International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Machine Learning Algorithm Comparison using Sampling Techniques for Car Insurance Claim Classification

Gerry Geraldo German[1], Dinar Ajeng Kristiyanti [2*]

Submitted:13/03/2024 **Revised**: 28/04/2024 **Accepted**: 05/05/2024

Abstract: This research aims to compare the performance of machine learning algorithms in the classification of car insurance claims using sampling techniques. The study focuses on addressing the issue of class imbalance in insurance claim data, which can result in errors when detecting fraudulent claims. Oversampling and Undersampling techniques are applied to Decision Tree, Random Forest, Naïve Bayes, K-Nearest Neighbors, and Logistic Regression algorithms. The research methodology follows Knowledge Discovery in Database principles and utilizes RapidMiner version 10.1 as the tool for constructing classification models using the insurance claim data. The evaluation results reveal that the K-Nearest Neighbor (K-NN) algorithm with Oversampling technique achieves the highest performance in predicting insurance claims, with an accuracy of 90.46%, recall of Yes class at 99.03%, recall of No class at 81.88%, precision of Yes class at 84.53%, precision of No class at 98.83%, and an AUC of 0.984. Furthermore, the evaluation and visualization of performance comparisons indicate that the Random Forest (RF) algorithm with Oversampling technique and the K-Nearest Neighbors (K-NN) algorithm with Oversampling technique exhibit the most promising results in predicting car insurance claims.

Keywords: *Car Insurance Claim, Classification, Decision Tree, K-Nearest Neighbors, Logistic Regression, Naïve Bayes, Random Forest, Sampling Techniques*

1. Introduction

Insurance is an agreement between two parties, namely the insurance company and the policyholder, which forms the basis for the acceptance of premiums by insurance companies in return for providing reimbursement to the insured or policyholder, providing payment [1]. The purpose of auto vehicle insurance is to take over the risks that may be borne by the owner of the car vehicle concerned with the finances suffered by motor vehicles due to various indeterminate reasons which is based on the death or life of the insured or policyholder [2]. According to the Insurance Information Institute in 2019, total vehicle claims filed in the United States were approximately 33.3 million claims. Of the total claims, about 3.0% or around 995,000 claims were rejected due to indications of fraud in vehicle claims [3]. Fraud in vehicle claims can occur by forging claim documents or reporting claims for damage that did not actually occur. Data mining has become an important topic in the field of information technology because it can help organizations or companies to understand more deeply about the data they have, including car insurance claim data. By using machine learning algorithms to predict fraud, car insurance companies can reduce the risk of financial losses due to fraudulent actions committed by their customers. Class imbalance has a bad impact on classification results where the minority class is often misclassified as the majority class. This can reduce the accuracy value of the classification results [4]. In classification problems, class imbalance is defined as having the majority of observations from one class, which makes it challenging for the classifier to detect the minority group. Many researchers studied class imbalance problem classification and presented solutions [5]. Data mining can be a useful tool in analyzing car insurance claim data. Data mining is the process of extracting and analyzing massive data to identify useful patterns or relationships in data [6].

Several previous studies have been conducted to support the comparison of machine learning algorithms. In 2022, using Vehicle Claim Fraud data comparing Logistic Regression, K-Nearest Neighbor, Random Forest and XGBoost algorithms, it shows that the Random Forest algorithm has the highest accuracy rate of 98.5% in handling data imbalances in Vehicle Claim Fraud data. [7]. In addition, in 2023 using Algerian Forest Fires data and Random Undersampling techniques with the Decision Tree algorithm resulted in an accuracy of 94.52% and an ROC value of 0.950 in handling data imbalances [8]. Furthermore, in 2018 using German credit data comparing Random Forest, K-Nearest Neighbor, Naïve Bayes and J48 showed that the Random Forest algorithm has an accuracy of 90.10% in handling data imbalances [9].

This study aims to compare SMOTE Oversampling and Undersampling sampling techniques on Decision Tree algorithms, Random Forest, Naïve Bayes, K-Nearest Neighbors and Logistic Regression. By selecting machine learning algorithms that are most effective at classifying

[1,2*] *Information Systems Study Program, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia*
[2*] *ORCID ID: 0000-0001-7887-9842*
* *Corresponding Author Email: dinar.kristiyanti@umn.ac.id*

fraud labels on auto insurance claim data, auto insurance companies can improve the effectiveness of predicting fraud and reduce the risk of financial loss. Class imbalances can have a negative impact on classification outcomes because often minority classes will be misclassified as majority classes [10].

2. Materials and Methods

2.1. Data Mining

Data Mining is a term that used to decipher knowledge discovery within a database [11].Data mining is the process of extracting useful knowledge or information from large, complex, and varied data. This algorithm uses probability theory to determine the probability of an event based on available data. This process involves several techniques, such as classification, grouping, association, and prediction, to find patterns, relationships, and trends hidden in the data [12]. According to Florin Gorunescu, classification is one of the main techniques in data mining used to build predictive models [12].

2.2. SMOTE and Undersampling

SMOTE (Synthetic Minority Over-sampling Technique) is a commonly used method to overcome data imbalance by producing synthetic samples from minority classes [13]. Undersampling involves reducing the number of samples from the majority class to align with the number of samples from the minority class [9].

2.3. Evaluation and Validation Techniques

2.3.1. Cross Validation

Cross-validation is a technique used in Machine Learning and statistics to evaluate the performance of predictive models using available data [14]. Cross-Validation is used to ensure that the model built can be well generalized to datasets not seen before. Cross-validation involves dividing a dataset into subsets and using those subsets to train and test models [15].

2.3.2. Confusion Matrix

A confusion matrix is a table used to measure the performance of classification models in predicting target class labels. Confusion matrix is usually used in Supervised Learning to evaluate the results of model predictions and to make comparisons between classification techniques, can be done through several evaluation matrices such as [16].

1 *Accuracy*

Accuracy is used to measure how accurately the model predicts the correct class from a given dataset. Accuracy is calculated as the number of correct predictions divided by the total number of predictions. Mathematical equations to calculate accuracy using equation 1 [17].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (1)$$

2 *Precision*

Precision measures how many predictions are correct from all positive predictions made by the model. Precision can be calculated as the ratio between the number of true positive predictions and the total number of positive predictions (true positives and false positives). Mathematical equations for calculating precision using equation 2 [18].

$$Precision = \frac{TP}{TP+FP} * 100\% \quad (2)$$

3 *Recall*

Recall measures how many correct predictions from all data are actually positive in a dataset. Recalls can be calculated as the ratio between the number of true positives and the total number of true positives and false negatives. Mathematical equation for calculating recall using equation 3 [19].

$$Recall = \frac{TP}{TP+FN} * 100\% \quad (3)$$

4 *T-Test*

The T-test is a statistical technique used to compare the average of two independent groups of data. The T-test is one of the commonly used methods in data analysis and experimentation, and can help determine whether or not the differences between two groups of data are statistically significant [14].

5 *ROC Curve*

The ROC (Receiver Operating Characteristic) curve is depicted in two dimensions, with True Positive (TP) levels plotted on the Y axis and False Positive (FP) levels plotted on the X axis. However, to determine a better classification, the AUC (Area Under the ROC Curve) calculation method is used which represents the area under the ROC curve. AUC is interpreted as probability [14].

AUC (Area Under the Curve) is an evaluation metric that measures the discriminatory performance of a model by estimating the output probability of a randomly selected sample of positive or negative populations [20]. The greater the AUC value, the stronger the classification capabilities possessed by the model. Since AUC is part of the area under the ROC curve, the AUC value is always between 0.0 and 1.0 [6]. The classification of AUC values is in the Table 1.

Table 1. *AUC value*

AUC value	*Classification*
0.90 - 1.00	Best
0.80 - 0.90	Good

0.70 - 0.80	Fair or Equal
0.60 - 0.70	Low
0.50 - 0.60	Fail

2.4. Naïve Bayes

Naïve Bayes is a classification algorithm based on probability theory and statistics [21]. This algorithm is capable of processing classification with probability and statistical methods based on Thomas Bayes theorem, which predicts future opportunities based on previously available historical data [22]. This method will calculate a set of probabilities by summing the frequencies and combinations of values from a given dataset [23].

2.5. Decision Tree

Decision tree is a predictive model used in data mining and machine learning to predict target values based on a set of rules and conditions [24]. This model is built like a tree with roots, branches, and leaves, where each branch and leaf represents a condition or decision [25].

