Journal of Information Systems Engineering and Management
2025, 10(20s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Implementation of Naïve Bayes Classifier & Support Vector Machine Algorithm for Sentiment Classification using Twitter Data on Indonesian Presidential Candidates In 2024

Gigih Forda Nama[1,2], Atiqah Hanifah Shalihah[3], Puput Budi Wintoro[4], Yessi Mulyani[5], Dikpride Despa[6]
[1] *Doctoral Program of Environmental Science, University of Lampung, Indonesia.*
[2,3,4,5] *Department of Informatics, University of Lampung, Indonesia.*
[6] *Department of Electrical Engineering, University of Lampung, Indonesia.*
Corespondent Email: gigih@eng.unila.ac.id

ARTICLE INFO

Received: 19 Dec 2024

Revised: 30 Jan 2025

Accepted: 15 Feb 2025

ABSTRACT

Indonesia is a democratic country, this is shown by the presidential election that is held every five years. In 2024, Indonesia is set to hold another presidential election, which has become a hot discussed topic among the public, particularly regarding the potential candidates who will be running for office. Many people voice their thoughts and opinion regarding the presidential candidates vying for office by means of tweets on the popular social media platform, Twitter. This classification model is built using machine learning algorithms, namely Naïve Bayes Classifier and Support Vector Machine. This study focuses on two presidential candidates, Anies Baswedan and Ganjar Pranowo. The dataset contains 3000 tweets in each dataset with an imbalanced class distribution. The percentage distribution for the labels in the Anies Baswedan dataset is 46.60% positive, 16.10% neutral, and 37.30% negative. Meanwhile, the percentage distribution for the labels in the Ganjar Pranowo dataset is 58.83% positive, 16.77% neutral, and 24.40% negative. The results show that using classification report evaluation method, the naïve bayes classifier algorithm has a higher performance on the Ganjar Pranowo, achieving an accuracy of 92%. Meanwhile, for the Anies Baswedan dataset, achieving an accuracy of 87%. For the SVM algorithm, an accuracy of 88% was obtained for the Anies Baswedan dataset, and an accuracy of 82% was obtained for the Ganjar Pranowo dataset. Based on the results, it can be observed that the naïve bayes classifier algorithm outperforms the SVM algorithm in sentiment classification.

Keywords: Sentiment Analysis, Naïve Bayes Classifier, Support Vector Machine, Twitter

1. INTRODUCTION

Indonesia is a legal state that adheres to democracy, where sovereignty is in the hands of the people. This is shown by the freedom for the people to express their opinions, aspirations, and choose presidential candidates in general elections (elections) according to their will without coercion from either side. In 2024 Indonesia will hold another general election to elect a president, of course this is a moment that is awaited and becomes a hot issue discussed by the public both on the internet and in the real world. In connection with the existence of this pilpres, many people have expressed their opinions about future presidential candidates who will nominate themselves both positive and negative opinions. Sentiment classification is useful for knowing the general picture of public opinion on a topic or problem, including the topic of presidential candidates who will run for the 2024 presidential election. Public opinion can shape the image of presidential candidates in the eyes of the public. However, in knowing the opinions of the public, there are problems, namely with the large amount of data related to public opinion, the classification process cannot be done manually, this is because it will take a very long time and if done manually it will not be effective in classifying. Therefore, to overcome this problem, the solution that can be done is to apply *machine learning* that can classify positive and negative sentiments related to public opinion on presidential candidates automatically using the appropriate algorithm. In expressing opinions about the 2024 presidential election, one of the media used by the public is through social media twitter by providing comments in the form of *tweets*. This research will focus on classifying the sentiment of presidential candidates Ganjar Pranowo and Anies Baswedan, then the names of these figures will be used as keywords or keywords in the search for *tweets*. Based on several studies that have been conducted, the most appropriate algorithms used for classification are the *Naïve Bayes Classifier* and *Support Vector Machine* algorithms. This is because the *Naïve Bayes Classifier* algorithm is easy to understand, faster in terms of calculation and only requires a small amount of *training* data.[1]. In addition, other studies have also been conducted using the *Support Vector Machine* algorithm. SVM has high accuracy and works very well with limited datasets.[2].

In this case, a sentiment classification will be carried out based on the opinions conveyed by the public through twitter social media related to the 2024 presidential candidates using the *Naïve Bayes Classifier* and *Support Vector Machine* methods, so that later the comparison of the two methods will be known. The *naïve bayes classifier* and *Support Vector Machine* algorithms can be used in determining the sentiment of *tweets* so that positive, neutral, and negative sentiments can be known which are the sentiment classes in the twitter text.

2. LITERATURE REVIEW

2.1. Twitter

Twitter is a communication technology that enables users to write and publish opinions and comments freely. In its use, Twitter allows users to read and upload messages known as "*tweets*", which can be text, images, or a video[3]. Twitter is a medium for data collection that is used for various purposes. Data collection from Twitter can be used for various needs such as, knowing the popularity of election candidates or elections, getting information about the popularity of a product or for a simple one can be used to see all mentions, retweets of a particular Twitter account.

2.2. *Natural Language Processing* (NLP)

Natural *Language* Processing is a theory-driven field of computing and artificial intelligence-based techniques for performing analysis and representation of natural human language with the goal of forming a strong relationship between computers and humans[4]. One of the most frequent implementations of text classification is sentiment classification where the goal is to determine the sentiment contained in the text. In the context of sentiment classification, texts such as product reviews, movie reviews, or social media posts are classified into different sentiment categories, such as positive or negative[5].

2.3. Sentiment Analysis

Sentiment analysis or also called opinion mining is a field of study that analyzes opinions, sentiments, evaluations, assessments, attitudes, individuals, issues, events, topics, and their attributes[6]. The use of sentiment analysis is widely employed as a means to obtain feedback or responses regarding services, products, or topics provided, so that it can be used for evaluation purposes [7].

2.4. *Text Mining*

Text Mining can be broadly defined as a knowledge-intensive process where users interact with a set of documents over time using a series of analysis tools. In general, the stages carried out in text mining are *preprocessing* which aims to clean the text. Then the feature extraction process that converts text data into structured data to be processed by mining operations with data mining algorithms. Additionally, in text mining, sentiment analysis can be performed based on a lexicon.[8].

2.5. *Text Preprocessing*

Text preprocessing is an initial stage in the text processing workflow to prepare the text for further analysis. It consists of several document cleaning steps, including *cleaning text, case folding, tokenizing, normalization, stopword removal,* and *stemming*[9].

2.6. Python

Python is a dynamic, high-level programming language and an interpreted language, meaning it converts source code into machine code directly when the program is run[10]. Python programming language utilizes various libraries and frameworks for data analysis. The libraries used in this research are Pandas, NLTK (*Natural Language Toolkit*), Sklearn, *Snscrape*, and Sastrawi.

