Comparison of the K-Nearest Neighbor (K-NN) and C4.5 Algorithms for Predicting the Graduation Status of Indonesia Smart Card (KIP) Scholarship Recipients at AMIK Medicom Medan

Meiarni Situkkir, Khairul, Zulham Sitorus, Muhammad Iqbal, Muhammad Syahputra Novelan

Abstract

The increasing number of students receiving the Smart Indonesia Card (KIP) Lecture requires universities to develop an early prediction system in monitoring student graduation status. This study aims to compare the performance of the K-Nearest Neighbor (K-NN) and C4.5 algorithms in predicting the graduation status of KIP Lecture recipients at AMIK Medicom Medan. The study used a comparative descriptive approach with a dataset of 310 students, where the Achievement Index (IP) from semester 1 to semester 4 was used as a predictor variable, while graduation status (Graduated or Drop Out) was used as the target class. The research stage includes data preprocessing consisting of data cleaning, coding, normalization, and data sharing. The K-NN model is optimized using 10-fold cross-validation to determine the best K-value, while the C4.5 model is built using a decision tree based on entropy criteria. Model performance was evaluated using confusion matrix, accuracy, precision, recall, and F1-score. The results showed that the K-NN algorithm with a value of K = 3 produced the best classification performance with an accuracy of 99.03%, a macro F1-score of 0.96, and a weighted F1-score of 0.99. Meanwhile, the C4.5 algorithm also showed excellent performance with an accuracy of 98.71%, a macro value of F1-score of 0.94, and a weighted F1-score of 0.99. Although K-NN results in slightly higher prediction performance, the C4.5 algorithm has an advantage in terms of interpretability through an explicit decision tree structure making it more suitable to support academic decision-making. These findings show that both algorithms are highly effective in predicting student graduation status, with K-NN excelling in terms of prediction accuracy, while C4.5 is superior in providing transparency for decision support systems in the education sector.

***Keywords**: K-Nearest Neighbor, C4.5, graduation prediction, KIP Lectures, machine learning, data mining education.*

[1] Meiarni Situkkir
[1] Master of Information Technology, Universitas Pembangunan Panca Budi, Indonesia
e-mail: situngkirmeiarni3@gmail.com

[2,3,4,5] Khairul, Zulham Sitorus, Muhammad Iqbal, Muhammad Syahputra Novelan
[2,3,4,5] Master of Information Technology, Universitas Pembangunan Panca Budi, Indonesia
e-mail:khairul@dosen.pancabudi.ac.id, zulham@dosen.pancabudi.ac.id , wakbalpb@yahoo.co.id, putranovelan@dosen.pancabudi.ac.id
3rd International Cofference on the Epicentrum of Economic Global Framework (ICEEGLOF)
Theme: Navigating The Future: Business and Social Paradigms in a Transformative Era.
https://proceeding.pancabudi.ac.id/index.php/ICEEGLOF

Introduction

The development of information technology has had a major impact on the world of education, especially in data-based decision-making. One of the rapidly evolving technologies is machine learning, which allows systems to learn from data and make decisions automatically with minimal human intervention. One of its relevant applications is in predicting the graduation status of students, especially for recipients of educational assistance such as the Smart Indonesia Card (KIP) for College. The Smart Indonesia Card (KIP) Lecture scholarship program provides tuition support to underprivileged students to receive education assistance from the government, so it is important to monitor and evaluate their graduation more precisely and accurately. (Rasyid et al., 2024)

In student data processing, data mining techniques can be used to extract important information from large data sets. Data mining itself is a process that involves machine learning techniques to analyze and gain knowledge from data automatically. Two algorithms that are often used in data classification (Iqbal & Efendi, 2023) (Indra & Nasution, 2024) are K-Nearest Neighbor (K-NN) and C4.5. The K-NN algorithm is a classification method that determines categories based on the majority of their closest neighbors, while the C4.5 algorithm is a decision tree-based method that divides data into branches to simplify the classification process. (Fawaz et al., 2024) (Sitorus & Indra, 2024)

The AMIK Medicom Medan campus, as one of the private higher education institutions, participates in organizing the Smart Indonesia Card (KIP) Lecture scholarship program, and 75% of students are recipients of KIP Lectures. However, there is no optimal prediction system to determine the graduation trend of the students receiving the assistance. Therefore, this study aims to compare the performance of the K-NN and C4.5 algorithms in predicting the graduation status of students receiving the Smart Indonesia Card (KIP) Lecture scholarship. The results of the research are expected to make a real contribution to academic policy making and the improvement of data-based education monitoring systems.