2.6. Random Forest

Random Forest is an ensemble algorithm in data mining and machine learning used for classification and regression tasks [26]. This algorithm is a combination of several decision trees that are randomly generated and taken on average (forest).

2.7. K-Nearest Neighbor

K-Nearest Neighbor is a machine learning classification and regression algorithm that uses nearby labeled data to predict a class or target value from unlabeled data [27]. Then, predictions are made by voting the majority of the class or target value of the k-nearest neighbors.

2.8. Logistic Regression

Logistic Regression is a classification method used to predict categories or classes of data based on several input variables or attributes [28]. This method uses logistic regression models to model the probability that certain data will fit into a single class or category.

3. METODOLOGY

3.1. Research Methodology

Comparative method is a method by comparing a variable between two or more groups or samples. This method is used to compare differences or similarities between different groups or samples. The results of the evaluation of model performance from the algorithm are collected and analyzed, using relevant metrics such as accuracy, precision and recall. Then, the results of the analysis are interpreted to determine which algorithm has better performance in solving the problem of classifying car insurance claim data.

3.1.1. Knowledge Discovery in Database

Many data mining model processes that can be used include Knowledge Discovery Databases (KDD), Cross Industry Standard Process (CRISP-DM), and Sample, Explore, Modify, Model, Access (SEMMA). this research uses Knowledge Discovery Databases [29]. Data mining has many stages and techniques that can be implemented [30].

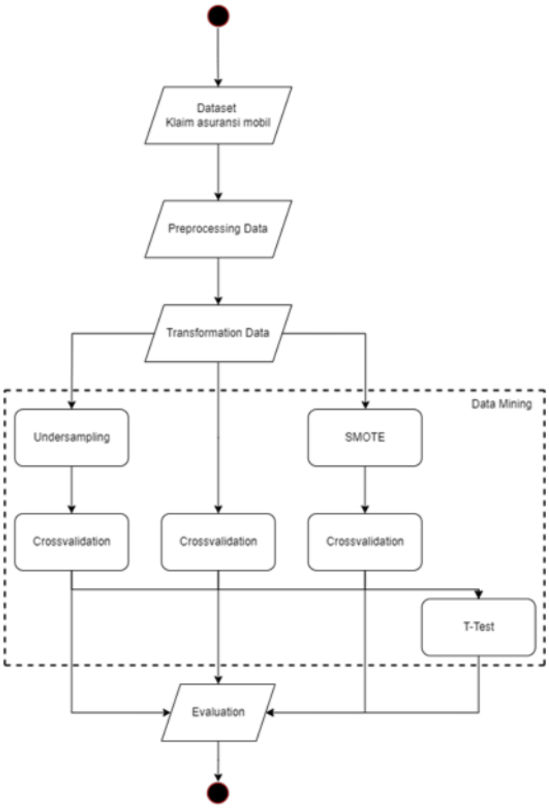


Fig 1. Research Flow

The KDD process is shown by Figure 1, involves steps such as data selection, data processing, data transformation, modeling and evaluation of the results in this study, namely:

1. Data Selection

The auto insurance claims dataset is selected from kaggle.com, an open-source site that provides a variety of data sets. The data was collected using Angoss Knowledge Seeker software from January 1994 to December 1996. There are 33 attributes in this dataset, and the class attribute indicates whether a claim is considered fraud or not. In total, there are 15,420 claim records in excel format.

2. Data Preprocessing

In this stage, clean the data from defects and objections, such as missing and duplicate data.

3. Data Transformation

At this stage, the data is transformed or transformed into a form that is easier for rapid miners to understand and process, such as changing the data format.

4. Data mining

At this stage, machine learning models are built to predict whether an insurance claim is fraudulent or not. Some machine learning algorithms that can be used, including Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbor and Logistic Regression.

5. Data Mining Evaluation

The model built is evaluated to find out how well it performs in predicting fraud labels. Using AUC and Matrix values that can be used, including accuracy, precision and recall.

3.1.2. Data Collection Techniques

Secondary data is data that has been collected by another party for other purposes and is available for use by another party for research or analysis purposes. In this study car insurance claim data was taken from kaggle.com, this data has 33 attributes and finally classification whether this is considered as fraud or not as a class attribute and contains 15,420 policy claim records and 33 features.

3.1.3. Population and Sample

The population in this study is all car insurance claim data that has a fraud label within a certain period of time with a total of 33 attributes. The sample in the study was a randomly selected amount of auto insurance claim data from a larger population. The sample is used to represent the characteristics and variability of overall auto insurance claims in that population.

3.1.4. Independent and Dependent Variables

Independent variables are variables that affect or influence the occurrence of fraud in car insurance claims. In this study, the independent variable is all attributes of car insurance claim data, while the dependent variable is the variable you want to research or explain, namely the occurrence of fraud.

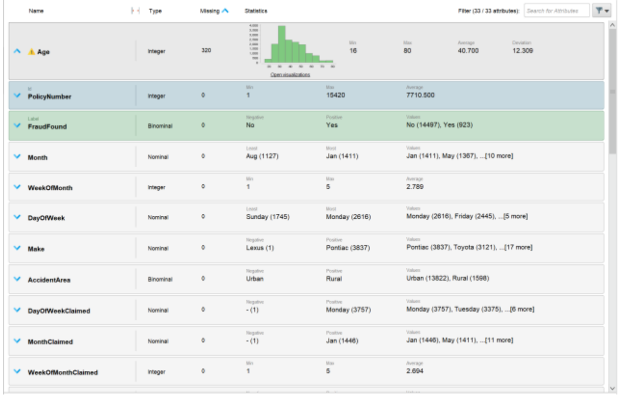
4. Results and Discussion

4.1. Data Selection

The data is taken from Kaggle which is an open-source site that has many datasets. This dataset was collected by Angoss Knowledge Seeker software from January 1994 to December 1996. This data has 33 attributes and is considered fraudulent or not as a class attribute and contains 15,420 claim records with excel format. Data that has 33 attributes has a missing value of 320 values in the age attribute is shown by Figure 2. To fill in the missing attribute values will be done at the Data Preprocessing stage.

4.2. Preprocessing Data

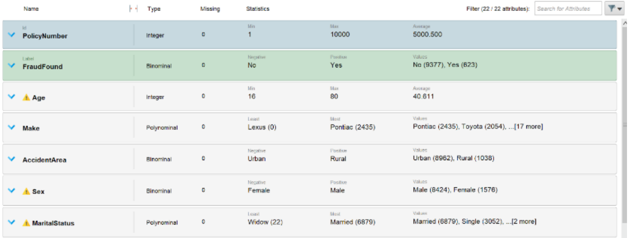
In the Data Preprocessing stage, the first stage is to remove duplicate data in the next dataset in the car insurance claim data there are 320 age attributes that have a value of 0 replaced with the average number in age so that there is no missing value in the dataset. The result of data preprocessing is shown by Figure 3.



Filter (33 / 33 attributes): Search for Attributes

Name	Type	Missing	Statistics			
Age	Integer	320	Min 16	Max 80	Average 40.700	Deviation 12.309
Id PolicyNumber	Integer	0	Min 1	Max 15420	Average 7710.500	
Label FraudFound	Binominal	0	Negative No	Positive Yes	Values No (14497), Yes (923)	
Month	Nominal	0	Least Aug (1127)	Most Jan (1411)	Values Jan (1411), May (1367), ...[10 more]	
WeekOfMonth	Integer	0	Min 1	Max 5	Average 2.789	
DayOfWeek	Nominal	0	Least Sunday (1745)	Most Monday (2616)	Values Monday (2616), Friday (2445), ...[5 more]	
Make	Nominal	0	Negative Lexus (1)	Positive Pontiac (3837)	Values Pontiac (3837), Toyota (3121), ...[17 more]	
AccidentArea	Binominal	0	Negative Urban	Positive Rural	Values Urban (13822), Rural (1598)	
DayOfWeekClaimed	Nominal	0	Negative - (1)	Positive Monday (3757)	Values Monday (3757), Tuesday (3375), ...[6 more]	
MonthClaimed	Nominal	0	Negative - (1)	Positive Jan (1446)	Values Jan (1446), May (1411), ...[11 more]	
WeekOfMonthClaimed	Integer	0	Min 1	Max 5	Average 2.694	