2.7. *Term Frequency Invers Document Frequency* (TF-IDF)

The Term Frequency-Inverse Document Frequency (TF-IDF) method is a way of weighting the relevance of a word (term) to a document. This method is used to assign a high weight to each term that frequently appears in a specific document but not in many documents within the corpus[11]. This method combines two concepts to calculate the weight, namely the frequency of occurrence of a word in a particular document (TF) and the inverse frequency of documents containing that word (IDF). Term Frequency (TF) refers to the frequency of a word's occurrence in each document. Meanwhile, Inverse Document Frequency (IDF) reduces the dominance of terms that frequently appear across various documents. Words that appear in fewer documents should be considered more important than words that appear in many documents[12]. The general formula for TF-IDF weighting is as follows[13]:

$$idf_t = log\frac{N}{df_t}$$

$$tf - idf_{d,t} = tf_{d,t}\ x\ idf_{d,t}$$

Where:

tf = the number of words searched in a document

idf_t= Inversed Dokumen Frequency

N = total number of documents

df_t= number of documents containing term t

d = document number d

t = word number t from the keywords

$tf - idf_{d,t}$= weight of document d for word-t

2.8. *Naïve Bayes Classifier*

Naïve Bayes Classifier is a classification with probability and statistical methods proposed by British scientist Thomas Bayes. It predicts future probabilities based on past data, known as Bayes' Theorem. The performance accuracy level of the system built using the *naïve bayes* algorithm depends on the data available and the data used as training data. If the data taken as training data can represent all or most of the data available, the classification system will have good performance. If a classification system has good performance, it can be implemented to classify larger datasets[14]. The general formula for Bayes' Theorem can be written as follows [15]:

$$\frac{P(H \mid X) = P(X \mid H)\, P(H)}{P(X)}$$

Where:

X = Data with unknown classs

H = Hypothesis X data is class specific

P(H|X) = Probability of hypothesis H based on condition x (a posteriori prob.)

P(H) = Probability of hypothesis H (prior prob)

P(X|H) = Probability of X based on the condition

P(X) = Probability of X

2.9. *Support Vector Machine (SVM)*

SVM is an abstract learning machine that learns from a set of training data and attempts to generalize and make correct predictions for new data[16]. SVM is used to find the best hyperplane by maximizing the distance between classes. Hyperplane is a function that can be used to separate classes [17]. The best separating hyperplane between the two classes can be found by measuring the margin of the hyperplane and finding its maximum point. Where the margin is the distance between the hyperplane and the data from each class.

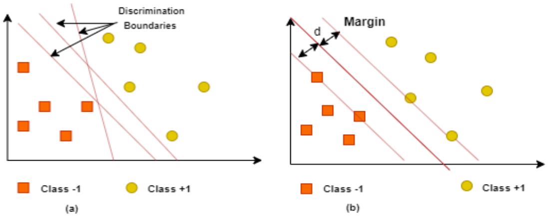


Figure 1. SVM process in finding the hyperlane between two classes

2.10. *Classification Report*

Classification report will display the performance results of *accuracy, precision, recall, fi-score and support of* the model that has been created. By using the classification report, it will provide a better understanding of the overall performance of the model that has been trained[18].

2.11. Evaluasi *Confusion Matrix*

Confusion matrix is a table that states the classification of the number of correct test data and the number of incorrect test data. *Confusion matrix* consists of a two-dimensional matrix, where each row of the matrix represents the actual class of the data, and each column represents the predicted class of the data (or vice versa)[19]. In the confusion matrix, there are four terms used to represent the results of the classification process, namely:

a. *True Positive* (TP), when the predicted class is positive and the fact is positive.
b. *True Negative* (TN), when the predicted class is negative, and is in fact negative.
c. *False Positive* (FP), when the predicted class is positive and the fact is negative.
d. *False Negative* (FN), when the predicted class is negative and the fact is positive

2.12. *K-Fold Cross Validation*

Cross Validation is one of the statistical methods used to evaluate and compare learning algorithms by dividing the data into two parts: one is used to learn or train the model and the other is used to validate the model[20]. The basic form of *cross-validation* is *k-fold cross validation* which works by splitting the data into k or equal (or nearly equal) sized segments.

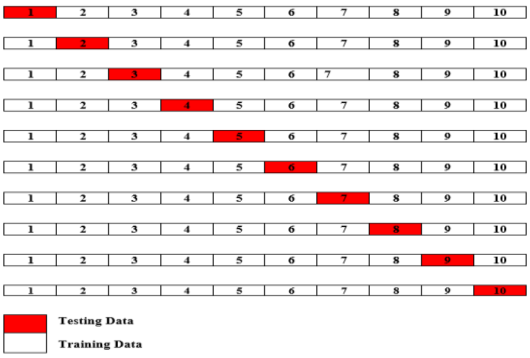


Figure 2. Schematic of 10-*Fold Cross Validation*

2.13. *Cross Industry Standard Process For Data Mining* (CRISP-DM)

Cross Industry Standard Process For Data Mining (CRISP-DM) is a method that is widely used to solve various business problems and has a relationship with data mining. This methodology consists of six phases, namely *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation,* and *Deployment*[21].

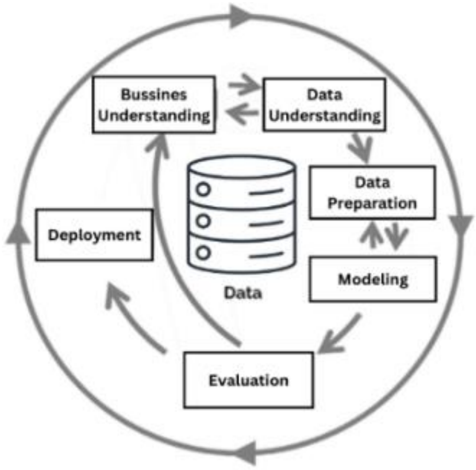


Figure 3. Stages of CRISP-DM Methodology

2.14. Related Work

There are previous studies that have a relationship with this research which is used as a reference and comparison of the methods used to get the expected results.

Hendy Syuhada, in 2022, conducted a study titled “Sentiment Analysis of the Corruption Eradication Commission (KPK) on Twitter Using the Naïve Bayes Classifier Algorithm,” achieving an accuracy rate of 64%[22].

Furthermore, Melati Indah Petiwi, Agung Triayudi, and Ira Diana Sholihati in 2022 conducted research with the title "Gofood Sentiment Analysis Based on Twitter Using *Naïve Bayes Classifier* and *Support Vector Machine* Methods". The result showed an accuracy of 83% for SVM, while the accuracy of Naïve Bayes was 74.6% .So it can be proven that the SVM method is more accurate as a clustering method for the sentiment analysis process [23].

Raihan Fais Sya'bani, Uktach Enri, and Tesa Nur Padilah conducted research in the year with the title "Sentiment Analysis of 2024 Presidential Candidates with *Naïve Bayes* Algorithm". The results of Ridwan's dataset obtained an accuracy value of 62.5% and AUC 0.65, then Prabowo's dataset with 60% accuracy and AUC 0.92, dataset from Anies with 71.43% accuracy and AUC 1.0, and dataset from Ganjar with an accuracy of 73.68% and AUC 0.74 [24].

Klaifer Garcia and Lilian Berton in 2020 conducted research with the title "Topic detection and sentiment analysis in Twitter content related to COVID 19 from Brazil and the USA". In this study, three algorithm methods were used, namely Support Vector Machine, Random Forest, and Logistic Regression. The results of the study show that in the negative class classification the three methods have the same F-1, but in the positive class classification Linear SVM shows the best performance with an F1 Score value of 0.66 compared to the other two methods [25].