The goal to be achieved through this research is to know and understand various aspects related to the topic discussed, which include various factors that affect and contribute to the results of this research. This research aims to achieve specific goals as listed below:

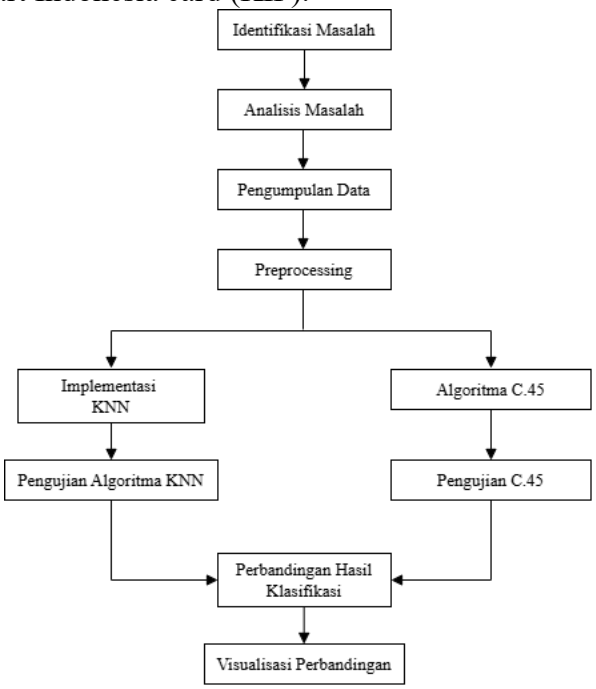
1. Applying the K-Nearest Neighbor (K-NN) algorithm in the process of classifying the graduation status of students receiving the Smart Indonesia Card (KIP) Lecture scholarship.
2. Applying the C4.5 algorithm in the process of classifying the graduation status of students receiving the Smart Indonesia Card (KIP) Lecture scholarship.
3. Compare the accuracy and performance levels of the two algorithms to determine the most effective classification method.

Methods

This type of research is a comparative descriptive research, which aims to compare the performance of two classification algorithms, namely K-Nearest Neighbor (K-NN) and C4.5, in predicting the graduation status of students who receive the Smart Indonesia Card (KIP) to study at the AMIK Medicom Medan Campus. The population in this study is all data on students who receive the Smart Indonesia Card (KIP) Lecture scholarship available at the AMIK Medicom Medan Campus.

The framework of this study is explained through the design flow described in Figure 3.1 which describes the stages to be followed in the research to achieve the goals that have

been set. The design flow includes systematic steps that include problem identification, data collection, data pre-processing, algorithm implementation, model performance evaluation, and analysis of the resulting results. Each step in the flow is designed to ensure that the research can run in a structured, effective, and valid way to compare the performance of the K-Nearest Neighbor (K-NN) and C4.5 algorithms in predicting the graduation status of students receiving the smart Indonesia card (KIP).



Data analysis in this study was carried out gradually and systematically using the Google Colab platform to evaluate and compare the performance of the K-Nearest Neighbor (K-NN) and Decision Tree C4.5 classification algorithms in predicting the graduation status of KIP scholarship recipients at AMIK Medicom Medan. The classification target is divided into two categories, namely "On Time" and "Late". The stages of data analysis carried out are as follows:

1. Data Cleaning and Validation: The initial stage is done by verifying the data using libraries such as *Pandas* and *NumPy* to ensure its integrity and validity. Data that contains missing values, duplications, or illogical values is identified and cleaned up. This process is important to ensure that only quality data is used in the modeling.
2. Data Transformation and Pre-Processing: Categorical data such as gender and scholarship status are converted into numerical data using encoding techniques such as *Label Encoding* or *One Hot Encoding* with the help of *scikit-learn*. Numerical features such as GPA and number of credits are normalized using *the Min-Max Scaling* method to support the performance of the attribute-scale sensitive K-NN algorithm.
3. Sharing of Training and Test Data: The processed data is divided into training data (80%) and test data (20%) using the train_test_split function of *scikit-learn*, with *random_state parameters* to ensure replication of results. This technique is known as *hold-out validation* and aims to avoid overfitting and provide a more objective evaluation of the model.
4. Application of Classification Algorithms: Two classification algorithms are applied using the *scikit-learn library*, namely:

a. K-Nearest Neighbor (K-NN): Using the KNeighborsClassifier, this algorithm classifies students based on attribute similarities to a number of nearby neighbors. K-values are tested in multiple configurations (e.g. K=3, 5, and 7) to determine the best performance.
b. Decision Tree (C4.5): In the C4.5 Algorithm stage, it is implemented with the data that has been provided.

Model Performance Evaluation

The model was evaluated using various binary classification evaluation metrics from *scikit-learn*, including:

a. Accuracy: The correct proportion of predictions of the entire test data.
b. Precision: The proportion of the correct "Pass" prediction compared to the entire "Pass" prediction.
c. Recall (Sensitivity): The proportion of "Graduated" students who successfully were correctly classified.
d. F1-Score: The harmonic mean between precision and recall.
e. Confusion Matrix: Using confusion_matrix to describe:
    1. TP (True Positive): Students who are predicted to be "Passing" correctly.
    2. TN (True Negative): The "Graduate" student who is predicted to be correct.
    3. FP (False Positive): "DO" students who are predicted to be "Passed".
    4. FN (False Negative): "On Time" students who are predicted as "DO".

Algorithm Analysis and Comparison

The results of the evaluation of both models were analyzed comparatively to determine the most effective algorithm. The analysis includes accuracy, model stability, ease of interpretation, and processing time. The algorithm with the best performance will be used as a reference in decision-making related to monitoring the graduation of scholarship recipients.

Result and Discussion

3.1 Pre-Processing of Case Study Data

Data pre-processing is an essential stage in the process of developing machine learning models, especially in the context of supervised classification. The quality and structure of the data used greatly determine the accuracy, generalization, and validity of the results obtained from the applied algorithm. In this study, the data used consisted of the Achievement Index (IP) score per semester and the graduation status of students in the Diploma Three (D3) Computer Vocational program. The data that was successfully collected consisted of 310 student entries, with four numerical attributes in the form of Achievement Index (IP) from semester 1 to semester 4, as well as one classification target attribute, namely the student's graduation status coded as "L" for Graduation and "D" for Drop Out (DO).

3.2 Implementation of the KNN Method

In the case study of manual calculation of this research, one test data was used, namely D304 which has an IP value presented in Table 4.1.x

Table 4.1. Test Data For Example Manual Calculation

Yes	**References**	**IP1 ($j=1$)**	**IP2 ($j=2$)**	**IP3 ($j=3$)**	**IP4 ($j=4$)**	**Class (y)**	**Prediction**

1	D304	3,04	3,00	2,84	2,97	OF	?

So that it can be known:

$$x(D304) = (3{,}04;\ 3{,}00;\ 2{,}84;\ 2{,}97)$$

Meanwhile, the training data consists of 20 student data with details of IP values presented in Table 4.2.

Table 4.2. Train Data For Calculation Examples

No	ID Data	$x^{(i)}$	IP1	IP2	IP3	IP4	Classes
1	D001	$x^{(1)}$	4,00	3,67	3,59	3,10	Lulus
2	D002	$y^{(2)}$	4,00	3,48	3,77	3,57	Lulus
3	D003	$y^{(3)}$	3,76	3,14	3,14	3,57	Lulus
4	D004	$y^{(4)}$	3,67	3,81	4,00	4,00	Lulus
5	D005	$y^{(5)}$	3,57	3,19	3,86	3,76	Lulus
6	D006	$y^{(6)}$	4,00	3,81	3,68	3,86	Lulus
7	D008	$y^{(7)}$	4,00	3,90	4,00	3,95	Lulus
8	D009	$y^{(8)}$	4,00	3,90	4,00	4,00	Lulus
9	D010	$y^{(9)}$	4,00	3,81	3,59	4,00	Lulus
10	D011	$y^{(10)}$	3,95	3,71	3,55	4,00	Lulus
11	D012	$y^{(11)}$	3,10	3,19	3,95	3,86	Lulus
12	D013	$y^{(12)}$	3,90	3,57	3,86	3,98	Lulus
13	D007	$y^{(13)}$	3,14	2,95	2,88	2,75	OF
14	D020	$y^{(14)}$	3,00	3,20	2,67	2,81	OF
15	D036	$y^{(15)}$	2,80	3,00	2,80	2,90	OF
16	D046	$y^{16)}$	3,02	3,03	3,05	2,65	OF
17	jD074	$y^{(17)}$	3,12	3,08	2,97	3,07	OF
18	D082	$y^{(18)}$	2,74	3,30	2,73	3,14	OF
19	D089	$y^{(19)}$	3,07	3,08	2,95	2,36	OF
…	…	…	…	…	…	…	…
310	D104	$y^{(20)}$	3,00	2,83	2,80	2,80	OF

