Journal of Soft Computing Exploration

Homepage: https://shmpublisher.com/index.php/joscex

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Comparison of the performance of naive bayes and support vector machine in sirekap sentiment analysis with the lexicon-based approach

Ramadhana Setiyawan[1], Zaenal Mustofa[2*]
[1,2]Department of Informatics Engineering Education, Universitas Negeri Yogyakarta, Indonesia

Article Info

Article history:

Received May 16, 2024
Revised May 30, 2024
Accepted June 5, 2024

Keywords:

Support vector machines
SIREKAP
Naive bayes
Classification
Lexicon based

ABSTRACT

The general public often uses the SiRekap application to see the progress of the election and to provide critical statements. Policies made by the government have good and bad outcomes, and users end up leaving their reviews and ratings on the Google Play Store, where the app can be downloaded. These reviews can be collected and processed into useful information such as sentiment analysis using Naïve Bayes and Support Vector Machine methods. Both methods have differences during training and during evaluation. The difference in results from the various scenarios tested was not much different. When training the Support Vector Machine model is able to process comment data labeled with a lexicon 10% better than the Naïve Bayes model by looking at the results of the accuracy of the two models. During the accuracy evaluation process, the two models produce the same accuracy of 72%. Although both models get the same accuracy during the evaluation process, there are differences in precision, recall, and f1 score. The difference is that the Support Vector Machine model is 5% better for precision, 8% for recall, and 3% for f1-score compared to the Naïve Bayes model. This research is limited to only knowing the performance of two machine learning models, namely the use of naive bayes and svm by using a label lexicon. The results obtained can be improved for the better. Improving the evaluation results can be done by adding data or using text data augmentation and there is creation from experts related to language sentiment.

Corresponding Author:

Zaenal Mustofa,
Department of Informatics Engineering Education,
Universitas Negeri Yogyakarta,
Jln. Colombo No. 1, Kab. Sleman, 55281, Daerah Istimewa Yogyakarta
Email: zaenalmustofa@uny.ac.id
https://doi.org/10.52465/joscex.v5i1.367

1. INTRODUCTION

An important part of a country's democratic process is general elections, also known as general elections. Free, fair, and transparent elections are essential for a strong democratic system. Many countries have switched to electronic systems as a result of technological advances to simplify the process of calculating election results and voting. A system called the Voice Recapitulation System (SIREKAP) is part of the technology used to speed up and automate the collection and processing of voice data. Therefore, this technology is in line with the global trend in which most countries are using technology in their election processes. If digital data are available directly, analysis and reporting can be done more quickly. Providing support to increase public confidence in the fairness and transparency of elections [1].

Today, the use of the internet is a place to express thoughts and opinions. The Internet has many different sources. One of them is the Google Play Store site, which has a review column for each mobile application listed there. Based on KPU Decree Number 66 of 2024, the KPU will use the Electronic Recapitulation System, or abbreviated as SiRekap, as the organizer of the 2024 Election. In 2019, the KPU used the Vote Counting System, or Situng, as the old mechanism[2]. The general public often uses the SiRekap application to see election developments and provide critical statements. Every user has a different opinion about the SIREKAP application recently launched by the government. The policies created by the government have both good and bad results, and users end up leaving their reviews and ratings on the Google Play Store, where the app can be downloaded. In this case, it is possible to collect and process these reviews into useful information such as sentiment analysis.

Sentiment analysis, also known as opinion mining, is a field of learning that analyzes sentiments, opinions, and emotions from written data [3]. Sentiment analysis can be performed after the data set is collected. Data set collection is done to determine the inclusion of something that has a positive or negative tendency. Sentiment analysis can be created using machine learning methods [4]. In the machine learning method, there are models called SVM (Support Vector Machine) and NB (Naive Bayes), both models have good performance when used for sentiment analysis [5].

Previous research entitled "Digital Payment Comparison for Sentiment Analysis Based on Reviews on Google Playstore using the Support Vector Machine Method" aims to compare the sentiment of user reviews of two payment applications on the Google Play Store using the Support Vector Machine method. The best results obtained in the fund application resulted in a 92% accuracy [6].

