A COMPARATIVE STUDY OF SENTIMENT CLASSIFICATION: TRADITIONAL NLP VS. NEURAL NETWORK APPROACHES

Djarot Hindarto[1)*]

[1] Prodi Informatika, Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional Jakarta
Jl. Sawo Manila No.61, RT.14/RW.7, Pejaten Bar., Ps. Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12520
e-mail: djarot.hindarto@civitas.unas.ac.id[1)] *e-mail korespondensi: djarot.hindarto@civitas.unas.ac.id

ABSTRACT

The current research compares traditional natural language processing methods, such as Naive Bayes and Support Vector Machine, to neural network approaches, particularly Multi-Layer Perceptron, to classify positive and negative sentiments regarding company customer service. This research is motivated by the need to understand the effectiveness of these two approaches in analyzing and classifying sentiment in customer reviews, a crucial aspect of enhancing the quality of customer service. The author evaluated accuracy, speed, and adaptability to complex and diverse review content using a dataset containing various business customer reviews. The findings of this study indicate that neural network approaches, particularly Multi-Layer Perceptron, tend to provide superior performance in classifying customer sentiment with greater precision, albeit at a higher computational cost. Traditional methods such as Naive Bayes and Support Vector Machine still apply in situations with limited resources. The results of this research provide valuable guidance for companies in selecting an appropriate approach to analyzing customer sentiment, with the potential to increase understanding of customer views and improve overall customer service. Nave Bayes achieves 68.75% accuracy, Support Vector Machine achieves 87.5% accuracy, and Multi-Layer Perceptron achieves 100% accuracy.

Keywords: *Customer Service; Naïve Bayes; Support Vector Machine; Multi-Layer Perceptron; Natural Language Processing*

I. INTRODUCTION

In the ever-expanding digital era, text data has become one of the most valuable and widely used information sources in various spheres of life. Society is inundated with customer reviews, social media comments, news articles, blogs, and other text documents in the current information ecosystem. These text forms reflect individuals' views, emotions, and opinions regarding various topics, including products, services, news events, and social and political issues. Sentiment classification is one of the most significant relevant applications of text analysis. Sentiment classification seeks to determine whether a text contains positive or negative sentiment. It provides valuable insight into how society feels or perceives things and significantly impacts numerous fields.

The impact is extensive. Companies use sentiment classification [1] to analyze customer reviews and feedback on products and services, which assists with product and service enhancement, brand management, and customer service policies. In addition to using sentiment classification to analyze external feedback, companies also use it to analyse internal feedback, such as employee responses to job satisfaction surveys. Social media platforms and news organizations use sentiment classification to comprehend people's reactions to news and social events, enabling them to understand ongoing trends and perspectives. Additionally, sentiment classification is utilized in public policy and politics to comprehend people's approval or disapproval of particular issues, which can impact political and policy decisions.

It is necessary to recognize the significance of technology in sentiment classification. Technology has transformed text analysis, particularly in terms of sentiment classification. Initially, manual and rule-based methods were employed for text analysis. With the advent of natural language processing (NLP) techniques, neural networks can now automate this task with greater precision. Using NLP and neural network approaches, it can extract essential features from text automatically, dealing with complex language and varying contexts. This allows for a more precise and scalable sentiment classification in the vast amounts of text data generated by social media, websites, and other online sources. In an increasingly interconnected and information-rich environment, sentiment analysis is essential for understanding people's perspectives and making informed decisions. Comparative studies between traditional NLP methods and neural network approaches, such as those proposed in this study, are an essential step toward understanding how to utilize this abundance of text data more effectively for various practical purposes.

Initial approaches to sentiment classification relied on conventional Natural Language Processing (NLP) techniques, such as Naive Bayes and Support Vector Machine (SVM) [2],[3],[4]. This classification method employs manual feature extraction and statistical models. However, as technology advances, neural network approaches, such as Multi-Layer Perceptron (MLP) [5], have become extremely popular. This neural network can process text from beginning to end, removing the need for manual feature extraction. Although traditional NLP approaches have been used with some success in sentiment classification, questions arise as to whether neural network approaches can provide superior results. This study aims to answer this question by comparing traditional NLP and neural network approaches in the context of sentiment classification. The author is interested in the following research questions:

- Do neural network approaches, such as MLP, perform better than conventional methods in classifying the sentiment of customer texts? (Research Question 1).
- How well does this approach handle the complex and variable text frequently found in customer reviews? (Research Question 2).
- Are the increased computational costs associated with neural network approaches justified by the improved accuracy? (Research Question 3).

Natural Language Processing (NLP) has employed various detection techniques to perform classification tasks using [6],[4] supervised learning approaches, including Naive Bayes, Support Vector Machine, and deep learning methodologies. The outcomes achieved by applying the NBC and SVM techniques demonstrate their efficacy in language identification. However, it is worth noting that this study does not incorporate the Natural Language Processing methodology. The accuracy of the SVM method is found to be 0.9634, which is higher than the accuracy value of 0.9378 obtained for the NBC method [7]. This conclusion is drawn solely based on the confusion matrix calculations.