The data above
Graduates = 12 Students
DO = 8 Students

Table 4.3. Summary of Euclidean Distances

No	ID Data	Class (y)	Distance $d(x, x^{(i)})$
1	D001	Lulus	1,396
2	D002	Lulus	1,542
3	D003	Lulus	0,994
4	D004	Lulus	1,86
5	D005	Lulus	1,408
6	D006	Lulus	1,754
7	D008	Lulus	2,009

No	ID Data	Class (y)	Distance $d(x, x^{(i)})$
8	D009	Lulus	2,034
9	D010	Lulus	1,789
10	D011	Lulus	1,702
11	D012	Lulus	1,437
12	D013	Lulus	1,768
13	D007	OF	0,25
14	D020	OF	0,31
15	D036	OF	0,253
16	D046	OF	0,384
17	D074	OF	0,199
18	D082	OF	0,47
19	D089	OF	0,626
20	D104	OF	0,247

Table 4.4. Order of Euclidean distance from the smallest

Peringkat	ID Data	Class (y)	Distance $d(x, y)$
1	D074	OF	0,199
2	D104	OF	0,247
3	D007	OF	0,25
4	D036	OF	0,253
5	D020	OF	0,31
6	D046	OF	0,384
7	D082	OF	0,47
8	D089	OF	0,626
9	D003	Lulus	0,994
10	D001	Lulus	1,396
11	D005	Lulus	1,408
12	D012	Lulus	1,437
13	D002	Lulus	1,542
14	D011	Lulus	1,702
15	D006	Lulus	1,754
16	D013	Lulus	1,768
17	D010	Lulus	1,789
18	D004	Lulus	1,86
19	D008	Lulus	2,009
20	D009	Lulus	2,034

3.3 KNN Implementation Results for All Data

In this section, the implementation of the K-Nearest Neighbors (KNN) algorithm is carried out to predict student graduation status based on academic data in the form of Achievement Index (IP) for semesters 1 to 4 for all data. The KNN model has characteristics that are non-parametric, easy to implement, and effective for small to medium-sized data, especially in the case of binary classifications such as in this study. The

dataset used consisted of 310 students, each with four numerical features (IP semesters 1–4) and one categorical label ("Pass" or "DO" status). Data is read from an Excel file using *the pandas* and *openpyxl* libraries in *Python*. Before training, the data was divided into two subsets: 217 data for training (70%) and 93 data for testing (30%), with class stratification to keep the distribution between the "Pass" and "DO" categories proportional.
With a moderate KNN model, the entire experiment can be completed efficiently without the need for GPU support or parallel processing.

a. Parameter K Selection Using 10-Fold *Cross-Validation*

The selection of parameter K in the KNN algorithm was carried out empirically through 10-fold *cross-validation* (CV=10). In this scheme, the training data is divided into 10 parts (folds) that are almost the same size. For each iteration, 9 sections are used for training and 1 section for validation. This process is repeated 10 times so that each part becomes validation data once. The average accuracy of these 10 iterations is calculated for each K value tested.

In this experiment, the K-value was varied from 1 to 20. The evaluation was carried out using the *cross_val_score()* function of the *scikit-learn library*, with the evaluation metric in the form of classification accuracy. The results of the experiment showed that the value of K=3 provided the highest average accuracy, thus being set as the optimal parameter for the training of the final KNN model. A visualization of the relationship between the K-value and the average accuracy of the 10-fold cross-validation can be seen in Figure 4.1.