In addition to this research, there is another study entitled "Sentiment Analysis on Amazon Shopping App Reviews on the Google Play Store Using Naive Bayes Classifier" which aims to analyze the sentiment of user reviews on the Amazon Shopping application using the Naive Bayes classifier method and several other machine learning methods. The best result obtained is the Naive Bayes method, which gets an accuracy of 86.74% [7]. Research related to sentiment analysis of an application has been carried out, but for research that focuses on comparing machine learning methods, not much has been done with object focus to the SiRecap application. It is hoped that this study will be able to provide good comparative results related to understanding the use of SVM and Naive Bayes[8], especially in the SIREKAP application.

2. METHOD

Research Flow

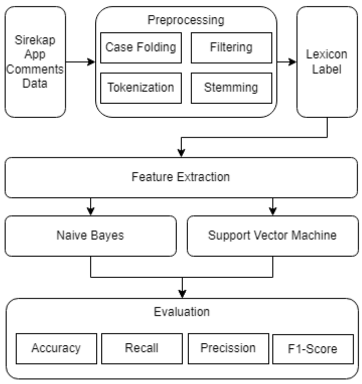

Figure 1. Research flow

Figure 1 is part of the research method carried out in this study. In the picture, part of each process as follows:

1) Comment data is part of the text data collection contained in the comment column of the sirekap mobile application.
2) Preprocessing is part of the process of cleaning and preparing raw text data to be suitable for further analysis. The main purpose of preprocessing is to improve data quality and reduce noise that can interfere with the analysis process.
3) Lexicon labels are part of labeling using the lexicon dictionary [9].

4) Feature extraction is part of the process of converting raw text into numerical representations that can be used by machine learning algorithms to model relationships between text documents[10].
5) Model Training using Naive Bayes and Support Vector Machine, in this section using two machine learning models to be compared.

Evaluation is part of the testing process of a pre-trained model. In this section, we will use a sentence that will produce an output in the form of aspects and sentiments of the sentence. This process is also part of the use of the confusion matrix method.

Sentiment Analysis

Text mining always involves document preprocessing, such as text categorization, information extraction, and word extraction. This technique uses interesting patterns to extract information from data sources. The use of text mining is often used for problems such as classification, clustering, data extraction, and data recovery. Text pre-processing, text mining procedures, and post-processing are three common steps in the text mining process. The text pre-processing process includes data selection, classification, and feature extraction to transform documents into intermediate forms suitable for various search purposes. Grouping, discovery of association rules, analysis trends, discovery patterns, and knowledge discovery algorithms are major parts of the work of text processing operations [11].

Preprocessing

Pre-processing is a step that needs to be done because at this stage the data text to be taken from an unstructured form is converted into a multidimensional and structured form [12].

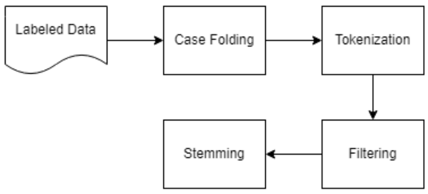

Figure 2. Preprocessing

In figure 2 the preprocessing stage consists of several stages, including case folding, tokenization, filtering, and stemming.

1) Case folding is the steps taken to transform unstructured words into structured or uniform ones. For example, "i" changes to "I" and "yoU" changes to "you".
2) Tokenization is the process of deciphering a sentence so that it splits into words, but also deciphers and groups separate words to create a higher word, called a "token". This process also removes unused characters or punctuation.
3) Filtering to retrieve important words, filtering, also known as last-word removal, takes less important words such as "in", "and", etc., and then stores them in a list of important words.