Based on the ROC curve, TF-IDF feature extraction combined with NB, SVM, KNN, RF, and LR on a 90:10 data split achieved an accuracy greater than 80% and belonged to the excellent classification category. DT is a poor classification with an accuracy of less than 70%. The combination of BoW and machine learning algorithms included in good variety are NB, SVM, RF, and LR (accuracy above 80%). The poor classification category is DT (accuracy below 70%), while the failure classification is KNN (accuracy below 60%) [8].

This study classifies text comments using the Support Vector Machine (SVM) algorithm with Linear, RBF, and Polynomial kernels. The test results demonstrate that SVM has high precision, notably the RBF kernel with 88% precision in sentiment classification. In classifying aspects, the RBF kernel achieves an accuracy of 78% [9].

Deep Belief Network applied to tweet classification to identify the sentiment class of tweet training data in Indonesian. The system test results indicate that the DBN method is the most accurate method for tweet data, with an accuracy of 93.31%, compared to the Naive Bayes method, which has an accuracy of 79.10%, and the Support Vector Machine method, which has an accuracy of 92.18% [10].

This study examines the performance of RNN and Nave Bayes by incorporating the TF-IDF (Term Frequency-Inverse Document Frequency) technique, which aims to give weight to the document's word relationships (terms). The test results indicate that RNN (TF-IDF) is more accurate than Nave Bayes (TF-IDF) by 97.77% versus 80% [11].

On the annotated dataset, a variety of machine-learning models are then implemented, including Support Vector Machine (SVM), Nave Bayesian (NB), and Random Forest (RF). As novel features, we have utilized various parts of speech (Nouns, Verbs, Adverbs, and Adjectives). According to our findings, we have outperformed the three competing models with an accuracy rate of 91% [12].

In specific literary works, there remains room for enhancing accuracy. Hence, this study is expected to result in a growth exceeding 97%. This research presents a potential area of investigation that may be considered a research gap. The primary objective of this study is to develop a model through a neural network algorithm. In a hypothetical scenario, it is postulated that the Neural Network algorithm exhibits a notable level of precision, contingent upon the condition that the training dataset must possess a substantial volume, thereby facilitating the attainment of heightened accuracy.

To address these inquiries, an extensive comparative analysis was undertaken utilizing multiple datasets encompassing diverse categories of customer reviews. This study aims to compare two primary types of approaches in the field of natural language processing (NLP) [13]: traditional NLP methods, such as Naive Bayes and Support Vector Machines (SVM), and neural network approaches, specifically Multilayer Perceptron (MLP). These two approaches' sentiment classification accuracy, processing speed, and adaptability will be quantitatively assessed across various text types. Furthermore, it is essential to ascertain circumstances where each approach may exhibit superiority. Conventional methods may be more pragmatic in scenarios characterized by restricted resources. Nevertheless, neural network methodologies might be more efficient in situations that demand precise accuracy and profound comprehension of sentiment. The anticipated outcome of this research is expected to yield significant insights for researchers, practitioners, and companies interested in examining customer sentiment and enhancing comprehension of customer perspectives and emotions. Moreover, this study can provide valuable insights for determining suitable methodologies for different sentiment classification endeavors in various circumstances.

II. Research Metodology

A. Proposed Method

Analyzing textual data within the realm of customer service plays a crucial role in developing a proficient sentiment classification model. The procedure above encompasses several sequential steps essential for transforming unprocessed textual data into a format comprehensible by machine learning algorithms. The initial phase involves tokenization, a process that divides the text into individual units known as tokens, typically corresponding to words. This enables additional analysis to be conducted at the granularity of individual words. Following the process of tokenization [14], the subsequent pre-processing step assumes significance. The pre-processing stage encompasses three primary components:

- the elimination of punctuation.
- the application of stemming and lemmatization techniques.
- the removal of stop words.

The initial step in pre-processing involves the elimination of punctuation. Punctuation marks, such as commas, periods, exclamation marks, and others, frequently need more significant sentiment-related information and can potentially obscure the intended message within the text. Hence, the omission of punctuation aids in maintaining the clarity and pertinence of the text. The subsequent steps involve stemming and lemmatization. Stemming refers to eliminating a word's initial or final components, thereby retaining only its core form. Lemmatization refers to transforming words into their fundamental or base form. This feature facilitates the management of lexical variations arising from inflectional and derivational processes. For instance, the term "food" would be transformed into "consume," while the verb "run" would remain unchanged as "run."

Subsequently, the removal of stop words was executed. Stop words are a set of commonly used words, such as "and," "or," and "this," that occur frequently in text but have a relatively low impact on sentiment analysis. The elimination of stop words aids in directing attention toward the more substantive keywords in the text.