Alexander Pak and Patrick Paroubek conducted a research entitled "Twitter as a Corpus for Sentiment Analysis and Opinion Mining" using the *naïve bayes* method, then compared it with the SVM and CRF algorithms. The final conclusions of this study include using *feature exctraction, unigram, bigram,* and *trigam*, an accuracy value of 80% is obtained [26]. On work [27] exploring SVM and Naive Bayes for sentiment analysis.

3. RESEARCH METHODOLOGY

In this research, the method used is the *Cross Industry Standard Process for Data Mining* (CRISP-DM). CRISP-DM provides a process in the problem-solving strategy of a business or research unit using appropriate data mining. The CRISP-DM method has six phases, as described in the flowchart below:

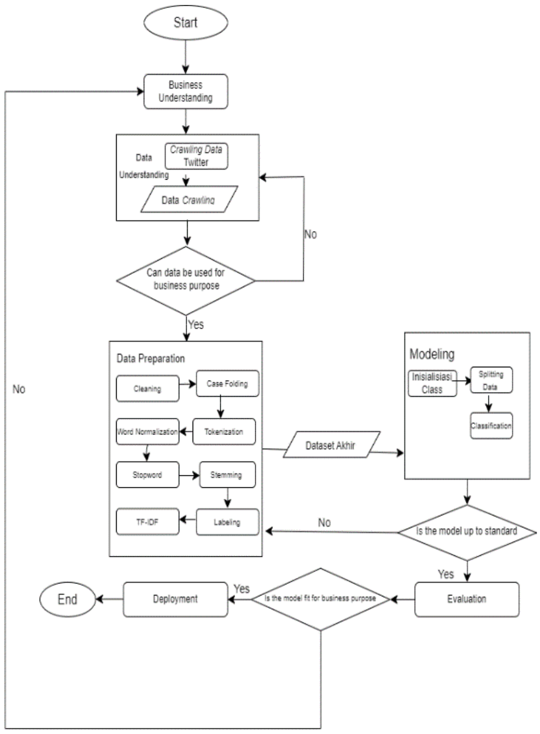


Figure 4. Research Flow Chart

Based on the flowchart of the CRISP-DM method, the phases carried out in this study are as follows:

3.1. *Business Understanding*

At this stage, the business objective of the research is to create a system to classify sentiment based on public *tweets* regarding the 2024 presidential candidates in Indonesia. Then classify sentiment using the *Naïve Bayes Classifier* and *Support Vector Machine* algorithms*,* then analyze the performance of the two methods in classifying *tweet* data, and create a website that is used to visualize the data.

3.2 *Data Understanding*

At this stage, an understanding of the data requirements is carried out. Where in this study, the dataset used is *tweets* from the public obtained through twitter social media. The *tweet* data is taken through the *crawling* process with the *snscrape library* from python. The data taken from twitter is data from January-May 2023 with the amount of data for each dataset is 3000 data. The dataset obtained at this stage is stored in csv form.

3.3 *Data Preparation*

In the data preparation stage, the final dataset is processed to be used in the modeling stage. To process the dataset, *pre-processing* and TF-IDF weighting are carried out. The *pre-processing* stage includes various processes such as *cleaning, case folding, tokenization, filtering (stopword removal).* Next, the data labeling process is carried out, namely giving labels or classes to user *tweets* manually which are then stored in the dataset to be included in the modeling stage. In this study using 3 labels, namely positive, neutral, and negative. Then after the data has been labeled, the next step is word weighting, word weighting in this study using the *Term Frequency Inverse Document Frequency* (TF-IDF) method.

3.4 *Modelling*

In the modeling stage, the algorithms used are *Naïve Bayes Classifier* and *Support Vector Machine.* Before entering the modeling stage, the dataset is first divided into two parts: training data and testing data. The ratio used for the training and testing data is 80:20, or 80% training data and 20% testing data.

3.5 *Evaluation*

In this study, the evaluation used to measure the performance of the *naïve bayes classifier* and *support vector machine* model is using a *classification report* that provides information about various evaluation metrics to measure how well the classification model can predict. These evaluation metrics show the level of accuracy, *precission, recall, and F1-Score*. Then the model that has been made will be tested using the *confusion matrix* and *cross validation* methods, to determine the performance of the model in classifying new data

3.6 *Deployment*

At this stage all the information that has been obtained is visualized on a website so the users can more easily and clearly access the information needed. This website was built using the django framework. On the website, a new dataset can be inputted, then the system will predict the sentiment of the dataset that has been inputted.

4 RESULT AND DISCUSSIONS

4.1 *Business Understanding*

Based on the results of assessing the situation regarding the upcoming presidential election in 2024 in Indonesia, comments and opinions—both positive and negative—about the presidential candidates who will run for 2024 presidential election in Indonesia are spread on Twitter social media. Thus, based on the assessment of the situation, the business objective is to create a sentiment classification system stemming from public tweets regarding the 2024 presidential candidates in Indonesia. Furthermore, text mining is applied to classify sentiment applying the Naïve Bayes classifier and Support Vector Machine algorithms and subsequently analyse the performance of the two algorithms in classifying data.

4.2 *Data Understanding*

Data understanding is done by the process of collecting datasets through the data crawling process applying the snscrape library from python with the keywords Anies Baswedan and Ganjar Pranowo. In this research, the dataset used is tweets from the public gained through Twitter media. The data taken from Twitter is data with a time span of January-May 2023.

```
import pandas as pd

import snscrape.modules.twitter as sntwitter

maxTweets = 3000

data_tweets = []


for i,tweet in
enumerate(sntwitter.TwitterSearchScraper('anie
s baswedan & ganjar pranowo since:2023-01-01
until:2023-06-01 lang:id').get_items()):

    if i>maxTweets:

        break

    data_tweets.append([tweet.date,
  tweet.username,tweet.content.encode("utf-
  8")])
```

Figure 5. Crawling Process of Anies Baswedan & Ganjar Pranowo Twitter Data

Here are the crawling results of both datasets:

Figure 6. The Result of Anies Baswedan Data Crawling

Figure 7. The Result of Ganjar Pranowo Data Crawling

Figures 6 and 7 show the results of data crawling with the keywords Anies Baswedan and Ganjar Pranowo. The successfully obtained data will be saved in a CSV file.

4.3 *Data Preparation*

The initial stage in the data preparation process is processing the results of *crawling* data that has been obtained previously through the *preprocessing* stage. Later on, the word weighting process is accomplished using the TF-IDF method. The data labelling process—which is a process intends to manually label or class the tweet data and then store them in the dataset—further, is performed after word weighting process finished.