Feature Extraction

Feature extraction is part of the machine that accepts text as numbers. Word vectorization or word embedding is the process of converting or mapping words or text to real value vectors. It is a feature extraction method in which a document is divided into sentences and then divided into words. Next, a matrix or feature map is created, where each row represents a sentence or document, each column represents a word in the dictionary, and the values in the feature map cells usually indicate the number of words present in the sentence or document [13]. There is a method that can be used, one of which is TF-IDF. TF-IDF is a word/term weighting method that assigns different weights to each word in a document based on the frequency of words in each document and the total frequency of words in the document. In this study, TF-IDF was used because it has better performance, especially in terms of increasing recall and precision values. The TF-IDF model consists of four steps [14]. The first step is to calculate the frequency of each word that appears in each document (TF). As in Equation (1).

$$\mathrm{tf_t} = 1 + \log\,(\mathrm{tf_t}) \quad (1)$$

Where:
$\mathrm{tf_t}$ = is part of the term.

Next, the second step is the calculation of the number of documents containing a certain word (DF). The third step is the calculation of the DF (IDF). The use of IDF can be seen in Equation (2).

$$idf_t = log\left(\frac{d}{df_t}\right) \quad (2)$$

where:
idf_t = merupakan inverse document frequency
d = is the sum of the document
$\mathrm{df_t}$ = is the number of documents containing the term t

The final step is the calculation of the TF-IDF. TF-IDF is the multiplication between the result of TF and the calculation of the result of the IDF for each word. The calculation is shown in Equation (3).

$$W_{t,d} = tf_t \times idf_t \quad (3)$$

Where:
W = is the weight of the term of the document
tf_t = is the number of occurrences of the term
idf_t = is the inverse of the frequency of documents containing the term

Lexicon

Lexicon-based approach This method uses a word dictionary in which each positive and negative word is assigned a sentiment value. Then, the sum or average of sentiment values is used to calculate the sentiment of the entire sentence or document. The dictionary-based approach and the corpus-based approach are two types of lexicon-based approaches based on the sentiment lexicon [13].

Naïve Bayes

Naive Bayes (NB) is a classification method that can predict the probability of a class to make decisions based on learning data. The advantages of NB include being easy to use, fast and very accurate when applied to large data [15]. The use of Naïve Bayes can be seen in equation (4).

$$P(Y_j\,|X_i) = \frac{P(Y_j\,|X_i)\,.\,P(Y_j)}{P(X_i)} \quad (4)$$

where:
X_i = is a feature of *sample Vector* $i, i \in \{1,2, \ldots, n\}$
Y_j = is class notation $j, j \in \{1,2, \ldots, n\}$
$P(Y_j\,|X_i)$ = represents the probability of *the sample* X_i belonging to the class W_j.

Support Vector Machine

Obtaining a model that maximizes performance of training data is the primary goal of pattern classification. Conventional training methods define the model in such a way that each input-output pair is correctly classified within a class to which it belongs. However, if the classifier is too suitable to teach training data, the model begins to memorize the training data rather than learning to generalize, which makes it more difficult to generalize. SVM allows the model to maximize its generalizability because its main purpose is to separate the various classes in the training set by a surface that maximizes the distance between them. This is the goal of the structural risk minimization (SRM) principle, which allows minimization tied to the generalization errors of a model. This is in contrast to the philosophy often used by empirical risk minimization methods [16]. The advantages of using SVM [17] are:

1) When there are minor problems in a particular case, this model can be used well.
2) Have the ability to generalize well.
3) Resolve if there are problematic dimensions.

4) If there are non-linear data, it is solved because this model is kernel-based.

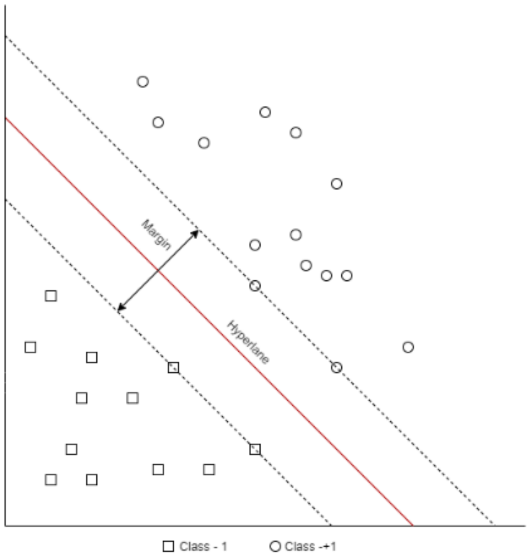

Figure 3. Hyperplane

In figure 3 it can be explained that there are classes separated by delimiters paired with parallel shapes. The delimiter has a function to distinguish the location of the positive class and negative class so as to obtain Equation 5 and Equation 6.