Fig 2. Car insurance claims dataset



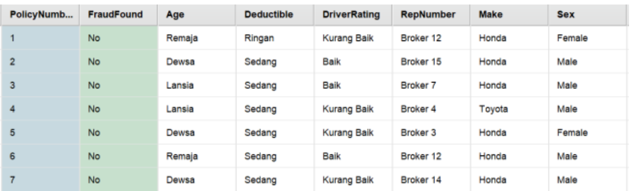
Filter (22 / 22 attributes): Search for Attributes

Name	Type	Missing	Statistics		
Id PolicyNumber	Integer	0	Min 1	Max 10000	Average 5000.500
Label FraudFound	Binominal	0	Negative No	Positive Yes	Values No (9377), Yes (623)
Age	Integer	0	Min 16	Max 80	Average 40.611
Make	Polynominal	0	Least Lexus (0)	Most Pontiac (2435)	Values Pontiac (2435), Toyota (2054), ...[17 more]
AccidentArea	Binominal	0	Negative Urban	Positive Rural	Values Urban (8962), Rural (1038)
Sex	Binominal	0	Negative Female	Positive Male	Values Male (8424), Female (1576)
MaritalStatus	Polynominal	0	Least Widow (22)	Most Married (6879)	Values Married (6879), Single (3052), ...[2 more]

Fig 3. Results of Data Preprocessing

4.3. Transformation Data

Data Transformation changes the values of the Age, Deductible, Driver Rating and RepNumber attributes with the generate operator. In the age attribute, it is divided into 3 age values, namely adolescents, adults and the elderly. Deductible attributes are divided into 3 values, namely light, medium and heavy. Attributes Driver Rating is divided into 2 values, namely good and less good and the Rep Number attribute is changed to broker 1 to broker 16.



PolicyNumb...	FraudFound	Age	Deductible	DriverRating	RepNumber	Make	Sex
1	No	Remaja	Ringan	Kurang Baik	Broker 12	Honda	Female
2	No	Dewsa	Sedang	Baik	Broker 15	Honda	Male
3	No	Lansia	Sedang	Baik	Broker 7	Honda	Male
4	No	Lansia	Sedang	Kurang Baik	Broker 4	Toyota	Male
5	No	Dewsa	Sedang	Kurang Baik	Broker 3	Honda	Female
6	No	Remaja	Sedang	Baik	Broker 12	Honda	Male
7	No	Dewsa	Sedang	Kurang Baik	Broker 14	Honda	Male

Fig 4. Results of Data Transformation

4.4. Data mining

4.4.1. Naïve Bayes

1 Naïve Bayes Algorithm Performance Without Sampling Technique

Table 2 and Figure 5 are the results of the performance of the Naïve Bayes algorithm in predicting fraud labels without sampling techniques resulting in an execution time of 0 seconds with recall no results of 97.26%, recall yes of 11.48%, precision no of 94.52% and precision yes of 21.07% with accuracy of 92.13% and AUC of 0.784. From these results show that the performance of the Naïve Bayes algorithm without using sampling techniques is able to detect and accurately predict class no but this model is not good at detecting and accurate in predicting class yes. From these results, the yes class is considered not optimal because the data is considered unbalanced so that it will be compared sampling techniques for the Naïve Bayes algorithm.

Table 2. Naïve Bayes Algorithm Performance Without Sampling Techniq**ue**

	True No	**True Yes**	**Class Precision**
Pred. No	14100	817	94.52%
Pred. Yes	397	106	21.07%
Class Recall	97.26%	11.48%	

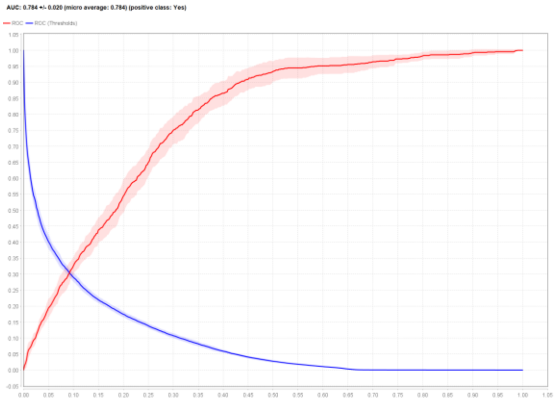


Fig 5. AUC Naïve Bayes algorithm without sampling technique

2 Naïve Bayes Algorithm Performance using Oversampling Technique

Table 3 and Figure 6 are the results of the performance of the Naïve Bayes algorithm using the SMOTE Oversampling Technique resulting in an execution time of 25 seconds with recall no results of 62.46%, recall yes of 89.51%, precision no of 85.62% and precision yes of 70.45% with an accuracy of 75.98% and AUC of 0.820. These results are considered to have the ability to distinguish between the two classes. From these results, it shows that the performance of the Naïve Bayes algorithm using the Oversampling technique is less capable of detecting class no but has a fairly good level of accuracy in predicting class no. In predicting yes class, this model has a fairly good value in detecting and its level of accuracy. From these results, recall no is considered not optimal and decreased from the technique without sampling, because the data is balanced so that the results show the lack of performance of the Naïve Bayes algorithm.

Table 3. Naïve Bayes Algorithm Performance using Oversampling Technique

	True No	**True Yes**	**Class Precision**
Pred. No	9055	1521	85.62%
Pred. Yes	5442	12976	70.45%
Class Recall	62.46%	89.51%	

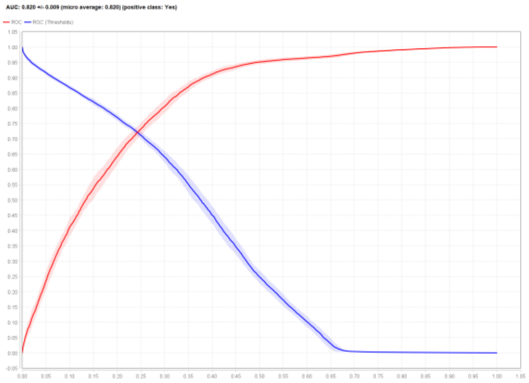


Fig 6. AUC Naïve Bayes Algorithm using Oversampling Technique

3 Naïve Bayes Algorithm Performance using Undersampling Technique

Table 4 and Figure 7 are the results of the performance of the Naïve Bayes algorithm using the Undersampling Technique resulting in an execution time of 0 seconds with the result of recall no value of 59.91%, recall yes of 86.02%, precision no of 81.09% and precision yes of 68.21% with an accuracy of 72.97% and AUC of 0.785. These results are considered to have little ability to distinguish between the two classes. From these results, it shows that the performance of the Naïve Bayes algorithm using undersampling techniques is less able to detect class no but has a very good level of accuracy in predicting class no. In predicting yes classes this model has a fairly good value in detecting and a very good value in its level of accuracy. From these results, recall no and precision yes are considered not optimal, because the data is balanced so that the results show the lack of performance of the Naïve Bayes algorithm.

Table 4. Naïve Bayes Algorithm Performance using Undersampling Technique

	True No	True Yes	Class Precision
Pred. No	553	129	94.52%
Pred. Yes	370	794	68.21%
Class Recall	59.91%	86.02%	

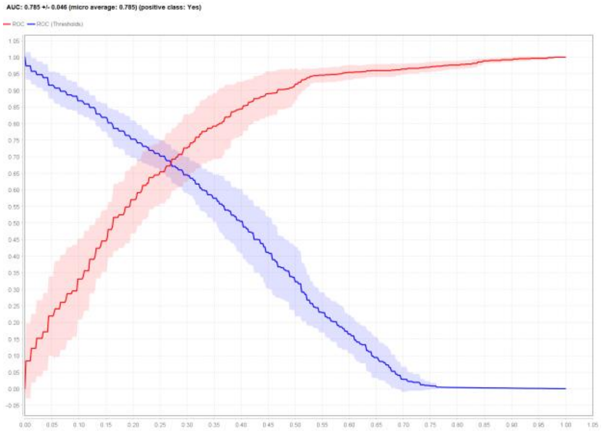


Fig 7. AUC Naïve Bayes Algorithm using UnderSampling Technique

4 *Evaluation of the Naïve Bayes Algorithm Model*

Table 5 is the result of the T-test resulting in the Naïve Bayes algorithm having a significant difference between using Oversampling and Undersampling techniques based on a probability of 0.05. The results of the comparison of Naïve Bayes using Oversampling and Undersampling techniques also have significant differences based on a probability of 0.05. Therefore, the significant difference in T-test results suggests that the use of Oversampling and Undersampling techniques in the Naïve Bayes algorithm has a significant influence on performance and classification results.

