Comparison of Iris dataset classification with Gaussian naïve Bayes and decision tree algorithms

Yasi Dani, Maria Artanta Ginting
Computer Science Department, School of Computer Science, Bina Nusantara University, Bandung Campus, Bandung, Indonesia

Article Info

Article history:

Received Aug 3, 2023
Revised Oct 16, 2023
Accepted Dec 13, 2023

Keywords:

Classification
Decision tree
Gaussian naïve Bayes
Iris dataset
Machine learning

ABSTRACT

In this study, we apply two classification algorithm methods, namely the Gaussian naïve Bayes (GNB) and the decision tree (DT) classifiers. The Gaussian naïve Bayes classifier is a probability-based classification model that predicts future probabilities based on past experiences. Whereas the decision tree classifier is based on a decision tree, a series of tests that are performed adaptively where the previous test affects the next test. Both of these methods are simulated on the Iris dataset where the dataset consists of three types of Iris: setosa, virginica, and versicolor. The data is divided into two parts, namely training and testing data, in which there are several features as information on flower characteristics. Furthermore, to evaluate the performance of the algorithms on both methods and determine the best algorithm for the dataset, we evaluate it using several metrics on the training and testing data for each method. Some of these metrics are recall, precision, F1-score, and accuracy where the higher the value, the better the algorithm's performance. The results show that the performance of the decision tree classifier algorithm is the most outperformed on the Iris dataset.

Corresponding Author:

Yasi Dani
Computer Science Department, School of Computer Science, Bina Nusantara University
Pasir Kaliki street No.25-27, Ciroyom, Bandung 40181, Indonesia
Email: yasi.dani@binus.ac.id

1. INTRODUCTION

Classification is a machine learning approach in data mining that is often used where many methods are chosen to classify a dataset [1], [2]. Classifications that involve two classes are called binary classifications [3], [4], while those that involve more than two classes are called multi-class classifications [5]–[7]. In real applications, classification techniques are needed such as medical disease analysis, text classification, user smartphone classification, and images [8]–[10].

In recent years, many researchers have studied machine learning classification methods using the Iris dataset. Wu *et al.* [11] compared the classification of the Iris dataset using the boosting tree, random forest, and GraftedTrees algorithms, where the performance of the algorithms was still below 0.85. Thirunavukkarasu *et al.* [12] classified the Iris dataset using the KNN algorithm and the performance of the algorithm uses one metric which is the accuracy value where the accuracy of the training data is 0.975 and the test data is 0.967. Swain *et al.* [13] studied neural networks to classify the Iris dataset and evaluated the performance of the algorithm using one metric which is the accuracy value where the value is in the interval [0.833, 0.967]. Ghazal *et al.* [14] compared three classification algorithms namely decision tree, neural networks, and naïve Bayes using WEKA software where these algorithms were evaluated using one metric which was the ROC curve whose value was in the interval [0.955, 0.941].

The Gaussian naïve Bayes (GNB) classifier is a classification technique that has independent assumptions on its features [15]. In general, the assumption of independence is a poor assumption, however, the GNB classifier is very competitive and superior in its application compared to other classifiers [16]–[19]. The decision tree (DT) classifier is an effective and relatively fast classification technique compared to other classification methods. This classifier is often used since its accuracy is almost similar to other classification methods, even outperforming other methods. The DT algorithm can be implemented serially or in parallel based on the volume of data, available memory space on computer resources, and the scalability of the algorithm. The use of DT classifiers is now widely studied in the field of machine learning [20]–[23].

In our work, we use two classification algorithms that are the GNB and DT classifiers on the Iris dataset using Python 3.7.4 software and we also evaluate the performance of the algorithms using several metrics, namely recall, precision, accuracy, and F1-score. This dataset is divided into two parts, namely training (67%) and testing data (33%). Next, the results of the algorithm performance of the two methods in the dataset are compared by calculating these metrics, then we determine which classification method is the most outperforming for this dataset.

The remainder of this paper is organized as follows. Section 2 explains the algorithms of the GNB and DT, the proposed method for the Iris dataset, and the performance evaluation of these algorithms on the dataset. Next, the results of the classification and performance evaluation are presented in section 3. Finally, section 4 summarizes our conclusions.

2. THE MATERIALS AND METHOD

In this section, we have implemented two classification techniques that are already popular in machine learning algorithms, namely GNB and DT classifiers. Then, we describe the dataset and research methods. Finally, we describe several metrics to evaluate the performance of the algorithm.