$$(w, x_i) + b \geq 0 \;,\; y_n = +1 \tag{5}$$
$$(w, x_i) + b < 0 \;,\; y_n = -1 \tag{6}$$

In Equations 5 and 6 there is **w** is the normal plane of the hyperlane, which has a perpendicular direction while **b** is the bias or alternative position of the plane of the coordinate center. In this case, SVM will find the maximum value of the hyperlane margin for decision makers.
The search for the maximum margin value uses a calculation formula related to optimization problems or often called lagrange multipier (Ld), so that Equation 7 can be made.

$$L_D = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j}^{n} y_i y_j a_i a_j (x_i . x_j) \;, \tag{7}$$
$$0 \leq a_i \leq C \;\&\; \sum_{i=1}^{n} a_i y_i = 0 \tag{8}$$

In equations 6 and 7, a is the weight of the data and y is the target value, while x is part of the kernel. The equation has conditions if the values 0 to C (constant) and the sum of a_i times by y_i 0. In real cases, most data often get errors in separating data linearly because it has non-linear properties [18]. This problem can be solved by using the Radial Basis Function (RBF) kernel function, because RBF is suitable for all types of data. The RBF formulas can be used as in Equation 9.

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right) \tag{9}$$

In equation 8 there is $||x - x'||$ is the Euclidean distance of the two different feature spaces in the data while σ (sigma) is a parameter in the RBF kernel as a determinant of the value or weight of the SVM kernel. The RBF kernel parameters have the same performance as linear kernels, namely the C and γ parameters. In this case, it requires a grid search method to get the best parameters so as to produce high accuracy when training data.

Confussion Matrix

.

Performance measurement for binary classification can use the confusion matrix method [19]. The confusion matrix is an important evaluation tool in statistical modeling, especially in the context of classifying machine learning models. It helps in evaluating the performance of the classification model by comparing the model's prediction results with the actual class of data. The confusion matrix has four main parts [20], namely:

1) True Positive (TP) is the portion of the number of samples that are actually positive and predicted correctly by the model.
2) True Negatives (TN) are the subset of sample numbers that are completely negative and predicted correctly by the model.
3) False positives (FP) are the subset of sample numbers that are actually negative but predicted as positive by the model (type I error).
4) False negatives (FN) are part of the number of samples that are actually positive but predicted as negative by the model (type II error).

The confusion matrix provides a clear understanding of the knowledge of the model's performance results, including good or failed categories. From this, several model performance evaluation metrics can be calculated, such as accuracy, precision, recall, and F1 score, all of which are based on a combination of elements in the confusion matrix.

1) Accuracy is the proportion of correct predictions. Accuracy measures how well a classification predicts a condition.

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{N}} \tag{10}$$

2) Precision is a measure used to evaluate relevant instances among the examples taken. It is calculated as the proportion between a truly positive prediction and the set of all positive values.

$$\mathrm{Precs} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{11}$$

3) Recall or Sensitivity is used to determine the ability of the model to predict positive cases. In this section, it will be calculated as the proportion between truly positive predictions and the set of all positive predictions.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{12}$$

4) Score is a metric that takes into account precision and Sensitivity in the same way. It is a harmonized average between precision and sensitivity.

$$\mathrm{FSC} = \frac{\mathrm{Precs} * \mathrm{Recall}}{\mathrm{Precs+Recall}} \tag{13}$$

3. RESULTS AND DISCUSSIONS

Dataset

At this stage, data collection of text sets from the sirecap application is carried out; the data collection uses thelibrary that has been provided, namely called Google Play Scrapper. The data obtained have a total of 2362 comments.

