Jurnal Teknik Informatika Vol. 15 No. 2, 2022 (169-177)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v15i2.29302

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Comparative Analysis of KNN, Naïve Bayes and SVM Algorithms for Movie Genres Classification Based on Synopsis.

Nurhayati[1], Lee Kyung Oh[2], Muhammad Hugo Athallah Hardy [3], Yusuf Wijaya[4]

[1,3,4]Informatics Department, Science and Technology Faculty
[2]Computer Engineering Department, Sun Moon University, Korea
[1,3,4]Syarif Hidayatullah State of Islamic University, Indonesia
[2]Asan-Si, Chungcheongnam-do, South Korea
[1,3,4]Jl. Ir. H. Juanda No. 95, Ciputat 15412, Telp. (021) 7401925
E-mail: [1]nurhayati@uinjkt.ac.id, [2]leeko@sunmoon.ac.kr, [3]muhammad.hugo19@mhs.uinjkt.ac.id, [4]yusuf.wijaya19@mhs.uinjkt.ac.id,

Article:
Accepted: August 22, 2022
Revised: July 13, 2022
Issued: November 15, 2022

***Correspondence Address:**
muhammad.hugo19@mhs.uinjkt.ac.id

ABSTRACT

Text classification is a process of categorizing a text into the correct label. Text classification in natural language processing is a challenging task that requires accuracy to get the correct results, manual text classification tends to be inefficient because it requires a lot of time and also experts. The utilization of machine learning for automatic text classification can be a solution to this problem. KNN, Naive Bayes, and SVM are known as some of the most algorithms to solve classification problems, especially text classification. In this study, we are trying to compare the KNN, Naive Bayes, and SVM algorithms for text classification with the problem of classifying movie genres based on a synopsis using datasets obtained from Kaggle.com and IMDB Dataset. The results of this study indicate that of the 12 experiments, Support Vector Machine (SVM) is the best-performing algorithm with an accuracy of 90%, 93%, 65%, and 63%. It is hoped that this research can help to determine the best algorithm in the text classification process.

Keywords: *Movie Genres, Text Classification, Natural Language Processing, KNN, Naïve Bayes, SVM*

I. INTRODUCTION

Movie is one of the media to communicate that has audio-visual properties where the message is implied by the film's creator. Movie have several different Movie genres, namely romance, horror, thriller, comedy, fantasy, and so on. Of the many film genres that exist. The process of sorting and categorizing film genres certainly takes time and resources. The use of machine learning for automatic film genre categorization with the text classification method can be a solution to this problem.

Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups [1]. A significant and increasing number of research problems in the center of the social sciences on text analysis [2], the fundamental task of which assigns text archives to at least one predefined class according to the substance and sample to

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v15i2.29302

which it is labeled [3]. The problem of Text classification has been considered extensively and tends to have a wide range of real applications over the last few years. [2]. Especially with recent breakthroughs in Natural Language Processing (NLP) and text mining. Classification is widely used for tasks such as Indonesian capital city relocation sentiment [4], spam mail detection [5], and news article classification [6]. Classification of natural language processing texts is a challenging task [7]. To get accurate results, a proper algorithm is needed for a model to carry out this task.

Many text classifiers have been proposed in the literature using machine learning techniques, probabilistic models, etc. They often differ in the approach adopted: decision trees, naıve-Bayes, rule induction, neural networks, k-nearest neighbors [8], and lately, support vector machines [9] Although there are already many approaches, automated text classification needs to be studied more deeply because there is still a lot that needs to be improved. This study aims to obtain accurate results in the classification process. K-Nearest Neighbor, Naive Bayes, and Support Vector Machine are some very popular algorithms [10] in solving classification problems.

These algorithms differ in their strengths and shortcomings, therefore contrasting them might reveal which algorithm could be the most useful for a particular text categorization problem. For instance, KNN is renowned for being straightforward and simple to use, yet it can have high computational complexity and may struggle with big data. SVM, on the other hand, is a more complicated method that can handle enormous datasets and high-dimensional data, but it may not be as simple to understand as other algorithms. In contrast, Naive Bayes is a probabilistic method that is renowned for being fast and accurate but also for relying heavily on the independence of characteristics, which may not always be true in actual use. Finally, contrasting these algorithms can help us gain a better knowledge of the advantages and disadvantages of various approaches to text categorization, which can then be used to build future algorithms that are more successful and efficient. Of the three algorithms, we will try to compare the performance of the three algorithms in the text classification process.

