JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS
ISSN 2518-8739
31 August 2019, Volume 4, Issue 2, JSEIS, CAOMEI Copyright © 2016-2019
www.jseis.org

COMPARATIVE STUDY OF CLASSIFIERS FOR SENTIMENT ANALYSIS IN BAHASA

ADINDA OCTADIA PUTRI, ANA KURNIAWATI

Department of Computer Science and Information Technology, Faculty of Information System,
Gunadarma University, Depok, Indonesia.
Email: adindaoctadia@gmail.com, ana@staff.gunadarma.ac.id

ABSTRACT

Academic Editor: Masitah Ghazali, Universiti Teknologi Malaysia, Johor, Malaysia

Published: 31st August, 2019

Funding: Authors have no funds.

Article Type: Research Article

As the use of the internet is continuously increasing, there are a huge amount of opinions available online. This is proven by the existence of a site specifically made for book reviews called Goodreads (www.goodreads.com). In this site, the user can freely express their opinions, give an assessment through star-rating, and write a review for specific book they read. The abundance of information from unstructured data like this encourages the emergence of knowledge in text analysis or also known as sentiment analysis. The complexity of analysis sentiment includes selecting appropriate classification algorithm. Other than that, a problem often found in classifying text is imbalanced dataset that can cause seriously negative effect on classifier's performance of machine learning algorithm. The purpose of this research is to compare 3 classifier's performance, which are Naïve Bayes, Random Forest and K-Nearest Neighbor, in dealing with imbalanced dataset. The classification is done for 3 different comparison ratios, which are 90%:10%, 80%:20%, and 70%:30%, and 3 different random sampling values or known as random states, which are 0, 10, and 20. The performance is assessed based on accuracy, confusion matrix, and classification report that includes precision, recall, and F1-score calculation.

Keywords: imbalanced dataset; K-nearest neighbor; Naïve Bayes; random forest; sentiment analysis;

1. INTRODUCTION

Internet today has become an important part of everyday human life, especially because it can be applied in various fields, like opinion submission. People are now able to access not only opinions from family members and friends, but also from strangers through internet that provides a virtual environment so that people could share their experiences via the electronic-of-mouth (WOM) [1]. Goodreads is the world's largest site for readers and book recommendations. It was launched in January 2007 and has mission to help people find and share books [2]. More than just a recommendation site, it is an online community for book reviews and ratings. On July 2019, it has 90 million members, 2.6 billion books added, and 90 million reviews.

Sentiment analysis is a process to determine opinions or feelings from a text [3] which can be classified as positive, negative, or neutral. Companies around the world have implemented machine learning to conduct sentiment analysis automatically in order to get insights from customer's opinions. But, getting an overall sense of those reviews can be time-consuming, however, if only few reviews were read the evaluation would be biased [1]. Complexity in sentiment analysis includes removing unnecessary data from raw datasets, selecting appropriate features or words to represent opinions, and selecting appropriate classification algorithm [4].

From a comparative study research of 5 different classifiers, which are Random Forest, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naïve Bayes, and Decision Tree, it is known that Random Forest generate the best result based on accuracy and processing time needed, which is 88.65% for 8 books and generate the highest accuracy for 6 books. Other than that, KNN generate accuracy up to 84.59% for 8 books and has the highest accuracy for 2 books. Those classifiers ranked first and second based on accuracy, outperformed the other 3 classifiers [5]. In 2017, authors did the same research using Naïve Bayes classifier to classify sentiment analysis in Bahasa and generate accuracy up to 86.67%

According to the description above, this research has purpose to give a comparative study of Naïve Bayes, Random Forest, and KNN classifiers to classify book reviews into positive or negative class based on model evaluation of each classifiers. The data used is 200 reviews from a book titled "Daun Yang Jatuh Tak Pernah Membenci Angin" written by Tere Liye that was posted on Goodreads. The model evaluation includes accuracy, confusion matrix, and classification report that consist of precision, recall, and F1-score. Variable of analysis used are comparison ratio which consist of 90%:10%, 80%:20%, and 70%:30%, and random state parameter to do the

Corresponding Author: Adinda Octadia Putri
Email Address: adindaoctadia@gmail.com

random sampling in the program, which consist of 0, 10, and 20. The comparison is done in Jupyter Notebook environment using Python libraries called Scikit-learn.

2. RESEARCH PROCESS

2.1 Sentiment analysis

Sentiment analysis has many names, usually often referred to as subjectivity analysis, opinion mining, and valuation extraction, with several connections to affective computing (computer recognition and emotional expression) [3]. Some of the most frequently studied research in sentiment analysis are product and film reviews [6][7]. The advantage of the data is that the topic is very clear and it is often assumed that the sentiment expressed in the review is related to the topic. Many also have a star-rating system that serves as a quantitive indication of that opinion. The general task aimed at research on sentiments is to find opinion on the products concerned in various web content [8].

2.2 Text Classification

Text classification is a process to classify a given data instance into pre-specified set of categories. It is the process of finding the correct topics for each document. There are two types of approaches to text categorization, rule based and machine learning based approaches. Machine learning based approach has much higher recall but a slightly lower precision than rule based approache. Therefore, this approach are replacing rule based one for text categorization [9].

2.3 Classifier Types

Classifier used in this research are Naïve Bayes, Random Forest, and K-Nearest Neighbor that will be described below.

2.3.1 Naïve Bayes Classifier

Naïve Bayes is a statistical classification technique based on Bayes theorem with the “naïve” assumption of conditional independence between every pair of features given the value of the class variable. Naïve Bayes equation is stated below [10].