Pre-processing

At this stage, the cleaning and preparation of the text data is carried out so that it can be processed to the next stage. At this stage there are several parts that need to be done including case folding, tokenization, filtering, and stemming.

Table 1. Case folding

No	Commentary

1	Create a Vote Report
2	Good application to immediately remove Playstore
3	Very useful
4	Quite helpful, even though the photo must be in an offline position &; the results must be edited one by one.. The upload of images to PDF is also quite fast.
5	Asshole app.... The trigger for the people's uproar...

In Table 1 it can be explained that all the data obtained will be converted to lowercase letters to have uniformity to avoid differences between upper and lowercase letters.

Table 2. Tokenization

No	Commentary
1	['create', 'report', 'gain', 'vote']
2	[apps', 'the' , 'good' , 'to' , 'soon' , 'in' , 'delete' , 'playstore']
3	['very', 'useful']
4	['passable', 'helpful' , 'alu' , 'the photo' , 'must' , 'position' , 'offline' , 'result' , 'must' , 'edited' , 'one' , 'by one' , 'by one' , 'upload' , 'image' , 'to' , 'pdf' , 'also' , 'passable' , 'fast']
5	['application', 'asshole', 'trigger', 'commotion', 'people']

In Table 2, it can be explained that the data that have gone through the case folding process will be broken down into word division and cleaning of unused punctuation.

Table 3. Filtering

No	Commentary
1	Create a Vote Report
2	good app immediately remove playstore
3	Very useful
4	Not bad for helping the photo offline position, the results are edited one by one, and the upload of pdf images is quite fast
5	The Jerk application triggers people's uproar

In Table 3 it can be explained that data that have gone through the tokenization process will be reunited and word deleted. Words removed from the data use a filtering method if any words are deemed irrelevant or undesirable for the analysis section.

Table 4.Mood

No	Commentary
1	Make Report by Voice
2	good app immediately remove playstore
3	very useful
4	Not bad help off-line position photos edited one one upload pdf images quite fast
5	Jerk Application Triggers People's Storm

In Table 4 it can be explained that data that have gone through the filtering process will be modified into the basic form of words in a sentence used.

Label Lexicon

At this stage using the Lexicon-Based Method, the method uses dictionaries that have been provided by other researchers. The word dictionary provided has weights for each word that exists. The basic determination for positive and negative labels is obtained by means of which the total positive word score of an existing sentence will be reduced by the total negative word score. If the word score is more than 0 then it becomes a "positive" sentence, otherwise it will be "negative". Based on this explanation, some of the results of this stage can be seen in Table 5.

.

Table 5. Lexicon based

No	Commentary	Label
1	Make Report by Voice	Positif
2	good app immediately remove playstore	Negatif
3	very useful	Positif
4	Not bad help offline position photos edited one one upload pdf images quite fast	Positif
5	Jerk Application Triggers People's Storm	Negatif

The data will also be split as much as 70:30 for model data and test data in this study. The results of the split can be seen in Table 6.

Table 6. Comment data

	Positive	Negative
Data Model	980	673
Test Data	440	269

Feature Extraction

At this stage, it is part of giving faithful weight to words using the TF-IDF method. The result obtained is that each word or term in a sentence will have its own weight. A sentence that already has the weight of each word will proceed to the stage of making a model.

Classification Model Training

At this stage, it is used for the creation of classification models. The model uses two methods, namely multinomial Naive Bayes and Support Vector Machine. At this stage, explain the results of the training obtained using the two existing methods. The quality of the model training results will affect the model's ability to generalize and perform tasks with high accuracy on never-before-seen data. The training model in this study will use 70% of the data obtained.