Following the completion of the pre-processing stage, the subsequent step involves feature engineering, wherein the "Bag of Words" methodology is employed. The numerical representation of text is a method in which each word in a document is encoded as a distinct feature. This results in forming a matrix in which every row corresponds to a document, while each column corresponds to a word in the dictionary. The frequency of each word in the document is represented by the numerical value assigned to each cell in the matrix. Construct a sentiment classification model once the data has been processed and transformed into a Bag of Words representation. In this stage, three frequently employed algorithms are Naive Bayes, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP). The Naive Bayes algorithm is recognized for its efficiency as a probability-based method. In contrast, the Support Vector Machine (SVM) algorithm is known for its robustness as a margin-based approach. Lastly, the Multi-layer Perceptron (MLP) [15] algorithm is a type of deep neural network. The process depicted in figure 1 can be observed.

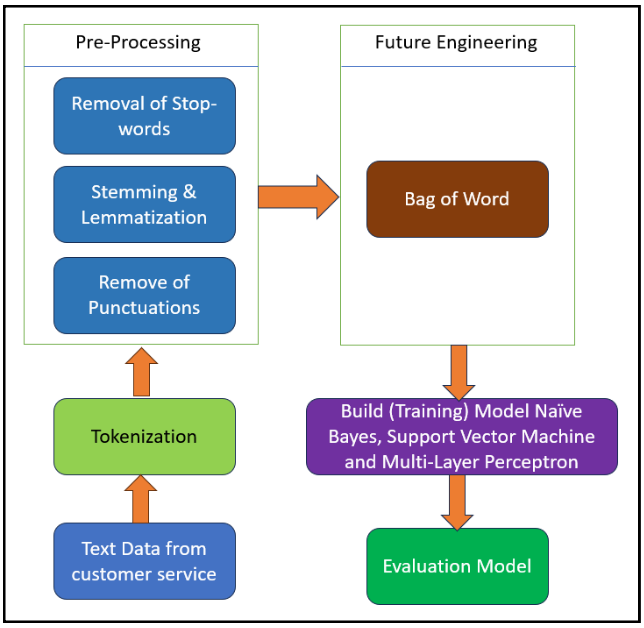


Figure 1. Proposed Method
Source: Researcher Property

B. Natural Language Processing

B.1. Tokenization

Tokenization is the process of dividing a sequence of text into smaller units, known as tokens. These tokens might be individual words. Tokenization is a process that involves dividing a given text into smaller parts, commonly referred to as tokens, to facilitate analysis. Tokens can encompass complete or partial words, thereby reducing the computational processing of textual data. An illustration of this is the sentence "I enjoy consuming pizza," which may be tokenized as "I," "enjoy," and "consume pizza," facilitating the analysis of word frequency.

B.2. Pre-processing Procedures

- The removal of punctuation marks is done to enhance clarity in written texts.
- Stemming and lemmatization are linguistic processes that facilitate the reduction of words to their fundamental forms, enhancing word uniformity.
- Stop word removal involves eliminating often used words, such as "and," that do not contribute significantly to the analysis and interpretation of a text.

B.3. Feature Engineering Utilizing the Bag of Words Approach

The Bag of Words (BoW) technique transforms textual data into a numerical representation. Every word is treated as a feature, and its occurrence frequency in each document is recorded. The Bag-of-Words (BoW) technique facilitates the comprehension of textual patterns and relationships by machine learning algorithms. One of the advantages of this approach is its simplicity, yet it is accompanied by a loss of word order and the potential for high-dimensional data.

The provided abridged explanations offer a succinct overview of tokenization, pre-processing procedures, and the utilization of Bag of Words in the analysis of textual data.

C. Naïve Bayes

The Naive Bayes method is a classification algorithm used in machine learning to predict the category or label of data using data-related features. Bayes' Theorem calculates probabilities based on prior information in this algorithm. In sentiment classification, Nave Bayes helps classify text or data into the appropriate positive, negative, or neutral sentiment categories. The fundamental concept of Nave Bayes is to calculate class or label probabilities based on observed data feature information. However, the primary assumption made by Nave Bayes is the assumption of feature independence, which means that the algorithm assumes that each feature is unrelated or unaffected by the others. The term "naive" is therefore used to describe this assumption.

The initial step in employing Nave Bayes is to calculate the prior probability, or the likelihood of each class's appearance, without considering the features. This can be determined using the training data utilized to teach the model. Then, conditional probabilities, including the likelihood of occurrence of features within each class, are calculated. This enables the model to determine how each feature affects a class's chance. Naive Bayes calculates the probability for each class based on the features observed in the data to be predicted when making predictions. By combining the features' prior and conditional probabilities, the model can predict the most likely categories or labels for the data. The outcome is the class with the most significant likelihood. One of the primary benefits of the Nave Bayes method is its ability to handle data with numerous features or dimensions, such as in text analysis involving thousands of words or word features. In addition, this method is typically effective even with limited training data.

Bear in mind, however, that Nave Bayes assumes feature independence, which is not always the case in the real world. When the relationships between features are highly complex or unpredictable, Nave Bayes' performance may not be optimal. More meticulous data processing or selecting a more suitable model can be implemented in this instance. The Naive Bayes method is an influential text analysis and sentiment classification instrument. It has been implemented in numerous applications, such as spam email management, product review analysis, and document classification. With a thorough understanding of this method's fundamentals, researchers and practitioners can employ it to comprehend better and categorize text data.