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)} \qquad \text{Eq. (1)}$$

Where :
$P(y \mid x_1, \dots, x_n)$: Posterior probability, or the probability of hypothesis h given the data x_1 to x_n.
$P(y)$: Prior probability of y, or the probability of hypothesis u being true (regardless of the data).
$P(x_1, \dots, x_n)$: Prior probability, or the probability of the data (regardless of the hypotesis).
$P(x_1, \dots, x_n \mid y)$: Posterior probability, or the probability of data x_1 to x_n given that the hypothesis was true.

2.3.2 Random Forest Classifier

Random Forest was the first paper which brought the concept of ensemble of decision trees which is composed by combining multiple decision trees. While dealing with the single tree classifier there may be the problem of noise or outliers which may possibly affect the result of the overall classification method, whereas Random Forest is a type of classifier which is very much robust to noise and outliers because of randomness it provides. Random Forest works as shown in below [11].

Algorithm 1. Random Forest

```
Input: B = Number of Trees, N = Training Data, F = Total Features, f = Subset of Features
Output: Bagged class label for the input data.

1   For each tree in Forest B:
    a   Select a bootstrap sample S of size N from training data.
    b   Create the tree Tb by recursively repeating the following steps for each internal node of the tree.
        i     Choose f at random from the F.
        ii    Select the best among f.
        iii   Split the node.
2   Once B trees are created, test instance will be passed to each tree and class label will be assigned based on
    majority of votes.
```

2.3.3 K-Nearest Neighbor Classifier

KNN algorithm is used to classify instances based on nearest training examples in the same frame space. It is known as lazy learning algorithm in which function is approximated locally and computations are delayed until classification. A majority of instances is used for classification process. Object is classified into the particular class which has maximum number of nearest instances. KNN works as shown below [12].

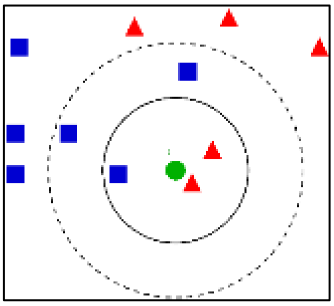

Figure. 1 Example of K-nearest neighbor

From Figure 1, the test instance (green circle) should be classified either into blue square class or into red triangle class. If *k*=3 (solid line circle), test object (green circle) is classified into res triangle class because there are 2 triangle instances and only 1 square instance in the inner circle. If *k*=5 (dashed line circle), test object (green circle) is classified into blue square class because there are 3 blue square instances and only 2 red triangle instances in the inner circle [12].

2.4 Evaluation

Algorithm are mainly compared on accuracy. It is a performance evaluation in the most general way of comparing algorithm, without focusing on each class. Thus, accuracy does not distinguish between the number of correct labels of different classes. The equation of accuracy is stated below [13]:

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad \text{Eq. (2)}$$

Confusion matrix, or error matrix, is a performance measurement for classification models in machine learning. This technique facilitates the identification of confusion between classes and gives an insight not only about errors made but most importantly on the type of error made.

Table. 1 Confusion matrix

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

From confusion matrix, the value of precision, recall and F1-score can be known. The equation of each of them are stated below [13][10].

$$precision = \frac{TP}{TP+FP} \quad \text{Eq. (3)}$$

$$recall = \frac{TP}{TP+FN} \quad \text{Eq. (4)}$$

$$F1 - score = \frac{2 \times recall \times precision}{recall+precision} \quad \text{Eq. (5)}$$

2.5 Research methodology

The research methodology consists of several steps to get final result of classification for each classifiers, which are data collecting, data labelling, data reading, data cleaning, data splitting, data resampling, data converting, data classification, and model evaluation. The steps illustrated in Figure 2 and further explanation will be delivered in 2.3.1 until 2.3.9.