Figure 4.1 shows the results of the evaluation of the selection of the K parameter on the K-Nearest Neighbors (KNN) algorithm using the 10-fold *cross-validation* technique. In this graph, the horizontal axis shows the value of K varied from 1 to 20, while the vertical axis shows the average classification accuracy obtained from the cross-validation process. Based on the results of visual observation of the graph, it can be seen that the accuracy of the model increases significantly at small K-values , with the highest peaks reached at K=3 and K=4, which is 99.0%. After reaching this peak, there was a fluctuating decrease in accuracy ranging from K=5 to K=15, although most accuracy values remained high, in the range between 96.8% to 98.4%.

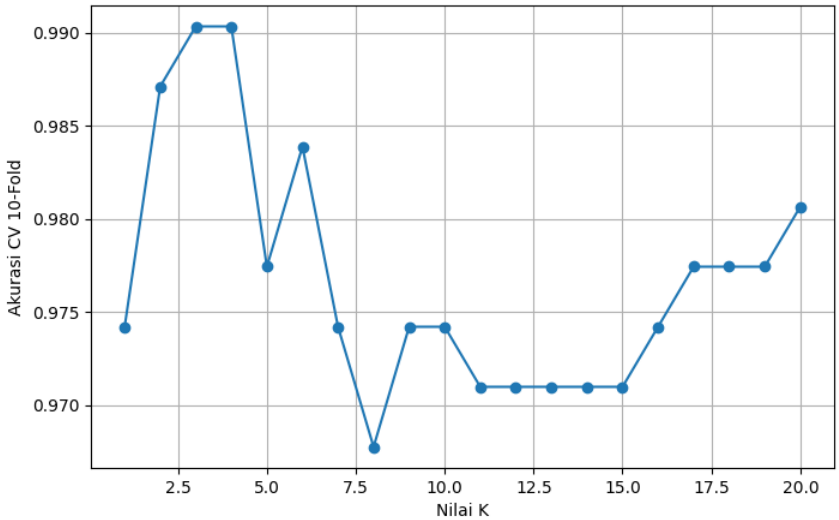


Figure 4.1. Best K Value Selection (*Cross Validation* 10-*fold*)

This phenomenon of performance decline can be explained by the tendency of the KNN model to experience an over-smoothing effect on too large a K-value, i.e. when classification decisions are based on too many neighbors, thus blurring the differences between local classes. On the other hand, a K value that is too small can lead to overfitting because the model is too sensitive to noise or outliers. Therefore, the selection of K-values must consider the balance between bias and variance. In this context, the value K=3 is chosen as the optimal parameter because it provides the highest accuracy and is within a stable range, and maintains the sensitivity of the model to local patterns in the data. This optimal value is then used in the final model training process and evaluation of the performance of the KNN model on the full dataset.

b. Evaluation of the KNN Model on Test Data

After obtaining an optimal value of K=3 from the 10-fold cross-validation process, the K-Nearest Neighbors (KNN) model was then trained and tested using an entire dataset of 310 student data. This step aims to test the final performance of the model against all available data, as well as get a comprehensive picture of the algorithm's classification ability in detecting student graduation status based on IP scores for semesters 1 to 4. The model evaluation was carried out using *a confusion matrix*, *classification report*, and measurement of four main metrics: accuracy, *precision*, *recall*, and f1-score. Table 4.4 presents the *confusion matrix* of the KNN model prediction results in two target classes, namely "DO" and "Pass", by showing the number of true and false predictions produced.

Based on Table 4.4, the KNN model with K=3 is able to classify all students who actually graduated (290 data) accurately without prediction errors (*class* recall "Passed" = 100%). This shows that the model is very effective in recognizing students who have the potential to complete their studies on time. For the "DO" class, the model successfully identified 17 out of 20 students who were actually dropouts (class recall "DO" = 85%), but there were still three misclassifications where DO students were predicted as "Passed".