2.1. GNB classifier

The GNB is a model that calculates probabilities and this model is based on the Bayes theorem [24]. The GNB is also an extension of naïve Bayes which follows the normal Gaussian distribution [25]–[28]. Suppose $X = (x_1, x_2, x_{3,}, \dots x_m)$ where x_i is the i-th feature and Y is the response or class variable, then the Bayes' theorem formula is

$$P(Y|X) = \frac{P(Y|X)P(Y)}{P(X)}. \tag{1}$$

Using Bayes' theorem (1), $P(Y|X)$ is the probability of event X (as a hypothesis) where Y (as a fact) has occurred. Then, in this case the assumptions of predictors or features are independent, that is, the existence of features does not affect other features, so this phenomenon is called naive. The formula is defined as (2).

$$P(Y|x_1, x_{2,}, x_{3,}, \dots x_m) = \frac{P(x_1|Y)P(x_2|Y)P(x_3|Y)\dots P(x_m|Y)P(Y)}{P(X)}, \tag{2}$$

From (2) it can be calculated the probability value for each data where for each entry in the dataset, the denominator is static, so the denominator can be removed. Thus, (2) can be written as (3).

$$P(Y|x_1, x_{2,}, x_{3,}, \dots x_m) \propto P(Y) \prod_{i=1}^{m} P(x_i|Y), \tag{3}$$

and the response or class variable can be obtained by using the maximum probability, which is as in (4) [29].

$$Y = argmax_Y\, P\,(Y) \prod_{i=1}^{m} P(x_i|Y). \tag{4}$$

In this study, the feature is the characteristic information of the Iris flower, while it is the type or species of Iris where the classification of this species is multivariate. The type of naïve Bayes classifier chosen in this study is GNB. Since the predictor is a continuous value (not discrete), the assumption is that these values are taken from the Gaussian distribution. Therefore, the equation for probability becomes as (5).

$$P(x_i|Y) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} exp(-\frac{(x_i-\mu_Y)^2}{2\sigma_Y^2}). \tag{5}$$

2.2. DT classifier

The DT is a learning model for classification. This classifier is based on the concept of a structured tree where internal nodes represent features, branches in the tree define decision rules, and each leaf node defines an outcome [30], [31]. The topmost node is called the root node in the DT. This technique learns to partition by feature value and partitions the tree recursively [32].

One of the reasons this classifier is often used is that the concept in a decision tree imitates the logic of human thinking when making decisions, so it is easy to understand since it can be visualized like a tree structure as shown in Figure 1 which is an example of a flowchart to make it easier to understand the DT [20], [33].

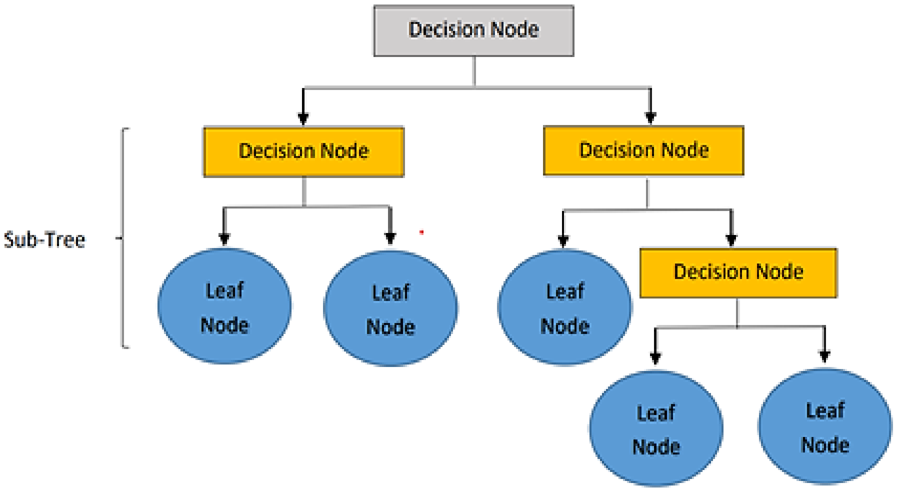


Figure 1. The DT flowchart

2.2.1. Attribute selection measures

Attribute selection measures (ASM) aims at the technique of selecting the best features for tree nodes. There are two ASM techniques [34], namely Information Gain and Gini Index. Information Gain is a method for measuring the weighting of a feature by maximizing entropy which computes the information in the response variable. According to these values, we divide the nodes and build a DT. In this algorithm, we choose the information gain from the highest node, then the node is split. The information gain (IG) formula (6).

$$IG(S,A) = E(S) - [\sum_{i=1}^{n} \frac{|A_i|}{|S|} E(A_i)], \quad (6)$$

where S is the case set, A is the feature set, n is the number of feature partitions, and E is the entropy which can be calculated as [35].

$$E(S) = \sum_{i=1}^{n} -p_i \, log_2 \, p_i, \quad (7)$$

where p is the probability of the event. Gini index (GI) is a cost function which aims to evaluate the separation in the dataset. This formula in (8)

$$GI(S) = 1 - \sum_{i=1}^{n} p_i^2. \quad (8)$$

2.2.2. DT algorithm

DT is a predictive modeling in data analysis that uses a tree structure. This DT aims to describe and make decisions based on a series of rules and conditions. The steps for classifying datasets with the DT algorithm are as follows.

Step 1: Start with the root node, suppose S comprises the entire dataset.
Step 2: Find the best features using ASM.
Step 3: Splits into subsets that comprise the possible values for the best features.
Step 4: Generate the nodes of the decision tree that consist of the best features.

