Scientific Journal of Informatics
Vol. 9, No. 2, Nov 2022
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Comparative Analysis Performance of K-Nearest Neighbor Algorithm and Adaptive Boosting on the Prediction of Non-Cash Food Aid Recipients

Yusi Yustikasari[1], Husni Mubarok[2*], Rianto[3]
[1,2,3]Department of Informatics, Faculty of Engineering,
Universitas Siliwangi, Indonesia

Abstract.
Purpose: The implementation of this manual system is considered less accurate in obtaining the results of social assistance recipients. From these problems to overcome this problem, systematic calculations are needed. In processing data, a model is needed that can explain the data with its application, so a machine learning model is made that can help process the data.
Methods: This study's classification of non-cash food social assistance receipts uses the K-Nearest Neighbor and Adaptive Boosting algorithms. This study will compare the performance of the two algorithms.
Result: The results obtained for Adaptive Boosting are the best classification results with a maximum accuracy of 100% and produce a high AUC value of 1.0. In comparison, the ROC curve for the K-Nearest Neighbor algorithm produces an accuracy of 96% with an AUC value of 0.94.
Novelty: ROC curves in the two algorithms are good classification results because the two graphs cross above the diagonal line and produce an AUC value included in the Excellent classification.

Keywords: Adaptive Boosting, BPNT, Confusion Matrix, K-Nearest Neighbor
Received October 2021 / **Revised** October 2021 / **Accepted** October 2022

INTRODUCTION

Poverty is a problem faced by several developing countries, including Indonesia [1], [2]. The problem of poverty is closely related to society, economy, education, and other aspects. It means that poverty is an important problem that needs to be addressed immediately [3], [4]. To help reduce poverty, the government implements an assistance program for the poor. The assistance is the Non-Cash Food Social Assistance or BPNT. BPNT is food assistance that is distributed and given in non-cash form from the government to recipient communities every month [5].

Based on the results of observations made at the Tasikmalaya City Social Service, information was obtained that the poverty rate in Tasikmalaya City was still high. Therefore, the Tasikmalaya City Social Service implemented a non-cash food social assistance program or BPNT. The provision of BPNT takes into account the economic condition of the community. It has several criteria, including ownership status of residential buildings, house conditions such as roof conditions, wall conditions, floor conditions, use of restrooms, water sources used, and so on. The process of determining the recipients of social assistance by the Tasikmalaya City Social Service is still done manually, even though a lot of data is processed. The application of this manual system is considered less accurate in obtaining results from social assistance recipients because it takes a relatively long time to make decisions.

Thus, to overcome these problems, a systematic calculation is needed to determine who is entitled to receive social assistance. In processing data, a model is needed to explain the data with its application, so a machine learning model can help process the data [6], [7]. In this process, many machine learning algorithms can be applied [8], [9]. This study's non-cash food assistance receipts classification uses the K-Nearest Neighbor and Adaptive Boosting algorithms. The K-Nearest Neighbor algorithm was chosen because it detects and

*Corresponding author.
Email addresses: yusiyustikasari0@gmail.com (Yustikasari), husni.mubarok@unsil.ac.id (Mubarok), rianto@unsil.ac.id (Rianto)
DOI: 10.15294/sji.v9i2.32369

analyzes the complex and non-linear problems [10], [11]. In comparison, the Adaptive Boosting algorithm is one variant of several boosting algorithms that can change the weak classification model into a strong one [12].

The Machine Learning model created will be evaluated using the Confusion Matrix calculation and the Area Under The Curve (AUC) value generated from the Receiver Operating Characteristic (ROC) curve. This is used to see the performance of the results of each algorithm and the extent to which the accuracy of the K-Nearest Neighbor and Adaptive Boosting algorithms differs in predicting social assistance recipients.

In research [13] in 2020 regarding comparing the K-Nearest Neighbor and Naive Bayes methods with a case study of recommendations for determining scholarship recipients, the K-Nearest Neighbor algorithm has a higher accuracy of 100% compared to the K-Nearest Neighbor algorithm. Naive Bayes algorithm, which produces an accuracy of 99.89%. Research on the comparison of classification algorithms was also carried out by [14] in 2020, namely comparing the K-Nearest Neighbor and Adaptive Boosting methods using the base learner, namely CART, in multi-class classification. The results of this comparison get the best model with the highest accuracy, namely the Adaboost method, with many values of accuracy, precision, and recall that are greater than the values in the K-Nearest Neighbor algorithm model.

This study aims to compare the performance of two machine learning algorithms that produce the highest accuracy from previous studies, namely the K-Nearest Neighbor and Adaptive Boosting algorithms. The K-Nearest Neighbor and Adaptive Boosting algorithms each have advantages and disadvantages. Therefore, in this study, the two algorithms will be compared to obtain the most optimal algorithm results in determining recipients of non-cash food assistance.

METHODS

Data Collection

The data used in this study is primary data, namely data on non-cash food social assistance recipients obtained from the Integrated Data on Social Welfare of the Tasikmalaya City Social Service. Non-cash in this study includes the ownership status of residential buildings, house conditions such as roof conditions, wall conditions, floor conditions, use of restrooms, water sources used, and so on.

Research Methods

The research stages in Figure 1 started with a literature study, data collection and analysis, data processing, K-Nearest Neighbor and Adaboost algorithm modeling, model evaluation, and the conclusion.