Table 4.4. *Confusion Matrix* KNN Model Results

		Prediction: DO	**Prediction: Pass**
Actual Grade: DO		17	3
Actual Class: Pass		0	290

Table 4.5. Evaluation of the KNN Model Classification

Classes	**Precision**	**Recall**	**F1-Score**	**Support**
OF	1,00	0,85	0,92	20
Lulus	0,99	1,00	0,99	290
Macro Avg	0,99	0,93	0,96	310
Weighted Avg	0,99	0,99	0,99	310

The overall performance of the model is shown by an accuracy of 99.03%, with a *precision value* of 0.989, a recall of 1,000, and an f1-score of 0.995. This indicates that the KNN model is able to produce highly accurate and balanced predictions for both classes. Not only is the *weighted average* high due to the dominance of the majority class, but the macro *average* also shows consistent performance against the minority class. With a significant improvement in *recall* and DO-class f1-score compared to previous evaluations, the KNN model in this experiment proved to be more resilient in handling unbalanced data distribution, especially after applying 10-fold cross-validation and optimal K selection.

3.4 Implementation of the C45 Method

3.4.1 Case Studies and Manual Calculations of the C4.5 Method

To demonstrate manual calculations in building a decision tree using the C4.5 algorithm, the first step is to select the *best threshold* and attributes as the data separator (*split*). The selection *of thresholds* and attributes is done through the calculation of *Split Info* for each potential *threshold* on each attribute. *Threshold* is obtained from the middle value between two feature values that are in a row and have different classes. For each of these thresholds, the *Entropy* subset, *Information Gain*, and *Split Info* values are calculated, which are then used to obtain *the Gain Ratio* value. This process is carried out for all attributes systematically.

In the case study of this study's manual calculation, one test data was used, namely D304 with IP and Class values as presented in Table 4.1 and 20 training data as presented in Table 4.2 previously. So that it can be known: x

$$x(D304) = (2{,}80, 3{,}00, 2{,}80, 2{,}90)$$

By notating the Pass class as L and the Drop Out class as D, it can be written that: there is as much training data as , with the amount of data with the class of Pass, which is and the amount of data with the class DO is $|S| = 20|S_L| = 12|S_D| = 8$.

Table 4.8. GainRatio recapitulation of all attributes

Attribution	**Best Split Points**	***Gain***	***SplitInfo***	***GainRatio***
IP1	3,355	0,74448	0,99277	0,74990
IP2	3,110	0,55678	0,88129	0,63178
IP3	3,095	0,97095	0,97095	**1,00000**
IP4	3,355	0,74448	0,99277	0,74990

Next, the attribute with the largest *GainRatio* value is selected . If there is more than one attribute that has *the best GainRatio*, then the attribute that appears first is chosen in a so-called *tie-breaking by order* (Zhang, & Jiang, 2020). In Table 4.8, the *largest* GainRatio value is found in the IP3 attribute, so the attribute chosen as the root of the decision tree is: IP3, with a split: 3.095. So the first branch of the decision tree is presented by Figure 4.2 as follows.

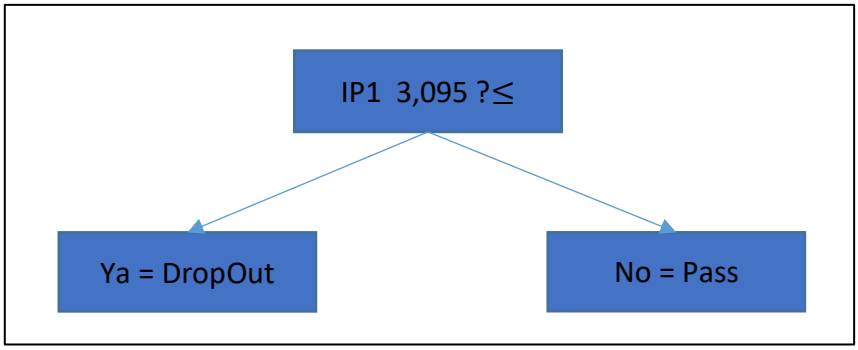


Figure 4.2. C4.5 Decision Tree on Case Studies

In the first branch, of the 20 data that *were split*, there were 8 data with an IP3 value that met 3.095 as a DO class, and 12 data with an IP3 value that did not meet 3.095 as a Pass class. Since each subset of data already has a uniform class, the split is not continued, so the decision tree consists of only 1 level.≤≤

Testing on Test Data:

The decision tree in Figure 4.2 is implemented to predict the D304 test data class. It was previously known that:

$$x(D304) = (3{,}04;\ 3{,}00;\ 2{,}84;\ 2{,}97),$$

where the IP3 attribute value of the D304 data is 2.84, as the value to be tested with a threshold value of 3.095 in the decision tree. Because $x(D304)_{IP3} = 2{,}84 \leq 3{,}095$, then enter the left branch, so that the class prediction for D304 students is "Drop out (D)", which is also in accordance with the actual label of the student.