We want to compare the model with the KNN, Naive Bayes [11], and SVM algorithms and will examine which one is more optimal for genre grouping based on a synopsis. In addition to comparing the algorithms of the model, the author also applies the vectorization method of count-vectorizer and tf-vectorizer to the model and compares the effect of vectorization on the three Machine Learning models. In this study, the author uses two datasets for model training which contains a synopsis of films with their genres taken from the website Kaggle and IMDB.

II. LITERATURE STUDY

As reference material for this study, the author uses 5 journals related to this research:

2.1 Text Classification between Multinomial Naïve Bayes and Bernoulli Naïve Bayes

This journal written by Gurinder Singh, Bhawna Kumar, Akriti Tyagi, and Loveleen Gaur/ 2019 [12] first the paper tried to predict using multinomial and Bernoulli naïve bayes the result of sentiment, whether it is negative or positive. After that these algorithms later are compared to each other to see which algorithm is better to do text classification. The paper used about 312 data records and has several steps which are preprocessing including lowercasing, punctuation removal, tokenization, and stopwords removal. From the accuracy graph, it can be seen that naïve bayes can be used for text classification because it exceeded 70% of accuracy and multinomial naïve bayes had better accuracy than the Bernoulli naïve bayes.

2.2 Text Classification on Twitter Data

Journal written by Dr. Priyanka Harjule, Astha Gurjar, Harshita Seth, Priya Thakur in 2020 [13] compared the performance of the Logistic Regression, Naive Bayes, RNN-LSTM, and SVM algorithms for text classification of Twitter data from the "Sentiment140" dataset, dataset from Stanford University and "Crowdflower's Data for Everyone library". The process carried out in this study begins with preprocessing the data using NLTK and splitting the data with a proportion of 70% for training data and 30% for testing data. Furthermore, the data is used to train each of these algorithms where the text in the dataset becomes the independent variable and positive and negative labels on the dataset become the dependent variable. Results show that the RNN-LSTM algorithm has the best

performance with an accuracy of 84% for dataset 1 and 66% for dataset 2.

2.3 Text Classification Using The K-Nearest Neighbor Algorithm in The Case of Government Performance on Twitter

Journal written by Octaryo Sakti Yudha Prakasa and Kemas Muslim Lhaksamana in 2018 [14] proved the sentiments of comments or tweets of Twitter users about the current government's performance. The dataset used in this research is 1000 tweets. The dataset is obtained through a crawling process using PHP. The dataset is needed to build a classification system using the KNN algorithm. At the feature extraction stage, preprocessed tweets are converted into values. To make tweets a value, the feature extraction that I use is binary TF. In this process, tweets that have been preprocessed will be separated into words to become attributes in the training and testing data. What must be considered is that the training and testing data attributes must be the same so that distance calculations can be carried out in the KNN Distance process.

2.4 Naïve Bayes Classifier Optimization For Text Classification in E-Government Using Particle Swarm Optimization

Journal written by Kuncahyo Setyo Nugroho and Fitri Marisa Istiadi in 2019 [15] proved the Optimization of the NBC Algorithm in text classification. The dataset used in this study comes from the Sambat Online portal of Malang City (www.sambat.malangkota.go.id) as well as, which was collected using the web scraping method. The dataset is text converted to .xlsx format. The dataset consists of 200 data with 7 categories as labels representing the responsible OPD.

The stages of text preprocessing carried out in this study include case folding, tokenizing, stemming, and filtering. The classification model is built based on the NBC and k-NN algorithms. In this study, the Naive Bayes Classifier algorithm was chosen as the standard method because it is a simple and efficient classification method.