Step 5: Make a new decision tree using the dataset subsets in step 3. Then, we keep on this process until the final node.

2.3. Dataset

Ronald Fisher is a biologist from England. In 1936, he researched and compiled a species dataset on Iris which was multivariate and used some of its measurements in taxonomic problems. There are three types of species in the Iris dataset, namely setosa, virginica, and versicolor. The dataset consists of 150 samples and this dataset is balanced since the ground truth data for each species is 50 samples. The four features that are informed of each data point are length, width of sepals, and petals, where the units are in centimeters. This dataset is taken from the UCI machine learning repository. For more details as shown in Table 1.

Table 1. Some sample instances of the Iris dataset

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	Setosa
4.9	3.0	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
7.0	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
6.3	3.3	6.0	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3.0	5.9	2.1	Virginica

To identify patterns and relationships between different variables in a dataset, we use pairwise plots. Figure 2 is a paired plot that visualizes data from the relationship between information features in the Iris flower dataset. For more details, the blue color represents the Iris-setosa species, the yellow color denotes the Iris-versicolor species, and the green color represents the Iris-virginica species.

2.4. Performance evaluation

There are many metrics that are applied to evaluate the ability of multi-class classifiers. Most metric calculations are based on the confusion matrix, as it includes all relevant information about algorithm performance and classification rules. The confusion matrix is a table for measuring performance in machine learning classification problems where the output can be two or more classes. This table records the number of data points for which the actual classification and predicted classification. This table is also used as a performance measure. Then this table consists of combinations of predicted values and actual values. First, a true positive (TP) is a result correctly identified by the algorithm as positive. Second, a false negative (FN) is the result of being incorrectly identified by the algorithm as negative. Third, a true negative (TN) is a result correctly identified by the algorithm as negative. Fourth, a false positive (FP) is a result incorrectly identified by the algorithm as positive. When the predicted value is a real number, we need to define a threshold [36]. Recall, precision, accuracy, and F1-score are four evaluation metrics that are very frequently used in machine learning when evaluating the performance of classifier algorithms. Recall is the proportion of the number of true positives to the total number of samples that are classified positively which can be seen in (9).

$$Recall = \frac{TP}{TP+FN}. \tag{9}$$

Precision is the proportion between samples that we predict correctly to all positive samples. In other words, precision interprets how reliable the algorithm is when the sample is identified as positive. This formula in (10).

$$Precision = \frac{TP}{TP+FP}. \tag{10}$$

Accuracy is an evaluation score used to measure the ratio of the number of correct predictions produced by an algorithm to the total predictions. In other words, accuracy is used to evaluate the proportion of correct predictions made by an algorithm in a classification problem. This formula can be written as (11).

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \times 100\%. \tag{11}$$

F1-score is a metric that combines recall and precision metrics where the calculation is based on the concept of a harmonic average. The range of F1-score values is between 0 and 1 where the score is close to 1 so that the algorithm performance is higher. For more details, the formula in (12).

$$F1 - score = 2 \times \left(\frac{precision \times recall}{precision + recall}\right). \tag{12}$$