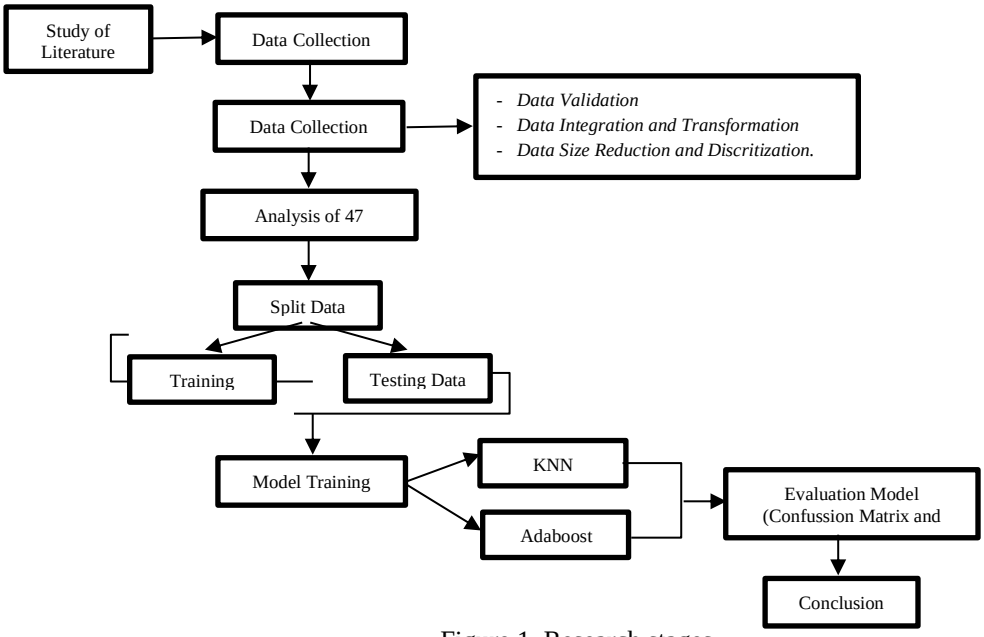

Figure 1. Research stages

RESULT AND DISCUSSION

In this section, the research results will be explained based on the stages described in the research methodology in the previous chapter. These stages start from data collection, data processing, variable analysis, model training process, and evaluation of the model with several predetermined schemes.

Data Collection

The data used in this study is primary data, namely data on non-cash food social assistance recipients obtained from the Integrated Data on Social Welfare of the Tasikmalaya City Social Service. Non-cash in this study includes the ownership status of residential buildings, house conditions such as roof conditions, wall conditions, floor conditions, use of restrooms, water sources used, and so on.

Data Processing

The sample data obtained amounted to 62,950 data with 47 attributes. The data is obtained in the form of an excel file which is still in the form of irregular data. For the data obtained to be of good quality and valid, it is necessary to carry out several stages of data processing, including:

a. Data Validation

At this stage, data cleaning is carried out by removing unnecessary data, such as data that is empty or null. The next step is to identify the little value as a missing value. A missing value is a condition where some attribute values in the dataset are empty or have no value[15]. At this stage, all attributes contained in the dataset are checked to determine whether there are inappropriate or null values. After being identified, seven attributes have missing values. These attributes are listed in Table 1 with each number of missing values.

Table 1. Attributes that have missing value

No	Attribute	Number of Missing Value
1	wall_condition	9590
2	presence_of_rt	6652
3	number_of_rooms	1356
4	toilet	164
5	roof_condition	148
6	power	3
7	business_sta_art	2

The right solution to overcome the missing value is necessary for imputation on each attribute. Before attributing the missing value, the skewness of the data is checked first to determine the correct imputation. Skewness is a measure of asymmetry in the distribution of values. To check the skewness of each data attribute with a missing value, a new variable is created containing the attributes with the missing value.

The skewness value is normal when the value is between -1 to 1 and -2 to 2. If the skewness value is normal, then the mean imputation is used, while if the skewness is not normal, then the median imputation is used. There is an imputation class consisting of the class mean, class median, and class mode.

Table 2. Results of the skewness attribute missing value

Out [12]:	wall_condition	-0.908196
	presence_of_rt	6.366525
	number_of_rooms	0.469626
	toilet	1.361469
	roof_condition	-1.339888
	power	1.445863
	business_sta_art	-1.309553
	dtype: float64	

The results of checking the skewness value show that only the presence_of_rt attribute, which has a skewness value of more than 2, means that the skewness value of the attribute is not normal. While others have a skewness value of less than 2, meaning that the skewness value of the attribute is normal. If the skewness is normal, then the imputation used is the mean imputation, while if the skewness is not normal, then the median imputation is used.

Missing Value imputation is a process used to determine and determine the replacement value for data that has a missing value. In this study, conventional imputation is the imputation used to overcome the missing value attribute data for social assistance recipients.

In conventional imputation, there is what is known as class-based imputation, which consists of class mean imputation, median class imputation, and class mode imputation. This research will determine imputation based on class, based on a class divided into two classes. The first class is based on "Recipient" target data and the second is based on "Non-Recipient" target data.

Table 3. Skewness grade 1

Out [14]:	wall_condition	-0.985573
	presence_of_rt	6.625487
	number_of_rooms	0.438348
	toilet	1.283937
	roof_condition	-1.426966
	power	1.411460
	business_sta_art	-1.401688
	dtype: float64	

The skewness value obtained in class 1 is only the presence_of_rt attribute which has an abnormal value.