4.3.1 *Preprocessing* Data

Several processes in data preprocessing stage such as data cleaning, case folding, tokenization, word normalization, stopword removal, and stemming are enforced. This stage is where the data are processed into more structured data to be entered into the next stage. The following are the stages of the preprocessing process:

a. *Cleaning* Data

```
def cleaning(Text):
    Text = re.sub(r'\$\w*', '', Text)
    Text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', ' ', Text)
    Text = re.sub('&quot;'," ", Text)
    Text = re.sub(r"\d+", " ", str(Text))
    Text = re.sub(r"\b[a-zA-Z]\b", "", str(Text))
    Text = re.sub(r"[^\w\s]", " ", str(Text))
    Text = re.sub(r'(.)\1+', r'\1\1', Text)
    Text = re.sub(r"\s+", " ", str(Text))
    Text = re.sub(r'#', '', Text)
    Text = re.sub(r'[^a-zA-z0-9]', ' ', str(Text))
    Text = re.sub(r'\b\w{1,2}\b', '', Text)
    Text = re.sub(r'\s\s+', ' ', Text)
```

Figure 8. The Cleaning Data Process

Figure 8 shows the process at the data cleaning stage, using the regex library to remove some non-alphanumeric characters in the text. Then remove all URLs or web links from the text, remove words with a length of one or two characters in the text, and remove emoticons and symbols in the text.

b. *Case Folding*

```
tweets['case_folding'] = tweets['cleaning'].str.lower()
tweets.head()
```

Figure 9. Case Folding Process

Figure 9 is the process of case folding, where at this stage all the results of cleaning the data that has previously been obtained will be changed from capital letters to lowercase letters using the str.lower() function.

c. *Tokenization*

```
nltk.download('punkt')
from nltk.tokenize import word_tokenize

def word_tokenize_wrapper(Text):
    return word_tokenize(Text)

tweets['tokenisasi'] = tweets['case_folding'].apply(lambda x: word_tokenize_wrapper(x.lower()))
tweets.head()
```

Figure 10. Tokenization Process

Figure 10 shows the process of tokenization, where this process uses the NLTK library to separate sentences into words that have been obtained from the previous case folding results.

d. *Word Normalization*

```
def normalization (Text):
  tweets_slang = pd.read_csv('colloquial-indonesian-lexicon.csv')
  dict_slang ={}
  for i in range(tweets_slang.shape[0]):
    dict_slang[tweets_slang["slang"][i]]=tweets_slang["formal"][i]

  drop_slang = []
  for teks in Text:
    normalisasi_teks = [dict_slang[word] if word in dict_slang.keys() else word for word in teks]
    drop_slang.append(normalisasi_teks)

  return drop_slang

tweets['normalisasi'] = normalization(tweets['tokenisasi'])
tweets.head()
```

Figure 11. *Word Normalization* Process

Figure 11 is the process of *word normalization*. In this process the words that have been obtained in the tokenization process and contain non-standard language will be converted into standard language according to the Colloquial Indonesian Lexicon dictionary.

e. *Stopword Removal*

```
nltk.download('stopwords')

from nltk.corpus import stopwords

list_stopwords =
stopwords.words('indonesian')

list_stopwords.extend(['yg', 'dg', 'rt', 'dgn',
'ny', 'gt', 'klo', 'kalo', 'amp', 'biar', 'gak', 'ga',
'krn', 'nya', 'nih', 'sih',  'si', 'tau', 'tdk', 'tuh',
'utk', 'ya','jd', 'jgn', 'sdh', 'nyg', 'hehe', 'pen',
'u', 'nan', 'loh', 'rt', '&', 'yah', 'no', 'je'])

list_stopwords = set(list_stopwords)

def stopwords_removal(Text):

 return [word for word in Text if word not
in list_stopwords]
```

Figure 12. *Stopword Removal* Process

Figure 12 is the process of stopword removal. In this process using the NLTK library, it will eliminate words that are not important or have no meaning contained in the document and have been processed previously at the word normalization stage.

Table 1. The Preprocessing Results of Anies Baswedan Dataset

Preprocessing	*Tweet*
Raw data	Maha karya orang bego gk bisa kerja @aniesbaswedan @AniesRelawan1 @Relawananies liat itu sumur resapan..
Cleaning Text	Maha karya orang bego kerja liat itu sumur resapan..
Case Folding	maha karya orang bego kerja liat itu sumur resapan
Tokenization	[‘maha’, ‘karya’, ‘orang’, ‘bego’, ‘bisa’, ‘kerja’, ‘liat’, ‘itu’, ‘sumur’, ‘resapan’]
Word Normalization	[‘maha’, ‘karya’, ‘orang’, ‘bego’, ‘bisa’, ‘kerja’, **‘lihat’,** ‘itu’, ‘sumur’, ‘resapan’]
Stopword Removal	[‘maha’, ‘karya’, ‘orang’, ‘bego’, ‘kerja’, ‘lihat’, ‘sumur’, ‘resapan’]
Stemming	[‘maha’, ‘karya’, ‘orang’, ‘bego’, ‘kerja’, ‘lihat’, ‘sumur’, **‘resap’**]

Table 2. The Preprocessing Results Ganjar Pranowo Dataset

Preprocessing	*Tweet*
Data Mentah	@HukumDan @ganjarpranowo Udah ngobrol sama novel tentang ektp ?
Cleaning Text	HukumDan ganjarpranowo Udah ngobrol sama novel tentang ektp
Case Folding	hukumdan ganjarpranowo udah ngobrol sama novel tentang ektp
Tokenization	[‘hukumdan’, ‘ganjarpranowo’, ‘udah’, ‘ngobrol’, ‘sama’, ‘novel’, ‘tentang’, ‘ektp’]
Word Normalization	[‘hukumdan’, ‘ganjarpranowo’, **‘sudah’, ‘mengobro**l’, ‘sama’, ‘novel’, ‘tentang’, ‘ektp’]
Stopword Removal	[‘hukumdan’, ‘ganjarpranowo’,’mengobrol’,’novel’, ‘ektp’]

Stemming	[‘hukumdan’, ‘ganjarpranowo’,’obrol’,’novel’, ‘ektp’]

4.3.2 Data Labeling

This process is carried out to label the tweets resulting from crawling data. Then it will be stored in the dataset for further use in the modeling process. This labeling process is done manually by categorizing words for sentiment based on connotations or common meanings associated with these words.

Table 3. Data Labeling Process

Usernam e	*Tweet*	*Stemmin g*	Label
SenoWer kudoro5	jadi mantan gubernur unfaedah BOTOL dki hahaaha...	mantan gubernu r unfaeda h botol dki hahaaha	Negatif
Aniesprsd n2024	Pak Anies Baswedan pilihan rakyat Indonesia	anies basweda n pilih rakyat indonesi a	Positif
haimasde p1	Dinanti apresiasi yg benar bersama telah menciptak an lagu untuk bersemang at dalam berjuang mendukun g...	nanti apresiasi cipta lagu semanga t juang dukung ...	Positif
Sarwoed5 8181867	Maaf klo sy bingung mau komen apa nyatanya yg diucapkan kadang gk sesuai fakta dilapangan ...	maaf bingung komen kadang sesuai fakta lapang...	Negatif
ahsancikn usi	Kita lihat saja, cara	lihat kerja	Netral

	kerja akan memperlihat hasil kok...	lihat hasil	

4.3.3 Word Weighting

Word weighting is carried out by assigning values to all terms found in tweets that have been preprocessed, where in this study, word weighting is performed using the TF-IDF (Term Frequency – Inverse Document Frequency) method. The use of the TF-IDF method is due to its ability to balance word frequency by assigning higher weights to words that frequently appear in a document, then the results of this word weighting will be visualized in the form of a wordcloud. The following are the results of wordcloud visualization on each dataset.