III. METHODOLOGY

3.1 Field of Study

The study areas in this research include the classification of natural language processing texts, the implementation of algorithms for predictive models related to these problems, and also the effect of vectorization methods on model performance. Matters related to this research study area will be discussed one by one starting with Text Classification, KNN, Naive Bayes, SVM, Text Vectorization (Count Vectorizer + TF-IDF), and Evaluation Metrics.

3.2 Text Classification

Text classification is a machine learning technique that assigns a set of predefined categories to text. Text classifiers can be used to organize and categorize almost any type of text from documents, such as medical studies and files. Text classification is one of the fundamental tasks in natural language processing with wide application fields such as sentiment analysis, topic labeling, to spam detection. Automated text classification has been considered a vital method to manage and process a vast amount of documents in digital forms that are widespread and continuously increasing [16].

3.3 K Nearest Neighbor

The k-nearest neighbor algorithm (k-NN or KNN) is a method for classifying objects based on the learning data that is closest to the object [17]. K-Nearest Neighbor algorithm is widely applied because of its effectiveness, non-parametric & easy implementation properties [18]. The k-Nearest Neighbor algorithm is a supervised algorithm learning where the results of the new instance are classified according to the majority of the k-nearest neighbor categories.

For example, there is a house that is right in the middle of the border between City A and City B. We can determine it using the k-NN algorithm, namely by involving the distance between the house and the houses around it (neighbors). First, we must determine the number of neighbors that we will count (k), for example, we determine the 3 nearest neighbors (k = 3). Next, we calculate the distance of each neighbor to the house, then sort the results by distance, starting from the smallest to the largest. After that, take the 3 closest neighbors, then we will see whether each of the neighbors is included in City A or City B. It will leave us with 2 possibilities. If from the 3 neighbors 2 houses are included in the A Area, then the house is included in the A Area. On the other hand, if from the 3 neighbors 2 houses are

Jurnal Teknik Informatika Vol. 15 No. 2, 2022 (169-177)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v15i2.29302

included in the B Area, then the house is included in the B Area.

The KNN (k-Nearest Neighbor) algorithm is a classification algorithm based on the nearest neighbor, to calculate the distance we can use the Euclidean Distance formula. Similar to Pythagoras, only the Euclidean Distance has more than 2 dimensions.

3.4 Naïve Bayes

Naive Bayes classification is a simple probability classification based on the application of Bayes' theorem [19], with the belief that the informative variables are independent. In this case, it's assumed that the presence or absence of a specific event from one cluster isn't related to the presence or absence of alternative events (Zhang and Li, 2007). For example, if we're trying to identify a fruit by color, shape, and taste, then an orange, round, and tangy fruit is most likely an orange. Even if these traits depend on each other or on the presence of other traits, all of these traits individually contribute to the possibility that this fruit is an orange and that is why it is known as "naive."

Naive Bayes calculates the likelihood of every category and so chooses the one with the best probability. Naives Bayes is a classification technique supported Bayes' Theorem. In the Naïve Bayes classification, the learning method is a lot emphasized on estimating probabilities. The advantage of this approach is that the classification can get a smaller error worth once the dataset is massive. In addition, the Naïve Bayes classification is proven to possess high accuracy and speed once applied to massive databases [20]. Recent studies conjointly demonstrate that Naive Bayes classifiers with word presence or absence values performed higher in predicting opinion polarities analysis of text classification method of movie reviews [21].

3.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning methodology that's typically used for classification and regression. In classification modeling, SVM encompasses an additional mature and clearer mathematical idea than alternative classification techniques. SVM can even solve classification issues on linear and non-linear data. SVM is employed to search out the most effective hyperplane by increasing the space between classes. The hyperplane is an operation that will be accustomed to separate between classes. This method uses a hypothesis in the form of linear functions in feature space with high dimensions, by implementing a learning bias derived from statistical learning theory [22]. The figure below shows how the data is classified with a support vector machine.