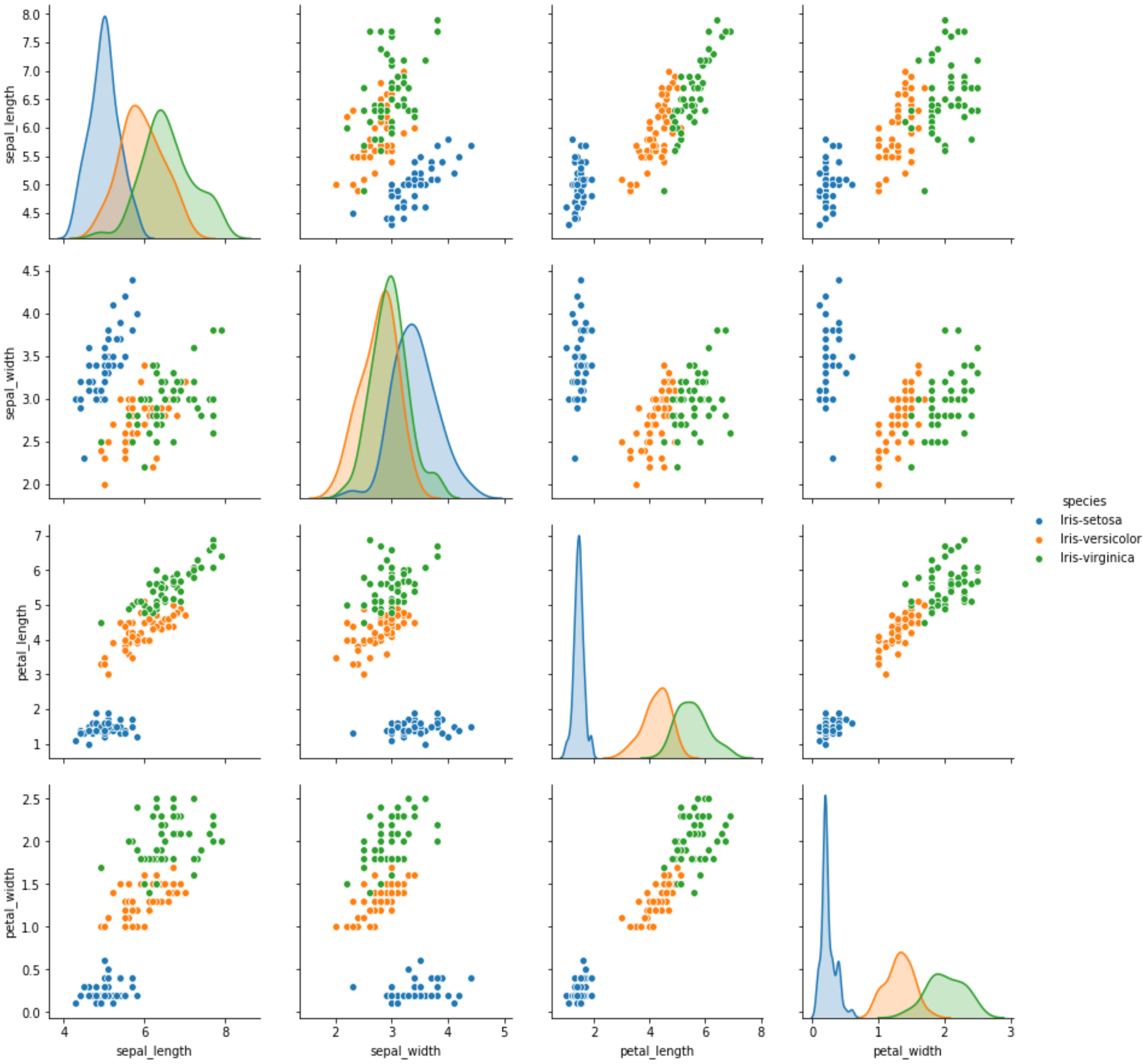


Figure 2. Paired plot of the Iris dataset

2.5. Methods

We use two classification algorithms: the GNB and DT classifiers. These classification algorithms are one of the processes of machine learning algorithms. More specifically, these algorithms are a type of supervised learning. In machine learning, classification is a grouping of data where the data used has a label class. These algorithms are simple and easy to implement in real life.

An architecture overview is shown in Figure 3. First, we loaded the Iris dataset. This dataset has been split into two parts, namely training and testing data. Next, we classify this dataset by two classifying techniques which are the GNB and DT algorithms. Furthermore, we construct a confusion matrix for each algorithm, then we calculate some metrics which are recall, precision, accuracy, and F1-score. Finally, we determine the best classifier for this dataset. To illustrate ease of use, Figures 4 to 6 are the code used (Python 3.7.3) for classifying the Iris dataset and evaluating the performance of the algorithm.

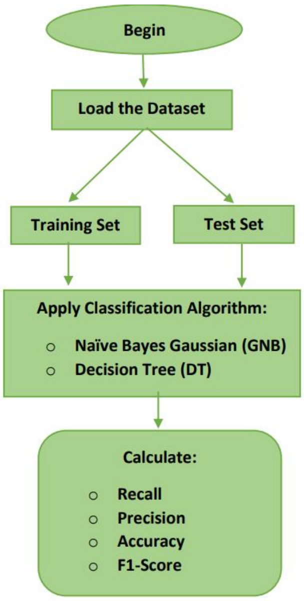
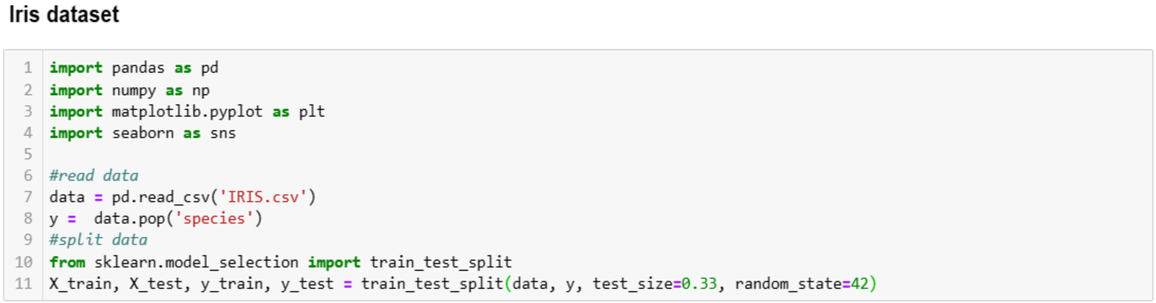