Table 5. Comparison Results of Naïve Bayes Algorithm T-test

	Naïve Bayes	NB Oversampling	NB Undersampling
Naïve Bayes		0.000	0.040
NB Oversampling			0.000
NB Undersampling			

After seeing significant results, compare the evaluation matrix to measure model performance. Table 6 is the result of prediction of the Naïve Bayes algorithm model in this study better by using the Oversampling technique which produces the highest detecting value and accuracy in the comparison of the Naïve Bayes algorithm model which means the Model with the Oversampling technique is the most optimal model in predicting fraud labels. The decline in accuracy of the Naïve Bayes algorithm shows the weakness of the dataset after the data is balanced that the attributes in the dataset are not related to each other or have no dependencies on each other.

Table 6. Performance Evaluation of Naïve Bayes Algorithm

Metric & AUC	Naïve Bayes	NB Oversampling	NB Undersampling
Accuracy	92.13%	75.98%	72.97%
AUC	0.784	0.820	0.785
Recall	11.49%	89.51%	86.03%
Recall Yes	11.48%	89.51%	86.02%
Recall No	97.26%	62.46%	59.91%
Precision	21.10%	70.46%	68.30%
Precision Yes	21.07%	70.45%	68.21%
Precision No	94.52%	85.62%	81.09%
Time	0 second	25 second	0 second

4.4.2. Decision Tree

1 *Decision Tree Without Sampling Technique*

Table 7 and Figure 8 are the results of the performance of the Decision Tree Algorithm without the Sampling technique resulting in an execution time of 0 seconds with recall no results of 100.00%, recall yes of 0.00%, precision no of 94.01% and precision yes of 0.00% with an accuracy of 94.01% and AUC of 0.500. These results are considered to have no ability to distinguish between the two classes, indicating that the model has the same performance as random prediction or a model that does not have predictive ability. From these results, the yes class is considered not optimal because the data is considered unbalanced so that it will be compared to sampling techniques for the Decision Tree algorithm.

Table 7. Decision Tree Algorithm Performance Without Sampling Technique

	True No	True Yes	Class Precision
Pred. No	14479	923	94.01%
Pred. Yes	0	0	0.00%
Class Recall	100.00%	0.00%	

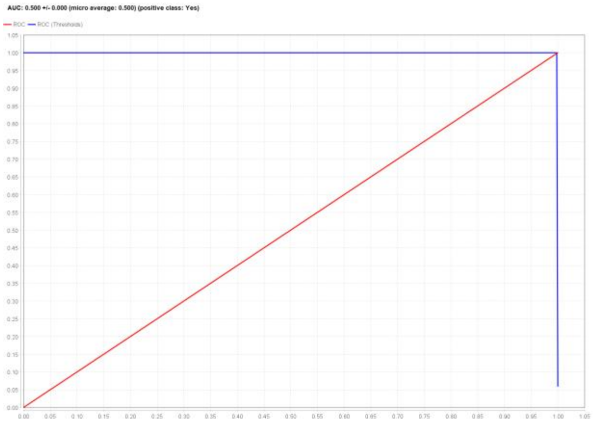


Fig 8. AUC Decision Tree Algorithm Performance Without Sampling Technique

2 Decision Tree Teknik Oversampling

Table 8 and Figure 9 are the results of the performance of the Decision Tree algorithm using the Oversampling technique resulting in an execution time of 28 seconds with recall no results of 61.45%, recall yes of 94.97%, precision no of 92.44% and precision yes of 71.13% with an accuracy of 78.21% and AUC of 0.811. From these results, it shows that the performance of the Decision Tree algorithm using the Oversampling technique is less able to detect class no but shows a very good accuracy value in predicting class no and this model is very good at detecting class yes with a fairly good accuracy value. From these results, recall no is considered not optimal and has decreased from the technique without sampling, because the data has been balanced so that the results show the lack of performance of the Decision Tree algorithm.

Table 8. Decision Tree Performance using Oversampling Technique

	True No	True Yes	Class Precision
Pred. No	8908	729	92.44%
Pred. Yes	5589	13768	71.13%
Class Recall	61.45%	94.97%	

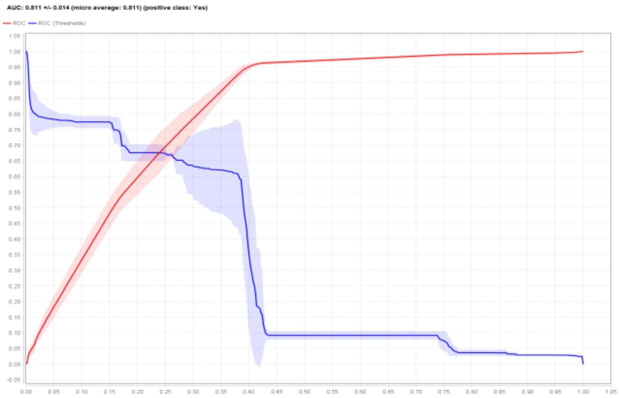


Fig 9. AUC Performance of Decision Tree Algorithm using Oversampling Technique

3 Decision Tree Undersampling Techniques

Table 9 and Figure 10 are the results of the performance of the Decision Tree algorithm using Undersampling techniques resulting in an execution time of 0 seconds with recall no results of 67.50%, recall yes of 78.66%, precision no of 75.98% and precision yes of 70.76% with an accuracy of 73.08% and AUC of 0.767. From these results, it shows that the performance of the Decision Tree algorithm using the Undersampling technique is less able to detect class no but shows a fairly good accuracy value in predicting class no and this model is quite good in detecting class yes with a fairly good accuracy value. From these results, recall no is considered not optimal, because the data is balanced so that the results show the lack of performance of the Decision Tree algorithm.

Table 9. Decision Tree Algorithm Performance using Undersampling Techniques

	True No	True Yes	Class Precision
Pred. No	623	197	75.98%
Pred. Yes	300	726	70.76%
Class Recall	67.50%	78.66%	

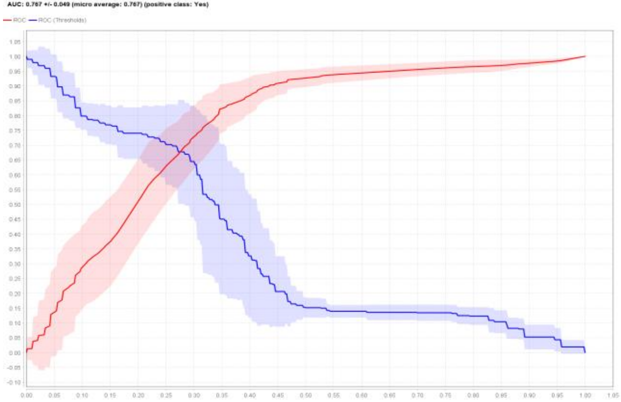


Fig 10. AUC Decision Tree using Undersampling Technique

4 Evaluasi Model Algoritma Decision Tree

Table 10 is the result of the T-test resulting in the Decision Tree algorithm has a significant difference between using Oversampling and Undersampling techniques based on a probability of 0.05. The results of the Decision Tree comparison using Oversampling and Undersampling techniques also have significant differences based on a probability of 0.05. Therefore, the significant difference in T-test results shows that the use of Oversampling and Undersampling techniques in the Decision Tree algorithm has a significant influence on performance and classification results.

Table 10. Decision Tree Algorithm T-test Comparison Results

	Decision Tree	DT Oversampling	DT Undersampling
DecisionTree	-	0.000	0.001
DT Oversampling	-	-	0.000
DT Undersampling	-	-	-

Table 11 is the result of the prediction metric of the Decision Tree model in this study better by using the Oversampling Technique which produces the highest detecting value and accuracy in the comparison of Naïve Bayes algorithm models which means the Model with Oversampling technique is the most optimal model in predicting fraud labels Then supported by the highest AUC value in the comparison of classification models which means the Model with Oversampling Techniques goes into good category. The accuracy results produced in this experiment resulted in a decrease compared to without sampling techniques, but the AUC results produced were more optimal than without sampling techniques so that using Oversampling and Undersampling techniques entered into fair or equal.