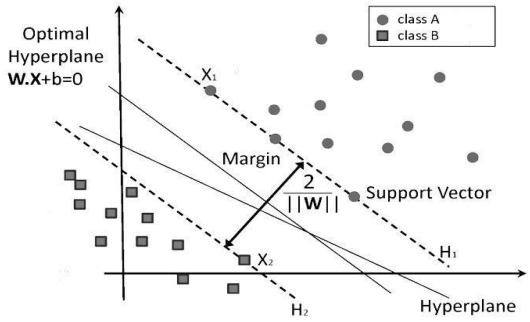

Figure 1. Support vector machine

In this experiment, we also use SGD as an optimization method. We use Linear SVM with Stochastic Gradient Descent (SGD) approach to fitting linear classifiers under convex loss functions. SGD is a simple and very efficient optimization method. SGD has been around within the machine learning community for quite a long time, and it's received a substantial quantity of attention only in the near past in the context of large-scale learning.

3.6 Text Vectorization

Feature Extracting is the method of reworking information into options that will be used for machine learning models. Typically, machine learning algorithms are programmed numerically. Therefore, the text or word is then mapped into a numeric feature vector (series of numbers).

This process is named Text Vectorization or “Bag of Words” representation. This feature is often extracted through exploitation (i) Count vectorizer - takes into account direct technique in feature extraction because it solely counts the number of times the word/token seems in an exceedingly given document, or (ii) term frequency - reversed document frequency (TF-IDF). It represents an applied mathematics measuring technique for evaluating vital words in an exceedingly given document. Term Frequency (tf) refers to how often the term appears in the document against the number of words in the document. So, the more frequent occurrences of the term, the greater the value.

Jurnal Teknik Informatika Vol. 15 No. 2, 2022 (169-177)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v15i2.29302

$$TF = \frac{(number\ of\ occurrences\ of\ terms\ in\ document)}{(total\ number\ of\ terms\ in\ document)} \quad (1)$$

Inverse document frequency (idf) is a measurement of the weight of the selected term in the document. The IDF method is a calculation of how the terms are widely distributed in the collection of documents in question.

$$IDF = \frac{(total\ number\ of\ given\ documents)}{(number\ of\ documents\ with\ selected\ words)} \quad (2)$$

Both Count Vectorizer and TF-IDF vectorizer consist of customizable n-gram size features and n-grams are consecutive words. So, 1 gram is just one word, it is also called a unigram, while 2 grams is called a bigram, 3 grams is called a trigram, and so on. When working with n-grams, all n-grams with degrees less than or equal to n are generated [23].

3.7 Evaluation Metrics

Evaluation Metrics are a measurement of the predictive quality of the system. To do this, we can measure the performance of the newly trained model on test data or new independent data. Comparison of model prediction results with actual data can be a measure of how good the quality of a model is. Several measurement metrics can be used to evaluate a model. Some of them can be seen in the following table:

Table 1. Evaluation metrics

Metrics	Formula
F1-Score	$2\ x\ \frac{Precission\ x\ recall}{Precission\ +\ recall}$
Precision	$\frac{TP}{TP\ +\ FP}$
Recall	$\frac{TP}{TP\ +\ FN}$
Accuracy	$\frac{TP\ +\ TN}{TP\ +\ FN\ +\ TN\ +\ FP}$

TP is True Positive / Positive data detected is correct, TN is True Negative / Detected Negative Data is correct, FP is False Positive / Negative data detected is wrong, and FN is False Negative / False positive data detected. These metrics will be used to evaluate each model to be tested. And the results in each model will be compared. Here we are not only using accuracy to quantify the performance of an algorithm, we also used precision, recall, and f1-score to quantify the classification performance of the algorithms, since it performed better than accuracy to quantify the classification performance of the algorithms for skewed datasets [24].

3.8 Research Stages

a) Data Acquisition

In this study, the authors used two datasets used in the implementation of the classification algorithm. The first dataset is movie genre data taken from the kaggle.com site and the second dataset is the result of parsing and merging of plot list and genre list data from IMDB site.

