JURNAL MATEMATIKA, STATISTIKA DAN KOMPUTASI e-ISSN: 2614-8811
Published by Departement of Mathematics, Hasanuddin University, Indonesia
https://journal.unhas.ac.id/index.php/jmsk/index
**Vol. 20, No. 3, May 2024, pp. 623-638**
**DOI: 10.20956/j.v20i3.32970**

# Performance Evaluation of Classification Methods on Big Data: Decision Trees, Naive Bayes, K-Nearest Neighbors, and Support Vector Machines

**Justin Eduardo Simarmata[1]*, Gerhard-Wilhelm Weber[2], Debora Chrisinta[3]**

[1]*Faculty of Teacher Training & Education, University of Timor, East Nusa Tenggara, Indonesia*
[2]*Faculty of Engineering Management, Poznan University of Technology, PUT, Poznań, Poland*
[3]*Faculty of Agriculture, Science and Health, University of Timor, East Nusa Tenggara, Indonesia*
***Email**: [1*)] justinesimarmata@unimor.ac.id, [2)] gerhard.weber@put.poznan.pl*
*[3)] deborachrisinta@unimor.ac.id*

*Received: 31 January 2024, revised: 1 April 2024, accepted: 2 April 2024*

**Abstract**

Performance evaluation of classification methods on big data is becoming increasingly important in addressing the challenges of data analysis at scale. This study aims to conduct a comparative evaluation of the classification method, namely Decision Trees (DT), Naive Bayes (NB), k-Nearest Neighbors (KNN), and Support Vector Machines (SVM), in analysis on big data evaluated from data simulation and application of real data available in the Rstudio package, namely ISLR. The simulation data used consisted of 2 types of datasets generated based on predictor variables that were normally distributed with different averages and variants and response variables generated in classes adjusted to the characteristics of predictor variables with different proportions. Real data are taken from two types of numeric variables and predictor variables available in the package. The number of sample sizes to be evaluated in each method is n = 500, n = 1000 and n = 5000. In real data, sample division is done randomly to maintain data representativeness. At the evaluation stage, the performance of the method is measured using accuracy metrics. The results of the evaluation of the simulation of Dataset 1 show that the methods that have an influence on the quality of the classification produced if applied to Big Data are the DT and KNN methods. However, in Dataset 2 there is a change in the results of the DT method, because of the influence on the number of classes and the proportion of class distribution in the data. The results obtained from data simulation, proven by applying to real data by showing that similar methods provide a quality influence if applied to Big Data, while the NB and SVM methods do not show a consistent influence when applied to Big Data. The results of observations in this study show that the DT and KNN methods have several advantages that make them suitable for application to Big Data.

**Keywords:** Performance Evaluation, Classification Method, Big Data.

# 1. INTRODUCTION

Classification methods are an important part of data science and machine learning because they can be used to understand patterns in data, make predictions, and make decisions based on existing data [7]. Some of the reasons why classification methods are important include decision making, prediction, data analysis, object classification, and allow automation of manual and repetitive tasks, thus saving time and human resources [2]. Some commonly used classification methods include Decision Trees, Naive Bayes, k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Neural Networks. Each method has advantages and disadvantages, and the choice of the right method depends on the nature of the data, the complexity of the problem, and the purpose of the analysis or prediction to be achieved. In today's conditions of technological advancement and rapid data growth, the development of efficient and accurate classification methods is becoming increasingly important to face the challenges of increasingly complex and large data [14], [8], [4].

Efforts to measure and understand the extent to which algorithms or classification methods used are successful in handling large and complex volumes of data are important. In the era of big data data, there are many applications that collect, store, and analyze huge amounts of data in real time. In this context, it is important to have efficient and accurate classification algorithms to retrieve valuable information from this large and diverse data. Several reasons why performance evaluation of classification methods in Big Data simulation is important such as the scalability aspect is that in the Big Data simulation process, the dataset size can reach several petabytes or more. Therefore, classification methods must be efficiently applied to data on a large scale, while still providing accurate and consistent results [10]. There are a wide variety of classification algorithms available, each with its own advantages and disadvantages. Performance evaluation allows us to compare several classification methods and choose the most appropriate one for a particular dataset and needs. In addition, resources such as memory, processor speed, and processing time can be constraints [17], [16]. Performance evaluation can help identify limitations and possible problems in implementing certain classification methods in Big Data scenarios [5].

In big data era, the reliability of classification results becomes very important. Performance evaluation helps ensure that the results provided by the classification algorithm are reliable and have a high degree of accuracy [13]. Performance evaluation can also help in the development of better classification methods and adjustments to Big Data conditions that change over time. It is important to note that Big Data simulation allows us to test classification methods on well-controlled scenarios and data that can be used for research purposes without disrupting the actual production environment [12]. Therefore, performance evaluation in Big Data simulation can be the first step before implementing classification methods in a real Big Data environment.

This study will evaluate the performance of Decision Trees, Naive Bayes, k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) classification methods in Big Data simulations. The dataset scenario used is based on the sample size in Big Data. However, the minimum sample size for big data can vary depending on specific analysis needs, data complexity, and desired level of accuracy. The selected sample must represent the entire dataset representatively. In some cases, a small sample of Big Data may be sufficient to reflect key trends and patterns in the data [15]. Such as the research conducted by Wang and Kim in evaluating sampling techniques for Big Data analysis using sample sizes n = 500 and n = 1000 [11]. Considering in this study only want to understand general patterns or major trends, then a smaller sample size is sufficient. In cases where hypothesis testing does not involve testing, if it involves testing specific hypotheses or analyzing smaller groups of data, a larger sample size may be required. The sample sizes used in this study were 500, 1000 and 5000. The simulation process

involves two predictor variables that are randomly generated and normally distributed and two types of response variables consisting of 2 and 3 labels.

Performance evaluation of classification methods in Big Data simulations involves accuracy metrics or known as accuracy methods in predicting data patterns. Moreover, the calculation of accuracy values helps in understanding as well as measuring how well the classification method can classify the data correctly.