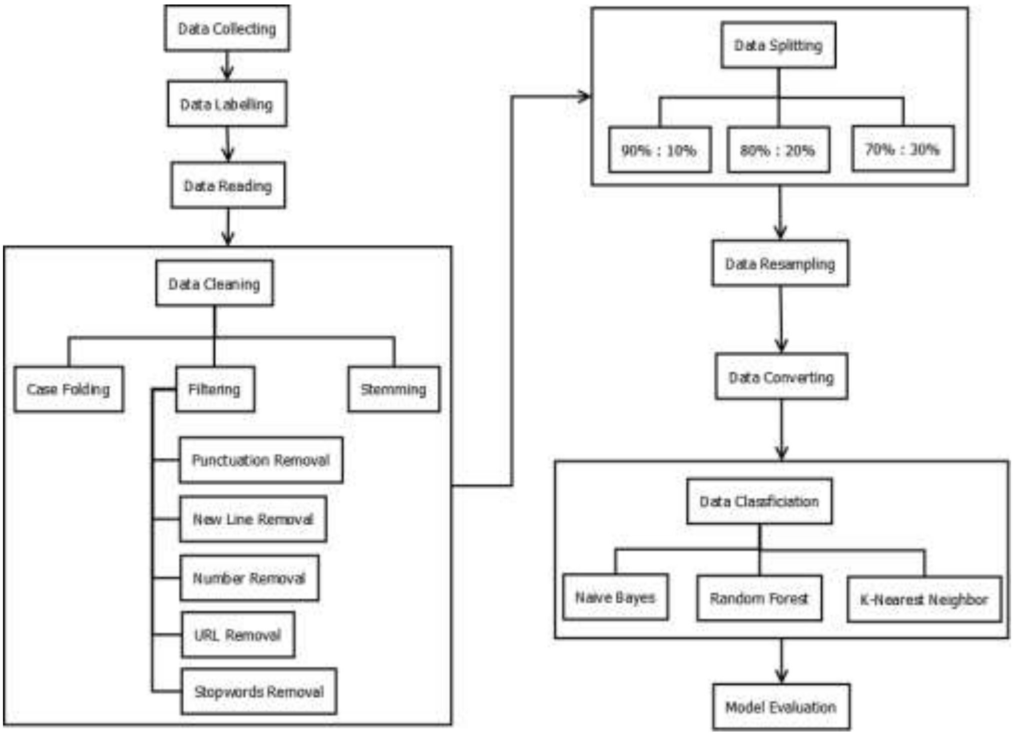

Figure. 2 Research methodology of comparative study of classifier for sentiment analysis in Bahasa

2.5.1 Data collecting

The data used in this research is 200 reviews from a book titled “Daun Yang Jatuh Tak Pernah Membenci Angin” written by Tere Liye that was posted on Goodreads (www.goodreads.com). The data was collected in May 2019 and saved in a CSV file. The information needs to do the classification is the reviews and ratings.

2.5.2 Data labelling

A classifier in supervised learning can only find attribute target if the target label has been defined previously. In this research, labelling is done using star-rating method. For 1 to 2 star-rate reviews will be defined as negative class and for 4 to 5 star-rate reviews will be defined as positive class [14].

2.5.3 Data reading

The data reading is done using Python library called pandas that will import the CSV file of data into Jupyter Notebook. The structure used is DataFrame that will display the data as table. After the data is read, the label must be converted from string into integer. In this research, the positive class will be defined as “1” and the negative class will be defined as “0”.

2.5.4 Data cleaning

The data cleaning consists of 3 main steps, there are case folding, filtering, and stemming. Case folding is done to change all letters into lowercase. Filtering is done to filter features in the corpus because the only needed attribute is words, so attributes other than words needs to be removed. In this research filtering consist of punctuation removal, new line removal, number removal, URL removal, and stop words removal. After that, stemming is done to change words into its root form.

2.5.5 Data splitting

This step is done to split data into testing and training set using train_test_split function in Scikit-learn. The splitting ratio used in this research are 90%:10%, 80%:20%, and 70%:30%. In splitting data using train_test_split function, parameter random state needs to be defined. It has purpose to define the internal random number generator which will decide the splitting of data training and testing. If the parameter is not defined, the splitting step will generate different training and testing data every time the program is run. Random state value can be defined personally by the user, but keep in mind that it will influence the classification result because of data differences for every random state value, so it become important to know the most optimal value of random state. In this research, the random state value that will be used are 0, 10, and 20.

2.5.6 Data resampling

In this research, the data used is unbalance. From 200 reviews, there are 166 positive reviews and 34 negative reviews. Most algorithm usually assume balanced class distributions. The imbalanced dataset will cause most standard machine learning algorithm to be biased toward the majority class because they try to optimize overall accuracy, which is overwhelmed by majority classes and ignore minority class [15]. To resample data, the technique used id over-sampling minority class by duplicating minority data randomly to adjust the number of data in majority class.

2.5.7 Data Converting

Unlike human, classifiers cannot understand text, it only can understand numbers. This step is done to convert all features (words in the corpus) into numbers by using TF-IDF method. TF-IDF calculation is arranged from two tern, TF (Term Frequency) and IDF (Inverse Document Frequency). TF has purpose to measure how often a word appears in the document. But, in analysis sentiment, it will be more profitable to know the more unique words than the most frequent words [8]. That is the purpose of IDF. The conversion is using TfidfVectorizer function in Scikit-learn.

2.5.8 Data Classification

As described above, the classification is done for 3 different ratio and 3 different random state. The classification scenario can be seen in Table 2.

Table. 2 Classification scenario

Classifiers	Scenario	Ratio	Random State
Naïve Bayes	1	90% : 10%	0
	2		10
	3		20
	4	80% : 20%	0
	5		10
	6		20
	7	70% : 30%	0
	8		10
	9		20
Random Forest	10	90% : 10%	0
	11		10
	12		20
	13	80% : 20%	0
	14		10
	15		20
	16	70% : 30%	0
	17		10
	18		20
K-Nearest Neighbor	19	90% : 10%	0
	20		10
	21		20
	22	80% : 20%	0
	23		10
	24		20
	25	70% : 30%	0
	26		10
	27		20

The variation of Naïve Bayes used in this research is Multinomial Naïve Bayes. This variation is suitable for discreate feature like number of words for text classification. In Scikit-learn, Multinomial Naïve Bayes can be processed using MultinomialNB() function. As for Random Forest, the Scikit-learn function used is RandomForestClassifiers(). In this function, parameter n_estimator defined is 100. This parameter stated a number of tree in the forest and 100 is the default value of Scikit-learn version 0.22 [10]. Other than that, parameter random_state is also needed which will be adjusted with random state defined in data splitting step. And as for KNN, the Scikit-learn function used is KNeighborsClassifier(). In this function, parameter n_neighbors defined is 3. This parameter stated the number of neighbors in the classifier.