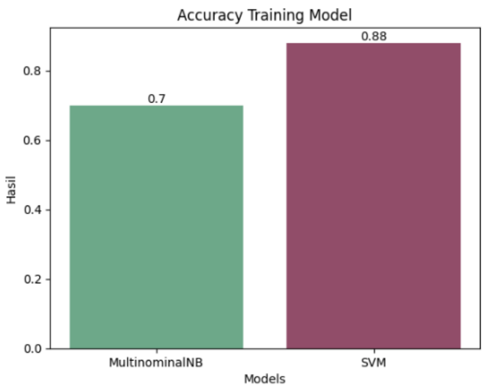

Figure 1. Accuracy of the training model

In Figure 4 it can be explained that the use of training data that have been set for model creation produces accuracy results for Naive bays of 70% and accuracy for svm of 88%.

Evaluation

At this stage, it is used to find out the performance results of the model that has been made. Model evaluation is an important process in the development and application of machine learning algorithms. It involves the use of various metrics and techniques to assess model performance against data used for training and testing. The ultimate goal is to ensure that the model built can generalize well to new data that have never been seen before. Model evaluation provides insight into how well the model can predict or classify previously unseen data.

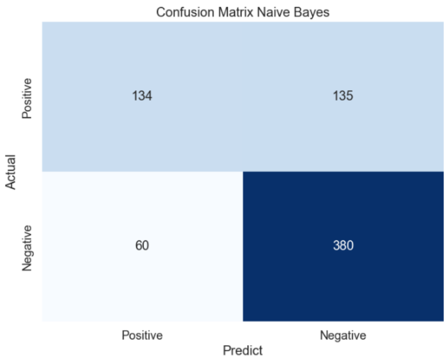

Figure 1. Confusion matrix NB

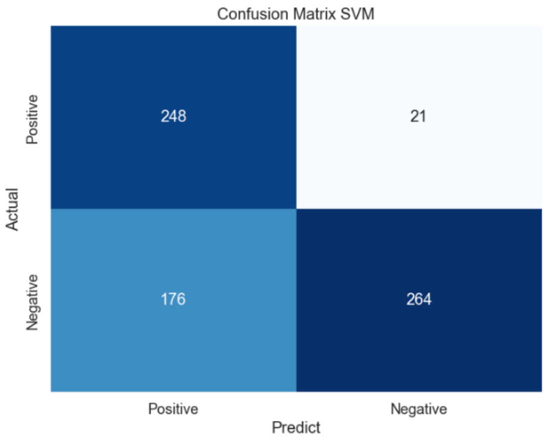

Figure 2. Confusion matrix SVM

In Figure 5 it can be explained that the image is the final result of the evaluation using confusion matrix for the Naïve Bayes model. In Figure 6 it can be explained that the gamber is the result of an evaluation using a confusion matrix for SVM mode. The results of the confusion matrix from both models will be used for measurements to find accuracy, precision, recall, and F1-Score.

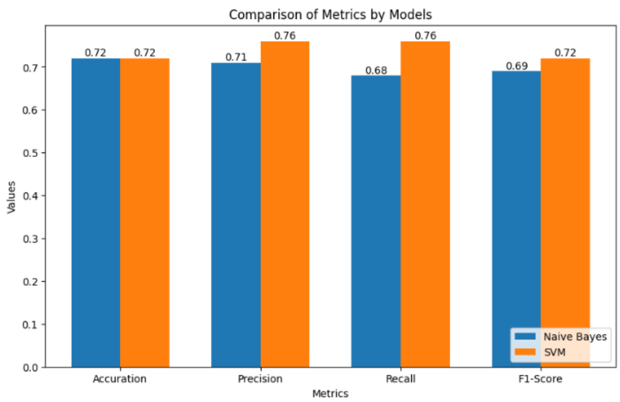

Figure 3. Comparison of evaluation results

In Figure 7 it can be explained that the comparison between the use of machine learning models between Naive Bayes and SVM has differences. The machine learning model using the Naïve Bayes model resulted in 72% accuracy, 71% precision, 68% recall, and 69% f1-score. Meanwhile, the machine learning model using the SVM model produces 72% accuracy, 76% precision, 76% recall, and an F1-score of 72%.

.