## 2. MATERIALS AND METHODS

### 2.1 Naive Bayes

Naïve Bayes is an algorithm in classification analysis that uses the Bayes probability rule [3]. The Naïve Bayes formula for numerical data is:

$$P(Y|X) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}} \quad 2.1$$

where $\sigma$ is the standard deviation of the observed variable, $\mu$ is the average value of observed variables, $x_i$ is object that is in the observed variable, $P(Y|X)$ is the data probability of variable $X$ on variable $Y$. The following is the Naïve Bayes calculation formula for categorical data [1]:

$$P(I|C) = \frac{P(I)|P(C|I)}{P(C)} \quad 2.2$$

where $I$ is the large number of opportunities for the appearance of objects for a particular category and $C|I$ is the number of certain categories that belong to a certain class. The steps in classifying using the Naive Bayes method are as follows:

1. Data Collection: Collect training data that contains examples of data with features want to use for classification. Each sample data must have a known class label.
2. Data Separation: Divide training data into two subsets: training data and testing data. Training data is used to train the classification model, while test data is used to test model performance.
3. Data Preprocessing: Preprocessing data such as removing missing values, normalizing, or transforming data if needed.
4. Calculating Class Probability: Calculate the probability of each class present in the practice data. This probability describes how often a particular class appears in the exercise data.
5. Calculating Feature Probability: Calculate the probability of each feature based on its class. It involves calculating how often feature values appear in data instances with a particular class.
6. Calculating Combined Probability: Combine class probability and feature probability to calculate compound probability, reflecting how likely it is that an example of data belongs to a particular class based on the value of the feature it has.
7. Class Prediction: Use combined probability to predict classes from examples of test data. The class with the highest combined probability becomes the class prediction for each sample data.
8. Model Evaluation: Evaluate the performance of a classification model by comparing class predictions with actual class labels in test data. The evaluation metric used in this study is accuracy.
9. If needed, it can perform model parameter adjustments to improve classification performance. This can involve selecting better features, or smoothing out probabilities to avoid absolute zero.
10. Model Usage: Once the model is declared good and has satisfactory performance, it can be used to classify new data.

## 2.2 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is one of the algorithms used in machine learning for classification and regression. This algorithm functions on the principle that similar data tend to be in close neighbors within the feature space [9]. Here are the main steps of the KNN algorithm:

1. Data Preparation: Prepare data to be used for training and testing. This data must contain features or attributes that define each instance and the target class or value want to predict.
2. Parameter K Selection: Menen determine the value of parameter K, i.e., the number of nearest neighbors to be used for prediction. The selection of K values usually involves experimentation and cross-validation to find the optimal K value.
3. Define Distance Metrics: Select the appropriate distance metric to measure how close or different two sample data are. A commonly used distance metric is
4. Euclidean distance with the following equation:

$$D_{ij} = |x_i - x_j| \tag{2.3}$$

where $x_i$ is the $i$-th data and $x_j$ is the $j$-th data.

5. Calculate Distance: For each sample data want to predict, calculate the distance to all sample data in the training dataset using the distance metric have selected.
6. Select K Nearest Neighbor: Specify K data examples with the smallest distance as the closest neighbors of the sample data want to predict. Typically, K sample data is selected based on the closest distance.
7. Prediction of Target Class or Value: For the classification task, use the majority of votes (modes) from the nearest neighboring class K as the prediction of the sample class data. That is, the class that appears most often among neighboring K's will be the prediction of its class. The estimation function used to predict classes is based on the following equation:

$$\hat{f}_k(x) = \frac{1}{k} W_{ki}(x) y_i \tag{2.4}$$

with $W_{ki}(x)$ is a sequence of weights defined as follows:

$$W_{ki}(x) = \begin{cases} 1 & x_i \in S_k \\ 0 & x_i \notin S_k \end{cases} \tag{2.5}$$

with $S_k$ is the closest data as much as k.

8. Model Evaluation: To measure the performance of the accuracy of the method in classifying observation objects.
9. Model Usage: Once the KNN model has been well studied and assessed, it can use it to predict target classes or values for new data that has never been seen before.

## 2.3 Decision Tree

Decision Tree is a machine learning algorithm used for data classification. This algorithm builds a decision tree based on a set of rules and decisions that are based on data features [6]. Here are the general steps for the Decision Tree algorithm:

1. Data Preparation: Prepare the data that will be used to build the decision tree. This data must contain the features or attributes that define each sample data and the target class or value want to predict.
2. Feature Selection: Determine the most informative or relevant feature as the root of the decision tree. This process is done using the following Gini impurity equation metrics:

$$Gini(t) = 1 - \sum_{i=1}^{C} \big(p(i|t)\big)^2 \tag{2.6}$$

where $C$ is the number of classes of data and $p(i|t)$ is the proportion of sample data from the class-$i$ at $t$-node.

3. Tree Formation: Starting from the root of the tree, divide the dataset into subsets based on the feature values selected in the previous step. Each subset will be a branch in the tree. Continue

this step recursively for each tree branch until it reaches a stop state or leaf node that expresses a target class or value.

4. Stop Condition: Set a stop condition that specifies when the tree formation process should stop. These conditions can be the minimum number of data instances in the node, the maximum depth of the tree, or other criteria that avoid overfitting.
5. Prediction: Once the decision tree is built, the data that is not yet visible can be predicted by following the path from the tree root until it reaches the leaf node corresponding to the data features. The majority class of the sample data on that leaf node will be the prediction class for the new data.
6. Pruning: Pruning the decision tree to reduce model complexity and avoid overfitting. Pruning removes insignificant tree branches to increase generalization on data not yet seen.
7. Evaluation: A decision tree evaluation is performed to measure model performance on data that is not yet visible. This can involve the use of accuracy metrics.

## 2.4 Support Vector Machines

Support Vector Machines (SVM) are one of the machine learning algorithms used for classification tasks that aim to find the optimal hyperplane (line, plane, or hyperfield) to separate data into two or more different classes. SVM is often used for classification tasks where data are separated by lines or hyperfields that are maximally spaced from different data class instances. Here are the general steps for classification using SVM [6]:

1. Data Preparation: Prepares training data that contains the features (attributes) of each data instance and the corresponding classes (labels). Make sure the data has been split into a training dataset and a validation or test dataset.
2. Kernel Selection: Select the appropriate SVM kernel type for the classification task. The kernel is a function used to transform data into a higher feature space, where data can be more linearly or non-linearly separated. The kernel function used in this study is a linear kernel with the following equation:

$$K(x, y) = x \cdot y \tag{2.7}$$

where $K(x, y)$ is the kernel value between the two feature vectors $x$ and $y$, as well $x \cdot y$ as is the dot product between both the feature vector $x$ and $y$as well $x \cdot y$ as is the dot product between both the feature vector $x$ and $y$.

3. Model Training: Train the SVM model by using the training dataset and kernel that have been selected. The training goal is to find a hyperplane (line or plane) that maximizes the margin (distance) between different classes and minimizes classification errors.
4. Parameter Tuning: Determine the relevant C parameters for SVM models. The parameter tuning process usually involves cross-validation to find the optimal parameter value. Mathematically, parameter C can be added to the destination function of SVM. If we symbolize C as a regulatory parameter, and $L$ as a function that calculates classification errors, then the destination function can be written as:

$$F = \frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{n} L\left(y_i(w \cdot x_i + b)\right) \tag{2.8}$$

where $w$ is SVM weight vector, $b$ is biased, $x_i$ is the $i$-th data, $y_i$ is the label (class) for the $i$-th sample data, $x_i$ is sample of the $i$-th data and $L$ is a function that calculates misclassification errors.