Table 4. Skewness grade 2

Out [15]:	wall_condition	-0.135066
	presence_of_rt	4.412013
	number_of_rooms	0.668274
	toilet	3.351195
	roof_condition	-0.487306
	power	2.028527
	business_sta_art	-0.413770
	dtype: float64	

Meanwhile, the skewness value obtained in class 2 contains two attributes with abnormal values: the toilet attribute and the presence_of_rt attribute. After the skewness value of each class is known, it can be seen the standard deviation of the two classes for imputing each attribute.

Table 5. Description of class 1

	wall_condition	presence_of_rt	number_of_rooms	toilet	roof_condition	power	business_sta_art
count	49292.000000	52310.000000	57228.000000	58359.000000	58382.000000	58518.000000	58519.000000
mean	1.721009	1.064003	2.044052	1.674892	1.790398	2.185806	1.786958
std	0.448507	0.433493	0.787592	1.113482	0.407028	1.885248	0.409461
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000
50%	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000	2.000000
75%	2.000000	1.000000	3.000000	2.000000	2.000000	2.000000	2.000000
max	2.000000	4.000000	8.000000	4.000000	2.000000	6.000000	2.000000

Table 6. Description of class 2

	wall_condition	presence_of_rt	number_of_rooms	toilet	roof_condition	power	business_sta_art
count	4068.000000	3988.000000	4366.000000	4427.000000	4420.000000	4429.000000	4429.000000
mean	1.533677	1.133902	2.287907	1.206912	1.618326	1.984872	1.601264
std	0.498926	0.619574	0.865675	0.643798	0.485852	1.444235	0.489693
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	1.000000	1.000000	2.000000	1.000000	1.000000	1.000000	1.000000
50%	2.000000	1.000000	2.000000	1.000000	2.000000	2.000000	2.000000
75%	2.000000	1.000000	3.000000	1.000000	2.000000	2.000000	2.000000
max	2.000000	4.000000	8.000000	4.000000	2.000000	6.000000	2.000000

The value of the standard deviation can be seen from the description of the two classes. In the attribute wall_condition, presence_of_rt, number_of_rooms, roof_condition, power, business_sta_art using Mean Imputation, the standard deviation value in class 1 obtained is not much different from the standard deviation value in class 2. While in the toilet attribute, the standard deviation value from

class 1 is 1.11, and the value of the standard deviation of class 2 is 0.64, so this toilet attribute uses Class Mean Imputation.

The imputation process on the 'toilet' attribute uses class mean imputation. Because the imputation used is class imputation, the imputation process is carried out in each class. Furthermore, the imputation process for the attributes' wall_condition, presence_of_rt, number_of_rooms, roof_condition, power, business_sta_art' uses mean imputation.

b. Data Integration and Transformation

Data integration and transformation are steps to improve the accuracy and efficiency of the algorithms used [16]. Data transformation is carried out at this stage, transforming recipient status attributes into two categories. Based on the regulations set by the agency that the status of recipients with a value range of 0 to 36 is included in the status of 'Recipient,' and data with a value range of 37 to 75 is the status of 'Non-Recipient.' The data transformation uses the NumPy library in the python programming language.

In Machine Learning, some things must be considered, namely that some algorithms work more optimally with numeric-type data to achieve better results [17]. Because the recipient's status has been transformed into text or categorical, it is necessary to convert the data to numeric to create a model. In this study, feature encoding is carried out through the Label Encoder contained in the sklearn preprocessing library to convert text or categorical variables into numeric values. This method refers to converting labels into a numeric form so as to convert them into machine-readable form. In the command above, the recipient status attribute with the label 'Recipient' is converted to a number, namely '1', and the label 'Not Recipient' is converted to '0.'

c. Data Size Reduction and Discretization

Data Size Reduction and Discretization Stages are used to obtain datasets with a small number of records but are informative [18]. In the training data, feature selection and deletion of duplication data are carried out.

d. K-Nearest Neighbor Algorithm Experiment Results

In this study, there were 62950 data from the dataset of social assistance recipients having 42 variables consisting of 41 independent variables and one dependent variable. In conducting classification analysis to build machine learning models, the data must first be divided into training and testing data [19], [20]. Sharing training and testing data are done by dividing some data into several scenarios. From several scenarios, a model and the best scenario will be selected based on the calculation of the Confusion Matrix and the AUC Value. Several scenarios for sharing training and testing data are listed in Table 7.

Table 7. Data sharing scenarios

No	Training Data	Testing Data
1	80%	20%
2	70%	30%
3	60%	40%
4	50%	50%

The next stage starts with building the K-Nearest Neighbor model. For the first step of modeling testing, the author wants to create a K-Nearest Neighbor model using the default parameters of the KneighborsClassifier [21]. In this study, the value of k that will be built based on the default result is 5.

The K-Nearest Neighbor model is a model for grouping data based on closest-neighbor data or Euclidean distance. The following is an example of a manual calculation to find the Euclidean distance between the new and the first data.