Table 11. Decision Tree Algorithm Performance Evaluation

Matrix & AUC	Decision Tree	DT Oversampling	DT Undersampling
Accuracy	94.01%	78.21%	73.08%
AUC	0.500	0.811	0.767
Recall	0.00%	71.21%	70.74%
Recall Yes	0.00%	94.97%	78.66%
Recall No	100.00%	61.45%	67.50%
Precision	unknown	94.97%	78.45%
Precision Yes	0.00%	71.13%	70.76%
Precision No	94.01%	92.44%	75.98%
Time	0 second	28 second	0 second

4.4.3. Random Forest

1 Random Forest Without Sampling Technique

Table 12 and Figure 11 are the results of the performance of the Random Forest algorithm without sampling techniques resulting in an execution time of 7 seconds with recall no results of 100.00%, recall yes of 0.11%, precision no of 94.02% and precision yes of 100.00% with accuracy of 94.02% and AUC of 0.681. These results are considered to have no ability to distinguish between the two classes, indicating that the model has the same performance as random prediction or a model that does not have predictive ability. From these results, recall yes is considered not optimal because the data is considered unbalanced so that it will be compared sampling techniques for the Random Forest algorithm.

Table 12. Performance of Random Forest Algorithm Without Sampling Technique

	True No	True Yes	Class Precision
Pred. No	14497	922	94.02%
Pred. Yes	0	1	100.00%
Class Recall	100.00%	0.11%	

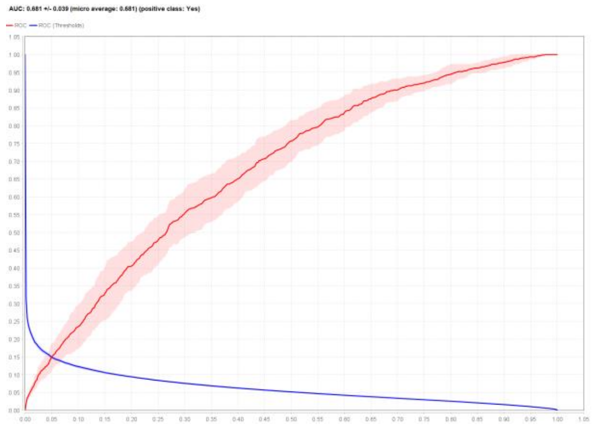


Fig 11. AUC Performance of Random Forest Algorithm Without Sampling Technique

2 Random Forest Teknik Oversampling

Table 13 and Figure 12, the results of the performance of the Random Forest algorithm using the Oversampling technique resulted in an execution time of 40 seconds with recall no results of 83.28%, recall yes of 97.25%, precision no of 96.80% and precision yes of 85.33% with an accuracy of 90.26% and AUC of 0.980. From these results, it shows that the performance of the Random Forest algorithm using the Oversampling technique is able to detect class no well and shows a very good accuracy value in predicting class no and this model is very good in detecting class yes with a good accuracy value.

Table 13. Performance of Random Forest Algorithm using Oversampling Technique

	True No	True Yes	Class Precision
Pred. No	12073	399	96.80%
Pred. Yes	2424	14098	85.33%
Class Recall	83.28%	97.25%	

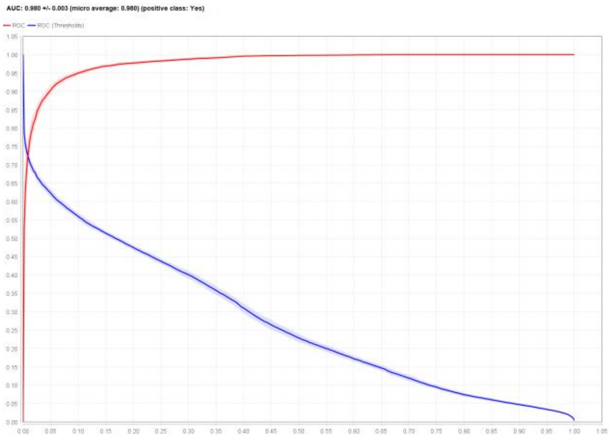


Fig 12. AUC Performance of Random Forest Algorithm using Oversampling Technique

3 *Random Forest Undersampling Techniques*

Table 14 and Figure 13 are the results of the performance of the Random Forest algorithm using Undersampling techniques resulting in an execution time of 1 second with recall no results of 64.03%, recall yes of 83.97%, precision no of 79.97% and precision yes of 70.01% with an accuracy of 74.00% and AUC of 0.789. From these results, it shows that the performance of the Random Forest algorithm using the Oversampling technique is less able to detect class no but shows a fairly good accuracy value in predicting class no and this model is good at detecting class yes with a fairly good accuracy value. From these results, recall no is considered not optimal, because the data is balanced so that the results show the lack of performance of the Random Forest algorithm.

Table 14. Performance of Random Forest Algorithm using Undersampling Technique

	True No	True Yes	Class Precision
Pred. No	591	148	79.97%
Pred. Yes	332	775	70.01%
Class Recall	64.03%	83.97%	

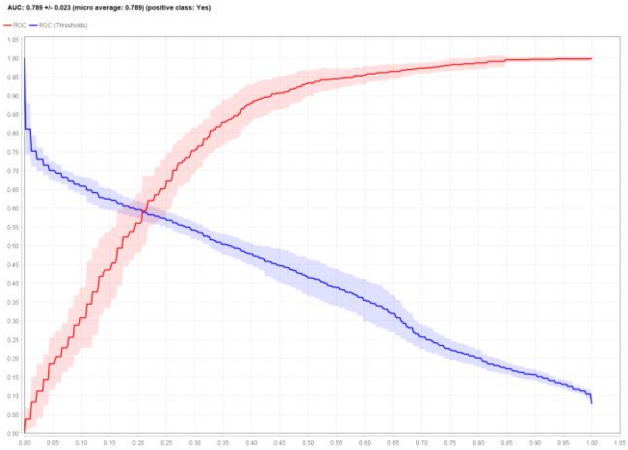


Fig 13. AUC Random Forest Algorithm using Undersampling Technique

4 *Evaluation of the Random Forest Algorithm Model*

Table 15 is the result of the T-test resulting in the Random Forest algorithm has a significant difference between using Oversampling and Undersampling techniques based on a probability of 0.05. The results of the Random Forest comparison using Oversampling and Undersampling techniques also have significant differences based on a probability of 0.05. Therefore, the significant difference in T-test results shows that the use of Oversampling and Undersampling techniques in the Random Forest algorithm has a significant influence on performance and classification results.

Table 15. Comparison Results of T-test Random Forest Algorithm

	Random Forest	RF Oversampling	RF Undersampling
Random Forest	-	0.000	0.000
RF Oversampling	-	-	0.000
RF Undersampling	-	-	

Table 16 is the result of the prediction metric of the Random Forest algorithm model in this study is better by using the Oversampling Technique which produces the highest detecting value and accuracy in the comparison of the Random Forest algorithm model which means the Model with Oversampling technique is the most optimal model in predicting fraud labels Then supported by the highest AUC value in the comparison of classification models which means Model with Technique Oversampling falls into the excellent category. The accuracy results produced in this experiment resulted in a decrease compared to without sampling techniques, but the AUC results produced were more optimal than without sampling techniques so that using Oversampling and Undersampling techniques entered the best and fairest or the same.

Table 16. Random Forest Algorithm Model Performance

Matrix & AUC	Random Forest	RF Oversampling	RF Undersampling
Accuracy	94.02%	90.26%	74.00%
AUC	0.681	0.980	0.789
Recall	0.11%	97.25%	83.96%
Recall Yes	0.11%	97.25%	83.97%
Recall No	100.00%	83.28%	64.03%
Precision	100.00%	85.33%	85.33%
Precision Yes	100.00%	85.33%	70.12%
Precision No	94.02%	96.80%	79.97%
Time	7 second	40 second	1 second

4.4.4. K-Nearest Neighbor

1 *K-Nearest Neighbor Without Sampling Technique*

Table 17 and Figure 14, the results of K-Nearest Neighbor performance without sampling techniques resulted in an execution time of 37 seconds with recall no results of 99.91%, recall yes of 1.41%, precision no of 94.09% and precision yes of 50.00% with accuracy of 94.01% and AUC of 0.676. These results are considered to have no ability to distinguish between the two classes, indicating that the model has the same performance as random prediction or a model that does not have predictive ability. From these results, recall yes and precision yes are considered not optimal because the data is considered unbalanced so that sampling techniques for the K-Nearest Neighbor algorithm will be compared.