D. Support Vector Machine

Support Vector Machine (SVM) is a potent classification algorithm utilized extensively in machine learning. This algorithm can solve various classification problems, such as sentiment classification in text analysis, character recognition in image processing, and even non-linear data separation in high dimensions. The foundation of SVM is a hyperplane search that can distinguish two data classes by a maximum margin. This hyperplane delineates the distinction between data classes. The margin is the distance between the hyperplane and each class's closest point. SVM attempts to identify the hyperplane with the most significant margin, thereby minimizing the possibility of misclassification with new data.

SVM's capacity to solve classification problems involving non-linear data is one of its primary advantages. This is accomplished by transforming the data into a higher-dimensional feature space where the data are separated linearly. This transformation can be achieved using kernel functions such as linear kernels, radial basis function (RBF) seeds, or polynomial kernels. SVM searches for a linear separating hyperplane in a higher feature space, which influences the non-linear data separation in the original space. Support vectors are another essential concept in support vector machines. These are the closest data points to the separating hyperplane, which play a crucial role in determining that hyperplane. In the case of an optimal hyperplane, only the support vectors contribute to the calculation and determination of the maximum margin.

SVM is available in two primary forms: SVM for binary classification and SVM for multi-class classification. In binary classification, SVM is used to distinguish between two data classes. However, in the case of multi-class classification, SVM can be expanded to handle more than two classes using approaches such as one-vs-all and one-vs-one. A further benefit of SVM is its ability to avoid overfitting, which occurs when a model is overly complex and fits the training data well but cannot generalize well to data it has never seen. This is achieved through setting parameters such as the regulation parameter C, which controls the extent to which the model will try to maximize the classification error margin or tolerance. In numerous applications, SVM is a potent and versatile algorithm. Nonetheless, it is essential to comprehend the involved parameters and select a kernel suited to your data to maximize performance. With a solid grasp of SVM's fundamental concepts, business owners, researchers, and practitioners can use this algorithm to solve various classification problems in several fields.