Euclidean = $\sqrt{((a1 - b1)^2 + \cdots + (a2 - b2)^2)}$

D(1,Baru)= $\sqrt{((4 - 1)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (2-6)^2 \cdots + (1 - 1)^2)}$

D(1, Baru) = $\sqrt{88}$ = 9.38083152

e. Evaluation of the Confusion Matrix of the K-Nearest Neighbor Algorithm

The method used to evaluate the model is to calculate the confusion matrix on the model that has been made to obtain accuracy, precision, recall, and f-measure scores. In Table 8, there is a Classification Report generated from the K-Nearest Neighbor Algorithm, with data sharing 80%:20%

Table 8. Classification report data sharing 80%:20%

	precision	recall	F1-score	support
Non-Recipient	0.88	0.53	0.66	668
Recipient	0.96	0.99	0.98	8699
Accuracy			0.96	9367
Macro avg	0.92	0.76	0.82	9367
Weighted avg	0.96	0.96	0.96	9367

The results of the classification of social assistance recipient data using the K-Nearest Neighbor algorithm with a data-sharing scenario of 80%:20% can be seen in Table 9 below:

Table 9. Conversion matrix conversion k-nearest neighbor algorithm 80%:20% data sharing

	True Recipient	True Non-Recipient	Class Precision
Pred Recipient	353	315	96%
Pred Non-Recipient	47	8652	88%
Class Recall	99%	53%	

Table 9 shows that from 9367 data, 353 data are classified as recipients according to the predictions made, while 315 data are predicted to be recipients. Still, the results are non-recipients, 8652 data are non-recipients predicted according to the system for the class, and 47 data are predicted to be non-recipients and recipients. Table 8 on the Classification Report and Table 9 on the Confusion Matrix conversion show that the K-Nearest Neighbor algorithm's accuracy with a data ratio of 80%:20% is 96%.

In Table 10. there is a Classification Report generated from the K-Nearest Neighbor Algorithm for 70%:30% data sharing. The Classification Report has values or percentages of precision, recall, and accuracy.

Table 10. Classification report data sharing 70%:30%

	precision	recall	F1-score	support
Non-Recipient	0.39	0.13	0.20	1059
Recipient	0.93	0.98	0.96	12992
Accuracy			0.92	14051
Macro avg	0.66	0.56	0.58	14051
Weighted avg	0.89	0.92	0.90	14051

The results of the classification of social assistance recipient data using the K-Nearest Neighbor algorithm with a 70%:30% data sharing scenario can be seen in Table 11 below:

Table 11. Conversion matrix k-nearest neighbor algorithm data sharing 70%:30%

	True Recipient	True Non-Recipient	Class Precision
Pred Recipient	138	921	93%
Pred Non-Recipient	214	12778	39%
Class Recall	98%	13%	

Table 11 shows that from 14051 data, 138 data are classified as Recipients according to the predictions made, while 921 data are predicted to be Recipients. Still, the results are Non-Recipients, the system for classes predicts 12778 Non-Recipient data, and 214 data are predicted to be Non-Recipients. Turns Receiver. Based on Table 10 in the classification report and Table 11 in the conversion confusion matrix, the K-Nearest Neighbor algorithm's accuracy with a data comparison ratio of 70%: 30% is 92%.

In Table 12. there is a Classification Report generated from the K-Nearest Neighbor Algorithm for 60%: 40% data sharing. The Classification Report has values or percentages of precision, recall, and accuracy.

Table 12. Classification report data sharing 60%:40%

	precision	recall	F1-score	support
Non-Recipient	0.38	0.13	0.20	1443
Recipient	0.93	0.98	0.96	17291
Accuracy			0.92	18734
Macro avg	0.65	0.56	0.58	18734
Weighted avg	0.89	0.92	0.90	18734

The results of the classification of social assistance recipient data using the K-Nearest Neighbor algorithm with a 60%: 40% data sharing scenario can be seen in Table 13 below:

Table 13. Conversion matrix k-nearest neighbor algorithm data sharing 60%:40%

	True Recipient	True Non-Recipient	Class Precision
Pred Recipient	192	1251	93%
Pred Non-Recipient	320	16971	38%
Class Recall	98%	13%	

Table 13 shows that from 18734 data, 192 data are classified as recipients according to the predictions made. In comparison, 1251 data are predicted to be the recipient. Still, the result is Non-Recipient, 16971 Non-Recipient data is predicted according to the system for the class, and as many as 320 are predicted to be Non-Recipient. The recipient turned out to be a recipient. Based on Table 12 in the classification report and Table 13 in the conversion of the confusion matrix shows that the level of accuracy produced by the K-Nearest Neighbor algorithm with a data comparison ratio of 60%: 40% is 92%.

In Table 14. there is a Classification Report generated from the K-Nearest Neighbor Algorithm for 50%:50% data sharing. The Classification Report has values or percentages of precision, recall and accuracy.

Table 14. Classification report data sharing 50%:50%

	precision	recall	F1-score	support
Non-Recipient	0.38	0.13	0.19	1814
Recipient	0.93	0.98	0.96	21604
Accuracy			0.92	23418
Macro avg	0.65	0.56	0.57	23418
Weighted avg	0.89	0.92	0.90	23418

The results of the classification of social assistance recipient data using the K-Nearest Neighbor algorithm with a 50%:50% data sharing scenario can be seen in table 15 below:

Table 15. Conversion matrix k-nearest neighbor algorithm data sharing 50%:50%

	True Recipient	True Non-Recipient	Class Precision
Pred Recipient	232	1582	93%
Pred Non-Recipient	383	21221	38%
Class Recall	98%	13%	

In Table 15 it is known that from 23418 data, 232 data are classified as recipient according to the predictions made, while 1582 data are predicted to be recipient. Still, the results are Non-Recipient, 21221 Non-Recipient data are predicted according to the system for the class, and 383 data are predicted to be Non-Recipient. Based on Table 14 in the classification report and Table 15 in the conversion of the confusion matrix, it shows that the level of accuracy produced by the K-Nearest Neighbor algorithm with a data comparison ratio of 50%:50% is 92%.