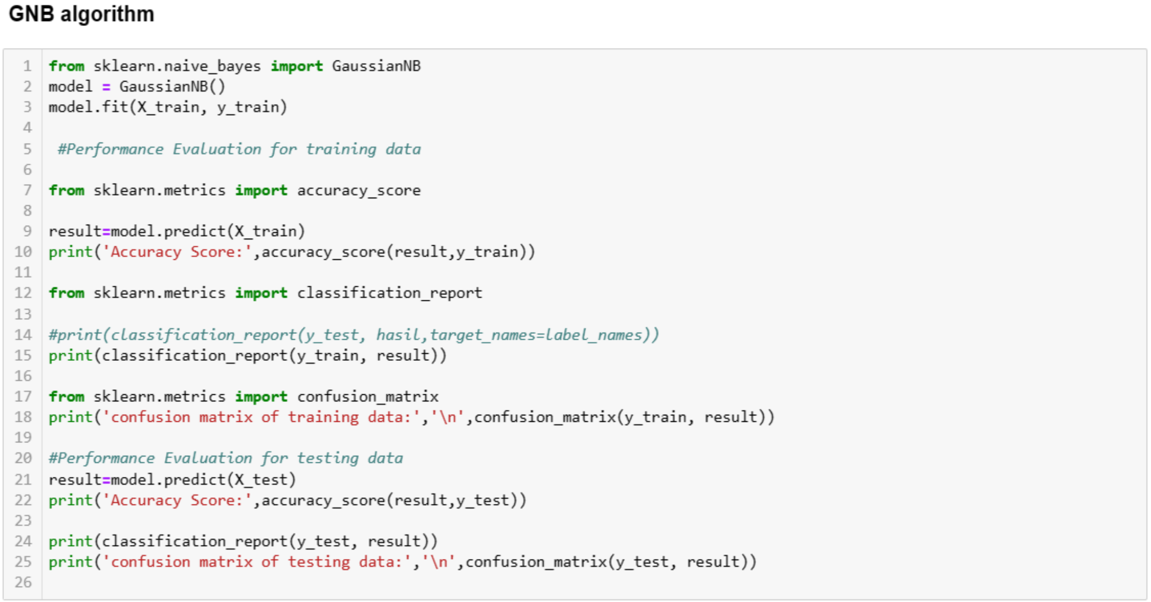
Figure 3. The proposed method flowchart



Iris dataset

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#read data
data = pd.read_csv('IRIS.csv')
y =  data.pop('species')
#split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data, y, test_size=0.33, random_state=42)
```

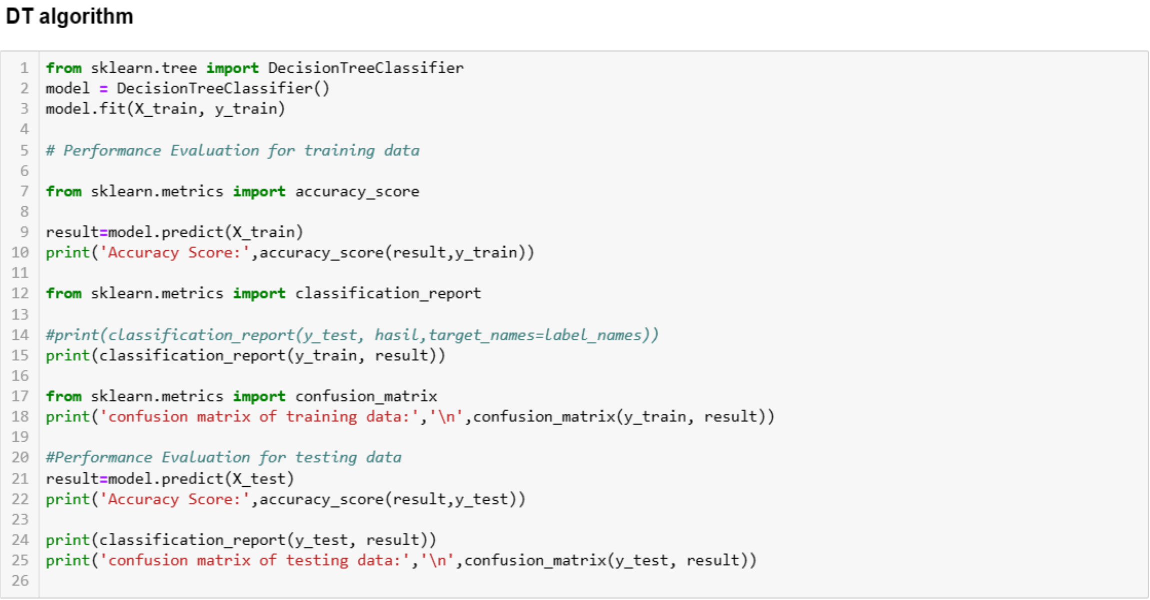
Figure 4. Python code for importing dataset



GNB algorithm

```python
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)

 #Performance Evaluation for training data

from sklearn.metrics import accuracy_score

result=model.predict(X_train)
print('Accuracy Score:',accuracy_score(result,y_train))

from sklearn.metrics import classification_report

#print(classification_report(y_test, hasil,target_names=label_names))
print(classification_report(y_train, result))

from sklearn.metrics import confusion_matrix
print('confusion matrix of training data:','\n',confusion_matrix(y_train, result))

#Performance Evaluation for testing data
result=model.predict(X_test)
print('Accuracy Score:',accuracy_score(result,y_test))

print(classification_report(y_test, result))
print('confusion matrix of testing data:','\n',confusion_matrix(y_test, result))

```

Figure 5. Python code for GNB algorithm



DT algorithm