Table 17. Performance of K-Nearest Neighbor Algorithm Without Sampling Technique

	True No	True Yes	Class Precision
Pred. No	14484	910	94.09%
Pred. Yes	13	13	50.00%
Class Recall	99.91%	1.41%	

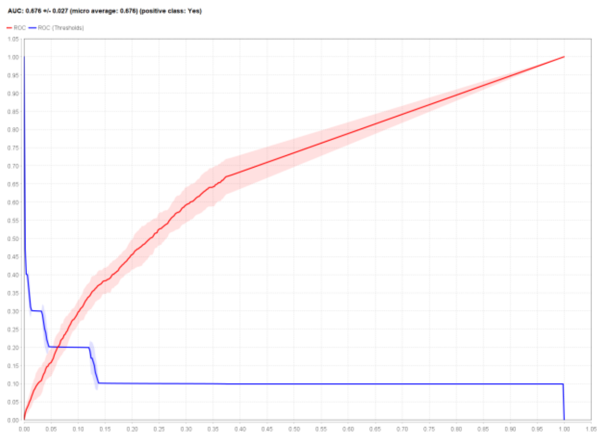


Fig 14. AUC Performance of K-Nearest Neighbor Algorithm Without Sampling Technique

2 *K-Nearest Neighbor Oversampling Technique*

Table 18 and Figure 15, the results of the performance of the K-Nearest Neighbor algorithm using the Oversampling technique resulted in an execution time of 2:35 minutes with recall no results of 81.88%, recall yes of 99.03%, precision no of 98.83% and precision yes of 84.53% with an accuracy of 90.46% and AUC of 0.984. From these results, it shows that the performance of the K-Nearest Neighbor algorithm using the Oversampling technique is able to detect class no well and shows a very good accuracy value in predicting class no and this model is very good in detecting class yes with a good accuracy value.

Table 18. Performance of K-Nearest Neighbor Algorithm using Oversampling Technique

	True No	True Yes	Class Precision
Pred. No	11870	140	98.83%
Pred. Yes	2627	14357	84.53%
Class Recall	81.88%	99.03%	

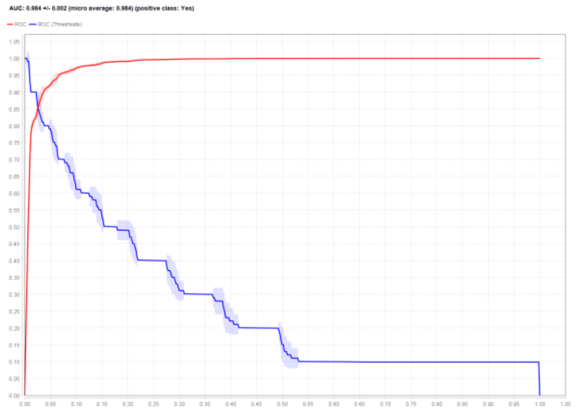


Fig 15. AUC Performance of K-Nearest Neighbor Algorithm using Oversampling Technique

3 K-Nearest Neighbor Undersampling Technique

Table 19 and Figure 16, the results of the performance of the K-Nearest Neighbor algorithm using the Undersampling technique resulted in an execution time of 0 seconds with recall no results of 60.56%, recall yes of 76.71%, precision no of 72.22% and precision yes of 66.04% with an accuracy of 68.64% and AUC of 0.758. From these results, it shows that the performance of the K-Nearest Neighbor algorithm using the Undersampling technique is not good in detecting class no and shows a fairly good accuracy value in predicting class no and this model is quite good at detecting class yes with a poor accuracy value. From these results, recall no and precision yes are considered not optimal and have decreased from the technique without sampling, because the data has been balanced so that the results show the lack of performance of the K-Nearest Neighbor algorithm.

Table 19. Performance of K-Nearest Neighbor Algorithm using Undersampling Technique

	True No	**True Yes**	**Class Precision**
Pred. No	559	216	72.2%
Pred. Yes	364	708	66.04%
Class Recall	60.56%	76.71%	

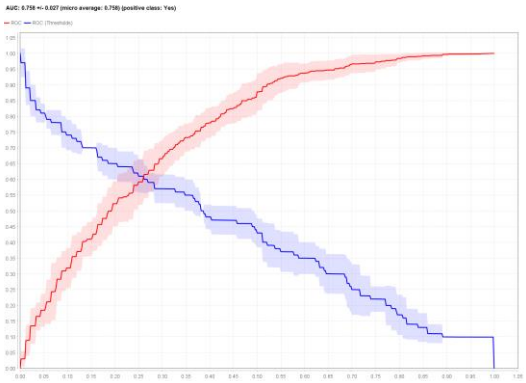


Figure 16. AUC K-Nearest Neighbor Using Undersampling Technique

4 Evaluation of the K-Nearest Neighbor Algorithm Model

Table 20 is the result of the T-test resulting in the K-Nearest Neighbor algorithm has a significant difference between using Oversampling and Undersampling techniques based on a probability of 0.05. The results of the K-Nearest Neighbor comparison using Oversampling and Undersampling techniques also have a significant difference based on a probability of 0.05. Therefore, the significant difference in T-test results shows that the use of Oversampling and Undersampling techniques in the K-Nearest Neighbor algorithm has a significant influence on performance and classification results.

Table 20. Comparison Results of K-Nearest Neighbor Algorithm T-test

	K-NN	**K-NN Oversampling**	**K-NN Undersampling**
K-NN	-	0.000	0.000
K-NN Oversampling	-	-	0.000
K-NN Undersampling	-	-	

Table 21 is the result of the prediction matrix of the K-Nearest Neighbor model in this study better by using the Oversampling Technique which produces the highest detecting value and accuracy in the comparison of the K-Nearest Neighbor algorithm model which means the Model with Oversampling technique is the most optimal model in predicting fraud labels Then supported by the highest AUC value in the comparison of classification models which means Model with Technique Oversampling falls into the excellent category. The accuracy results produced in Oversampling and Undersampling techniques produce a decrease compared to without sampling techniques, but the AUC results produced are more optimal than without sampling techniques so that using Oversampling and

Undersampling techniques falls into the best and fairest or the same category.

Table 21. Performance Evaluation of K-Nearest Neighbor Algorithm Model

Matrix & AUC	K-NN	K-NN Oversampling	K-NN Undersampling
Accuracy	94.01%	90.46%	68.64%
AUC	0.676	0.984	0.758
Recall	94.01%	99.03%	74.87%
Recall Yes	1.41%	99.03%	76.71%
Recall No	99.91%	81.88%	60.56%
Precision	44.33%	84.54%	63.68%
Precision Yes	50.00%	84.53%	66.04%
Precision No	94.09%	98.83%	72.22%
Time	37 second	2:35 minute	0 second

4.4.5. Logistic Regression

1 *Logistic Regression Without Sampling Technique*

Table 22 and Figure 17 the results of Logistic Regression performance without sampling techniques resulted in an execution time of 4 seconds with Recall no results of 99.82%, Recall yes of 1.19%, precision no of 94.07% and precision yes of 28.73% with an accuracy of 93.92% and AUC of 0.791. These results are considered to have no ability to distinguish between the two classes, indicating that the model has the same performance as random prediction or a model that does not have predictive ability. From these results, recall yes and precision yes are considered not optimal because the data is considered unbalanced so that sampling techniques for the Logistic Regression algorithm will be compared.