A. Word Weighting on Anies Baswedan Dataset

Figure 13. Visualization of Positive Sentiment Wordcloud on Anies Baswedan Dataset

Figure 14. Visualization of Neutral Sentiment Wordcloud on Anies Baswedan Dataset

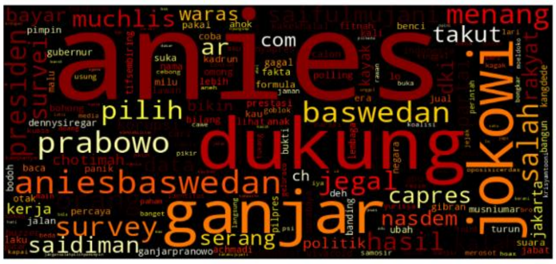


Figure 15. Visualization of Negative Sentiment Wordcloud on Anies Baswedan Dataset

Figures 13, 14, and 15 are the results of *wordcloud* visualization for Anies Baswedan dataset. Figure 13 shows the visualization of positive sentiment with the frequency of words that often appear, namely the words “Anies”, “Fairplay”, dan “Pimpin”. Meanwhile, Figure 14 shows a visualization of neutral sentiment with the frequency of words that often appear, namely “Anies”, Ganjar”, dan” survei”. While Figure 15 shows a visualization for negative sentiment with words that often appear, namely “Anies”, “dukung”, dan “salah”.

B. Word Weighting on Ganjar Pranowo Dataset

Figure 16. Visualization of Positive Sentiment Wordcloud on Ganjar Pranowo Dataset

Figure 17. Visualization of Neutral Sentiment Wordcloud on Ganjar Pranowo Dataset

Figure 18. Visualization of Negative Sentiment Wordcloud on Ganjar Pranowo Dataset

Figures 16, 17, and 18 is the result of *wordcloud* visualization for Ganjar Pranowo dataset. Figure 16 shows the visualization of positive sentiment with the frequency of words that often appear, namely the words "ganjarpranowo,", "dukung", dan "Presiden". While in Figure 17 shows the visualization of neutral sentiment with the frequency of words that often appear, namely "ganjarpranowo", "jalan", dan "manutjokowi". ". While Figure 18 shows the visualization for negative sentiment with words that often appear are "Ganjarpranowo", "Jokowi", dan "kalah".

4.4 *Modelling*

The *Naïve Bayes classifier* and *Support Vector Machine* algorithms are adopted for analysing the data. The *modelling* process begins with the initialisation of values or labels to identify and assign categories or labels that the classification model wants to predict or identify. Afterwards, the data *splitting* process is fulfilled by dividing the dataset into two parts, *training* data and *testing* data. Where the data ratio used in this study is 80:20 or 80% *training* data and 20% *testing* data from data taken with a total of 3000 tweet data from each dataset. The next process is classification which involves learning from previously created *training* data. The classification process utilises the *Naïve Bayes classifier* and *Support Vector Machine* algorithms. Later, the *machine learning* process will be carried out, where the training results of the *Naïve Bayes classifier* and *Support Vector Machine* algorithms that have been trained will be evaluated to measure accuracy in predicting new data.

```
model = MultinomialNB()

model.fit(features, label)

probability_to_be_positive =
model.predict_proba(features)
```

Figure 19. Data Classification Process Using Naïve Bayes Classifier Algorithm

```
model = SVC (probability=True)
model.fit(features, label)
probability_to_be_positive = model.
predict_proba (features) predict_output =
model.predict(features)
```

Figure 20. Data Classification Process Using *Support Vector Machine* Algorithm

4.5 *Evaluation*

The evaluation stage aims to determine the performance of the model that has been made before. This evaluation stage applies a classification report assessment to determine the performance of the model that has been made. By using the classification report method that displays several evaluation matrices, it, hence, can make better decisions about how to improve the model, specifically as the model is tested on diverse datasets. The following is a table of classification report results on both datasets using the *naïve bayes classifier* and *support vector machine* algorithm:

	Precision		*Recall*		*F1-Score*		*Accuracy*	
	Anies	Ganjar	Anies	Ganjar	Anies	Ganjar	Anies	Ganjar
Positif	91%	94%	86%	96%	89%	95%	87%	92%
Netral	81%	78%	82%	70%	82%	74%		
Negatif	82%	80%	89%	74%	85%	77%		
Weighted avg	87%	91%	87%	92%	87%	92%		

Figure 21. Classification Report Result of Naïve Bayes Classifier Algorithm

	Precision		*Recall*		*F1-Score*		*Accuracy*	
	Anies	Ganjar	Anies	Ganjar	Anies	Ganjar	Anies	Ganjar
Positif	92%	80%	88%	99%	90%	88%	88%	82%
Netral	95%	94%	68%	28%	79%	44%		
Negatif	84%	95%	93%	52%	89%	67%		
Weighted avg	89%	84%	88%	82%	88%	80%		

Figure 22. Classification Report Result of Support Vector Machine Algorithm

In the evaluation results applying the classification report method, it is shown that each dataset has a different accuracy value for each algorithm. In Anies Baswedan figure dataset, the accuracy value of the Support Vector Machine algorithm has a value that is superior to the *Naïve Bayes classifier* algorithm, where the accuracy value of the *Support Vector Machine* algorithm is 88% and the accuracy value of the *Naïve Bayes classifier* is 87%. Whilst, in the dataset of Ganjar Pranowo figure, the results depict that the *Naïve Bayes classifier* algorithm has a superior accuracy than the *Support Vector Machine* algorithm, where the accuracy value of the *Naïve Bayes classifier* algorithm is 92% and the accuracy of the *Support Vector Machine* algorithm is 82%. Based on the accuracy results obtained, it points out that the dataset has an influence on the accuracy results of the two algorithms. In this case, the number of positive, neutral, or negative sentiment labels in the dataset has an unbalanced or disproportionate value so that it can affect the accuracy value, where the Anies Baswedan figure dataset has a percentage distribution for positive labels of 46.60%, neutral of 16.10%, and negative of 37.30%. The distribution of labels on the Anies Baswedan figure dataset shows that the neutral label has a much smaller number of samples compared to the positive and negative labels. Simultaneously, the Ganjar Pranowo character dataset has a percentage distribution for positive labels of 58.83%, neutral labels of 16.77%, and negative labels of 24.40%. The distribution of labels in Ganjar Pranowo figure dataset displays that positive labels dominate with almost 58% of the total samples, while neutral and negative labels have a smaller number of samples. However, as it is observed as a whole from both datasets, the *Naïve Bayes classifier* algorithm tends to have superior accuracy performance compared to the *Support Vector Machine* algorithm.

In addition, classification testing was also carried out by taking a new dataset with the previous model. The data that had previously been preprocessed were employed to make predictions utilising the new dataset. The new data used is 11,407 for the Anies Baswedan character dataset. While for the Ganjar Pranowo character dataset using 6,226 new data. Based on the prediction of sentiment classification, the data results are as revealed below:

Table 4. Sentiment Classification Results on Anies Baswedan Dataset

Algoritma	Sentiment Classification		
	Positif	Negatif	Netral
Naïve Bayes Classifier	6263	4415	729
Support Vector Machine	5420	4877	1110

Table 5. Sentiment Classification Results on Ganjar Pranowo Dataset

Algorithm	Sentiment Classification		
	Positif	Negatif	Netral
Naïve Bayes Classifier	5120	886	220
Support Vector Machine	4302	1418	506

As predicting classification on new datasets, evaluation was also conducted to determine the performance of the *Naïve Bayes classifier* and *Support Vector Machine* classification models. In this case, the evaluation methods used are k- fold cross validation and confusion matrix.