The SVM model is superior in several key metrics. It achieves higher precision (76% compared to 71%), higher recall (76% compared to 68%), and a higher F1-score (72% compared to 69%) than the Naïve Bayes model. This indicates that the SVM model is more effective in correctly identifying positive instances and minimizing false positives and false negatives. Although both models share the same accuracy, the superior precision, recall, and F1 score of the SVM model suggest that it provides more reliable and balanced performance in classification tasks. This makes the SVM model a better choice for applications where the cost of false positives and false negatives is significant.The results of the evaluation are part of the use of the confusion matrix with the mean " Macro".

4. CONCLUSION

In this study, many processes used to analyze the sentiment of the SiRekap application. The process of analyzing comment data on the Sirekap app begins with data collection using Google Play Scrapper. This tool accesses the Google Play Store to extract comment data from the Sirekap app page. The data obtained in this study was 2362 comments. The data will be used for the creation of the model. After the data are collected, preprocessing is carried out which includes several stages case folding to convert all text into lowercase letters, tokenization to break the text into individual words, filtering to remove unimportant words, and stemming to convert words to their base form. These stages aim to clean and prepare the data to make it more ready for sentiment analysis. After pre-processing, the data is labeled using a lexicon dictionary containing words along with their sentiment scores to determine whether the comments have positive and negative. It will be labeled using lexicon labels so that positive comment data of 1420 and negative comments of 942 are obtained.

Machine learning models are then built using Naive Bayes and Support Vector Machine (SVM) algorithms to classify the sentiment of the comments. Model evaluation is performed using metrics such as accuracy, precision, recall, and F1-score to assess classification performance. Accuracy measures the proportion of correct predictions, precision assesses the proportion of truly positive predictions, recall measures the proportion of correctly detected positive cases, and the F1 score provides the harmony between precision and recall, giving an overall picture of the model's performance in classifying user comment sentiment.

Making classification models using two methods, namely Naive Bayes and Support Vector Machine. The results of making the model during the training session resulted in 70% accuracy for Naive Bayes and 80% for the Support Vector Machine. Apart from the training session, both models were evaluated using the confusion matrix method. The Naive Bayes machine learning model delivers 72% precision, 71% precision, 68% recall, and 69% f1-score. Meanwhile, the machine learning model using the SVM model produces a precision of of 72%, a precision 76%, a recall of 76%, and a f1 score of 72%.

Referring to the process that has been passed from this study, the two models have differences during training and during evaluation. The difference in results from the various scenarios tested was not much different. When training the Support Vector Machine model, it was able to process comment data labeled lexicon 10% better than the Naive Bayes model by looking at the results of the accuracy of both models. In the accuracy evaluation process, the two models produced the same accuracy, which was 72%. Although both models get the same accuracy during the evaluation process, there are differences in precision, recall, and f1 score. The difference was obtained by the Support Vector Machine model which is 5% better for precision, 8% for recall, and 3% for f1-score compared to the Naïve Bayes model.

REFERENCES

[1] I. A. Pradesa, “Analisis Penggunaan Sistem Rekapitulasi Suara (Sirekap) Dalam Menghadapi Problematika Pemilu 2024,” *Triwikrama J. Multidisiplin Ilmu Sos.*, vol. 03, no. 04, pp. 47–57, 2024.

[2] M. Nurkamiden, “SiRekap : Tantangan dan Potensi Kekeliruan Proses Rekapitulasi Pemilu Serentak di Indonesia SiRekap : Challenges and Potential Errors in the Recapitulation Process of Simultaneous Elections in Indonesia,” vol. 1, no. c, pp. 101–110, 2024.

[3] R. Naquitasia, D. H. Fudholi, and L. Iswari, “Analisis Sentimen Berbasis Aspek pada Wisata Halal dengan Metode Deep Learning,” *J. Teknoinfo*, vol. 16, no. 2, p. 156, 2022, doi: 10.33365/jti.v16i2.1516.

[4] A. Permana Putra and A. Farrah Syafira, “Analisis Sentimen Data Twitter Topik Politik Dengan Metode Naive Bayes Dan Convolutional Neural Networks (Cnn),” *J. Ilm. Wahana Pendidik.*, vol. 9, no. 20, pp. 36–41, 2023.