5. Model Validation: Use validation or test datasets to evaluate the performance of SVM models. It involves calculating evaluation metrics namely accuracy, to measure the extent to which the model can correctly predict classes.

6. Prediction: Once the SVM model has been properly trained and validated, use this model to predict classes from data that has not been seen before.
7. Advanced Tuning and Evaluation: If the model performance is not satisfactory, it can perform further adjustments to the parameters or try another kernel to improve classification performance.
8. Model Applicability: Once an SVM model has been deemed good and accurate, it can apply it to real-world data to predict classes from unknown data.

## 2.5 Simulation Study

Data simulation is a useful technique for generating datasets with certain characteristics, so that with the benefit of being able to test and understand how classification methods behave in certain situations [9]. Here are some data simulation scenarios used for classification methods in this study:

1. Overlapping Classes: The scenario is based on two data classes overlapping each other. This will create challenges in properly separating the two classes.
2. Data with Variance: Add variance into the data to make it more realistic. Noise can describe uncertainty or variation in real-world data. This will make it possible to test the robustness of the classification method to interference or imperfect data.
3. Unbalanced Classes: Create scenarios where classes have an unbalanced distribution. For example, one class has a significantly larger number of sample data than another class. This will make it possible to test classification methods in the face of unbalanced class problems.
4. Multiclass Data: Simulate datasets with more than two classes. This makes it possible to test classification methods that can handle multiclass classification tasks.

The following types of datasets are required in evaluating classification methods:

**Table 2.1.** Data Simulation Scenarios

| Skenario | Variable | Numbers of Data | Class Distribution |
|---|---|---|---|
| Dataset 1 | $X1 \sim \begin{pmatrix} 10; 5 \\ 20; 10 \end{pmatrix}$<br>$X2 \sim \begin{pmatrix} 100; 10 \\ 200; 50 \end{pmatrix}$<br>$Y \sim \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ | $n = \{500,1000,5000\}$ | $p_{Y_1} = \left\{\frac{1}{2}, \frac{1}{2}\right\}$<br>$p_{Y_2} = \left\{\frac{1}{4}, \frac{3}{4}\right\}$<br>$p_{Y_3} = \left\{\frac{1}{3}, \frac{2}{3}\right\}$ |
| Dataset 2 | $X1 \sim \begin{pmatrix} 0.7; 10.2 \\ 18.3; 22.6 \\ 56.7; 30.2 \end{pmatrix}$<br>$X2 \sim \begin{pmatrix} 117; 38.3 \\ 382; 101.4 \\ 752; 67.7 \end{pmatrix}$<br>$Y \sim \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$ | $n = \{500,1000,5000\}$ | $p_{Y_1} = \left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$<br>$p_{Y_2} = \left\{\frac{1}{4}, \frac{2}{4}, \frac{1}{4}\right\}$<br>$p_{Y_3} = \left\{\frac{1}{6}, \frac{2}{6}, \frac{3}{6}\right\}$ |

## 2.6 Performance Criteria

A confusion matrix is a matrix in the form of a 2×2 contingency table that presents the actual data class and the prediction class. This matrix can be used as a method to calculate the performance of classification algorithms in data mining. The accuracy value can be obtained from

the confusion matrix. Accuracy is the percentage proportion of the correct prediction classification results. The higher the accuracy value, the better the model is at correctly classifying. This metric measures how precise the model is in predicting classes from data that has not been seen before. The formula for determining the accuracy value based on Table 2.2:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad 2.9$$

**Table 2.2.** *Confusion Matrix*

| **Actually** | **Predictions** | |
|---|---|---|
| | **Positive** | **Negative** |
| **Positive** | True Positive | False Negative |
| **Negative** | False Positive | True Negative |

## 3. RESULTS AND DISCUSSION

Data simulation for classification methods is the process of creating synthetic datasets with certain characteristics to test the performance and effectiveness of various classification algorithms. In this simulation, we can control the distribution of data, the number of classes, class patterns, and the complexity of the data according to the needs of the experiment.

The process of data simulation can better understand the characteristics and performance of classification algorithms in various situations, and this helps in making better decisions in choosing the right classification method for a particular problem. Data simulations that have been built will be applied to Big Data. The main purpose of simulating classification data on big data is to test the performance of classification algorithms on a large scale and see how they behave when faced with large volumes of data. Simulating classification data on big data can provide valuable insight into how well classification algorithms perform at scale, and aid in the development and customization of algorithms to address big data challenges.

Visualization of classification data in Big Data involves graphical representation of very large datasets with complex features and classes. It aims to provide clear and easy-to-understand insights into class distribution, classification patterns, and overall data characteristics. This visualization helps to understand large and complex data in a more intuitive way, so as to identify patterns, anomalies, and trends that may not be visible through numerical analysis alone.

This study presents an initial data visualization using a Scatter Plot. Scatter plots are visualizations commonly used to plot two features on the x-axis and y-axis and represent classes with different colors or symbols. Scatter plots can help identify classification patterns and see if classes can be clearly separated or not.

Classification data analysis in big data is a powerful tool for understanding data structures, but it needs to be balanced with statistical analysis and appropriate classification methods to gain a more comprehensive insight into the data. The following is given in Figure 3.1 which is a visualization of the initial data on Dataset 1:

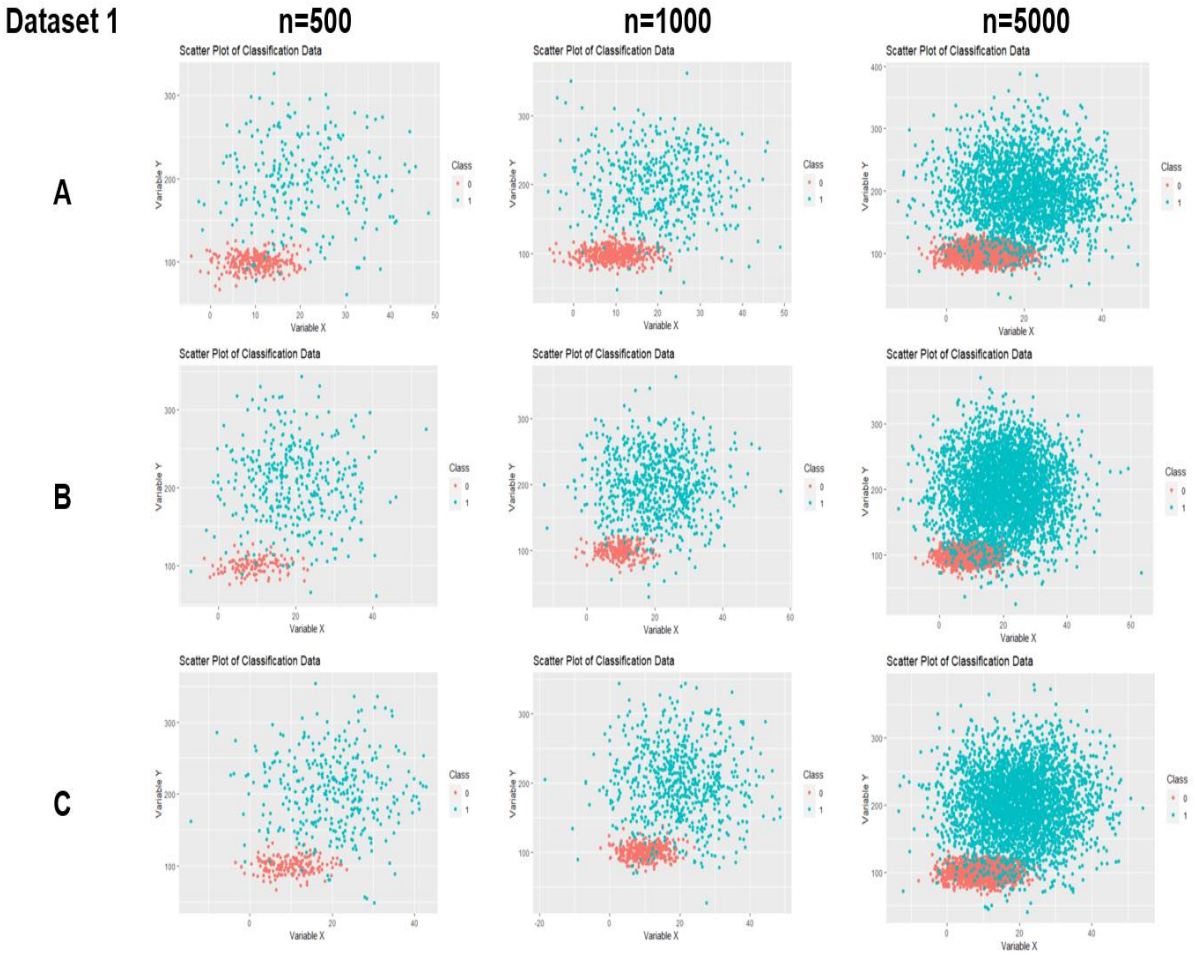


**Figure 3.1.** Compartment diagram

In general, there is class overlap in data distribution. Class 0 data shows a smaller value than class 1. The influence of data variations in class distribution can be a consideration in evaluating the performance of classification methods. In addition, the influence of the amount of data also affects the performance of the method in carrying out the accuracy of classification. The notations A, B and C indicate the type of data generated based on the distribution of class proportions. The simulation was carried out with 100 repeats, with the results of the distribution of accuracy values of each method given in Figure 3.2 below:

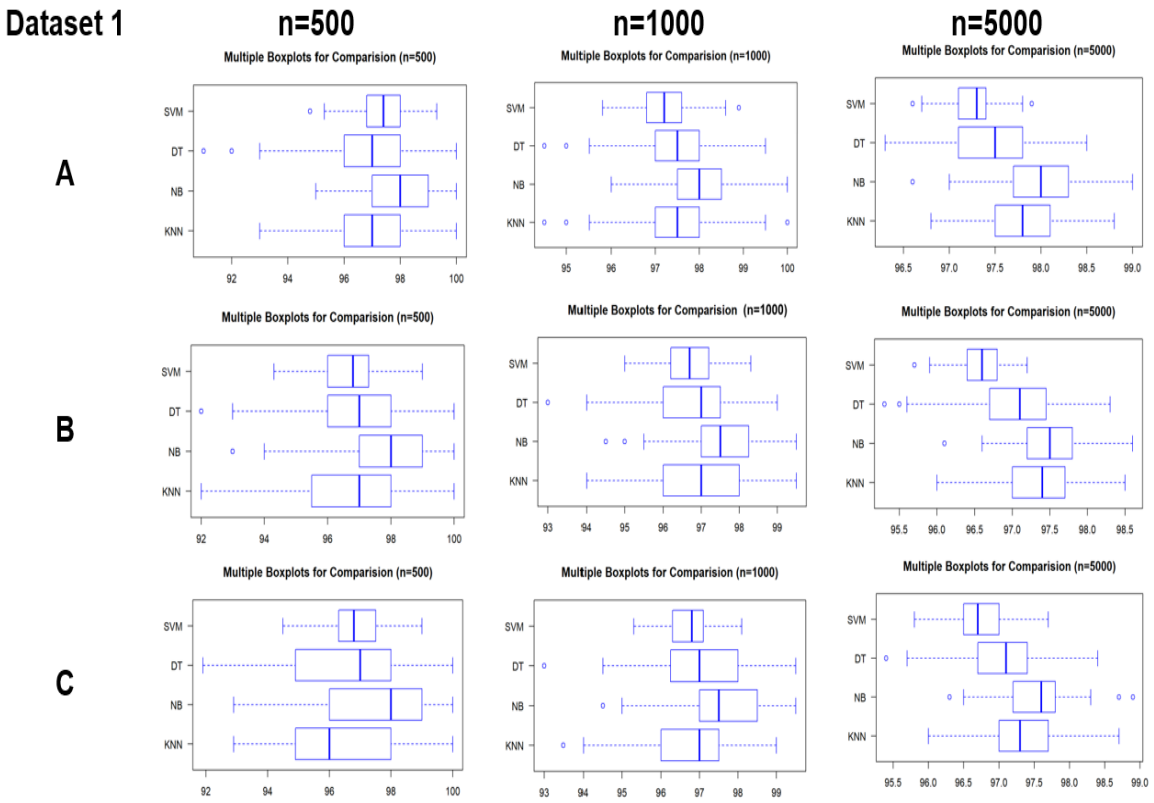


**Figure 3.2.** Distribution of Accuracy Values on The Performance of Classification Methods

When viewed in Figure 3.2, the SVM method tends to decrease in accuracy value when applied to a larger amount of data. DT produces accuracy values that tend to be stable both on the larger amount of data and on different proportions of class distribution in the data. NB shows the greatest accuracy value among all other classification methods. However, the influence of differences in class proportions that provide changes in the distribution of accuracy values is seen in the proportion of classes $\left\{\frac{1}{3}, \frac{2}{3}\right\}$. This shows that the larger the data used, the more accurate it will produce an accuracy value that does not have a large variation. Similarly, this same thing is found in the KNN method.