Table 6. Evaluation of K-Fold Cross Validation on Anies Baswedan DatasetAnies Baswedan Dataset

Algorithm	Algorithm K-Fold Cross Validation Accuracy Results
Naïve Bayes Classifier	87.5%
Support Vector Machine	86%

Table 7. Evaluation of K-Fold Cross Validation on Anies Baswedan DatasetGanjar Pranowo

Algorithm	Algorithm K-Fold Cross Validation Accuracy Results
Naïve Bayes Classifier	90.4%
Support Vector Machine	86.6%

Tables 6 and 7 are the results of the average accuracy value utilising 10 k-fold cross validation on each dataset. As seen in the tables, the *Naïve Bayes classifier* algorithm on each dataset has a higher accuracy value than the *Support Vector Machine* algorithm as it tested using the k-fold cross validation method. Applying the k-fold cross validation method will assist to evaluate the overall performance of the model whether the model that has been created can generalise sentiment patterns well on new data that has never been seen before. Again, this helps avoid the possibility of overfitting on certain data. Based on the accuracy results above, it is known that the performance of the model created is good and has been successful in classifying new datasets that have never been recognised by the model before.

Furthermore, the confusion matrix method is also used to evaluate the performance of the model by comparing the model predictions with the actual values of the data. This supports in illustrating the extent to which the model actually predicts the class correctly and where the model makes mistakes. The following are the results of the confusion matrix on both datasets applying the *Naïve Bayes classifier* and *Support Vector Machine* algorithms.

		Assigned Class	
		Positive	Negative
Actual Class	Positive	1070 (TP)	158 (FN)
	negative	92 (FP)	800 (TN)

Figure 23. Confusion Matrix Table for *Naïve Bayes Classifier* Algorithm on Anies Baswedan Dataset

		Assigned Class	
		Positive	Negative
Actual Class	Positive	973 (TP)	34 (FN)
	negative	50 (FP)	141 (TN)

Figure 24. Confusion Matrix Table for *Naïve Bayes Classifier* Algorithm on Ganjar Pranowo Dataset

Figure 23 and Figure 24 depict the results from the confusion matrix for each dataset using the Naïve Bayes classifier algorithm. The result of confusion matrix for Anies Baswedan dataset is shown in Figure 23, while in Figure 24 is the result of confusion matrix for Ganjar Pranowo dataset. Based on the data prediction results that have been acquired in the confusion matrix above, the accuracy, precision, recall, and f1-score values of the *Naïve Bayes classifier* algorithm can be obtained as well for each of the datasets above. The following are the results of the evaluation metrics of each dataset applying the Naïve Bayes classifier algorithm:

Metrik Evaluasi	Anies Baswedan	Ganjar Pranowo
Precision	92%	95%
Recall	87%	96%
F1-Score	92%	95%
Accuracy	89%	95%

Figure 25. Evaluation Metrics for *Naïve Bayes Classifier* Algorithm

Figure 25 contains the results of the evaluation metrics. The higher precision value is gained on the Ganjar Pranowo dataset which is 95%, this represents that the model on the Ganjar Pranowo dataset is better at identifying true positive values compared to the Anies Baswedan dataset, it occurs as the model predicts a positive class, the Ganjar Pranowo dataset has a higher probability of being correct than the Anies Baswedan dataset. Afterwards, the recall value is also higher on the Ganjar Pranowo dataset than on the Anies Baswedan dataset. This indicates that on the Ganjar Pranowo dataset, the model is able to identify positive classes correctly which means that there are only a few positive classes that are not detected by the model (low false negative). Moreover, there is an f1-score value which is the combined result of precision and recall. F1-score gives an idea of how well the model classifies both positive and negative classes accurately. A higher F1-score is reached by the Ganjar Pranowo dataset which points out a better balance between precision and recall compared to the Anies Baswedan dataset. Additionally, there is an accuracy value, where on the Ganjar Pranowo dataset has a higher value of 95% compared to the Anies Baswedan dataset which recovered an accuracy of 89%. Besides, on the Ganjar Pranowo dataset, it contains a higher accuracy value with total of 95% than on the Anies Baswedan dataset with 89% accuracy. This specifies that the model makes more correct predictions on the Ganjar Pranowo dataset.

Further, testing was carried out by applying the Support Vector Machine algorithm with the confusion matrix method. The following is a confusion matrix table on the *Support Vector Machine* algorithm for each Anies Baswedan and Ganjar Pranowo character dataset:

		Assigned Class	
		Positive	Negative
Actual Class	Positive	943 (TP)	126 (FN)
	negative	61 (FP)	934 (TN)

Figure 26. Confusion Matrix Table of *Support Vector Machine* Algorithm on Anies Baswedan Dataset

		Assigned Class	
		Positive	Negative
Actual Class	Positive	837 (TP)	5 (FN)
	negative	143 (FP)	157 (TN)

Figure 27. Confusion Matrix Table of *Support Vector Machine* Algorithm on Ganjar Pranowo Dataset

Figure 26 and Figure 27 reveal the results of the confusion matrix table in the *Support Vector Machine* algorithm for each dataset. Table 26 is the Confusion Matrix table for the Anies Baswedan dataset. The result of Confusion Matrix for Ganjar Pranowo dataset is depicted in Figure 27 Based on the data prediction results in the confusion matrix above, the accuracy, precision, recall, and f1-score values of the Support Vector Machine algorithm can also be gathered for each of the datasets above. The following are the results of the evaluation metrics of each dataset using the *Support Vector Machine* algorithm:

Metrik Evaluasi	Anies Baswedan	Ganjar Pranowo
Precision	93%	85%
Recall	88%	99%
F1-Score	90%	91%
Accuracy	90%	87%

Figure 28. Evaluation Metrics for *Support Vector Machine* Algorithm

Based on the results of the evaluation metrics in Figure 28, it is represented that the higher precision value is contained on the Anies Beswedan character dataset with a value of 93%, while the Ganjar Pranowo dataset had a precision value of 85%. Again, the recall value with a higher value found on the Ganjar Pranowo dataset which is 99% compared to the Anies Baswedan dataset with 88% recall value. This identifies that in Ganjar Pranowo dataset the model is able to detect almost all positive classes in the dataset. Furthermore, there is an f1-score value which is the combined result of precision and recall. Ganjar Pranowo dataset holds a higher f1-score by the number of 91% which shows a better balance between precision and recall compared to the Anies Baswedan dataset with 90% value. Likewise, there is the accuracy value where on the Anies Baswedan dataset has a higher value of 90% compared to the Ganjar Pranowo dataset which obtained an accuracy of 87%. On the Anies Baswedan dataset, the model has higher precision and accuracy values, indicating that the model is better at correctly identifying positive classes and has a good balance between the number of correct and incorrect predictions. While on the Ganjar Pranowo dataset the model has a very high recall and a slightly higher F1-Score, indicating that the model is very good at recognising positive classes.

4.6 *Deployment*

At the deployment stage, the model that has been successfully created is poured into a website to visualise the data. In this research, the analysis system is made based on the model from the results of the previous training dataset which has been stored in the form of a pickle. Furthermore, the model is used to predict new data inputted on the website.