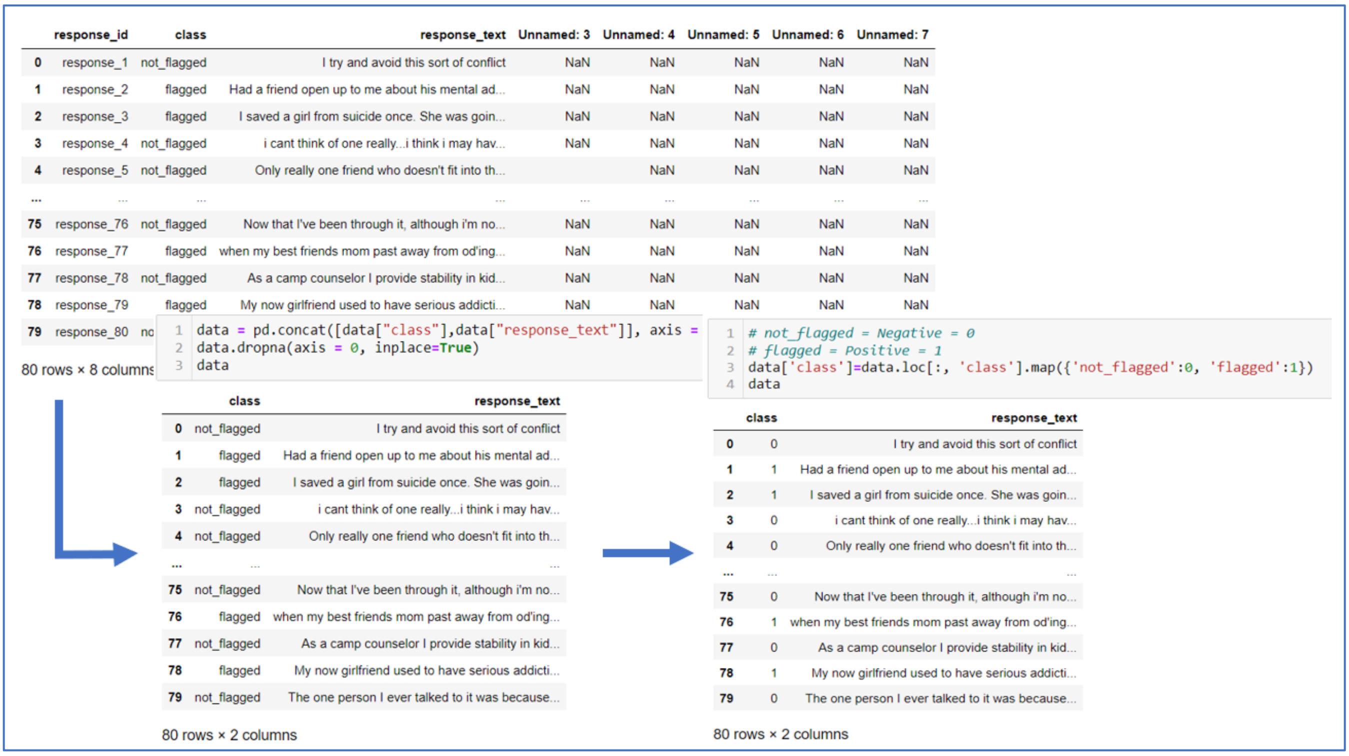
E. Multi-Layer Perceptron

MLP is an artificial neural network employed in machine learning and deep learning. This algorithm is called "multi-layer" because it contains multiple layers of interconnected neurons or nodes. MLP is a crucial component of deep learning and has been implemented in numerous applications, including image recognition, natural language processing, text analysis, and more. MLP's fundamental concept is to generate models that can comprehend and extract complex patterns from data. Each MLP layer comprises neurons or nodes with weights and activation functions. Each neuron in the hidden layer performs calculations based on its weights and activation function in response to incoming data. In the MLP training process, a backpropagation learning algorithm is used to adjust the neurons' weights so that the model can minimize prediction errors on the training data. Throughout this process, the model examines the training data multiple times, predicts the results, compares the predictions with the actual values, and adjusts the weights to decrease the error. This is one of the reasons why MLP can solve non-linear and complex classification problems.

MLP's capacity to model complex patterns and adapt to changing data is one of its greatest strengths. In regression tasks, MLP can also predict continuous values, not just categories or classes. This makes it highly adaptable to numerous types of problems. In addition to any hidden layers, MLPs typically consist of input and output layers. The input layer receives input data, such as images or text, whereas the output layer generates prediction outcomes. The number of neurons in the input layer is proportional to the number of data features. In contrast, the number of neurons in the output layer is proportional to the number of classes or categories to be predicted. Neurons utilize different types of activation functions to compute the output of MLPs. This activation function enables the model to represent non-linear input-output relationships. Rectified Linear Activation (ReLU), Sigmoid, and Hyperbolic Tangent (tanh) are frequently utilized activation functions. In application, MLP requires the selection of a suitable network architecture, number of hidden layers, number of neurons in each layer, activation function, and other parameters. The proper selection of these parameters is crucial for achieving desirable outcomes. MLP can be a potent tool for solving complex classification and regression problems in machine learning when appropriately configured.

III. RESULT AND DISCUSSION

A. Dataset



	response_id	class	response_text	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
0	response_1	not_flagged	I try and avoid this sort of conflict	NaN	NaN	NaN	NaN	NaN
1	response_2	flagged	Had a friend open up to me about his mental ad...	NaN	NaN	NaN	NaN	NaN
2	response_3	flagged	I saved a girl from suicide once. She was goin...	NaN	NaN	NaN	NaN	NaN
3	response_4	not_flagged	i cant think of one really...i think i may hav...	NaN	NaN	NaN	NaN	NaN
4	response_5	not_flagged	Only really one friend who doesn't fit into th...		NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
75	response_76	not_flagged	Now that I've been through it, although i'm no...	NaN	NaN	NaN	NaN	NaN
76	response_77	flagged	when my best friends mom past away from od'ing...	NaN	NaN	NaN	NaN	NaN
77	response_78	not_flagged	As a camp counselor I provide stability in kid...	NaN	NaN	NaN	NaN	NaN
78	response_79	flagged	My now girlfriend used to have serious addicti...	NaN	NaN	NaN	NaN	NaN
79	response_80	nc						

80 rows × 8 columns

```
data = pd.concat([data["class"],data["response_text"]], axis =
data.dropna(axis = 0, inplace=True)
data
```

	class	response_text
0	not_flagged	I try and avoid this sort of conflict
1	flagged	Had a friend open up to me about his mental ad...
2	flagged	I saved a girl from suicide once. She was goin...
3	not_flagged	i cant think of one really...i think i may hav...
4	not_flagged	Only really one friend who doesn't fit into th...
...	...	...
75	not_flagged	Now that I've been through it, although i'm no...
76	flagged	when my best friends mom past away from od'ing...
77	not_flagged	As a camp counselor I provide stability in kid...
78	flagged	My now girlfriend used to have serious addicti...
79	not_flagged	The one person I ever talked to it was because...

80 rows × 2 columns

```
# not_flagged = Negative = 0
# flagged = Positive = 1
data['class']=data.loc[:, 'class'].map({'not_flagged':0, 'flagged':1})
data
```

	class	response_text
0	0	I try and avoid this sort of conflict
1	1	Had a friend open up to me about his mental ad...
2	1	I saved a girl from suicide once. She was goin...
3	0	i cant think of one really...i think i may hav...
4	0	Only really one friend who doesn't fit into th...
...	...	...
75	0	Now that I've been through it, although i'm no...
76	1	when my best friends mom past away from od'ing...
77	0	As a camp counselor I provide stability in kid...
78	1	My now girlfriend used to have serious addicti...
79	0	The one person I ever talked to it was because...