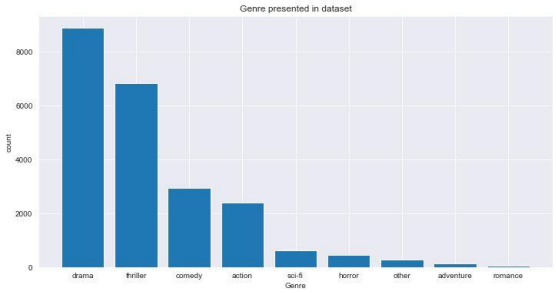

Figure 2. First Dataset Data Distribution

The first dataset has roughly a total of 22.000 data which has 3 columns and 22,579 rows. where the 3 columns consist of id, text, and genre columns. In this dataset, there are 9 film genres, namely thriller, comedy, drama, action, sci-fi, other, romance, horror, and adventure.

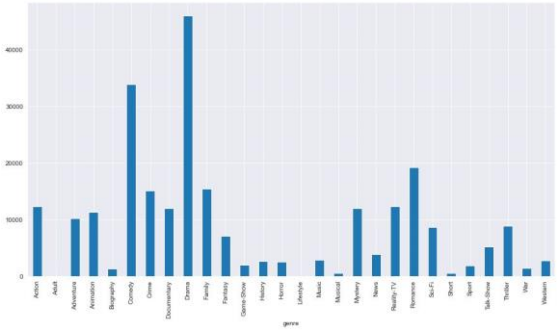

Figure 3. Second Dataset Data Distribution

The Second Dataset has roughly a total of 117.000 data which has 30 columns and 117,352 rows. The column in this dataset consists of the Title column, Plot, and 28 film genres ranging from Action, Thriller, to western, where the contents of the column will be worth 1 if the plot represents the column's genre and will be 0 if the plot is not the column's genre.

b) Data Preprocessing

Both datasets must first be preprocessed. Because the two datasets used in this study have

Jurnal Teknik Informatika Vol. 15 No. 2, 2022 (169-177)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v15i2.29302

different formats, the preprocessing of the two datasets will have different stages as well. Data preprocessing involves the process of data cleaning and/or data integration and/or data reduction and/or data addition and/or data transformation [25]. In this study, we will transform the data where we will map categorical to numerical values in both datasets and change the shape of the columns in the second dataset.

c) Text Cleaning

The text column in the first dataset and the plot column in the second dataset containing a synopsis will later be used as features or input for the algorithm to be used. The synopsis data in the form of text will be converted into a number matrix so that it can be inputted into the algorithm. But before vectorization is done, the text must be cleaned first. For the text cleaning process, there are four stages, the first is changing every character in the text to lowercase. Second removes special characters in text such as using a regex (regular expression) with pattern [^a-zA-Z0-9]. Third Removing stopwords to minimize unnecessary information. And the last one is Stemming.

d) Text Vectorization

After the text is cleaned, then vectorization is carried out on the text which aims to change the text into a number matrix so that the text which is a feature can be inputted into the model. The vectorization method used in this research is the CountVectorizer and TF-IDF Vectorizer from the sklearn library. CountVectorizerand TF-IDF Vectorizer was applied to the two datasets that had been cleaned of text.

e) Data Splitting

The data resulting from the vectorization is then divided into data for training and data for model validation with a proportion of 80% for training data and 20% for testing data. The data split process uses the train_test_split function from the sklearn library.

f) Making Models

The training and testing data will be used to train and evaluate the model. The algorithm that will be used in this research is K-Nearest Neighbor, Naive Bayes, and Support Vector Machine. The three algorithms will be used to build a film genre classification model based on the labels of each dataset. The following is the design model that will be used. The KNN model is created using the KNeighborsClassifier from the sklearn library. The model uses the default settings with parameters. The model built is as follows n_neighbors = 5, metric = Minkowski, weights = uniform, algorithm = auto. Next for the Naive Bayes model used is multinomial naive bayes, multinomial naive bayes algorithm is a popular algorithm for text classification. The Naive Bayes model in this study uses MultinomialNB from the sklearn library. The model uses the default settings with the following parameters alpha = 1.0, fit_prior = True, and class_prior = None. The last model is SVM algorithm, this model uses SVM linear classifier with SGD training optimization. Models are created using the SGDClassifier from the sklearn library. The model uses the default settings with the following parameters Loss = hinge, Penalty = l2, Alpha = 0.0001, Epsilon = 0.1, Learning rate = optimal

g) Model Evaluation

After the model is created, we evaluate the model's performance using the evaluation metrics that have been presented above. The metrics that we will use include f1-score, precision, recall, and accuracy. We use the evaluation results to compare the performance of each model that we have created.

