-NN (Percentage)

Scenario	**Dataset Imbalanced**				**Dataset Balanced**			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
1	72.55	74.73	93.15	82.93	**79.41**	**84.21**	**87.69**	**85.91**
2	**73.91**	**75.61**	**93.94**	**83.78**	78.26	81.94	89.39	85.51
3	73.17	75.34	93.22	83.33	74.39	80.65	84.75	82.64
4	72.22	74.60	92.16	82.46	77.78	81.82	88.24	84.91
5	73.77	76.92	90.91	83.33	77.05	85.71	81.82	83.72

4.2. Discussion

This step is to make a comparison of the three algorithms. The Support Vector Machine and K-nearest neighbour algorithms use similarities between objects, while Naïve Bayes uses probabilities. This study uses the Confusion Matrix to calculate the Accuracy, Precision, Recall, and F1-Score described previously. A comparison of the three algorithms takes accuracy as a reference. The results of the accuracy comparison of the Support Vector Machine, Naïve Bayes, and K-nearest neighbour algorithms are in Table 12.

Table 12. Comparison Accuracy Algorithms (Percentage)

Scenario	Support Vector Machine		Naïve Bayes		K-nearest neighbor	
	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced
1	77.45	78.43	69.61	80.39	72.55	**79.41**
2	**79.35**	84.78	70.65	77.17	**73.91**	78.26
3	78.05	79.27	**70.73**	79.27	73.17	74.39
4	76.39	84.72	69.44	**80.56**	72.22	77.78
5	77.05	**86.89**	70.49	80.33	73.77	77.05

Based on Table 12, the results of the accuracy of sentiment analysis on Lampung Robusta Coffee. It already has promising results on all three algorithms. The algorithm that gives the best accuracy results is the Support Vector Machine using the Synthetic Minority Over-sampling Technique (SMOTE) of **86.89%** in the 5th scenario, Naïve Bayes of **80.56%** in the

2nd scenario, and K-nearest neighbour of **79.41%** in the 1st scenario. The three algorithms, with the addition of the balanced Synthetic Minority Over-sampling Technique (SMOTE), provide better accuracy results compared to imbalanced datasets.

5. CONCLUSION

This research has successfully implemented the Support Vector Machine, Naïve Bayes, and K-nearest neighbour algorithms for Sentiment Analysis of Lampung Robusta Coffee we used 203 data obtained from YouTube Lampung Robusta Coffee was dominated by positive sentiment. 145 had a positive sentiment or 71.4%, and 48 had a negative sentiment or 28.6%. The best accuracy results are the Support Vector Machine method by balancing Synthetic Minority Over-sampling Technique (SMOTE) data at 86.89%, then in Naïve Bayes at 80.56% and K-nearest neighbours at 79.41%. Thus, this research would helps the public decide on future marketing and development of coffee products.

6. ACKNOWLEDGMENTS

The author expresses his deepest appreciation to Universitas Mitra Indonesia (UMITRA) which has provided internal research grand funding.

REFERENCES

[1] D. Budi, W. Mushollaeni, Y. Yusianto, and A. Rahmawati, "CHARACTERIZATION OF TULUNGREJO ROBUSTA COFFEE POWDER (Coffea canephora) FERMENTED WITH Saccharomyces cerevisiae YEAST," *J. Agroindustri*, vol. 10, no. 2, pp. 129–138, 2020, doi: 10.31186/j.agroindustri.10.2.129-138.

[2] H. R. Santosa, C. Cucu Suherman, and S. Rosniawaty, "Growth Response of Robusta Coffee Plants (Coffea robusta L.) Stressed by Aluminum in Sengon Vegetated Former Coal Mine Reclamation Land (El Nino Period)," *Agrikultura*, vol. 27, no. 3, pp. 124–131, 2016, doi: 10.24198/agrikultura.v27i3.10871.

[3] S. Setyani, S. Subeki, and H. A. Grace, "Evaluation of Defect Value and Flavour Robusta Coffee (Coffea canephora L.) Produced by Small and Medium Industri Sector of Coffee in Ta," *J. Teknol. Ind. Has. Pertan.*, vol. 23, no. 2, p. 103, 2018, doi: 10.23960/jtihp.v23i2.103-114.

[4] F. Romadoni, Y. Umaidah, and B. N. Sari, "Text Mining for Customer Sentiment Analysis towards Electronic Money Services Using the Support Vector Machine Algorithm," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 2, pp. 247–253, 2020, doi: 10.32736/sisfokom.v9i2.903.

[5] E. Retnoningsih and R. Pramudita, "Machine learning with supervised and unsupervised learning techniques using Python," *Bina Insa. Ict J.*, vol. 7, no. 2, p. 156, 2020, doi: 10.51211/biict.v7i2.1422.

[6] N. P. A. Anesca, K. Muludi, and D. A. Shofiana, "Sentiment Analysis Protokol Kesehatan Virus Corona Dari Tweet Menggunakan Word2Vec Model Dan Recurrent Neural Network Learning," *J. Pepadun*, vol. 2, no. 3, pp. 432–439, 2021, doi: 10.23960/pepadun.v2i3.86.

[7] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "Sentiment Analysis of the Ruang Guru Application on Twitter Using a Classification Algorithm," *J. Teknoinfo*, vol. 14, no. 2, p. 115, 2020, doi: 10.33365/jti.v14i2.679.

[8] J. D. C. Aruan, B. Rahyudi, and A. Ridok, "Analysis of Public Opinion Sentiment towards Regional General Hospital Services using the Support Vector Machine Method and Term Frequency - Inverse Document Frequency," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 5, pp. 2072–2078, 2022.