80 rows × 2 columns

Figure 2. Dataset

Source: Kaggle

Figure 2 depicts the initial data pre-processing phase before applying the machine learning classification algorithm. The dataset is sourced from Kaggle and contains 80 rows and 8 columns. This stage's primary objective is to prepare the dataset for training and evaluating classification models. The first stage is identifying and addressing missing values or NaNs in the dataset. This is crucial because missing values can impact the model's quality. Therefore, a "drop" operation eliminates rows with NaN values. After completing this step, the dataset will be cleansed and available for further analysis. Next, concentrate on choosing pertinent attributes for the classification assignment. In this context, only two columns, "class" and "response_text," are maintained. The "class" column will serve as the label or classification objective, while "response_text" will classify the data. In the final data pre-processing phase, the values in the "class" column are replaced with 0 and 1. This is done so that machine learning classification algorithms, which typically require numeric target variables, can utilize the dataset. "not_flagged" is replaced with

0 to represent an unflagged class, and "flagged" is replaced with 1 to represent a flagged class. This pre-processing step has effectively primed the dataset for the next phase, training the machine learning classification model. The model will have a solid foundation for comprehending and predicting the classes in the dataset according to the "response_text" attribute if the data are clean and ready for use.

Figure 2, NLP datasets from Kaggle are used for natural language processing (NLP) tasks available on the Kaggle platform, a popular resource for machine learning (ML) researchers and practitioners. These datasets may include product reviews, news articles, medical texts, and social tweets, which are used to train and evaluate NLP models. Typically, these datasets are organized in formats that facilitate research in numerous fields, such as sentiment analysis, language modeling, and translation. Due to its diversity, size, and quality, Kaggle's data is frequently used. Researchers and data scientists can use it to develop and improve NLP algorithms and answer text analysis-based questions.

B. Natural Language Processing

Figure 3. Dataset Preprocessing

Figure 3, Preprocessing is a crucial Natural Language Processing Toolkit (NLP) step for preparing text data before analysis or modeling. Natural Language Toolkit (NLTK) is a Python toolkit frequently utilized in preprocessing. This procedure includes the removal of stopwords (common words that are commonly overlooked), lemmatization (converting words to their essential forms), and tokenization (breaking text into words). The NLTK corpus can also access additional resources, such as dictionaries and dictionaries of commonly used words. In general, dataset preprocessing aims to clean and organize text data so that it can be utilized effectively in NLP analysis, yielding more accurate and informative results.

C. Naïve Bayes

Training a model using the Naïve Bayes algorithm is essential in data analysis. The algorithm employed in this study leverages probabilistic techniques and machine learning methodologies to make predictions about novel data categories or labels based on the analysis of pre-existing data. Naïve Bayes has demonstrated effectiveness in numerous practical scenarios despite its simplistic attribute independence assumption. During training, the model utilizes available data to estimate the likelihood of an event, thereby enabling it to generate appropriate predictions when presented with novel data. The Naïve Bayes algorithm facilitates machine learning by allowing a computer to acquire knowledge from data, thereby enhancing its capacity to generate predictions or classifications using the provided information.

TABLE 1. Experiment Naïve Bayes
Source: Source: Researcher Property

Experiment	Split (Data Training : Data Testing)	Accuracy
1	90:10	62.5%
2	**80:20**	**68,75%**
3	70:30	58,33%

Table 1, Experiments utilizing an 80:20 split between training and testing data yielded the highest level of accuracy, reaching 68.75%. This indicates that a model trained with training data can generalize well when presented with testing data it has never seen before. This high level of accuracy illustrates the model's ability to make precise and relevant predictions on new data. This result is deemed the best because it demonstrates the model's ability to classify or predict based on existing data. These results show the efficacy of the method utilized in this experiment.

D. Support Vector Machine

Model training using the Support Vector Machine (SVM) algorithm is an essential step in machine learning. SVM is a classification and regression method. To evaluate the performance of a model, data are typically divided into two parts: training data and testing data. A split ratio, such as 90:10, 80:20, or 70:30, indicates the proportion of data allocated to training versus testing. 90% of the data is used to train the model, while 10% is used to evaluate its performance. The 80:20 division assigns 80% of the data to training and 20% to testing, whereas the 70:30 division assigns 70% to training and 30% to testing. The choice of this split ratio impacts how the model learns patterns from the data and its ability to generalize to new data. A higher percentage of training data can improve the model's understanding of the data but can also lead to overfitting. To ensure a good model, the choice of split ratio must consider the trade-off between robust training and reliable testing. The experimental results are presented below:

TABLE 2. Experiment Support Vector Machine
Source: Source: Researcher Property

Experiment	Split (Data Training:Data Testing)	Accuracy
1	90:10	75%
2	**80:20**	**87.5%**
3	70:30	70.83%

Table 2, In Support Vector Machine (SVM) experiments on specific datasets, the optimal results were obtained when the data was split 80:20 between training and testing data. 80% of the data is used to train the SVM model, while 20% is used to assess its performance. A result of 87% accuracy demonstrates the extent to which the model can correctly classify the data. This 80:20 ratio shows a successful balance between robust learning and accurate evaluation. Large enough training data enables SVM to comprehend data patterns effectively, while large enough testing data provides a complete review of the model's ability to generalize to new data. The selection of an appropriate split ratio is crucial to the success of this model, as evidenced by the fact that the SVM, in this instance, can produce excellent predictions.