3.9 Framework

All stages of the research are described by a framework. This framework describes the research flow that we have discussed above. The Framework of this research can be seen in the following figure below:

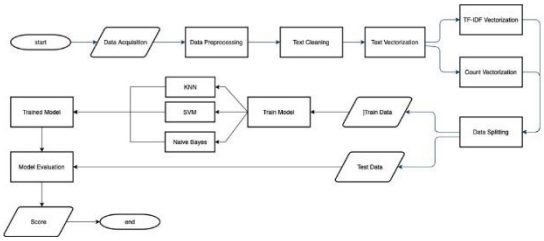

Figure 4. Research workflow

IV. RESULTS AND DISCUSSION

4.1 Experiment on First Dataset

The first experiment was conducted using a movie genre dataset from the kaggle.com site. In this experiment, 6 experiments were conducted using 6 models with different algorithms and vectorization methods. The experimental results can be seen in Table 1.

Jurnal Teknik Informatika Vol. 15 No. 2, 2022 (169-177)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v15i2.29302

Table 2. Results of experimental accuracy dataset 1

Vectorizer	KNN	SVM	Naïve Bayes
Count Vectorizer	82.04%	90.23%	90.08%
TF-IDF Vectorizer	92.87%	93.05%	68.25%

The experimental results show that the model with the algorithm achieves a fairly high accuracy above 90% on the KNN model with TF-IDF Vectorizer, SVM with Count Vectorizer and TF-IDF Vectorizer, and Naive Bayes with Count Vectorizer. While the worst accuracy was obtained by Naive Bayes with TF-IDF Vectorizer. The results of measuring the F1-score, Precision, and Recall metrics in the first dataset can be seen in the table below:

Table 3. Precision, recall, f1-Score first Dataset

Algorithm	Precision	Recall	F1-Score
KNN - Count Vectorizer	0.84	0.82	0.82
SVM- Count Vectorizer	0.92	0.86	0.89
Naïve Bayes – Count Vectorizer	0.87	0.92	0.89
KNN – TF-IDF Vectorizer	0.93	0.91	0.92
SVM – TF-IDF Vectorizer	0.96	0.89	0.92
Naïve Bayes – TF-IDF Vectorizer	0.71	0.68	0.61

SVM model with TF-IDF Vectorizer becomes the model with the highest F1-Score with a value of 0.92 followed by KNN with TF-IDF Vectorizer with an f1-Score value of 0.91del with TF-IDF Vectorizer becomes the model with the highest F1-Score with a value of 0.92 followed by KNN with TF-IDF Vectorizer with an f1-Score value of 0.91.

4.2 Experiment on Second Dataset

The second experiment was carried out using a parsed dataset and combining plot list and genre list data from the IMDB site. In this experiment, 6 experiments were conducted using 6 models with different algorithms and vectorization methods. The experimental results can be seen in the table below:

Table 4. Results of experimental accuracy of second dataset

Vectorizer	KNN	SVM	Naïve Bayes
Count Vectorizer	31.31%	65.26%	61.42%
TF-IDF Vectorizer	61.78%	62.94%	59.41%

The experimental results show that the model with the SVM algorithm with Count Vectorizer and TF-IDF vectorizer has the best accuracy with an accuracy of 65.26% and 62.94%, respectively. These results are quite far from the accuracy of the first dataset. This can be caused by several things such as the amount of noise in the second dataset, the number of labels that are quite a lot, and the length of the text on each row being different. The results of the F1-score, Precision, and Recall metrics measurements in the second dataset can be seen in Table 4 below:

Table 5. Precision, recall, f1-Score second dataset

Algorithm	Precision	Recall	F1-Score
KNN - Count Vectorizer	0.50	0.31	0.26
SVM- Count Vectorizer	0.65	0.65	0.65
Naïve Bayes – Count Vectorizer	0.63	0.61	0.61
KNN – TF-IDF Vectorizer	0.62	0.62	0.61
SVM – TF-IDF Vectorizer	0.63	0.62	0.62
Naïve Bayes – TF-IDF Vectorizer	0.62	0.59	0.57

SVM model with Count Vectorizer is the model with the highest F1-Score with a value of 0.65 followed by SVM with TF-IDF Vectorizer with an f1-Score value of 0.62

V. CONCLUSION

Based on the research results, it can be concluded that the Support Vector Machine (SVM) algorithm is the best-performing algorithm with an accuracy of 90.23% using CountVectorizer and 93.05% using TF-IDF Vectorizer in the first dataset and 65.26% using Count Vectorizer and 62, 94% used TF-IDF Vectorizer in the second dataset.

The Support Vector Machine algorithm also works well using both text vectorization methods, namely Count Vectorizer and TF-IDF Vectorizer. Meanwhile, the Naive Bayes algorithm works better with the Count Vectorizer and the KNN algorithm works better with the TF-IDF Vectorizer. By using this vectorization method, the Naive Bayes algorithm and KNN have similar accuracy to the Support Vector Machine Algorithm with SGD training.

From the conclusions, it is hoped that it can help to determine the appropriate algorithm and approach to solve the problem of classifying natural language processing texts. As a suggestion for future research, the author hopes to apply Grid Search to get optimal hyperparameters for each model to get more accurate prediction results.

BIBLIOGRAPHY

[1] I. Rasheed, V. Gupta, H. Banka, and C. Kumar, "Urdu Text Classification: A comparative study using machine learning techniques," *2018 Thirteen. Int. Conf. Digit. Inf. Manag.*, pp. 274–278, 2018.

[2] P. Barberá, A. E. Boydstun, S. Linn, R. McMahon, and J. Nagler, "Automated Text Classification of News Articles: A Practical Guide," *Polit. Anal.*, vol. 29, no. 1, pp. 19–42, 2021, doi: 10.1017/pan.2020.8.

[3] Y. Wang *et al.*, "A clinical text classification paradigm using weak supervision and deep representation," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, pp. 1–13, 2019, doi: 10.1186/s12911-018-0723-6.

[4] E. Sutoyo and A. Almaarif, "Twitter sentiment analysis of the relocation of Indonesia's capital city," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1620–1630, 2020, doi: 10.11591/eei.v9i4.2352.

[5] P. Verma, A. Goyal, and Y. Gigras, "Email phishing: text classification using natural language processing," *Comput. Sci. Inf. Technol.*, vol. 1, no. 1, pp. 1–12, 2020, doi: 10.11591/csit.v1i1.p1-12.

[6] L. Al Qadi, H. El Rifai, S. Obaid, and A. Elnagar, "Arabic text classification of news articles using classical supervised classifiers," *2019 2nd Int. Conf. New Trends Comput. Sci. ICTCS 2019 - Proc.*, pp. 1–6, 2019, doi: 10.1109/ICTCS.2019.8923073.

[7] H. Fausk and D. C. Isaksen, "Improving Language Understanding by Generative Pre-Training," *Homol. Homotopy Appl.*, vol. 9, no. 1, pp. 399–438, 2007, doi: 10.4310/HHA.2007.v9.n1.a16.

[8] N. Buslim and A. . Rahman, "Implementation of Naive Bayes and K-Nearest Neighbor Algorithm for Diagnosis of Diabetes Mellitus," *Proc. 13th Int. Conf. Appl. Comput. Appl. Comput. Sci.*, 2014.

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

[10] Nurhayati, F. Agustian, and M. D. I. Lubis, "Particle Swarm Optimization Feature Selection for Breast Cancer Prediction," *2020 8th Int. Conf. Cyber IT Serv. Manag. CITSM 2020*, pp. 5–10, 2020, doi: 10.1109/CITSM50537.2020.9268865.