From the results of the confusion matrix calculation, it can be calculated to find the values of accuracy, specificity, sensitivity, ppv, and npv in the following equation:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (1)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (2)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{PPV} = \frac{TP}{TP+FP} \quad (4)$$

$$\text{NPV} = \frac{TN}{TN+FN} \quad (5)$$

Based on these equations, the different values of accuracy, specificity, sensitivity, ppv, and npv are obtained from each data sharing ratio as shown in Table 16.

Table 16. Values of accuracy, specificity, sensitivity, ppv, and npv

Data Scenario(%)	Accuracy(%)	Specificity(%)	Sensitivity(%)	Ppv(%)	Npv(%)
80:20	96.13	99.45	52.84	88.25	96.48
70:30	91.92	98.35	13.03	39.20	93.27
60:40	91.61	98.14	13.30	37.50	93.13
50:50	91.60	98.22	13.30	37.72	93.06

From the calculations produced on the 80%:20% data distribution, the highest percentage values of accuracy, specificity, sensitivity, ppv, and npv were obtained. The resulting percentage of correct predictions of social assistance recipients is predicted to be 'Recipients' and 'Non-Recipients' of the total data, which is 96.13%. In specificity, the percentage generated is calculated from the correct prediction of social assistance recipients who are predicted to be 'Non-Recipient' compared to the total data that are actually 'Non-Recipient' of 99.45%. Meanwhile, the sensitivity resulting from the calculation of the prediction of social assistance recipients who are predicted to be 'Recipients' is compared to the overall data which actually becomes 'Recipients' of 52.84%. In Table 16, the percentage of PPV or Positive Predictive Value generated is 88.25%. This Positive Predictive Value is the percentage of the possibility or chance that someone will become a recipient of social assistance if the prediction result is 'Recipient'. While NPV or Negative Positive Value is the possibility or probability of someone becoming a Non-Recipient of social rock if the prediction results are 'Non-Recipient'. The NPV value generated in the prediction of the recipients of this social assistance is 96.48%.

f. Adaptive Boosting Algorithm Experiment

This sub-chapter describes the implementation of Adaptive Boosting for classifying. The implementation of Adaptive Boosting is done using the scikit learn library in python.

The process of sharing training data and data testing is done by dividing the data into several scenarios. From the existing scenarios, the best scenario will be selected based on the calculation of the confusion matrix calculation and the AUC value. Several scenarios for sharing training and test data are listed in Table 17.

Table 17. Data scenarios

No	Training Data(%)	Testing Data(%)
1	80	20
2	70	30
3	60	40
4	50	50

After the data is divided into training and test data, the training data will be used in the modeling stage. Base estimator or basic classifier used in modeling with Adaptive Boosting using Decision Tree.

g. Confusion Matrix Adaptive Boosting Evaluation

The second Confusion Matrix model uses an adaptive boosting classification, then inputs the prepared training data into the Confusion Matrix so that the results from the Classification Report are shown in Table 18.

Table 18. Classification report adaboost data sharing 80%:20%

	precision	recall	F1-score	support
Non-Recipient	1.00	1.00	1.00	668
Recipient	1.00	1.00	1.00	8699
Accuracy		1.00	1.00	9367
Macro avg	1.00	1.00	1.00	9367
Weighted avg	1.00	1.00	1.00	9367

The results of the classification of social assistance recipient data using Adaptive Boosting with 80%:20% data sharing scenarios can be seen in Table 19 below:

Table 19. Conversion matrix adaboost data sharing 80%:20%

	True Recipient	True Non-Recipient	Class Precision(%)
Pred Recipient	668	0	100
Pred Non-Recipient	0	8699	100
Class Recall(%)	100	100	

In Table 19 it is known that from 9367 data, 668 data are classified as Recipients according to the predictions made, while 0 data are predicted as Recipients, but the result is Non-Recipients, 8699 Non-Recipient data are predicted according to the system for class, and 0 data are predicted to be Non-Recipients. Based on Figure 18 on the classification report and Table 19 on the confusion matrix conversion, it shows that the level of accuracy produced by the Adaboost algorithm with a data ratio of 80%: 20% is 100%.

In Table 20 there is a Classification Report generated from the Adaboost Algorithm for 70%:30% data sharing. The Classification Report has values or percentages of precision, recall and accuracy.

Table 20. Classification report adaboost data sharing 70%:30%

	precision	recall	F1-score	support
Non-Recipient	0.61	0.15	0.24	1059
Recipient	0.93	0.99	0.96	12992
Accuracy			0.93	14051
Macro avg	0.77	0.57	0.60	14051
Weighted avg	0.91	0.93	0.91	14051

The results of the classification of social assistance recipient data using Adaptive Boosting with a 70%:30% data sharing scenario can be seen in Table 21 below:

Table 21. Conversion matrix adaboost data sharing 70%:30%

	True Recipient	True Non-Recipient	Class Precision(%)
Pred Recipient	161	898	93
Pred Non-Recipient	102	12890	61
Class Recall(%)	99	15	

In Table 10 it is known that from 1,4051 data, 161 data are classified as recipients according to the predictions made, while 898 data are predicted to be recipients. Recipient but the result turns out to be Non-Recipient, 12,890 Non-Recipient data is predicted according to the system for the class, and as many as 102 data predicted Non-Recipient turns out to be recipient. Based on Table 20 in the classification report and Table 21 in the conversion of the confusion matrix, it shows that the level of accuracy produced by the Adaboost algorithm with a data comparison ratio of 70%:30% is 93%.