4.6.1 System Design Using Django Framework

Django framework is applied in the process of creating a website, where Django implements MTV architecture or Model, View, and Template. The View part will manage requests from HTTP, it will take input from the user, then process the input and send a response. The View contains functions or classes which will receive the HTTP request as input and return the HTTP response as output. Subsequently, the Template section contains HTML pages that contain data from the view. URL configuration will map the URL to the appropriate view. The complete code of this system can be accessed in the barcode below:

4.6.2 System Tests

On the website, there is a feature to classify data utilising a new dataset. Datasets in CSV format can be inputted on the website, the preprocessing process is later carried out automatically on the website, then the system will display the results of sentiment classification in the form of positive, neutral, or negative sentiment from both algorithms, *namely the Naïve Bayes classifier* and *Support Vector Machine* algorithm.

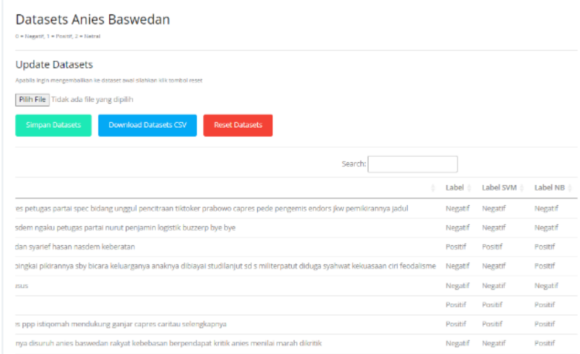


Figure 29. Display of Sentiment Classification Results on Anies Baswedan Dataset

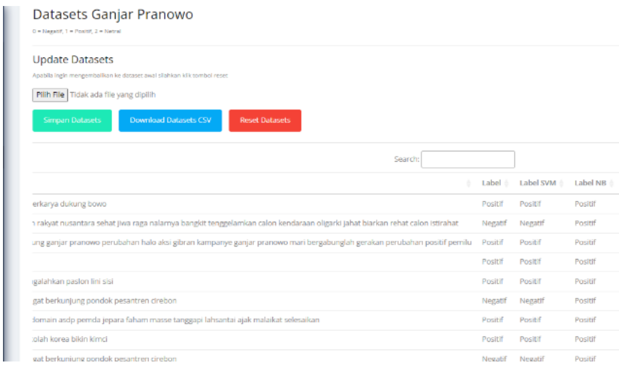


Figure 30. Display of Sentiment Classification Results on Ganjar Pranowo Dataset

Figures 29 and 30 depict the results of sentiment classification applying the Naïve Bayes classifier algorithm and Support Vector Machine with the new dataset inputted on the website. This page features the tweets that have gone through the preprocessing process along with sentiment results in the form of positive, neutral, or negative sentiment. On the data visualisation page, the number of sentiment classification results will be visualised.

4.6.3 Data Visualization

The visualisation represents the number of data classification results from both algorithms and from each dataset. The following is an example of the results of the data visualisation display on the website after inputting a new dataset:

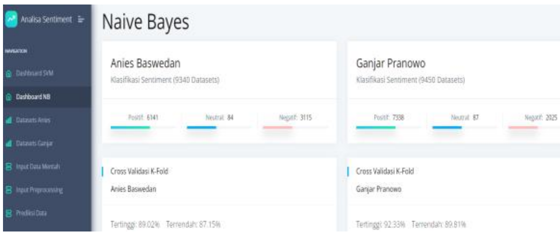
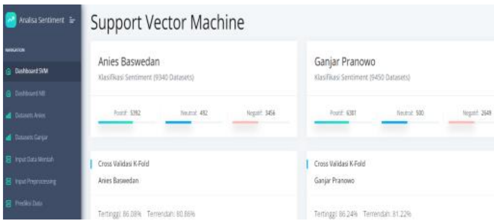


Figure 31. Data Visualization for *Naïve Bayes* Classifier Algorithm



Gambar 32. Data Visualization for *Support Vector Machine* Algorithm

Figures 31 and 32 show the number of results from sentiment classification using the Naïve Bayes classifier algorithm and Support Vector Machine on each dataset. Figure 31 is the result of visualisation applying the Naive Bayes classifier algorithm. It reveals that on the Anies Baswedan dataset—from the total existing data, there are 9340 data—with details where the first 3000 data is the initial training data when making the model with a time span of January 2023-May 2023 and 6340 data is a new dataset inputted with a time span of June 2023-December 2023. Whereas, on the Ganjar Pranowo dataset of the total data available is 9450 data; with details where the first 3000 data is the initial training data when making the model with a time span of January 2023-May 2023; and 6450 data is a new dataset inputted with a time span of July 2023-December 2023. The classification results of the Support Vector Machine algorithm employing the same data details as in the previous Naïve Bayes classifier algorithm are depicted in Figure 32

4.6.4 New Data Deployment

The model in this test always changes (makes a new model) as the new data is successfully inputted and will be combined with the previous initial data which will result in an increase in the number of datasets, so that the model undergoes training again based on the dataset. The dataset is a dataset with a time range of July 2023-December 2023. This new data will be compared with the initial data to determine the accuracy comparison of the initial data and new data after the model training process is carried out again. The following is a comparison of evaluations on the initial data and new data:

Table 8 Accuracy Value of Initial Data

Initial Dataset	Model Accuracy	
	Naïve bayes Classifier	*Support Vector Machine*
Anies Baswedan	87%	88%
Ganjar Pranowo	92%	82%

Table 9 Accuracy Value of New Testing Data

New Testing Dataset	Model Accuracy	
	Naïve bayes Classifier	*Support Vector Machine*
Anies Baswedan	89,02%	86,08%
Ganjar Pranowo	92,33%	86,24%

Table 8 represents the accuracy values of the model from the initial data, using a training dataset of 3,000 data points. As seen in the table that for the Anies Baswedan dataset, the accuracy of the *Support Vector Machine* algorithm is 88% and the accuracy of the *Naïve Bayes classifier* is 87%. Also, for the Ganjar Pranowo dataset, the *Naïve Bayes classifier* algorithm achieves an accuracy of 92% and the *Support Vector Machine* algorithm achieves an accuracy of 82%. As compared with the accuracy results from a new test dataset, where the Anies Baswedan dataset contains a total of 9,340 data points, the *Naïve Bayes classifier* algorithm achieves an accuracy of 89.02% and the *Support Vector Machine* algorithm achieves an accuracy of 86.08%. Besides, for the Ganjar Pranowo dataset—with a total of 9,450 test data points—the *Naïve Bayes classifier* achieves an accuracy of 92.33% and the *Support Vector Machine* algorithm achieves an accuracy of 86.24%. Since the model has undergone continuous learning due to the increase in the amount of data input, there is an increase in accuracy with the new data. This presents that the more data is used in the learning process, the higher the accuracy obtained.