Table 22. Performance of Logistic Regression Algorithm Without Sampling Technique

	True No	True Yes	Class Precision
Pred. No	14471	912	94.07%
Pred. Yes	26	11	29.73%
Class Recall	99.82%	1.19%	

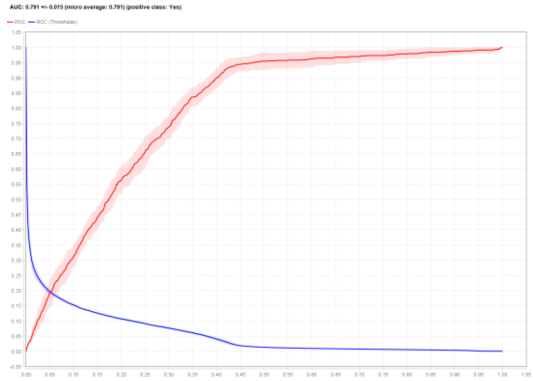


Fig 17. AUC Logistic Regression Performance Without Sampling Technique

2 *Logistic Regression Oversampling Technique*

Table 23 and Figure 18 are the results of the performance of the Logistic Regression algorithm using the Oversampling technique resulting in an execution time of 33 seconds with the results of Recall no of 65.12%, Recall yes of 89.58%, precision no of 86.21% and precision yes of 71.98% with an accuracy of 77.35% and AUC of 0.834. From these results, it shows that the performance of the Logistic Regression algorithm using the Oversampling technique is not good in detecting class no and shows a good accuracy value in predicting class no and this model is good at detecting class yes with a fairly good accuracy value. From these results, recall no is considered not optimal and has decreased from the technique without sampling, because the data is balanced so that the results show the lack of performance of the Logistic Regression algorithm.

Table 23. Performance of Logistic Regression Algorithm using Oversampling Technique

	True No	True Yes	Class Precision
Pred. No	9441	1510	86.21%
Pred. Yes	5056	12987	71.98%
Class Recall	65.12%	89.58%	

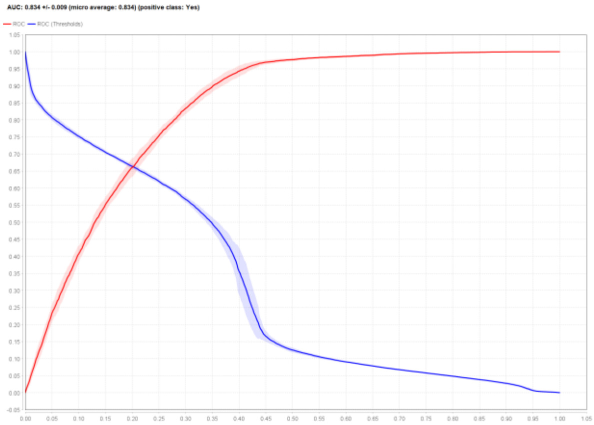


Fig 18. AUC Performance of Logistic Regression Algorithm using Oversampling Technique

3 *Logistic Regression Undersampling Technique*

Table 24 and Figure 19, the results of the performance of the Logistic Regression algorithm using Undersampling techniques resulted in an execution time of 0 seconds with recall no results of 64.46%, ecall yes of 83.86%, precision no of 79.97% and precision yes of 70.24% with an accuracy of 74.16% and AUC of 0.789. From these results, it shows that the performance of the Logistic Regression algorithm using Undersampling techniques is not good at detecting class no and shows a fairly good accuracy value in predicting class no and this model is good at detecting class yes with a fairly good accuracy value. From these results, recall no is considered not optimal and has decreased from the technique without sampling, because the data is balanced so that the results show the lack of performance of the Logistic Regression algorithm.

Table 24. Performance of Logistic Regression Algorithm using Undersampling Technique

	True No	True Yes	Class Precision
Pred. No	595	149	79.97%
Pred. Yes	328	774	70.24%
Class Recall	64.46%	83.86%	

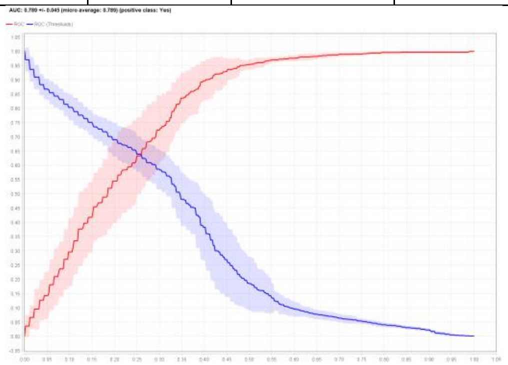


Fig 19. AUC Logistic Regression using Undersampling Technique

4 *Evaluation of the Logistic Regression Algorithm Model*

Table 25 is the result of the T-test resulting in a Logistic Regression algorithm has a significant difference between using Oversampling and Undersampling techniques based on a probability of 0.05. The results of the comparison of Logistic Regression using Oversampling and Undersampling techniques also have significant differences based on a probability of 0.05. Therefore, the significant difference in T-test results shows that the use of Oversampling and Undersampling techniques in the Logistic Regression algorithm has a significant influence on performance and classification results.

Table 25. Comparison Results of T-test Logistic Regression Algorithm

	Logistic Regression	LR Oversampling	LR Undersampling
Logistic Regression		0.000	0.007
LR Oversampling			0.000
LR Undersampling			

Table 28 is the result of the prediction matrix of the Logistic Regression model in this study better by using the Oversampling Technique which produces the highest detection value and accuracy in the comparison of the Logistic Regression algorithm model which means the Model with Oversampling technique is the most optimal model in predicting fraud labels Then supported by the highest AUC value in the comparison of classification models which means Model with Technique Oversampling falls into the good category. The accuracy results produced in Oversampling and Undersampling techniques result in a decrease compared to without sampling techniques, but the AUC results produced are more optimal than without sampling techniques so that using Oversampling techniques falls into the fair or equal category.

Table 26. Performance Evaluation of Logistic Regression Algorithm Model

Matrix & AUC	LR	LR Oversampling	LR Undersampling
Accuracy	93.92%	77.35%	74.16%
AUC	0.791	0. 834	0.789
Recall	1.19%	89.58%	83.86%
Recall Yes	1.19%	89.58%	83.86%
Recall No	99.82%	65.12%	64.46%
Precision	39.43%	71.98%	70.35%
Precision Yes	28.73%	71.98%	70.24%
Precision No	94.07%	86.21%	79.97%
Time	4 second	33 second	0 second

4.4.6. Machine Learning Algorithm Evaluation

Table 27 is the evaluation result it can be concluded that the Random Forest (RF) algorithm and K-Nearest Neighbors (K-NN) algorithm with Oversampling models have excellent performance in predicting fraud cases, with high accuracy, AUC, recall, and precision. The Decision Tree (DT) algorithm and the Logistic Regression (LR) algorithm also provide good results in several evaluation metrics. However, the Naïve Bayes (NB) algorithm has relatively lower performance compared to other algorithms.

Table 27. Machine learning Algorithm Model Performance Evaluation

Matrix & AUC	NB OS	DT OS	RF OS	K-NN OS	LR OS
Accuracy	75.98%	78.21%	90.26%	90.46%	77.35%
AUC	0.820	0.811	0.980	0.984	0.834
Recall	89.51%	94.97%	97.25%	99.03%	89.58%
Recall Yes	89.51%	94.97%	97.25%	99.03%	89.58%
Recall No	62.46%	61.45%	83.28%	81.88%	65.12%
Precision	70.46%	71.21%	85.33%	84.54%	71.98%
Precision Yes	70.45%	71.13%	85.33%	84.53%	71.98%
Precision No	85.62%	92.44%	96.80%	98.83%	86.21%
Time	25 second	28 second	40 second	2:35 Minute	33 second

The selection of the best algorithm for predicting fraud cases needs to be considered based on preferences, specific needs. However, in the context of predicting fraud cases, the most accurate and superior algorithm from the performance table results is the K-Nearest Neighbors (K-NN) algorithm because based on the results of the K-NN matrix it is more optimal than the Random Forest (RF) algorithm even though it has the longest execution time.

5. Conclusion

Conclusion of Machine learning Algorithm Comparison Research Using Car Insurance Claim Data Based on this study. The performance of Decision Tree, K-Nearest Neighbor, Naïve Bayes, Random Forest, and Logistic Regression algorithms has been compared in predicting insurance claims. Performance evaluation is performed using evaluation matrices such as accuracy, AUC, recall, and precision. Based on the evaluation results, algorithms that provide good performance in predicting insurance claims can be identified the K-Nearest Neighbor algorithm and the Random Forest algorithm with the results of evaluating the accuracy of the K-Nearest Neighbor algorithm (90.46%), recall Yes (99.03%), recall No (81.88%), precision Yes (84.53), Precision No (98.83%) and AUC (0.984) then the Random Forest algorithm (90.26%), recall Yes (97.25%), recall No (83.28%), precision Yes (85.33%), Precision No (96.80%) and AUC (0.980), which can be used for better decision making in the vehicle insurance industry.