In Table 22 there is a Classification Report generated from the Adaboost Algorithm for 60%:40% data sharing. The Classification Report has values or percentages of precision, recall, and accuracy.

Table 22. Classification report adaboost data sharing 60%:40%

	precision	recall	F1-score	support
Non-Recipient	0.56	0.16	0.25	1443
Recipient	0.93	0.99	0.96	17291
Accuracy			0.93	18734
Macro avg	0.75	0.58	0.61	18734
Weighted avg	0.91	0.93	0.91	18734

The results of the classification of social assistance recipient data using Adaptive Boosting with a 60%: 40% data sharing scenario can be seen in Table 23 below:

Table 23. Conversion matrix adaboost data sharing 60%:40%

	True Recipient	True Non-Recipient	Class Precision(%)
Pred Recipient	234	1209	93
PredNon-Recipient	184	17107	56
Class Recall(%)	99	16	

In Table 23 it is known that from 18734 data, 234 data are classified as recipient according to the predictions made, while 1209 data is predicted to be recipient, but the result is Non-Recipient, 17107 Non-Recipient data is predicted according to the system for class, and 184 data is predicted to be Non-Recipient. Based on Figure 22 in the classification report and Table 23 in the conversion of the confusion matrix, it shows that the level of accuracy produced by the Adaboost algorithm with a data comparison ratio of 60%: 40% is 93%.

In Table 24 there is a Classification Report generated from the Adaboost Algorithm for 50%:50% data sharing. The Classification Report has values or percentages of precision, recall and accuracy.

Table 24. Classification report adaboost data sharing 50%:50%

	precision	recall	F1-score	support
Non-Recipient	0.55	0.17	0.26	1814
Recipient	0.93	0.99	0.96	21604
Accuracy			0.92	23418
Macro avg	0.74	0.58	0.61	23418
Weighted avg	0.90	0.92	0.91	23418

The results of the classification of social assistance recipient data using Adaptive Boosting with a 50%:50% data sharing scenario can be seen in Table 25 below:

Table 25. Conversion matrix adaboost data sharing 50%:50%

	True Recipient	True Non-Recipient	Class Precision(%)
Pred Recipient	309	1505	93
Pred Non-Recipient	253	21351	55
Class Recall(%)	99	17	

In Table 25 it is known that from 23418 data, 309 data are classified as recipient according to the predictions made, while 1505 data is predicted to be recipient, but the result is Non-Recipient, 21351 Non-Recipient data is predicted according to the system for class, and 253 data is predicted to be Non-Recipient. Based on Table 24 in the classification report and Table 25 in the confusion matrix conversion, it shows that the level of accuracy produced by the Adaboost algorithm with a data comparison ratio of 50%:50% is 92%.

Based on these equations, different values of accuracy, specificity, sensitivity, ppv, and npv are obtained from each data sharing ratio as shown in Table 26.

Table 26. Value of accuracy, specificity, sensitivity, ppv, and npv

Data Scenario(%)	Accuracy(%)	Specificity(%)	Sensitivity(%)	Ppv(%)	Npv(%)
80:20	100	100	100	100	100
70:30	92.88	93.48	61.21	15.20	99.21
60:40	92.56	93.39	55.98	16.21	98.93
50:50	92.49	93.41	54.98	17.03	98.82

From the calculations produced on the 80%:20% data distribution, the highest percentage values for accuracy, specificity, sensitivity, ppv, and npv are obtained. The percentage generated on the correct prediction of social assistance recipients is predicted to be 'Recipient' and 'Non-Recipient' from the overall data, which is 100%. In the specificity, the percentage generated is calculated from the prediction of the correct social assistance recipient which is predicted to be 'Non-Recipient' which is compared to the overall data which is actually 'Non-Recipient' of 100%. Meanwhile, the sensitivity generated by the calculation of the prediction of the recipient of social assistance which is predicted

to be 'Recipient' is compared to the overall data which actually becomes 'Recipient' of 100%. Table 26 shows the PPV or Positive Predictive Value percentage generated is 100%. This Positive Predictive Value is the percentage of the possibility or probability of a person becoming a recipient of social assistance if the prediction results are 'Recipient'. While the NPV or Negative Positive Value is the possibility or probability of a person being a Non-Recipient of social rock if the prediction results are 'Non-Recipient'. The NPV value generated in the prediction of this social assistance recipient is 100%.

The results of the Classification Report and Confusion matrix conversion above show that the accuracy values obtained are different in several scenarios of sharing training data and testing data. The comparison of the accuracy can be seen in Table 27. The table shows that the highest accuracy value is in the scenario of 80%:20% data sharing, which is 100%.