2.5.9 Model evaluation

Model evaluation is done in 3 ways; accuracy, confusion matrix, and classification report. Basically, accuracy is the most common way to evaluate the model. But predictive accuracy may not be suitable for use when the data is unbalance [16]. Confusion matrix is another way that can be used to describe the breakdown of errors in predictions for an unseen dataset. Other than that, the exactness, completeness, and the balance between the two can be seen through precision, recall, and F1-score that presented as classification report in Scikit-learn. All of the model evaluation described can be done using Scikit-learn functions which are score() to calculate accuracy, confusion_matrix() to calculate confusion matrix, and classification_report() to calculate precision, recall, and F1-score.

3. RESULTS

3.1 Accuracy result

This section describes the result of the first model evaluation; accuracy. The result is displayed in Table 3 for all of the scenarios.

Table. 3 The result of model evaluation – accuracy

Scenario	Accuracy	Scenario	Accuracy	Scenario	Accuracy
1	0.60	**10**	0.75	**19**	0.75
2	0.55	**11**	0.75	**20**	0.65
3	0.80	**12**	1.00	**21**	0.80
4	0.70	**13**	0.73	**22**	0.70
5	0.65	**14**	0.85	**23**	0.68
6	0.73	**15**	0.80	**24**	0.80
7	0.63	**16**	0.77	**25**	0.73
8	0.62	**17**	0.88	**26**	0.78
9	0.67	**18**	0.82	**27**	0.82

Scenario 1 to 9 displays accuracy for Naïve Bayes classifier. It shows that for 90%:10% ratio, the highest accuracy is achieved when the random state value is 20 (scenario 3). For 80%:20% ratio, the highest accuracy is achieved when the random state value is 20 (scenario 6). And for 70%:30% ratio, the highest accuracy is achieved when the random state value is 20 (scenario 9).

Scenario 10 to 18 displays accuracy for Random Forest classifier. It shows that for 90%:10% ratio, the highest accuracy is achieved when the random state value is 20 (scenario 12). For 80%:20% ratio, the highest accuracy is achieved when the random state value is 10 (scenario 14). And for 70%:30% ratio, the highest accuracy is achieved when the random state value is 10 (scenario 17).

Scenario 19 to 27 displays accuracy for KNN classifier. It shows that for 90%:10% ratio, the highest accuracy is achieved when the random state value is 20 (scenario 21). For 80%:20% ratio, the highest accuracy is achieved when the random state value is 20 (scenario 24). And for 70%:30% ratio, the highest accuracy is achieved when the random state value is 20 (scenario 27).

3.2 Confusion matrix result

This section describes the result of the second model evaluation; confusion matrix. The result is displayed in Table 4 for all of the scenarios.

Table. 4 The result of model evaluation – confusion matrix

Scenario	Confusion Matrix				Scenario	Confusion Matrix				Scenario	Confusion Matrix			
	TN	FP	FN	TP		TN	FP	FN	TP		TN	FP	FN	TP
1	5	0	8	7	**10**	0	5	0	15	**19**	1	4	1	14
2	2	3	6	9	**11**	0	5	0	15	**20**	1	4	3	12
3	0	0	4	16	**12**	0	0	0	20	**21**	0	0	4	16
4	5	6	6	23	**13**	0	11	0	29	**22**	2	9	3	26
5	2	4	10	24	**14**	0	6	0	34	**23**	1	5	8	26
6	6	2	9	23	**15**	0	8	0	32	**24**	2	6	2	30
7	11	3	19	27	**16**	0	14	0	46	**25**	3	11	5	41
8	4	4	19	33	**17**	1	7	0	52	**26**	3	5	8	44
9	4	7	13	36	**18**	0	11	0	49	**27**	3	8	3	46

As described above, confusion matrix has purpose to show the breakdown of errors in the process. The description of each scenario with the highest accuracy will be stated below.

a Scenario 3 : there are 0 negative data correctly predicted as negative (TN), 0 negative data wrongly predicted as positive (FP), 4 positive data wrongly predicted as negative (FN), and 16 positive data correctly predicted as positive (TP).

b Scenario 6 : there are 6 negative data correctly predicted as negative (TN), 2 negative data wrongly predicted as positive (FP), 9 positive data wrongly predicted as negative (FN), and 23 positive data correctly predicted as positive (TP).

c Scenario 9 : there are 4 negative data correctly predicted as negative (TN), 7 negative data wrongly predicted as positive (FP), 13 positive data wrongly predicted as negative (FN), and 36 positive data correctly predicted as positive (TP).

d Scenario 12 : there are 0 negative data correctly predicted as negative (TN), 0 negative data wrongly predicted as positive (FP), 0 positive data wrongly predicted as negative (FN), and 20 positive data correctly predicted as positive (TP).

e Scenario 14 : there are 0 negative data correctly predicted as negative (TN), 6 negative data wrongly predicted as positive (FP), 0 positive data wrongly predicted as negative (FN), and 34 positive data correctly predicted as positive (TP).

f Scenario 17 : there are 1 negative data correctly predicted as negative (TN), 7 negative data wrongly predicted as positive (FP), 0 positive data wrongly predicted as negative (FN), and 52 positive data correctly predicted as positive (TP).

g Scenario 21 : there are 0 negative data correctly predicted as negative (TN), 0 negative data wrongly predicted as positive (FP), 4 positive data wrongly predicted as negative (FN), and 16 positive data correctly predicted as positive (TP).

h Scenario 24 : there are 2 negative data correctly predicted as negative (TN), 8 negative data wrongly predicted as positive (FP), 2 positive data wrongly predicted as negative (FN), and 30 positive data correctly predicted as positive (TP).

i Scenario 27 : there are 3 negative data correctly predicted as negative (TN), 8 negative data wrongly predicted as positive (FP), 3 positive data wrongly predicted as negative (FN), and 46 positive data correctly predicted as positive (TP).