[5] D. F. Salsabillah, D. E. Ratnawati, and N. Y. Setiawan, “PERBANDINGAN ALGORITMA SUPPORT VECTOR MACHINE DENGAN NAÏVE BAYES (STUDI KASUS : AYAM GORENG NELONGSO CABANG SINGOSARI , MALANG) SENTIMENT ANALYSIS OF RESTAURANT REVIEWS USING COMPARISON OF SUPPORT VECTOR MACHINE ALGORITHM WITH NAÏVE BAYES (CASE STUDY :,” vol. 11, no. 1, pp. 107–116, 2024, doi: 10.25126/jtiik.20241117584.

[6] R. A. A. Renal, Syariful Alam, and Moch Hafid T, “Komparasi Payment Digital Untuk Analisis Sentimen Berdasarkan Ulasan Di Google Playstore Menggunakan Metode Support Vector Machine,” *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 2, no. 3, pp. 118–128, 2023, doi: 10.55123/storage.v2i3.2337.

[7] Ernianti Hasibuan and Elmo Allistair Heriyanto, “Analisis Sentimen Pada Ulasan Aplikasi Amazon Shopping Di Google Play Store Menggunakan Naive Bayes Classifier,” *J. Tek. dan Sci.*, vol. 1, no. 3, pp. 13–24, 2022, doi: 10.56127/jts.v1i3.434.

[8] A. Palanivinayagam, C. Z. El-Bayeh, and R. Damaševičius, “Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review,” *Algorithms*, vol. 16, no. 5, pp. 1–28, 2023, doi: 10.3390/a16050236.

[9] Fajri Koto and Gemala Y. Rahmaningtyas, “InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in

Microblogs," *2017 Int. Conf. Asian Lang. Process.*, pp. 391–394, 20AD.

[10] Z. A. Khan and V. Rekha, "Fake News Detection Using TF-IDF Weighted with Word2Vec: An Ensemble Approach," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 3, pp. 1065–1076, 2023.

[11] A. F. Firdaus and W. I. Firdaus, "Text Mining Dan Pola Algoritma Dalam Penyelesaian Masalah Informasi : (Sebuah Ulasan)," *JUPITER J. Penelit. Ilmu dan Teknol. Komput.*, vol. 13, no. 1, pp. 66–78, 2021.

[12] I. Ho, H. N. Goh, and Y. F. Tan, "Preprocessing Impact on Sentiment Analysis Performance on Malay Social Media Text," *J. Syst. Manag. Sci.*, vol. 12, no. 5, pp. 73–90, 2022, doi: 10.33168/JSMS.2022.0505.

[13] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, pp. 1–19, 2021, doi: 10.1007/s13278-021-00776-6.

[14] 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.

[15] S. Styawati, A. R. Isnain, N. Hendrastuty, and L. Andraini, "Comparison of Support Vector Machine and Naïve Bayes on Twitter Data Sentiment Analysis," *J. Inform. J. Pengemb. IT*, vol. 6, no. 1, pp. 56–60, 2021, doi: 10.30591/jpit.v6i1.3245.

[16] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020, doi: 10.1016/j.neucom.2019.10.118.

[17] M. Luo and L. Luo, "Feature selection for text classification using OR+SVM-RFE," *2010 Chinese Control Decis. Conf. CCDC 2010*, pp. 1648–1652, 2010, doi: 10.1109/CCDC.2010.5498331.

[18] O. Jing, "Research on English Text Information Filtering Algorithm Based on SVM," *Proc. 2020 IEEE Int. Conf. Power, Intell. Comput. Syst. ICPICS 2020*, pp. 1001–1004, 2020, doi: 10.1109/ICPICS50287.2020.9202016.

[19] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput. Oper. Res.*, vol. 152, no. December 2022, p. 106131, 2023, doi: 10.1016/j.cor.2022.106131.

[20] H. Yun, "Prediction model of algal blooms using logistic regression and confusion matrix," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 2407–2413, 2021, doi: 10.11591/ijece.v11i3.pp2407-2413.

.