CONCLUSIONS

The conclusions obtained based on the results of the research that have been carried out are as follows:

1. The *Naive Bayes Classifier* and *Support Vector Machine* algorithms have been successfully used to classify data using three sentiments, namely positive, neutral, and negative, according to the research's results. It is well recognised that high-dimensional data sets, like data from Twitter, can be used to apply the *Naive Bayes Classifier* and *Support Vector Machine* methods.
2. The *Naïve Bayes Classifier* algorithm has a greater level of accuracy and weighted *f1-score* value when compared to the *support vector machine* algorithm, according to the accuracy results produced using the Classification Report evaluation method. This demonstrates that the *Naïve Bayes Classifier* algorithm outperforms the *Support Vector Machine* approach in classification because it can effectively handle multiclass data without the requirement for supplementary methods. By including multiclass approaches, the *Support Vector Machine* algorithm will be used to multiclass data more effectively in the interim.
3. The *Naive Bayes Classifier* algorithm achieved the highest average accuracy value for each dataset, as determined by the accuracy results of testing the new dataset using the k-fold cross validation evaluation method. The *Naïve Bayes Classifier* algorithm yielded an accuracy value of 90.4% in the Ganjar Pranowo dataset, compared to 87.5% in the Anies Baswedan dataset. In the meantime, the *Support Vector Machine* algorithm's accuracy value was 86% for the Anies Baswedan dataset and 87.6% for the Ganjar Pranowo dataset. Other than that, the accuracy values in the Ganjar Pranowo dataset are often greater than those in the Anies Baswedan dataset.
4. Based on evaluation results obtained from testing the new dataset using the Confusion Matrix method, it was found that the *Naïve Bayes Classifier* algorithm outperformed the Ganjar Pranowo dataset with higher accuracy, precision, recall, and f1-score. This demonstrates that the *Support Vector Machine* approach outperforms the Anies Baswedan dataset with greater accuracy, whereas the Naïve Bayes Classifier algorithm is well appropriate for this dataset.
5. A visualisation of the newly input dataset is shown on the website. *Naïve Bayes Classifier* and *Support Vector Machine* are the two algorithms used to classify the data into positive, neutral, and negative sentiment. Then the dataset that has been input can be downloaded and produce the classification data in CSV form.

Acknowledgements

The author extends gratitude to the stakeholders that involved and provided support for this research.

REFERENCES

[1] Rahman, F. Rahmat, M. Y. Fariqi, and S. Adi, "Metode Naive Bayes untuk Menganalisis Akurasi Sentimen Komentar di Youtube," J. EECCIS (electrics, Electron. Commun. Control. informatics, Syst., vol. 14, no. 1, pp. 31–34, 2020.
[2] Y. Julianto, D. H. Setiabudi, and S. Rostianingsih, "Analisis Sentimen Ulasan Restoran Menggunakan Metode Support Vector Machine," J. Infra, vol. 10, no. 1, pp. 1–7, 2022.
[3] K. Weller, Twitter and Society. New York: Peter Lang, 2014.
[4] E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," IEEE Comput. Intell. Mag., vol. 9, no. 2, pp. 48–57, 2014, doi: 10.1109/MCI.2014.2307227.
[5] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, "Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends," Nat. Lang. Process. J., vol. 4, p. 100026, 2023, doi: https://doi.org/10.1016/j.nlp.2023.100026
[6] S. Bhatia, M. Sharma, and K. K. Bhatia, "Sentiment Analysis and Mining of Opinions," Stud. Big Data, vol. 30, no. May, pp. 503–523, 2018, doi: 10.1007/978-3-319-60435-0_20.
[7] Sasmita, G. A. Pradnyana, and D. G. H. Divayana, "Pengembangan Sistem Analisis Sentimen Untuk Evaluasi Kinerja Dosen Universitas Pendidikan Ganesha Dengan Metode Naïve Bayes," JST (Jurnal Sains dan Teknol., vol. 11, no. 2, pp. 451–462, 2022, doi: 10.23887/jstundiksha.v11i2.44384.
[8] M. R. Faisal, D. Kartini, T. H. Saragih, and A. Arrahimi, Belajar Data Science: Text Mining Untuk Pemula I. 2022.
[9] Y. Findawati, Buku Ajar Text Mining. Sidoarjo: UMSIDA PRESS, 2020. doi: 10.21070/2020/978-623-6833-19-3.
[10] Suharto, Fundamental Bahasa Pemrograman Python. Jawa Tengah: Eureka Media Aksara, 2023.
[11] C. Müller and S. T. A.-T. T.- Guido, "Introduction to machine learning with Python : a guide for data scientists." O'Reilly Media, Inc Sebastopol, CA, Sebastopol, CA, 2017. doi: LK- https://worldcat.org/title/960211579.
[12] S. Qaiser and R. Ali, "Text mining: use of TF-IDF to examine the relevance of words to documents," Int. J. Comput. Appl., vol. 181, no. 1, pp. 25–29, 2018.
[13] C. Manning, P. Raghavan, and H. Schutze, An Introduction to Information Retrieval. England: Cambridge University Press, 2009.
[14] T. M. Mitchell, Machine Learning. New York: McGraw-Hill, 1997.
[15] C. Aggarwal, Data Mining The textbook. London: Springer, 2015.
[16] Campbell and Y. Ying, Learning with Support Vector Machines. Morgan & Claypool Publishers series, 2011. doi: 10.2200/S00324ED1V01Y201102AIM010.
[17] Mahendro and D. Abimanto, Algoritma Support Vector Machine Untuk Menganalisa Kepuasan Mahasiswa Terhadap Pembelajaran E-Learning. Semarang: Cv Pustaka Stimart Amni, 2023.
[18] Kharwal, "Classification Report in Machine Learning." Accessed: Dec. 20, 2023. [Online]. Available: https://thecleverprogrammer.com/2021/07/07/classification-report-in-machine-learning/
[19] M. Ting, "Confusion Matrix BT- Encyclopedia of Machine Learning," C. Sammut and G. I. Webb, Eds., Boston, MA: Springer US, 2010, p. 209. doi: 10.1007/978-0-387-30164-8_157..
[20] P. Refaeilzadeh, L. Tang, and H. Liu, Encyclopedia of Database Systems. Boston, MA: Springer, 2009. doi: https://doi.org/10.1007/978-0-387-39940-9_565.
[21] S. and CP. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz and R. Wirth, CRISP-DM 1.0. The CRISP-DM consortium, 2020.
[22] H. Syuhada, "Analisis Sentimen Mengenai Komisi Pemberantasan Korupsi (Kpk) Pada Media Sosial Twitter Dengan Menerapkan Algoritma Naïve Bayes Classifier," Universitas Lampung, 2022.
[23] M. I. Petiwi, A. Triayudi, and I. D. Sholihati, "Analisis Sentimen Gofood Berdasarkan Twitter Menggunakan Metode Naïve Bayes dan Support Vector Machine," J. Media Inform. Budidarma, vol. 6, no. 1, p. 542, 2022, doi: 10.30865/mib.v6i1.3530.
[24] M. R. F. Sya'bani, U. Enri, and T. N. Padilah, "Analisis Sentimen Terhadap Bakal Calon Presiden 2024 Dengan Algoritme Naïve Bayes," JURIKOM (Jurnal Ris. Komputer), vol. 9, no. 2, pp. 265–273, 2022.
[25] Garcia and L. Berton, "Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA," Appl. Soft Comput., vol. 101, p. 107057,2021,doi:10.1016/j.asoc.2020.107057
[26] Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.," in LREc, 2010, pp. 1320–1326.

[27] Putri, D.D., Nama, G.F. and Sulistiono, W.E., 2022. "Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier. Jurnal Informatika dan Teknik Elektro Terapan, 10(1).