[11] T. Laksana, P. Rizqiyah, R. Ramadhani, and N. S.N, "Classification of Twitter Comments About the Image of the People's Representative Council (DPR) Using the K-Nearest Neighbor (K-NN) Method and Naïve Bayes," no. April 2020, 2020, doi: 10.4108/eai.14-3-2019.2292042.

[12] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification," *2019 Int. Conf. Autom. Comput. Technol. Manag. ICACTM 2019*, pp. 593–596, 2019, doi: 10.1109/ICACTM.2019.8776800.

[13] P. Harjule, A. Gurjar, H. Seth, and P. Thakur, "Text Classification on Twitter Data," *Proc. 3rd Int. Conf. Emerg. Technol. Comput. Eng. Mach. Learn. Internet Things, ICETCE 2020*, no. February, pp. 160–164, 2020, doi: 10.1109/ICETCE48199.2020.9091774.

[14] O. S. Y. Prakasa and K. M. Lhaksmana, "Klasifikasi Teks Dengan Menggunakan Algoritma K-nearest Neighbor Pada Kasus Kinerja

Pemerintah Di Twitter," *eProceedings Eng.*, vol. 5, no. 3, pp. 8237–8248, 2018.

[15] K. S. Nugroho, I. Istiadi, and F. Marisa, "Naive Bayes classifier optimization for text classification on e-government using particle swarm optimization," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 1, pp. 21–26, 2020, doi: 10.14710/jtsiskom.8.1.2020.21-26.

[16] R. R. Kumar, M. B. Reddy, and P. Praveen, "Text classification performance analysis on machine learning," *Int. J. Adv. Sci. Technol.*, vol. 28, no. 20, pp. 691–697, 2019.

[17] M. P. Vaishnnave, K. Suganya Devi, P. Srinivasan, and G. Arutperumjothi, "Detection and classification of groundnut leaf diseases using KNN classifier," *2019 IEEE Int. Conf. Syst. Comput. Autom. Networking, ICSCAN 2019*, pp. 1–5, 2019, doi: 10.1109/ICSCAN.2019.8878733.

[18] Y. Tan, "An Improved KNN Text Classification Algorithm Based on K-Medoids and Rough Set," *Proc. - 2018 10th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2018*, vol. 1, pp. 109–113, 2018, doi: 10.1109/IHMSC.2018.00032.

[19] F. J. Yang, "An implementation of naive bayes classifier," *Proc. - 2018 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2018*, pp. 301–306, 2018, doi: 10.1109/CSCI46756.2018.00065.

[20] Y. A. Gerhana, I. Fallah, W. B. Zulfikar, D. S. Maylawati, and M. A. Ramdhani, "Comparison of naive Bayes classifier and C4.5 algorithms in predicting student study period," *J. Phys. Conf. Ser.*, vol. 1280, no. 2, 2019, doi: 10.1088/1742-6596/1280/2/022022.

[21] S. Siddiqui, M. A. Rehman, S. M. Daudpota, and A. Waqas, "Opinion mining: An approach to feature engineering," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 3, pp. 159–165, 2019, doi: 10.14569/IJACSA.2019.0100320.

[22] I. M. Parapat and M. T. Furqon, "Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.* , vol. 2, no. 10, pp. 3163–3169, 2018, [Online]. Available: http://j-ptiik.ub.ac.id.

[23] A. F. Ab Nasir *et al.*, "Text-based emotion prediction system using machine learning approach," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 769, no. 1, 2020, doi: 10.1088/1757-899X/769/1/012022.

[24] N. Cahyana, S. Khomsah, and A. S. Aribowo, "Improving Imbalanced Dataset Classification Using Oversampling and Gradient Boosting," *Proceeding - 2019 5th Int. Conf. Sci. Inf. Technol. Embrac. Ind. 4.0 Towar. Innov. Cyber Phys. Syst. ICSITech 2019*, pp. 217–222, 2019, doi: 10.1109/ICSITech46713.2019.8987499.

[25] U. N. Dulhare, K. Ahmad, K. Amali Bin Ahmad, and M. Sharif Hossen, "Big Data and Pattern Recognition," 2020.