The results of the accuracy value obtained in 100 repetitions, in order to get a clearer conclusion on the good performance of the classification method, the average value of the overall accuracy value obtained by each method is calculated in Figure 3.3 below:

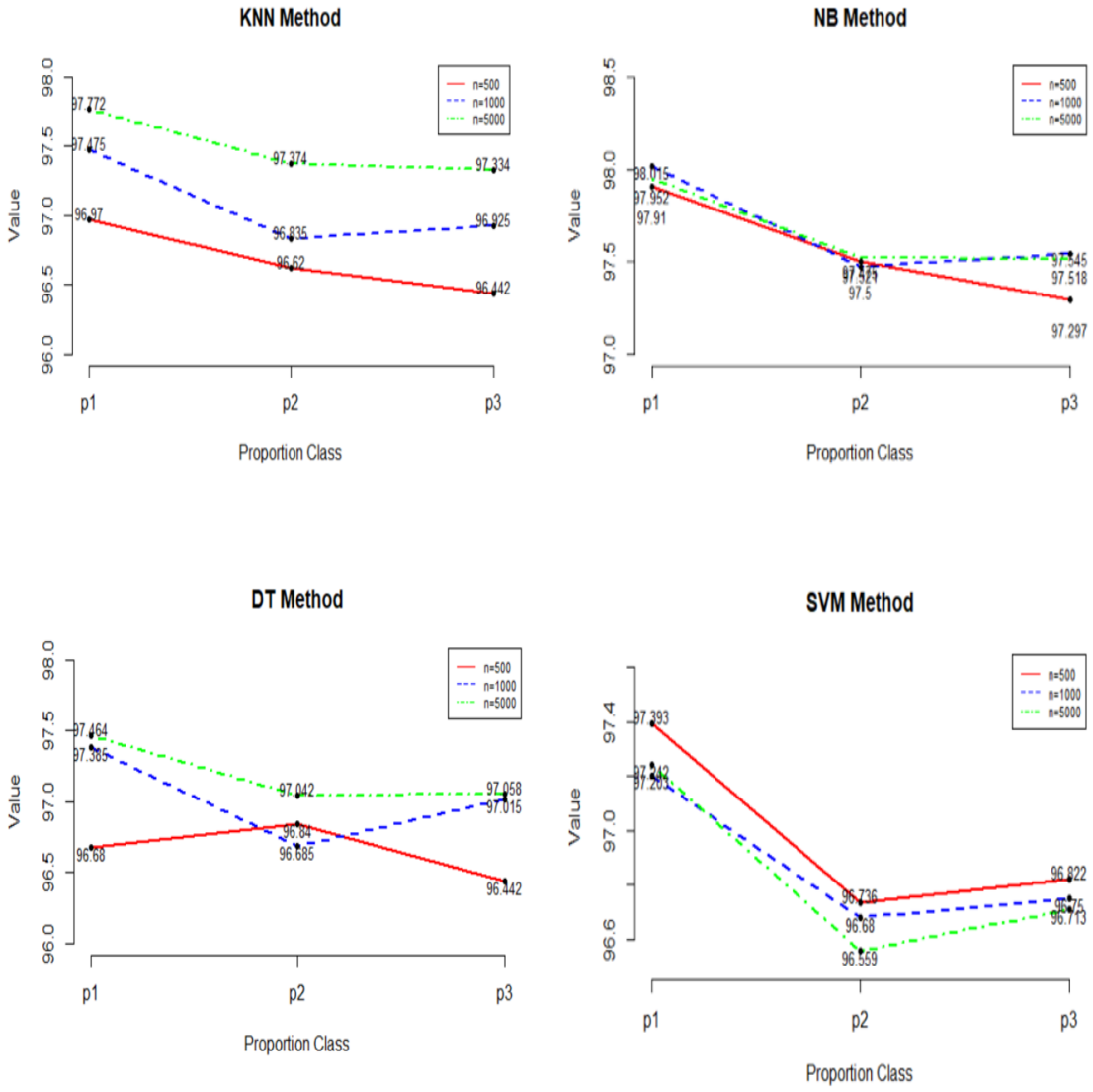


**Figure 3.3.** Performance of Classification Method Based on Accuracy Value

The classification methods that provide the best performance when applied to Big Data are DT and KNN. There is a green line that shows the highest performance in the data of 5000, with the distribution of class proportions $\left\{\frac{1}{2};\frac{1}{2}\right\}$ giving the highest accuracy value. This shows that the method gives the best classification result if the class distribution has the same proportions. SVM methods tend to have decreased performance when applied to Big Data. However, the NB method provides almost the same performance on the data type. This indicates that the NB method does not have a significant effect if applied to small or large samples.

Next, to prove the findings in dataset 1 can be trusted, a comparison is needed in the simulation scenario dataset 2. The following is given in Figure 3.4, namely the initial data visualization on Dataset type 2:

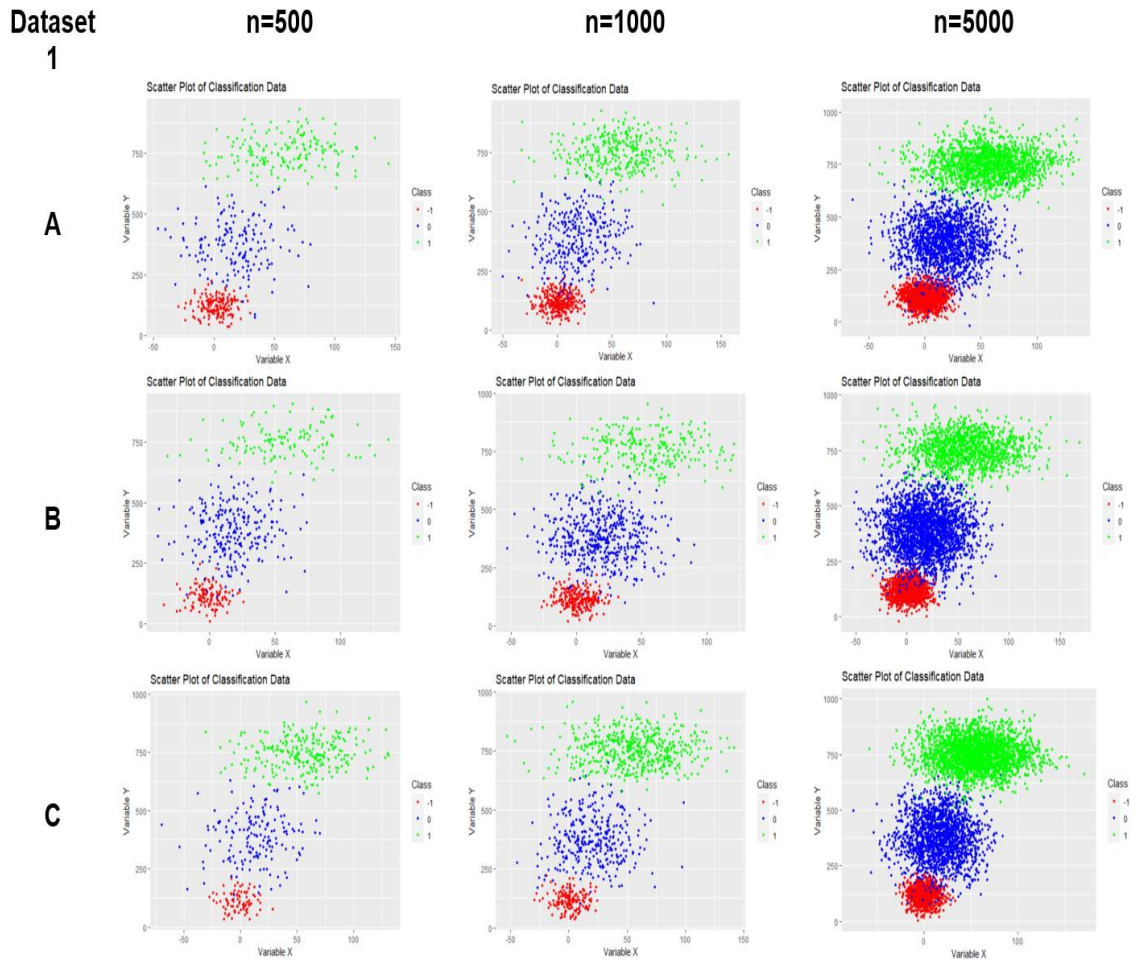


**Figure 3.4.** Visualization Dataset 2

Dataset type 2 brings up data that is divided into classes and has a considerable variance influence. In addition, data is generated from 3 types of classes with the proportion of classes, some are the same and some are different. Similarly, this type of data also has the nature of overlapping data in different classes. Even if look at Figure 3.4, the percentage of data that occurs overlaps slightly. But what we want to show in this type of data is the influence of variance. Most real data always have variance in data collection.

In the context of evaluating classification performance with simulated data, the influence of variance can play an important role in evaluating the quality of the classification model built. Variance refers to the extent to which different data in a simulated dataset can affect the results of the same classification model. When performing simulations, sometimes natural or chance variations in the simulation data can affect the performance of the classification model. Different data in simulations can produce varying results when applied to the same model. This can be challenging, as it wants to ensure that the classification model built has stable and consistent performance against these variations.

JURNAL MATEMATIKA, STATISTIKA DAN KOMPUTASI
Justin Eduardo Simarmata, Gerhard-Wilhelm Weber, Debora Chrisinta

There is a different type of data from Dataset 1 so that the performance of the classification method performance in Dataset 2 will be shown. The results of the distribution of accuracy values of each method are given in Figure 3.5 below:

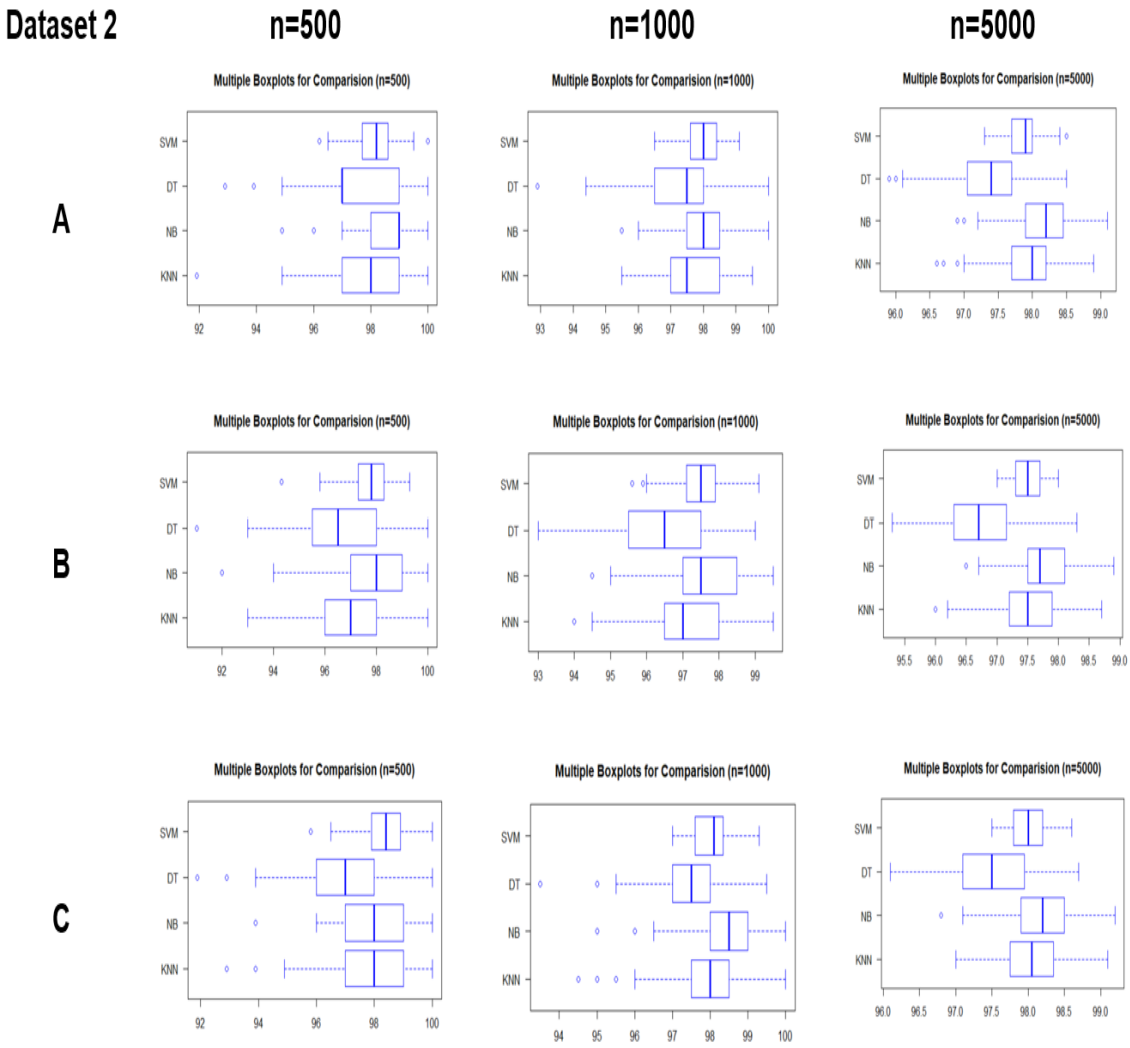


**Figure 3.5.** Distribution of Accuracy Values from Dataset 2

In Figure 3.5, the DT method tends to provide a smaller distribution of accuracy values than other methods. NB still provides a distribution of values that tend to be the highest. When viewed based on the difference in the distribution of class proportions, it appears that there is no significant difference in the performance of the method classification because it provides almost the same distribution pattern to the number of data samples.