6. Acknowledgments

The author wishes to express gratitude to Universitas Multimedia Nusantara for providing funds and support for this research.

References

[1] A. A. Firdaus and A. Komarudin, “Klasifikasi Pemegang Polis Menggunakan Metode XGBoost,” *Pros. Stat.*, vol. 7, no. 2, pp. 704–710, 2021, [Online]. Available: http://dx.doi.org/10.29313/.v0i0.30320

[2] R. Ramlah, “Penerapan Ganti Rugi Asuransi Mobil Pada Kasus Kecelakaan Dan Pencurian PT. Asuransi Tri Pakarta,” *Optim. J. Ekon. dan Manaj.*, vol. 2, no. 2, pp. 223–232, 2022, doi: 10.55606/optimal.v2i2.643.

[3] I. I. Information, “Facts + Statistics: Auto insurance.”

[Online]. Available: https://www.iii.org/fact-statistic/facts-statistics-auto-insurance

[4] F. D. Astuti and F. N. Lenti, “Implementasi SMOTE untuk mengatasi Imbalance Class pada Klasifikasi Car Evolution menggunakan K-NN,” *JUPITER (Jurnal Penelit. Ilmu dan Teknol. Komputer)*, vol. 13, no. 1, pp. 89–98, 2021.

[5] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. S. Hacid, and H. Zeineddine, “An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection,” *IEEE Access*, vol. 7, pp. 93010–93022, 2019, doi: 10.1109/ACCESS.2019.2927266.

[6] A. P. Ayudhitama and U. Pujianto, “JIP (Jurnal Informatika Polinema) ANALISA 4 ALGORITMA DALAM KLASIFIKASI PENYAKIT LIVER MENGGUNAKAN RAPIDMINER”.

[7] A. I. Alrais, “Fraudulent Insurance Claims Detection Using Machine Learning by A Capstone Submitted in Partial Fulfilment of the Requirements for the,” 2022.

[8] I. Kurniawan, D. C. P. Buani, A. Abdussomad, W. Apriliah, and E. Fitriani, “Penerapan Teknik Random Undersampling untuk Mengatasi Imbalance Class dalam Prediksi Kebakaran Hutan Menggunakan Algoritma Decision Tree,” *Acad. J. Comput. Sci. Res.*, vol. 5, no. 1, p. 1, 2023, doi: 10.38101/ajcsr.v5i1.617.

[9] A. Syukron and A. Subekti, “Penerapan Metode Random Over-Under Sampling dan Random Forest Untuk Klasifikasi Penilaian Kredit,” *J. Inform.*, vol. 5, no. 2, pp. 175–185, 2018, doi: 10.31311/ji.v5i2.4158.

[10] L. Fadilah, *Klasifikasi Random Forest pada Data Imbalanced Program Studi Matematika Universitas Islam Negeri Syarif Hidayatullah 2018 / 1439 H Klasifikasi Random Forest*. 2018.

[11] D. P. Utomo and M. Mesran, “Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung,” *J. MEDIA Inform. BUDIDARMA*, vol. 4, no. 2, p. 437, Apr. 2020, doi: 10.30865/mib.v4i2.2080.

[12] F. Gorunescu, *Data mining: concepts, models and techniques*, vol. 21, no. 1. 2011. [Online]. Available: http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203

[13] A. Saputra and Suharjito, “Fraud detection using machine learning in e-commerce,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 9, pp. 332–339, 2019, doi: 10.14569/ijacsa.2019.0100943.

[14] P. Agustia Rahayuningsih, R. Maulana, P. Studi Komputerisasi Akuntansi, A. BSI Pontianak, and J. Abdurahman Saleh, “Analisis Perbandingan Algoritma Klasifikasi Data Mining Untuk Dataset Blogger Dengan Rapid Miner,” vol. VI, no. 1, 2018.

[15] D. Y. Mohammed, “Detection of Vehicle Insurance Claim Fraud: A Fraud Detection Use-Case for the Vehicle Insurance Industry,” *Int. J. Progress. Sci. …*, vol. 30, no. March, pp. 504–507, 2021, [Online]. Available: http://ijpsat.es/index.php/ijpsat/article/view/3919%0A http://ijpsat.es/index.php/ijpsat/article/download/3919/2405

[16] F. D. Pramakrisna, F. D. Adhinata, and N. A. F. Tanjung, “Aplikasi Klasifikasi SMS Berbasis Web Menggunakan Algoritma Logistic Regression,” *Teknika*, vol. 11, no. 2, pp. 90–97, 2022, doi: 10.34148/teknika.v11i2.466.

[17] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, “Comparing different supervised machine learning algorithms for disease prediction,” *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, Dec. 2019, doi: 10.1186/s12911-019-1004-8.

[18] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, and A. Saboor, “Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare,” *IEEE Access*, vol. 8, no. Ml, pp. 107562–107582, 2020, doi: 10.1109/ACCESS.2020.3001149.

[19] A. Maghfiroh, Y. Findawati, and U. Indahyanti, “Klasifikasi Penipuan pada Rekening Bank menggunakan Pendekatan Ensemble Learning,” vol. 4, no. 4, pp. 1883–1891, 2023, doi: 10.47065/bits.v4i4.3212.

[20] D. A. Kristiyanti, “Analisis sentimen review produk kosmetik melalui komparasi feature selection,” *Konf. Nas. ilmu Pengetah. dan Teknol.*, vol. 2, no. 2, pp. 74–81, 2015.

[21] R. Indrayani, “ANALISA PERBANDINGAN ALGORITME NAÏVE BAYES DAN DECISION TREE PADA KLASIFIKASI DATA TRANSFUSI DARAH,” 2018.

[22] D. Retno Utari and A. Wibowo, “Pemodelan Prediksi Status Keberlanjutan Polis Asuransi Kendaraan dengan Teknik Pemilihan Mayoritas Menggunakan Algoritma-Algoritma Klasifikasi Data Mining,” *Pros. Semin. Nas. Teknoka*, vol. 5, no. 2502, pp. 19–24, 2020, doi: 10.22236/teknoka.v5i.391.

[23] R. N. Devita, H. W. Herwanto, and A. P. Wibawa, “Perbandingan Kinerja Metode Naive Bayes dan K-Nearest Neighbor untuk Klasifikasi Artikel Berbahasa indonesia,” *J. Teknol. Inf. dan Ilmu*

Komput., vol. 5, no. 4, p. 427, Oct. 2018, doi: 10.25126/jtiik.201854773.

[24] S. Panigrahi and B. Palkar, "Comparative Analysis on Classification Algorithms of Auto-Insurance Fraud Detection based on Feature Selection Algorithms," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 9, pp. 72–77, 2018, doi: 10.26438/ijcse/v6i9.7277.

[25] A. Rohman and A. Rufiyanto, "Implementasi Data Mining Dengan Algoritma Decision Tree C4 . 5 Untuk Prediksi Kelulusan Mahasiswa Di Universitas Pandaran," *Proceeding SINTAK 2019*, pp. 134–139, 2019.

[26] A. I. Lubis, U. Erdiansyah, and R. Siregar, "Komparasi Akurasi pada Naive Bayes dan Random Forest dalam Klasifikasi Penyakit Liver," *J. Comput. Eng. Syst. Sci.*, vol. 7, no. 1, pp. 81–89, 2022.

[27] J. Alzubi, A. Nayyar, and A. Kumar, "Machine Learning from Theory to Algorithms: An Overview," *J. Phys. Conf. Ser.*, vol. 1142, no. 1, 2018, doi: 10.1088/1742-6596/1142/1/012012.

[28] F. Handayani *et al.*, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Komparasi Support Vector Machine, Logistic Regression Dan Artificial Neural Network dalam Prediksi Penyakit Jantung," vol. 7, no. 3, pp. 329–334, 2021.

[29] Y. A. Suwitono and F. J. Kaunang, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Daun Dengan Metode Data Mining SEMMA Menggunakan Keras," *J. Komtika (Komputasi dan Inform.*, vol. 6, no. 2, pp. 109–121, 2022, doi: 10.31603/komtika.v6i2.8054.

[30] Y. I. Kurniawan, "Perbandingan Algoritma Naive Bayes dan C.45 dalam Klasifikasi Data Mining," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, p. 455, Oct. 2018, doi: 10.25126/jtiik.201854803.