3.3 Classification Report

This section describes the result of the third model evaluation; classification report. As described above, classification report has purpose to show the exactness (precision), completeness (recall), and the balance between the two (F1-score). The result is displayed in Table 5 for all of the scenarios.

Table. 5 The result of model evaluation – classification report

Scenario	Precision		Recall		F1-score	
	Positive	Negative	Positive	Negative	Positive	Negative
1	1.00	0.38	0.47	1.00	0.64	0.56
2	0.75	0.25	0.60	0.40	0.67	0.31
3	1.00	0.00	0.80	0.00	0.89	0.00
4	0.79	0.45	0.79	0.45	0.79	0.45
5	0.86	0.17	0.71	0.33	0.77	0.22
6	0.92	0.40	0.72	0.75	0.81	0.52
7	0.90	0.37	0.59	0.79	0.71	0.50
8	0.89	0.17	0.63	0.50	0.74	0.26
9	0.84	0.24	0.73	0.36	0.78	0.29
10	0.75	0.00	1.00	0.00	0.86	0.00
11	0.75	0.00	1.00	0.00	0.86	0.00
12	1.00	0.00	1.00	0.00	1.00	0.00
13	0.72	0.00	1.00	0.00	0.84	0.00
14	0.85	0.00	1.00	0.00	0.92	0.00
15	0.80	0.00	1.00	0.00	0.89	0.00
16	0.77	0.00	1.00	0.00	0.87	0.00
17	0.88	1.00	1.00	0.12	0.94	0.22
18	0.82	0.00	1.00	0.00	0.90	0.00
19	0.78	0.50	0.93	0.20	0.85	0.29
20	0.75	0.25	0.80	0.20	0.77	0.22
21	1.00	0.00	0.80	0.00	0.89	0.00

Scenario	Precision		Recall		F1-score	
	Positive	Negative	Positive	Negative	Positive	Negative
22	0.74	0.40	0.90	0.18	0.81	0.25
23	0.84	0.11	0.76	0.17	0.80	0.13
24	0.83	0.50	0.94	0.25	0.88	0.33
25	0.79	0.38	0.89	0.21	0.84	0.27
26	0.90	0.27	0.85	0.38	0.87	0.32
27	0.85	0.50	0.94	0.27	0.89	0.35

3.4 Summary of classification results

This section describes summary of classification results that is done using 3 model evaluation as stated above. To simplify comparison, the summary is presented using a chart each for accuracy, precision in positive and negative class, recall in positive and negative class, and F1-score in positive and negative class. The comparison is done with calculating the average of value generated for each ratio in each classifier.

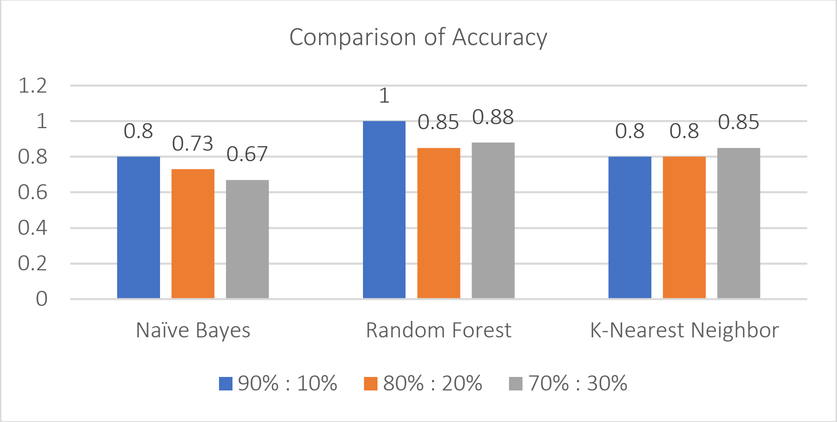

Figure. 3 Comparison of accuracy

From the data shown in Figure 3, the average value of accuracy each for Naïve Bayes, Random Forest, and KNN are 0.73, 0.91, and 0.82. So it can be concluded that Random Forest has the most optimal performance based on accuracy.

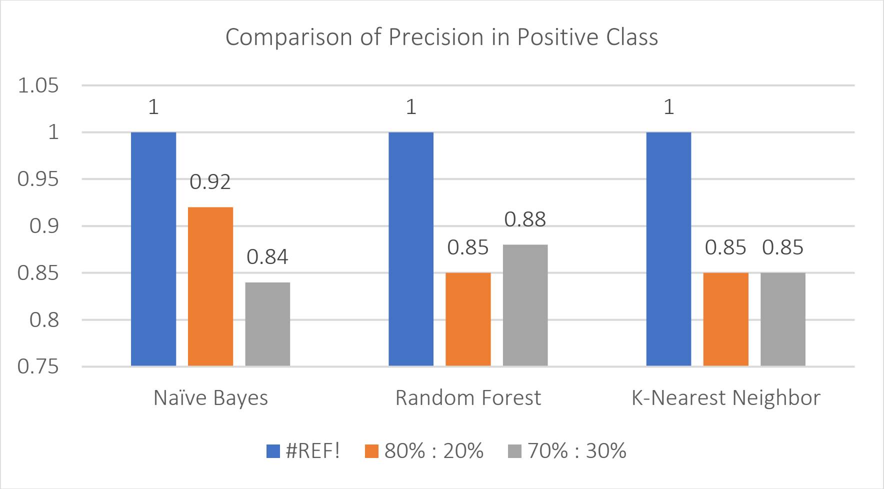

Figure. 4 Comparison of precision in positive class

From the data shown in Figure 4, the average value of precision in positive class each for Naïve Bayes, Random Forest, and KNN are 0.92, 0.91, and 0.90. Thus, it can be concluded that Naïve Bayes has the most optimal performance to classify the positive data correctly into positive class.