Next, to get a clearer conclusion on the good performance of the classification method, the average value of the overall accuracy value obtained by each method for the components of Dataset 2 is calculated which is given in Figure 3.6 below

JURNAL MATEMATIKA, STATISTIKA DAN KOMPUTASI
Justin Eduardo Simarmata, Gerhard-Wilhelm Weber, Debora Chrisinta

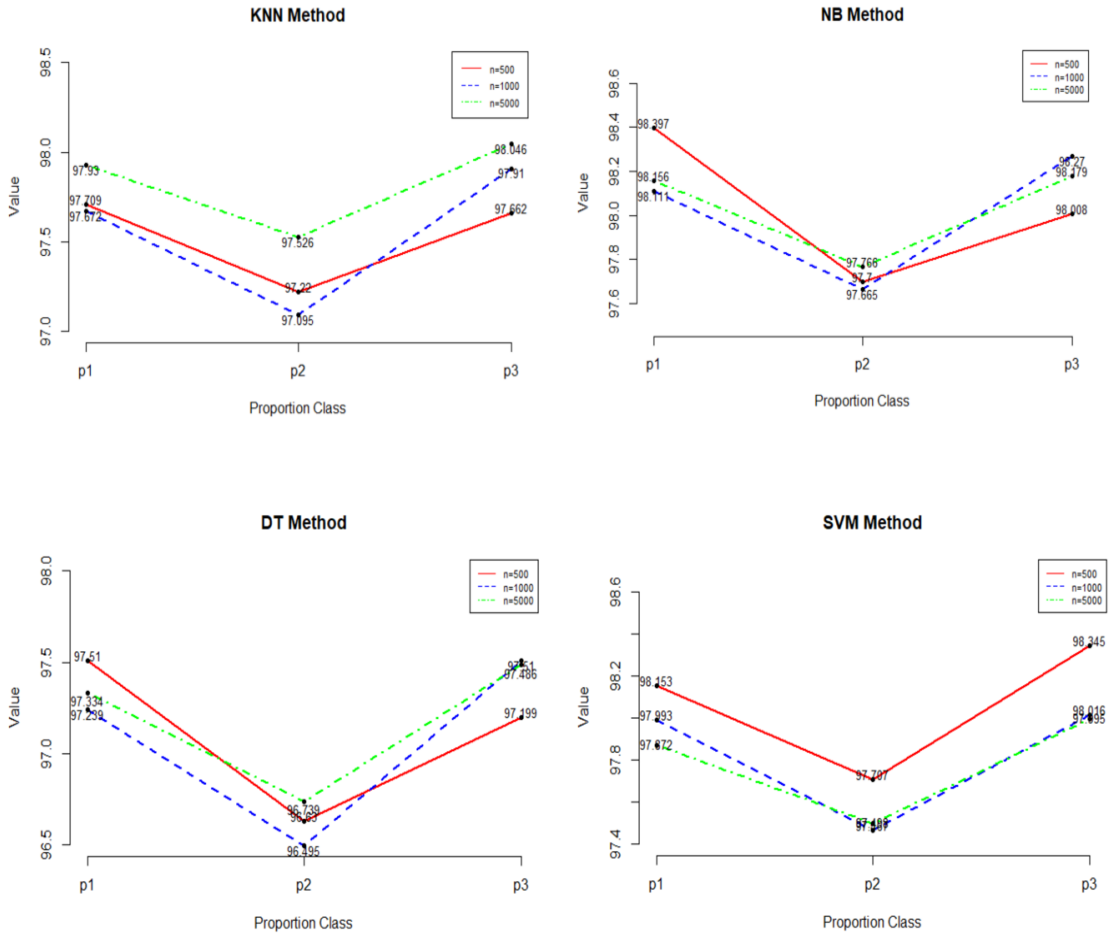


**Figure 3.6.** Performance of Classification Method Based on Accuracy Value from Dataset 2

In Dataset type 2, the classification method that provides the best performance when applied to Big Data is also generated by KNN. However, the DT method does not show similar results because the proportions of the first and last type classes provide smaller accuracy values compared to sample sizes n = 500 and n = 1000. Although at n = 1000 does not show a significant difference in value.

The NB method still shows almost the same performance quality as Dataset 1. Similarly, the SVM method provides better performance quality when applied to small data.

Overall, what influences the quality of classification when applied to Big Data is the DT method. However, when evaluated against the quality of the best classification results by ignoring the number of samples used, the NB method is given.

Therefore, for further investigation, it is necessary to pay attention to the good results of the methods that have been obtained when applied to real data. This study will provide evaluation results if applied to real data available in the RStudio package, namely "ISLR" with the "Default" dataset. The variables used only involve numerical variables according to the characteristics used in the simulation data. The application is carried out by sampling data as much as n = 500, n = 1000 and n = 5000. The accuracy calculation results are given in Figure 3.7 below