E. Multi-layer Perceptron

In machine learning, model training using the Multi-layer Perceptron (MLP) algorithm is essential. In this procedure, the data is divided into training data and testing data, with ratios such as 90:10, 80:20, and 70:30. This ratio indicates how much more data is used to train the model compared to the data used to evaluate its performance. A 90:10 data split, with 90% of the data used for training and 10% for testing, provides the model with excellent access to data patterns. However, this division can make the model susceptible to overfitting if not correctly tuned. The additional testing data provided by the 80:20 and 70:30 splits can be used to evaluate the model's ability to generalize to new data. The selection of the split ratio must take into account the trade-off between rigorous training and precise evaluation. The data is presented in Table 3.

TABLE 3. Experiment Multi-layer Perceptron
Source: Source: Researcher Property

Experiment	Split (Data Training:Data Testing)	Accuracy
1	**90:10**	**100%**
2	80:20	81.25%
3	70:30	70.83%

F. Results Summary

TABLE 4. TEST RESULTS SUMMARY

Algorithm	Split (Data Training : Data Testing)	Accuracy (%)
Naïve Bayes	80:20	68,75
Support Vector Machine	80:20	87,5
Multi-layer Perceptron	**90:10**	**100**

Test Result Summary:

a. An accuracy of 68.75% was achieved through training with the Nave Bayes algorithm using an 80:20 split of training data and testing data. This demonstrates that the Nave Bayes model can classify this dataset with a high success rate.
b. Using the Support Vector Machine (SVM) algorithm with an 80:20 split of training data and testing data resulted in an accuracy of 87.5%, demonstrating that the SVM model can classify data exceptionally well on the dataset used.
c. Using the Multi-layer Perceptron (MLP) algorithm with a 90:10 split of training and testing data, it was possible to achieve 100 percent accuracy. This demonstrates that the MLP model effectively recognizes data patterns and can produce highly accurate predictions. Neural Networks, such as MLP, can increase their level of precision as training data grows larger because they have more information to comprehend the data's complexity.

DISCUSSION

Do neural network approaches, such as MLP, perform better than conventional methods in classifying the sentiment of customer texts? (Research Question 1).
Neural network methodologies, specifically Multi-Layer Perceptron (MLP), have demonstrated superior performance in classifying customer text sentiment compared to traditional techniques. This can be attributed to their capability to extract intricate and contextual patterns from textual data, thereby enabling more precise outcomes in sentiment analysis.

How well does this approach handle the complex and variable text frequently found in customer reviews? (Research Question 2).
Approaches based on neural networks, such as MLP, have the advantage of being able to handle the complex and diverse text frequently found in customer reviews. They can deal with subtleties, changes in context, and various sentence structures that are typically difficult for conventional methods to comprehend. This makes it very effective at identifying more nuanced and accurate sentiments in customer review text, which may contain a variety of writing styles and expressions. With its high adaptability, the neural network approach to analyzing views on complex and diverse text data can provide more accurate and timely results.

Are the increased computational costs associated with neural network approaches justified by the improved accuracy? (Research Question 3).
The increased computational costs associated with neural network approaches, such as using the Multilayer Perceptron (MLP), are frequently justified by the significant accuracy enhancements such models can offer. Although neural network approaches require more computational resources, they can extract more complex patterns and comprehend subtleties that are difficult to access with conventional methods. In many instances, this increased

precision can result in more valuable insights, facilitate better decision-making, and replace additional costs with far more significant benefits in data analysis.

IV. CONCLUSION

The findings derived from the three outcomes of model training utilizing the Naive Bayes algorithm, Support Vector Machine (SVM), and Multi-layer Perceptron (MLP), along with the distribution of distinct training and testing data, can be summarized as follows:

The Naive Bayes algorithm attains a classification accuracy of 68.75% when the training and testing datasets are partitioned in an 80:20 ratio. This demonstrates that the Naive Bayes model exhibits a reasonably high success rate in classifying the dataset, albeit with potential for further enhancement in terms of accuracy.

Furthermore, the Support Vector Machine (SVM) algorithm demonstrates a classification accuracy of 87.5% when the dataset is partitioned into 80% for training and 20% for testing purposes. The findings of this study show that Support Vector Machines (SVM) exhibit high proficiency in accurately classifying data within the specific dataset employed. Consequently, SVM holds considerable promise as a robust option for undertaking classification tasks.