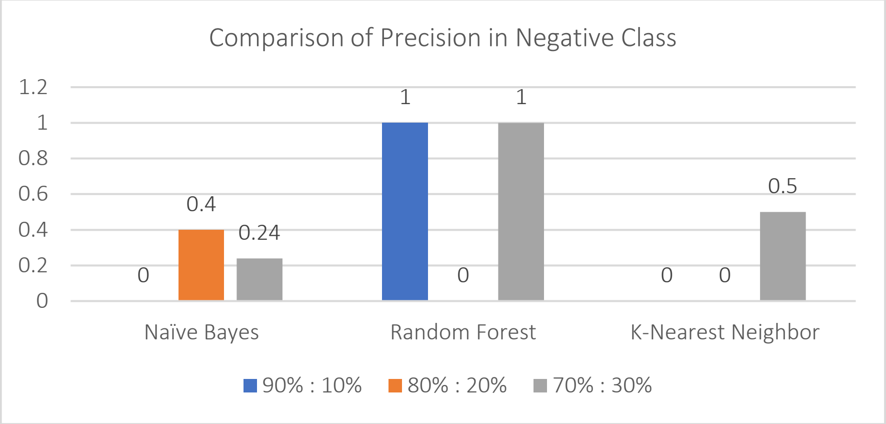

Figure. 5 Comparison of precision in negative class

From the data shown in Figure 5, the average value of precision in negative class each for Naïve Bayes, Random Forest, and KNN are 0.21, 0.67, and 0.17. So it can be concluded that Random Forest has the most optimal performance to classify the negative data correctly into negative class.

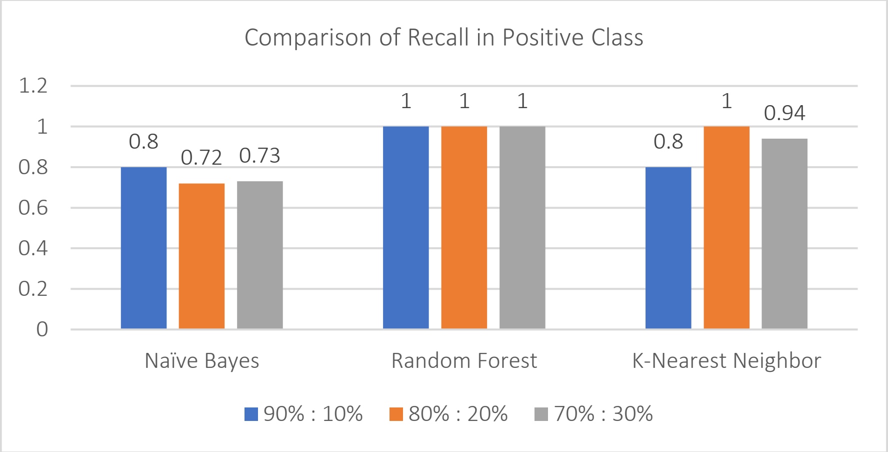

Figure. 6 Comparison of recall in positive class

From the data shown in Figure 6, the average value of recall in positive class each for Naïve Bayes, Random Forest, and KNN are 0.75, 1.00, and 0.91. So it can be concluded that Random Forest has the most optimal performance to classify all the positive data correctly.

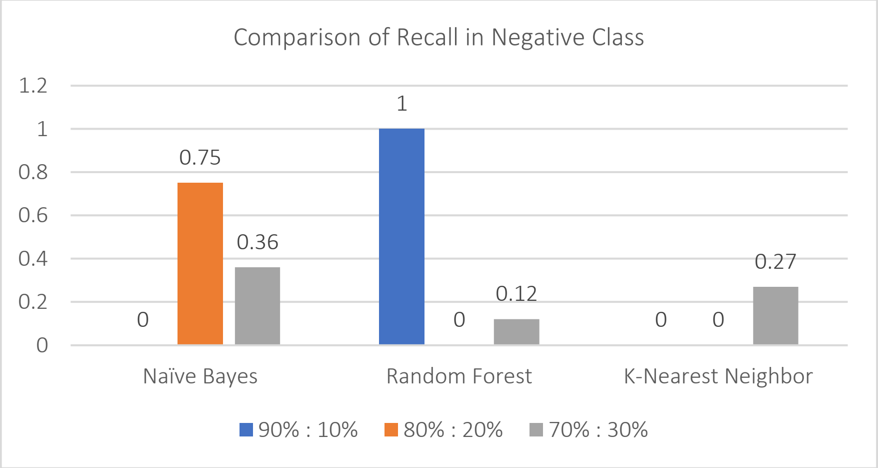

Figure. 7 Comparison of recall in negative class

From the data shown in Figure 7, the average value of recall in negative class each for Naïve Bayes, Random Forest, and KNN are 0.37, 0.09, and 0.04. So it can be concluded that Naïve Bayes has the most optimal performance to classify all the negative data correctly.