JURNAL MATEMATIKA, STATISTIKA DAN KOMPUTASI


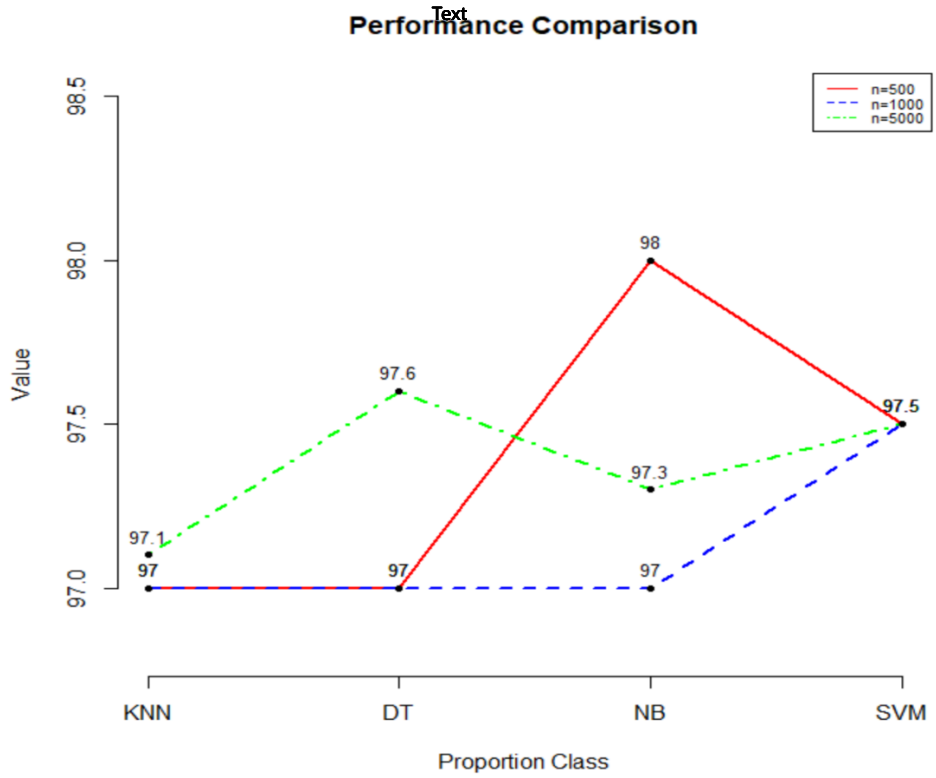


**Figure 3.7.** Performance of Classification Method Based on Accuracy Value

Similar results such as data simulation are also shown by the performance of DT and KNN methods, which provide high accuracy values when applied to Big Data. Similarly, the same results are given the NB method where small samples provide higher accuracy values. In contrast to the SVM method which provides the same accuracy value on all three types of sample sizes. In addition, the existence of large data effects does not prove to affect SVM performance.

Based on the consideration of the application of classification methods to simulation and real data, it has been proven that the effect of large data sizes on classification quality is given by the DT and KNN methods. However, the DT method has a quality influence on the proportion of classes scattered in the data.

## 4. CONCLUSION

The results of the evaluation of the simulation of Dataset 1 show that the methods that have an influence on the quality of the classification produced if applied to Big Data are the DT and KNN methods. However, in Dataset type 2 there is a change in the results of the DT method. This is because of the influence on the number of classes and the proportion of class distribution in the data. The results obtained from data simulation, proven by applying to real data by showing that similar methods provide a quality influence if applied to Big Data. While the NB and SVM methods do not show a consistent influence when applied to Big Data. The results of observations in this study show that the DT and KNN methods have several advantages that make them suitable for application to Big Data. Here are the reasons why both of these methods are good to apply to big data. DT has an easy-to-understand and visualizable structure, which helps in the

understanding of patterns in big data. While KNN is a flexible nonparametric method and does not have certain assumptions about data distribution. However, it is important to remember that each method also has its limitations. KNN can be computationally intensive when calculating distances to nearest neighbors, whereas Decision Trees are prone to overfitting on very large and complex datasets. Therefore,
the application of this method to Big Data must consider the balance between advantages and limitations related to the characteristics of the data and the purpose of the analysis.

## REFERENCES

[1]. Boris, M. & Milovic, M., 2012. Prediction and decision making in health care using data mining. *Kuwait chapter of arabian journal of business and management review*, Vol. 1, No. 12, 1–11.
[2]. Chrisinta, D. & Simarmata, J.E., 2023. Analisis Sentimen Penilaian Masyarakat Terhadap Pejabat Publik Menggunakan Algoritma Naïve Bayes Classifier. *Komputika: Jurnal Sistem Komputer*, Vol. 12, No. 1, 93–101.
[3]. Chen, C.L.P. & Zhang, C.-Y., 2014. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Inf Sci (N Y)*, 314–347.
[4]. Fathi, M., Haghi Kashani, M., Jameii, S.M., & Mahdipour, E., 2022. Big data analytics in weather forecasting: A systematic review. *Archives of Computational Methods in Engineering*, Vol. 29, No. 2, 1247–1275.
[5]. Gaye, B., Zhang, D., & Wulamu, A., 2021. Improvement of support vector machine algorithm in big data background. *Mathematical Problems in Engineering*, 1–9.
[6]. Ginting, R., 2022. *Analisis Big Data*. Klaten: CV. Penerbit Lakeisha.
[7]. Jin, X., Wah, B.W., Cheng, X., & Wang, Y., 2015. Significance and challenges of big data research doi: 10.1016/j.bdr.2. *Big data research*, Vol. 2, No. 2, 59–64.
[8]. Kramer, O., 2013. K-nearest neighbors. *In*: *Dimensionality reduction with unsupervised nearest neighbors*. 13–23.
[9]. Kumar, N. & Maurya, V., 2020. A review on machine learning (feature selection, classification and clustering) approaches of big data mining in different area of research. *Journal of Critical Reviews*, Vol. 7, No. 19, 2610–2626.
[10]. Kwang, K.J. & Wang, Z., 2019. Sampling techniques for big data analysis. *International Statistical Review*, Vol. 87, S177–S191.
[11]. Pham, Q. V., Nguyen, D.C., Huynh-The, T., Hwang, W.J., & Pathirana, P.N., 2020. Artificial intelligence (AI) and big data for coronavirus (COVID-19) pandemic: a survey on the state-of-the-arts. *IEEE access*, Vol. 8, 130820–130839.
[12]. Robert, N., Elder, J., & Miner, G.D., 2009. *Handbook of statistical analysis and data mining applications*. Academic press.
[13]. Rojas, J.A.R., Kery, M.B., Rosenthal, S., & Dey, A., 2017. Sampling techniques to improve big data exploration. *In*: *IEEE 7th symposium on large data analysis and visualization (LDAV)*. 26–35.
[14]. Saadoon, M., Hamid, S.H.A., Sofian, H., Altarturi, H.H., Azizul, Z.H., & Nasuha, N., 2022. Fault tolerance in big data storage and processing systems: A review on challenges and solutions. *Ain Shams Engineering Journal*, Vol. 13, No. 2, 101538.
[15]. Sujatha, R., Chatterjee, J.M., Jhanjhi, N., & Brohi, S.N., 2021. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocess Microsyst*, 80 (103615).

[16]. Sunil, K. & Mohbey, K.K., 2022. A review on big data based parallel and distributed approaches of pattern mining. *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 5, 1639–1662.
[17]. Tanveer, M., Rajani, T., Rastogi, R., Shao, Y.H., & Ganaie, M.A., 2022. Comprehensive review on twin support vector machines. *Annals of Operations Research*, 1–46.