[9] K. Muludi, M. S. Akbar, D. A. Shofiana, and A. Syarif, "Sentiment Analysis Of Energy Independence Tweets Using Simple Recurrent Neural Network," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 4, p. 339, 2021, doi: 10.22146/ijccs.66016.

[10] K. Munawaroh, "Performance Comparison of SVM, Naïve Bayes , and KNN Algorithms for Analysis of Public Opinion Sentiment Against COVID-19 Vaccination on Twitter," vol. 4, no. October, pp. 113–125, 2022.

[11] Y. Fu and Y. Yu, "Research on text representation method based on improved TF-IDF," *J. Phys. Conf. Ser.*, vol. 1486, no. 7, 2020, doi: 10.1088/1742-6596/1486/7/072032.

[12] C. Cahyaningtyas, Y. Nataliani, and I. R. Widiasari, "Sentiment Analysis on Shopee Application Ratings Using the SMOTE-Based Decision Tree Method," *Aiti*, vol. 18, no. 2, pp. 173–184, 2021, doi: 10.24246/aiti.v18i2.173-184.

[13] S. L. Gaol, A. Herdiani, and I. Asror, "Sentiment Analysis of Public Comments on 2019 Presidential Candidates on YouTube Social Media Using SVM and Lexicon Based Methods," 2019.

[14] N. Anggraini and M. J. Tursina, "Sentiment Analysis of School Zoning System on Youtube Social Media Using the K-Nearest Neighbor with Levenshtein Distance Algorithm," *2019 7th Int. Conf. Cyber IT Serv. Manag. CITSM 2019*, no. May, pp. 1–4, 2019, doi: 10.1109/CITSM47753.2019.8965407.

[15] A. Wahid and G. Saputri, "Sentiment Analysis of YouTube Comments About Patwal Ambulance Volunteers Using Naïve Bayes and Decision Tree Algorithms," *J. Sist. Komput. dan Inform.*, vol. 4, no. 2, p. 319, 2022, doi: 10.30865/json.v4i2.4941.

[16] A. Salma and W. Silfianti, "Sentiment Analysis of User Reviews on COVID-19 Information Applications Using Naive Bayes Classifier , Support Vector Machine , and K-Nearest Neighbor," vol. 6, no. 4, pp. 158–162, 2021.

[17] A. M. Rahat, A. Kahir, A. Kaisar, and M. Masum, "Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset," no. June 2020, 2019, doi: 10.1109/SMART46866.2019.9117512.

[18] E. F. Saputra and M. R. Pribadi, "Comment Sentiment Analysis on The Lazy Monday Youtube Channel Using the Naive Bayes Algorithm," *MDP Student Conf.*, vol. 2, no. 1, pp. 17–23, 2023, doi: 10.35957/mdp-sc.v2i1.4283.

[19] S. Thomas and P. Noviyanti, "Analysis Study of Sentiment Analysis Methods on YouTube," vol. 1, no. 1, 2021.

[20] V. O. Tama, Y. Sibaroni, and Adiwijaya, "Labeling Analysis in the Classification of Product Review Sentiments by using Multinomial Naive Bayes Algorithm," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012036.

[21] M. Işik and H. Dağ, "The impact of text preprocessing on the prediction of review ratings," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, pp. 1405–1421, 2020, doi: 10.3906/elk-1907-46.

[22] D. E. Cahyani and I. Patasik, "Performance comparison of tf-idf and word2vec models for emotion text classification," *Bull. Electr. Eng. Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021, doi: 10.11591/eei.v10i5.3157.

[23] A. M. Pravina, "Sentiment Analysis of Delivery Service Opinions on Twitter Documents using K-Nearest Neighbor," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 2, pp. 996–1012, 2022, doi: 10.35957/jatisi.v9i2.1899.

[24] R. Risnantoyo, A. Nugroho, and K. Mandara, "JITE (Journal of Informatics and Telecommunication Engineering)," vol. 4, no. 1, pp. 86–96, 2020.

[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "snopes.com: Two-Striped Telamonia Spider," *J. Artif. Intell. Res.*, vol. 16, no. Sept. 28, pp. 321–357, 2002, [Online]. Available: https://arxiv.org/pdf/1106.1813.pdf%0Ahttp://www.snopes.com/horrors/insects/telamonia.asp

[26] D. A. Pisner and D. M. Schnyer, "Support vector machine," *Mach. Learn. Methods Appl. to Brain Disord.*, pp. 101–121, 2019, doi: 10.1016/B978-0-12-815739-8.00006-7.

[27] F. R. Lumbanraja, E. Fitri, Ardiansyah, J. Akmal, and P. Rizky, "Machine Algorithm (Case Study : Abstract in a Computer Science Journal) Abstract Classification Using Support Vector Machine Algorithm (Case Study : Abstract in a Computer Science", doi: 10.1088/1742-6596/1751/1/012042.

[28] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive Bayes classification algorithm for traffic risk management," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, 2021, doi: 10.1186/s13634-021-00742-6.

[29] D. A. Kristiyanti, A. H. Umam, M. Wahyudi, R. Amin, and L. Marlinda, "Comparison of SVM Naïve Bayes Algorithm for Sentiment Analysis Toward West Java Governor Candidate Period 2018-2023 Based on Public Opinion on Twitter," *2018 6th Int. Conf. Cyber IT Serv. Manag. CITSM 2018*, no. June 2021, pp. 1–6, 2019, doi: 10.1109/CITSM.2018.8674352.

[30] N. Hidayati and A. Hermawan, "K-Nearest Neighbor (K-NN) algorithm with Euclidean and Manhattan in classification of student graduation," *J. Eng. Appl. Technol.*, vol. 2, no. 2, pp. 86–91, 2021, doi: 10.21831/jeatech.v2i2.42777.