Table 27. Comparison of the accuracy of each data sharing scenario on adaboost

No	Training Data(%)	Testing Data(%)	Adaboost Accuracy(%)
1	80	20	100
2	70	30	93
3	60	40	93
4	50	50	92

h. Comparison of Classification Results

The next step after getting the accuracy, precision, and recall values on the K-Nearest Neighbor and Adaptive Boosting algorithms are to compare the accuracy of the classification of the two algorithms. The results of the accuracy of the K-Nearest Neighbor and Adaptive Boosting algorithms can be seen in Table 28 below:

Table 28. Results of comparison accuracy of KNN and adaBoost

No	Training Data(%)	Testing Data(%)	KNN Accuracy(%)	Adaboost Accuracy(%)	Training Time(s)
1	80	20	96	100	9.29
2	70	30	92	93	6.43
3	60	40	92	93	4.60
4	50	50	92	92	2.75
Average accuracy (%)			93	94.5	

Based on the test results above, the comparison of the two classification algorithms, namely the K-Nearest Neighbor and Adaptive Boosting algorithms, the results of measuring accuracy using the confusion matrix proved that the test results of the Adaboost algorithm have a higher accuracy value than the K-Nearest Neighbor algorithm. Both algorithms produce the highest accuracy at 80%:20% data sharing ratio. The K-Nearest Neighbor algorithm produces an accuracy of 96% while the accuracy value for the test results of the Adaboost model is 100%. The average accuracy obtained from several data sharing ratios, namely the K-Nearest Neighbor algorithm, produces an average accuracy of 93% and the average accuracy produced by Adaboost is 94.5%.

i. Evaluation of the ROC Curve at the Highest Classification Result

The results of the evaluation of the confusion matrix show that the accuracy obtained for each data sharing ratio is different. The highest accuracy is obtained from the results of data sharing 80%:20%. The highest accuracy is obtained from the Adaptive Boosting algorithm, which is 100% and the K-Nearest Neighbor algorithm produces an accuracy of 96%.

In addition to using the confusion matrix, the model evaluation process in this study uses the ROC Curve function. ROC Curve or ROC Curve is a visualization to compare classification results and accuracy. The ROC curve is in the form of a two-dimensional curve or graph with the true positive rate (TPR) as a vertical line and the false positive rate (FPR) as a horizontal line.

The results of the evaluation of the ROC Curve calculation in this study used the sklearn metrics library in python with the roc_curve function and the roc_auc_score function to display the ROC Curve and AUC Value. The results of the visualization of the calculation evaluation with the ROC curve on two algorithms, namely K-Nearest Neighbor and Adaptive Boosting can be seen in Figure 2.

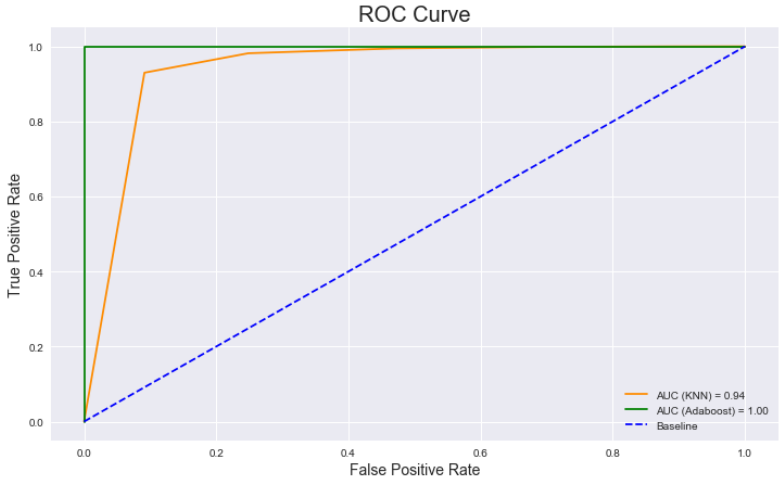

Figure 2. KNN and adaboost ROC curves

Figure 2 shows that the ROC curve for Adaptive Boosting is the best classification result with a maximum accuracy of 100% and produces a high AUC value of 1.0. While the ROC curve for the K-Nearest Neighbor algorithm produces an accuracy of 96% with an AUC value of 0.94. ROC curves in the two algorithms are good classification results because the two graphs cross above the diagonal line and produce an AUC value which is included in the Excellent classification.

CONCLUSION

Based on the research that has been done, it can be concluded that the comparison of Machine Learning algorithms between K-Nearest Neighbor and Adaptive Boosting on the prediction of non-cash food social assistance recipients has different results from each data sharing ratio and both algorithms produce good classifications. The results of the evaluation of the confusion matrix show that the accuracy obtained for each data sharing ratio is different. The highest accuracy is obtained from the results of data sharing 80%:20%. Adaptive Boosting obtains more accurate accuracy than the K-Nearest Neighbor algorithm in predicting recipients of non-cash food assistance. The average accuracy obtained from several data sharing ratios, namely the K-Nearest Neighbor algorithm produces an average accuracy of 93% and the average accuracy produced by Adaboost is 94.5%. The evaluation results from the ROC Curve for Adaptive Boosting are the best classification results with a maximum accuracy of 100% and a high AUC value of 1.0. While the ROC curve for the K-Nearest Neighbor algorithm produces an accuracy of 96% with an AUC value of 0.94. The ROC curve in both algorithms is a good classification result because the two graphs cross above the diagonal line and produce an AUC value which is included in the Excellent Classification. Thus, Adaptive Boosting is a fairly good method in predicting recipients of non-cash food assistance or BPNT.