```python
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Performance Evaluation for training data

from sklearn.metrics import accuracy_score

result=model.predict(X_train)
print('Accuracy Score:',accuracy_score(result,y_train))

from sklearn.metrics import classification_report

#print(classification_report(y_test, hasil,target_names=label_names))
print(classification_report(y_train, result))

from sklearn.metrics import confusion_matrix
print('confusion matrix of training data:','\n',confusion_matrix(y_train, result))

#Performance Evaluation for testing data
result=model.predict(X_test)
print('Accuracy Score:',accuracy_score(result,y_test))

print(classification_report(y_test, result))
print('confusion matrix of testing data:','\n',confusion_matrix(y_test, result))

```

Figure 6. Python code for DT algorithm

3. RESULTS AND DISCUSSION

In this section, we have compared the classification simulation results of the two algorithms, namely the GNB and DT algorithms on the Iris dataset where these simulations were carried out with the help of Python 3.7.3. The Iris dataset is divided into two parts, namely 67% of the training data (100 data points) and 33% of the testing data (50 data points). Next, the simulation results are evaluated for the performance of the algorithm using several metrics, namely recall, precision, F1-score, and accuracy. Before calculating these metrics, we calculated the confusion matrices of each of these algorithms. Below are tables of the confusion matrix and performance evaluation of the two algorithms for each training and testing data.

Figure 7 shows the confusion matrix of the GNB classifier from the training data. This confusion matrix explains that 31 data points are correctly classified in the Iris-setosa class, 32 data points are correctly classified in the Iris-versicolor class, and 32 data points are correctly classified in the Iris virginica class. Figure 8 shows the confusion matrix of the DT classifier from the training data. This confusion matrix explains that 31 data points are correctly classified in the Iris-setosa class, 35 data points are classified in the Iris-versicolor class correctly, and 34 data points are classified in the Iris virginica class correctly.

		Predicted Value		
		Setosa	Versicolor	Virginica
Actual Values	Setosa	31	0	0
	Versicolor	0	32	3
	Virginica	0	2	32

Figure 7. Confusion matrix using GNB on data training

		Predicted Value		
		Setosa	Versicolor	Virginica
Actual Values	Setosa	31	0	0
	Versicolor	0	35	0
	Virginica	0	0	34

Figure 8. Confusion matrix using DT on data training

Figure 9 shows the confusion matrix of the GNB classifier from the testing data. This confusion matrix explains that 19 data points are correctly classified in the Iris-setosa class, 14 data points are correctly classified in the Iris-versicolor class, and 15 data points are correctly classified in the Iris-virginica class. Figure 10 shows the confusion matrix of the DT classifier from the testing data. This confusion matrix explains that 19 data points are correctly classified in the Iris-setosa class, 15 data points are classified in the Iris-versicolor class correctly, and 15 data points are classified in the Iris-virginica class correctly.

		Predicted Value		
		Setosa	Versicolor	Virginica
Actual Values	Setosa	19	0	0
	Versicolor	0	14	1
	Virginica	0	1	15

Figure 9. Confusion matrix using GNB on data testing

		Predicted Value		
		Setosa	Versicolor	Virginica
Actual Values	Setosa	19	0	0
	Versicolor	0	15	0
	Virginica	0	1	15

Figure 10. Confusion matrix using DT on data testing

Tables 2 to 3 show the performance evaluation of both algorithms on the training data. The performance of the GNB classifier is very high on training data since the precision and recall values are in the range [0.91,1.00], the F1-score values are in the range [0.93,1.00], and also the accuracy value is 95%. Then, the performance of the DT algorithm is also perfect on the training data since the value of precision, recall, and F1-score values are 1.00, then also the accuracy value is 100%.

Table 2. Performance evaluation results of the GNB algorithm on training data

Training	Precision	Recall	F1-score	Accuracy
Setosa	1.00	1.00	1.00	0.95
Versicolor	0.94	0.91	0.93	
Virginica	0.91	0.94	0.93	

Table 3. Performance evaluation results of the DT algorithm on training data

Training	Precision	Recall	F1-score	Accuracy
Setosa	1.00	1.00	1.00	1.00
Versicolor	1.00	1.00	1.00	
Virginica	1.00	1.00	1.00	

Tables 4 to 5 show the performance evaluation of the two algorithms on testing data. The performance of the GNB algorithm is very high on data testing because the values for precision, recall, and F1-score are in the range [0.93, 1.00] and also the accuracy value is 96%. Then, the performance of the DT algorithm is also very high on the training data because the precision and recall values are in the range [0.94, 1.00], the F1-score values are in the range [0.97, 1.00] and also the accuracy value is 98%. Overall, the implementation method of DT classification outperforms training and test data.