Ultimately, when employing the Multi-layer Perceptron (MLP) algorithm and partitioning the data into a 90% training set and a 10% testing set, the outcome is a flawless % accuracy rate of 100%. This demonstrates the high efficacy of the MLP model in pattern recognition and generating precise predictions. Neural networks, such as Multi-layer Perceptron (MLP), exhibit an enhanced capacity for precision augmentation as the volume of training data expands. This is attributed to the increased availability of information, enabling the network to comprehend the intricate nature of the data. Therefore, when considering the ultimate objective and the model's requirements, it is crucial to carefully consider the algorithm selection and the proportion of training and test data allocation. In this context, the Multi-layer Perceptron (MLP) emerges as an exact option.

Future research on this topic is recommended to investigate ensemble learning techniques that combine Naive Bayes, SVM, and MLP models to achieve even higher classification precision. In addition, it would be helpful for practical implementations to investigate the effect of dataset size variation on model performance and scalability.

REFERENCE

[1] V. A. Fitri, R. Andreswari, and M. A. Hasibuan, "Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm," *Procedia Comput. Sci.*, vol. 161, pp. 765–772, 2019, doi: 10.1016/j.procs.2019.11.181.

[2] S. Farhana, "Classification of Academic Performance for University Research Evaluation by Implementing Modified Naive Bayes Algorithm," *Procedia Comput. Sci.*, vol. 194, pp. 224–228, 2021, doi: 10.1016/j.procs.2021.10.077.

[3] R. Haque, N. Islam, M. Tasneem, and A. K. Das, "Multi-class sentiment classification on Bengali social media comments using machine learning," *Int. J. Cogn. Comput. Eng.*, vol. 4, no. December 2021, pp. 21–35, 2023, doi: 10.1016/j.ijcce.2023.01.001.

[4] I. Kaczmarek, A. Iwaniak, A. Świetlicka, M. Piwowarczyk, and A. Nadolny, "A machine learning approach for integration of spatial development plans based on natural language processing," *Sustain. Cities Soc.*, vol. 76, 2022, doi: 10.1016/j.scs.2021.103479.

[5] C. Savojardo, P. L. Martelli, and R. Casadio, "Finding functional motifs in protein sequences with deep learning and natural language models," *Curr. Opin. Struct. Biol.*, vol. 81, p. 102641, 2023, doi: 10.1016/j.sbi.2023.102641.

[6] E. Hagberg *et al.*, "Semi-supervised learning with natural language processing for right ventricle classification in echocardiography—a scalable approach," *Comput. Biol. Med.*, vol. 143, no. September 2021, 2022, doi: 10.1016/j.compbiomed.2022.105282.

[7] D. Tuhenay and E. Mailoa, "Perbandingan Klasifikasi Bahasa Menggunakan Metode Naïve Bayes Classifier (Nbc) Dan Support Vector Machine (Svm) Comparison of Language Classification Using Naive Bayes Classifier (Nbc) and Support Vector Machine (Svm) Method," vol. 4, no. 2, pp. 105–111, 2021, doi: 10.33387/jiko.

[8] L. Efrizoni, S. Defit, M. Tajuddin, and A. Anggrawan, "Komparasi Ekstraksi Fitur dalam Klasifikasi Teks Multilabel Menggunakan Algoritma Machine Learning," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 3, pp. 653–666, 2022, doi: 10.30812/matrik.v21i3.1851.

[9] Y. Refo and S. Rostianingsih, "Penerapan SVM untuk Klasifikasi Sentimen pada Review Comment Berbahasa Indonesia di Online Shop," 2021.

[10] I. Zulfa and E. Winarko, "Sentimen Analisis Tweet Berbahasa Indonesia Dengan Deep Belief Network," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 11, no. 2, p. 187, 2017, doi: 10.22146/ijccs.24716.

[11] Merinda Lestandy, Abdurrahim Abdurrahim, and Lailis Syafa'ah, "Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naïve Bayes," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 802–808, 2021, doi: 10.29207/resti.v5i4.3308.

[12] I. Ihsan, H. Rahman, A. Shaikh, A. Sulaiman, K. Rajab, and A. Rajab, “Improving in-text citation reason extraction and classification using supervised machine learning techniques,” *Comput. Speech Lang.*, vol. 82, p. 101526, 2023, doi: https://doi.org/10.1016/j.csl.2023.101526.

[13] J. Ryu, S. Heisig, C. McLaughlin, M. Katz, H. S. Mayberg, and X. Gu, “A natural language processing approach reveals first-person pronoun usage and non-fluency as markers of therapeutic alliance in psychotherapy,” *iScience*, vol. 26, no. 6, p. 106860, 2023, doi: 10.1016/j.isci.2023.106860.

[14] M. Aso, S. Takamichi, N. Takamune, and H. Saruwatari, “Acoustic model-based subword tokenization and prosodic-context extraction without language knowledge for text-to-speech synthesis,” *Speech Commun.*, vol. 125, no. September, pp. 53–60, 2020, doi: 10.1016/j.specom.2020.09.003.

[15] D. Hindarto and A. Djajadi, “Android-manifest extraction and labeling method for malware compilation and dataset creation,” vol. 13, no. 6, pp. 6568–6577, 2023, doi: 10.11591/ijece.v13i6.pp6568-6577.