REFERENCES

[1] F. Salignac, J. Hanoteau, and I. Ramia, "Financial Resilience: A Way Forward Towards Economic Development in Developing Countries," *Soc. Indic. Res.*, vol. 160, no. 1, pp. 1–33, 2022.

[2] T. Pham and A. Nugroho, "Tourism-induced Poverty Impacts of COVID-19 in Indonesia," *Ann. Tour. Res. Empir. Insights*, vol. 3, no. 2, p. 100069, 2022.

[3] E. Firasari, N. Khasanah, U. Khultsum, D. N. Kholifah, R. Komarudin, and W. Widyastuty, "Comparation of K-Nearest Neighboor (K-NN) and Naive Bayes Algorithm for the Classification of the Poor in Recipients of Social Assistance," *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020.

[4] Z. Manshor, S. Abdullah, and A. B. Hamed, "Poverty and the Social Problems," *Acad. Res. Bus. \& Soc. Sci.*, vol. 10, no. 3, pp. 614–617, 2020.

[5] N. Vedy and V. Juwono, "Analysis of The Implementation of Non-cash Food Assistance (BPNT) Program for Reducing Poverty in Sub-district Panjang, Bandar Lampung 2018," *Proc. 3rd Int. Conf. Adm. Sci. Policy Gov. Stud.*, 2020.

[6] V. Siddesh Padala, K. Gandhi, and D. V. Pushpalatha, "Machine Learning: The New Language for Applications," *IAES Int. J. Artif. Intell.*, vol. 8, no. 4, pp. 411–421, 2019.

[7] V. Jain, G. S. Narula, and M. Singh, "Implementation of Data Mining in Online Shopping System Using," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 1, pp. 47–58, 2013.

[8] Y. Resti, F. Burlian, I. Yani, D. A. Zayanti, and I. M. Sari, "Improved the Cans Waste Classification Rate of Naïve Bayes Using Fuzzy," *Sci. Technol. Indones.*, vol. 5, no. 3, pp. 3–6, 2020.
[9] B. Sierra, E. Lazkano, J. M. Mart, and A. Astigarraga, "Combining Bayesian Networks , k Nearest Neighbours Algorithm and Attribute Selection for Gene Expression Data Analysis," in *Australas. Jt. Conf. Artif. Intell.*, 2004, pp. 86–97.
[10] E. N. Stankova and E. V. Khvatkov, "Using Boosted K-Nearest Neighbour Algorithm for Numerical Forecasting of Dangerous Convective Phenomena," in *Int. Conf. Comput. Sci. Appl.*, 2019, pp. 802–811.
[11] E. Z. Ferdousy, M. Islam, and M. A. Matin, "Combination of Naïve Bayes Classifier and K-Nearest Neighbor (KNN) in the Classification Based Predictive Models," *Comput. Inf. Sci.*, vol. 6, no. 3, 2013.
[12] J. Deshmukh and U. Bhosle, "A Study of Mammogram Classification Using Adaboost with Decision Tree, KNN, SVM And Hybrid SVM-KNN as Component Classifiers," *J. Inf. Hiding Multimed. Signal Process.*, vol. 9, no. 3, pp. 548–557, 2018.
[13] A. Sumiah and N. Mirantika, "Perbandingan Metode K-Nearest Neighbor dan Naive Bayes untuk Rekomendasi Penentuan Mahasiswa Penerima Beasiswa Pada Universitas Kuningan," *Buffer Inform.*, vol. 6, no. 1, pp. 1–10, 2020.
[14] A.I. Prianti, R. Santoso and A. R. Hakim, "Perbandingan Metode K-Nearest Neighbor dan Adaptive Boosting Pada Kasus Klasifikasi Multi Kelas 1,2,3," *J. Gaussian*, vol. 9, no. 2018, pp. 346–354, 2020.
[15] D.M. Farid, N. Harbi and M. Z. Rahman, "Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection," *arXiv*, vol. 2, no. 2, pp. 12–25, 2010.
[16] A. Bharathi and E. Deepankumar, "Survey on Classification Techniques in Data Mining," *Int. J. Recent Innov. Trends Comput. Commun.*, pp. 1983–1986, 2014.
[17] M. N. Baby and L. T. Priyanka, "Customer Classification and Prediction based on Data Mining Technique," *Sci. Inf. Database*, vol. 2, no. 12, 2012.
[18] B. Tran, B. Xue, and M. Zhang, "A New Representation in PSO for Discretization-Based Feature Selection," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1733–1746, 2017.
[19] Y. F. Safri, R. Arifudin, and M. A. Muslim, "K-Nearest Neighbor and Naive Bayes Classifier Algorithm in Determining The Classification of Healthy Card Indonesia Giving to The Poor," *Sci. J. Informatics*, vol. 5, no. 1, pp. 9–18, 2018.
[20] M. R. Hidayah, I. Akhlis, and E. Sugiharti, "Recognition Number of The Vehicle Plate Using Otsu Method and K-Nearest Neighbour Classification," *Sci. J. Inform*atics., vol. 4, no. 1, pp. 66–75, 2017.
[21] R. Andrian, D. Maharani, M. A. Muhammad, and A. Junaidi, "Butterfly Identification Using Gray Level Co-Occurrence Matrix (Glcm) Extraction Feature and K-Nearest Neighbor (KNN) Classification," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 6, no. 1, pp. 11–21, 2020.