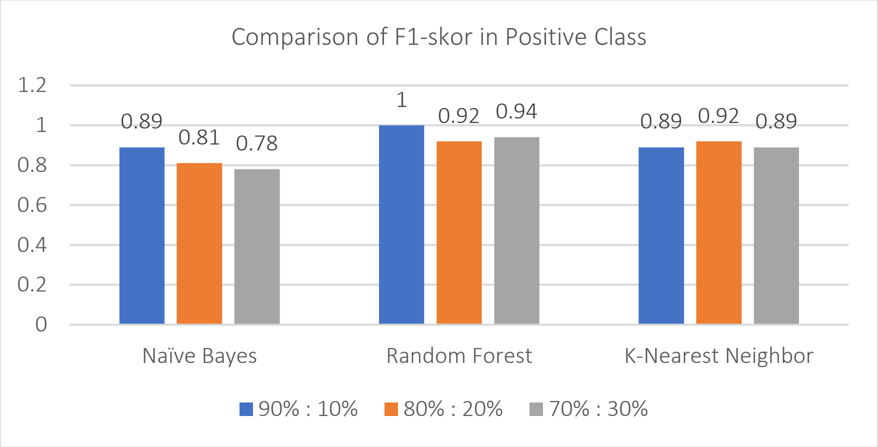

Figure. 8 Comparison of F1-score in positive class

From the data shown in Figure 8, the average value of F1-score in positive class each for Naïve Bayes, Random Forest, and KNN are 0.83, 0.95, and 0.90. So it can be concluded that Random Forest has the most balance value of precision and recall for positive class.

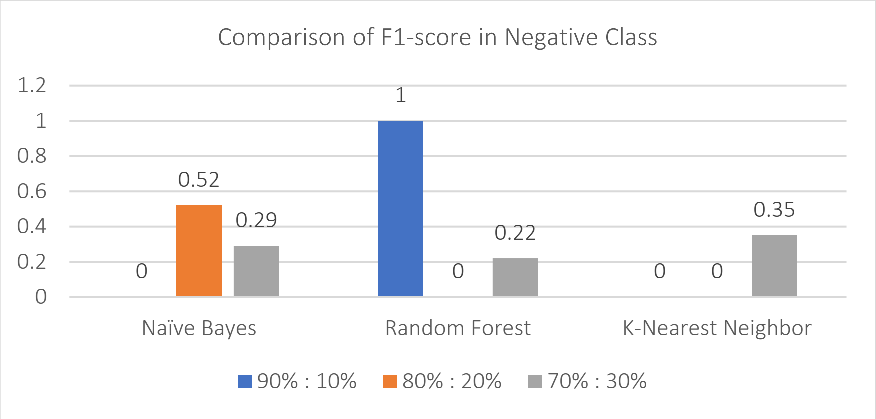

Figure. 9 Comparison of F1-score in negative class

From the data shown in Figure 9, the average value of F1-score in positive class each for Naïve Bayes, Random Forest, and KNN are 0.27, 0.41, and 0.12. So it can be concluded that Random Forest has the most balance value of precision and recall for negative class.

3.5 Analysis of results

This section describes several points of analysis achieved from the summary of result in previous section. As stated above, the analysis is done based on the average of comparison of each model evaluation which are accuracy, precision, recall, and F1-score. To facilitate the data reading and the making of analysis conclusion, the average of comparison summary of classification result is presented in Table 6.

Table. 6 Average of comparison summary of classification result

Classifiers	Accuracy	Classification Report					
		Precision		Recall		F1-score	
		Positive	Negative	Positive	Negative	Positive	Negative
Naïve Bayes	0.73	0.92	0.21	0.75	0.37	0.83	0.27
Random Forest	0.91	0.91	0.67	1.00	0.04	0.95	0.41
K-Nearest Neighbor	0.82	0.90	0.17	0.91	0.09	0.90	0.12

From Table 7, the analysis of result can be concluded as stated below.

a Naïve Bayes classifier generate the highest value for precision in positive class, but the lowest for recall in positive class. It means that Naive Bayes is able to classify positive data correctly as positive class, but a lot of positive data is missing or not predicted. For the negative class, Naïve Bayes is on the second place for precision and on the first place for recall. It means that Naïve Bayes is able enough to classify negative data correctly as negative class and a lot of negative data is able to be predicted. As for F1-score, Naïve Bayes is on the third place for positive class with average value of 0.83 and on the second place for negative class with average value of 0.27.

b Random Forest classifier is on the second place for precision in positive class and on the first place for recall in positive class. It means that Random Forest is able enough to classify a lot of positive data correctly as positive class even though there are still positive data that is wrongly classified as negative, and a lot of positive data is able to be predicted by the classifier. For negative class, Random Forest is on the first place for precision and on the third place for recall. It means that Random Forest is able to predict negative data correctly as negative class, but a lot of negative data is missing or not predicted. As for F1-score, Random Forest is on the first place both for positive and negative class with average value of each 0.95 and 0.41.

c K-Nearest Neighbor classifier is on the third place both for precision and recall in positive class. It means that KNN is poor in classifying positive data as positive class and a lot of positive data is missing or not predicted. For negative class, KNN is on the third level both for precision and recall. It means that KNN is poor in classifying negative data as negative class and a lot of negative data is missing or not predicted. As for F1-score, KNN is on the second place for positive class with average value of 0.90 and on the third place for negative class with average of 0.12.