Table 4. Performance evaluation results of the GNB algorithm on testing data

Testing	Precision	Recall	F1-score	Accuracy
Setosa	1.00	1.00	1.00	0.96
Versicolor	0.93	0.93	0.93	
Virginica	0.94	0.94	0.94	

Table 5. Performance evaluation results of the DT algorithm on testing data

Testing	Precision	Recall	F1-score	Accuracy
Setosa	1.00	1.00	1.00	0.98
Versicolor	0.94	1.00	0.97	
Virginica	1.00	0.94	0.97	

4. CONCLUSION

Based on the results of this research, we have concluded that the performance of the GNB and DT algorithms has very high performance for the classification of the Iris dataset where this dataset is balanced. This is indicated by all the metric values of recall, precision, and F1-score which are above 0.90, and also the accuracy metric values which are above or equal to 95%. Furthermore, the best algorithm performance of both algorithms is the DT classifier algorithm on this Iris dataset. We recommend further research to classify types of imbalanced datasets with machine learning techniques.

REFERENCES

[1] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica (Ljubljana)*, vol. 31, no. 3. 2007.

[2] F. J. Yang, "An implementation of naive Bayes classifier," *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, pp. 301–306, 2018, doi: 10.1109/CSCI46756.2018.00065.

[3] W. Zhu and H. Wu, "CTL model checking based on binary classification of machine learning," *International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 249–260, 2022, doi: 10.34028/iajit/19/2/12.

[4] M. Smith and F. Alvarez, "A machine learning research template for binary classification problems and Shapley values integration [Formula presented]," *Software Impacts*, vol. 8, 2021, doi: 10.1016/j.simpa.2021.100074.

[5] S. M. Sherwood, T. B. Smith, and R. S. W. Masters, "Decision reinvestment, pattern recall and decision making in rugby union," *Psychology of Sport and Exercise*, vol. 43, 2019, doi: 10.1016/j.psychsport.2019.03.002.

[6] J. Lee, J. Kang, S. Park, D. Jang, and J. Lee, "A multi-class classification model for technology evaluation," *Sustainability*, vol. 12, no. 15, Jul. 2020, doi: 10.3390/su12156153.

[7] N. Endut, W. M. A. F. W. Hamzah, I. Ismail, M. K. Yusof, Y. A. Baker, and H. Yusoff, "A systematic literature review on multi-label classification based on machine learning algorithms," *TEM Journal*, vol. 11, no. 2, pp. 658–666, 2022, doi: 10.18421/TEM112-20.

[8] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: a survey," *Information (Switzerland)*, vol. 10, no. 4. 2019, doi: 10.3390/info10040150.

[9] P. Kalia, Y. K. Dwivedi, and Á. Acevedo-Duque, "Cellulographics©: A novel smartphone user classification metrics," *Journal of Innovation & Knowledge*, vol. 7, no. 2, Apr. 2022, doi: 10.1016/j.jik.2022.100179.

[10] G. Shao, L. Tang, and H. Zhang, "Introducing image classification efficacies," *IEEE Access*, vol. 9, pp. 134809–134816, 2021, doi: 10.1109/ACCESS.2021.3116526.

[11] Y. Wu *et al.*, "Enhanced classification models for Iris dataset," in *Procedia Computer Science*, 2019, vol. 162, pp. 946–954, doi: 10.1016/j.procs.2019.12.072.

[12] K. Thirunavukkarasu, A. S. Singh, P. Rai, and S. Gupta, "Classification of IRIS dataset using classification based KNN Algorithm in supervised learning," *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018*, 2018, doi: 10.1109/CCAA.2018.8777643.

[13] M. Swain, "An approach for Iris plant classification using neural network," *International Journal on Soft Computing*, vol. 3, no. 1, pp. 79–89, Feb. 2012, doi: 10.5121/ijsc.2012.3107.

[14] T. M. Ghazal, M. A. Afifi, and D. Kalra, "Data mining and exploration: A comparison study among data mining techniques on Iris data set," *Talent Development & Excellence*, vol. 12, no. 1, pp. 3854–3861, 2020.

[15] M. V. Anand, B. KiranBala, S. R. Srividhya, K. C., M. Younus, and M. H. Rahman, "Gaussian naïve Bayes algorithm: A reliable technique involved in the assortment of the segregation in cancer," *Mobile Information Systems*, vol. 2022, pp. 1–7, Jun. 2022, doi: 10.1155/2022/2436946.

[16] E. K. Ampomah, G. Nyame, Z. Qin, P. C. Addo, E. O. Gyamfi, and M. Gyan, "Stock market prediction with Gaussian naïve Bayes machine learning algorithm," *Informatica*, vol. 45, no. 2, pp. 243–256, Jun. 2021, doi: 10.31449/inf.v45i2.3407.

[17] I. Rish, "An empirical study of the naive Bayes classifier," *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41–46, 2001.

[18] F. Itoo, Meenakshi, and S. Singh, "Comparison and analysis of logistic regression, naïve Bayes and KNN machine learning algorithms for credit card fraud detection," *International Journal of Information Technology*, vol. 13, no. 4, pp. 1503–1511, Aug. 2021, doi: 10.1007/s41870-020-00430-y.

[19] S. C. Hsu, I. C. Chen, and C. L. Huang, "Image classification using naive Bayes classifier with pairwise local observations," *Journal of Information Science and Engineering*, vol. 33, no. 5, pp. 1177–1193, 2017, doi: 10.6688/JISE.2017.33.5.5.

[20] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[21] C. El Morr, M. Jammal, H. Ali-Hassan, and W. El-Hallak, "Decision Trees," in *International Series in Operations Research and Management Science*, 2022, pp. 251–278.

[22] T. A. Assegie and P. S. Nair, "Handwritten digits recognition with decision tree classification: A machine learning approach," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 4446–4451, Oct. 2019, doi: 10.11591/ijece.v9i5.pp4446-4451.

[23] H. Ismaeil, S. Kholeif, and M. A. Abdel-Fattah, "Using decision tree classification model to predict payment type in NYC yellow Taxi," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, 2022, doi: 10.14569/IJACSA.2022.0130330.

[24] D. Berrar, "Bayes' theorem and naive Bayes classifier," in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1–3, 2018.

[25] D. Joshi, A. Mishra, and S. Anand, "A naïve Gaussian Bayes classifier for detection of mental activity in gait signature," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 15, no. 4, pp. 411–416, 2012, doi: 10.1080/10255842.2010.539562.

[26] S. Jayachitra and A. Prasanth, "Multi-feature analysis for automated brain stroke classification using weighted Gaussian naïve Bayes classifier," *Journal of Circuits, Systems and Computers*, vol. 30, no. 10, Aug. 2021, doi: 10.1142/S0218126621501784.

[27] S. A. Sushma and K. K. TG, "Comparative study of naive Bayes, Gaussian naive Bayes classifier and decision tree algorithms for prediction of heart diseases," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 3, pp. 475–486, Mar. 2021, doi: 10.22214/ijraset.2021.33228.

[28] I. Sulistiani, W. Wulandari, F. D. Astuti, and Widodo, "Breast cancer prediction using random forest and Gaussian naïve Bayes algorithms," in *2022 1st International Conference on Information System & Information Technology (ICISIT)*, Jul. 2022, pp. 170–175, doi: 10.1109/ICISIT54091.2022.9872808.

[29] A. H. Jahromi and M. Taheri, "A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features," in *19th CSI International Symposium on Artificial Intelligence and Signal Processing, AISP 2017*, 2017, vol. 2018-January, pp. 209–212, doi: 10.1109/AISP.2017.8324083.

[30] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991, doi: 10.1109/21.97458.

[31] A. Trabelsi, Z. Elouedi, and E. Lefevre, "Decision tree classifiers for evidential attribute values and class labels," *Fuzzy Sets and Systems*, vol. 366, pp. 46–62, Jul. 2019, doi: 10.1016/j.fss.2018.11.006.

[32] D. Lavanya and K. U. Rani, "Performance evaluation of decision tree classifiers on medical datasets," *International Journal of Computer Applications*, vol. 26, no. 4, pp. 1–4, 2011, doi: 10.5120/3095-4247.

[33] M. Barbareschi, S. Barone, and N. Mazzocca, "Advancing synthesis of decision tree-based multiple classifier systems: An approximate computing case study," *Knowledge and Information Systems*, vol. 63, no. 6, pp. 1577–1596, 2021, doi: 10.1007/s10115-021-01565-5.

[34] A. S. Bhatt, "Comparative analysis of attribute selection measures used for attribute selection in decision tree induction," in *2012 International Conference on Radar, Communication and Computing, ICRCC 2012*, 2012, pp. 230–234, doi: 10.1109/ICRCC.2012.6450584.

[35] C. S. Lee, P. Y. S. Cheang, and M. Moslehpour "Predictive analytics in business analytics: decision tree," *Advances in Decision Sciences*, vol. 26, no. 1, pp. 1–30, 2022, doi: 10.47654/v26y2022i1p1-30.

[36] D. Chicco, N. Tötsch, and G. Jurman, "The Matthews correlation coefficient (Mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Mining*, vol. 14, pp. 1–22, 2021, doi: 10.1186/s13040-021-00244-z.

BIOGRAPHIES OF AUTHORS

Yasi Dani received her bachelor's and master's degrees in mathematics from Institut Teknologi Bandung, Bandung, Indonesia, in 2010 and 2015 respectively. Now, she joins as a lecturer in Computational Mathematics at University of Bina Nusantara and is currently studying for a PhD degree in Mathematics at Institut Teknologi Bandung for Industrial and Financial Mathematics Research Group. Her research interests include outlier or anomaly detection, machine learning, applied mathematics and statistics. She can be contacted at email: yasi.dani@binus.ac.id.

Maria Artanta Ginting is a lecturer in computational mathematics at Computer Science Department, Bina Nusantara University. Her research focuses on applied mathematics, numerical modeling and simulation, particularly in water wave modeling. She received her doctoral degree from Institut Teknologi Bandung, where she studied in Industrial and Financial Mathematics Research Group. Her email address is maria.ginting@binus.ac.id.