d From point 1 to 3, it can be concluded that Random Forest has the best performance to classify data used in the research. Then followed by Naïve Bayes in the second place and KNN in the third place.

e From point 1 to 3, it can be concluded that all three classifiers work better in classifying positive class than in negative class. This is caused by data imbalanced happened in the research. As described above, this research has unbalanced dataset that has positive data a lot more than the negative data, so resample technique is needed to be done to balance the data.

f The imbalanced of data can influence the classification result. The classifier's performance tends to decrease by allocating all cases to majority class.

g According to the result, from 9 scenario that has the highest performance value for each classifier and ratio, 7 of them is using random state value of 20, while the other 2 is using random state value as 10 which only happened in Random Forest classifier. Even though Random Forest is the only classifier that works well by using various value of random state (10 and 20), but this classifier generates the highest value of model evaluation compared to the other 2 classifiers. It means that a classifier can still perform well even though by using various data splitting combination.

h According to the result, Random Forest is on the first place for all of the comparison ratio. For 90%:10% ratio, the accuracy is up to 1.00 for random state value of 20. For 80%:20% ratio, the accuracy is up to 0.85 for random state value of 10. And for 70%:30%, the accuracy is up to 0.88 for random state value of 10.

4. CONCLUSION

The comparative study and analysis of three classifiers, which are Naïve Bayes, Random Forest, and K-Nearest Neighbor by using Python libraries called Scikit-learn in Jupyter Notebook environment is successfully done. The result obtained from model evaluation and its analysis shows that Random Forest has the best performance in classifying data used in this research. Then followed by Naïve Bayes in the second place and K-Nearest Neighbor in the third place. There are a lot of factors that can affect the classification result and classifier's performance, there are the number of data used for classification, dataset balance, completeness of data cleaning steps done, comparison ratio, and parameters used in the classifiers such as random state to define random sampling in the document.

ACKNOWLEDGEMENT

Special thanks to Gunadarma University of Indonesia for providing resources to carry out this research.

REFERENCES

1 Zhang Z, Ye Q, Zhang Z, Li Y. *Sentiment classification of Internet restaurant reviews written in Cantonese*. Expert Systems with Applications. 2011. 38(6):7674-82.

2 Chandler, O. *Goodreads*. Available at: http://www.goodreads.com/. [Accessed 28 July 2019].

3 Pang B, Lee L. *A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts*. In Proceedings of the 42nd annual meeting on Association for Computational Linguistics (p. 271). 2004. Association for Computational Linguistics.

4 Lestari NM, Putra IK, Cahyawan AK. *Personality types classification for indonesian text in partners searching website using naïve bayes methods*. IJCSI International Journal of Computer Science Issues, 10:1-8. 2013.

5 Srujan KS, Nikhil SS, Rao HR, Karthik K, Harish BS, Kumar HK. *Classification of amazon book reviews based on sentiment analysis*. In Information Systems Design and Intelligent Applications (pp. 401-411). 2018. Springer, Singapore.

6 Liu B, Hu M, Cheng J. *Opinion observer: analyzing and comparing opinions on the web*. InProceedings of the 14th international conference on World Wide Web (pp. 342-351). 2005. ACM.

7 Popescu AM, Etzioni O. *Extracting product features and opinions from reviews*. In Natural language processing and text mining (pp. 9-28). 2007. Springer, London.

8 Mejova Y. *Sentiment analysis: An overview*. University of Iowa, Computer Science Department. 2009.

9 Pawar PY, Gawande SH. *A comparative study on different types of approaches to text categorization*. International Journal of Machine Learning and Computing, 2(4):423. 2012.

10 Pedregosa et al. *Scikit-learn: Machine Learning in Python*. JMLR 12. pp. 2825-2830. 2011.

11 Parmar H, Bhanderi S, Shah G. *Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyperparameters.* In International Conference on Information Science. Kerala. 2014.

12 Khamar K. *Short text classification using kNN based on distance function*. International Journal of Advanced Research in Computer and Communication Engineering, 2(4):1916-9. 2013.

13 Sokolova M, Japkowicz N, Szpakowicz S. *Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation*. In Australasian joint conference on artificial intelligence (pp. 1015-1021). 2006. Springer, Berlin, Heidelberg.

14 Dave K, Lawrence S, Pennock DM. *Mining the peanut gallery: Opinion extraction and semantic classification of product reviews*. In Proceedings of the 12th international conference on World Wide Web (pp. 519-528). 2003. ACM.

15 Sonak A, Patankar RA. *A survey on methods to handle imbalance dataset*. Int. J. Comput. Sci. Mobile Comput. 4(11):338-43. 2015.

16 Chawla NV. *Data mining for imbalanced datasets: An overview*. In Data mining and knowledge discovery handbook (pp. 875-886). 2009. Springer, Boston, MA.

AUTHORS PROFILE

Adinda Octadia Putri, S.Kom received a Bachelor degree in Computer Science and Information Technology from Gunadarma University, Depok, Department of Information System in 2017. Currently continuing Master degree in Gunadarma University at Department of Business Information System.

Dr. Ana Kurniawati, S.T., MMSI received a Bachelor degree of Engineering from Gunadarma University, Depok, in 1998. Received Master of Information System Management from Gunadarma University, Depok, in 2002. Received a Doctorate from Gunadarma University, Depok, in 2010. Has research interests in Design of Human Body Balancing Detection System Computer Based Personal, Data Mining, Database, and Document Similarity Detection. Currently is a Secretary of Information System Department at Gunadarma University.