4.3.3. C4.5 Implementation Results for All Data

The decision tree model in this study was implemented using the C4.5 algorithm approach, which was represented through the use of *DecisionTreeClassifier* from the *scikit-learn library with* entropy-based *attribute selection criteria*. The training and testing process was carried out on data of 310 students, consisting of Semester Achievement Index data (IP1 to IP4) and final graduation status (class "Graduate" or "DO"). The dataset was divided into 70% of the training data (217 students) and 30% of the test data (93 students) using *a stratified split* technique to maintain class proportions.

Table 4.7 presents the confusion matrix of the C4.5 model's prediction results for the two target classes. The model correctly predicted 289 out of 290 students who actually graduated (*true positive*), with only one misclassification (*false positive*) predicting that students did not graduate as graduates. Meanwhile, of the 20 students who were really DO, as many as 17 were classified correctly (*true negative*), and 3 students were wrongly classified as "Passed" (*false negative*).

Table 4.6. *Confusion Matrix* Model C.45 Results

	DO Prediction	**Pass Prediction**
Current DO	17	3
Actual Pass	1	289

Table 4.7. Evaluation of Model C.45 Classification

Classes	*Precision*	*Recall*	F1-*Score*	*Support*
OF	0,94	0,85	0,89	20
Lulus	0,99	1,00	0,99	290
Macro Avg	0,97	0,92	0,94	310
Weighted Avg	0,99	0,99	0,99	310

The overall accuracy of the model was recorded at 98.71%, indicating that the majority of the model's predictions corresponded to the actual label. From Table 4.8, *the weighted average precision* reaches 0.99 which means that most of the predictions are absolutely on point. *An overall recall* of 0.99 indicates the model's very high sensitivity in recognizing both classes, especially the "Pass" class. An F1-score of 0.99 indicates an excellent balance between precision and sensitivity. The support columns in the classification evaluation report show the actual amount of data for each class, which is important for understanding the proportion of class distributions in the dataset as well as assessing the balance of data that has a direct impact on the model's performance interpretation.

Specifically, the model's performance against the majority class of "Pass" is very impressive. With a recall value of 1.00, the model was able to identify all students who graduated without a single classification error. *An accuracy* of 0.99 in this class indicates that almost all of the "Pass" predictions do indeed correspond to reality, which is further reflected in the f1-score of 0.99. On the other hand, for the minority "DO" class, the model showed strong performance, with a precision of 0.94, a *recall* of 0.85, and an f1-score of 0.89, reflecting that although there were still some misclassifications, the model was still able to recognize the majority of DO students precisely.

Figure 4.3 shows the structure of the decision tree generated by the C4.5 algorithm after being trained using student academic data based on IP from semester 1 to semester 4. This model divides the decision space hierarchically using the entropy criterion as a measure of impurity and selects the best separator attribute based on the gain *ratio*, as is the basic principle of the C4.5 algorithm.

The root of the tree starts from the IP attribute of semester 3 (IP3) with a threshold of ≤ 2.98. If the IP3 value is less than or equal to the threshold (True), then the branch is pointed to the left, while if it is larger (False), it is pointed to the right. At this root node, there were 217 samples with an entropy value of 0.345, and the majority were classified into the "Pass" class (203 out of 217). This suggests that IP3 is being the most informative attribute for initial separation.

The left branch of the root node further breaks down the data based on the 1st semester IP (IP1 $\leq$ 3,262). At this node, a total of 18 samples were divided into 13 "DO" class data and 5 "Pass" data, with a fairly high entropy (0.852), indicating a significant mix of classes. From here, the model performs further separation based on the 4th semester IP (IP4 $\leq$ 3,209). As a result, 14 samples were classified as "DO" with high accuracy (13 true, 1 false), and the remaining 4 samples were classified into the "Pass" class with zero entropy (perfect classification). This shows that the combination of IP1 and IP4 plays a significant role in differentiating students who drop out in this group.

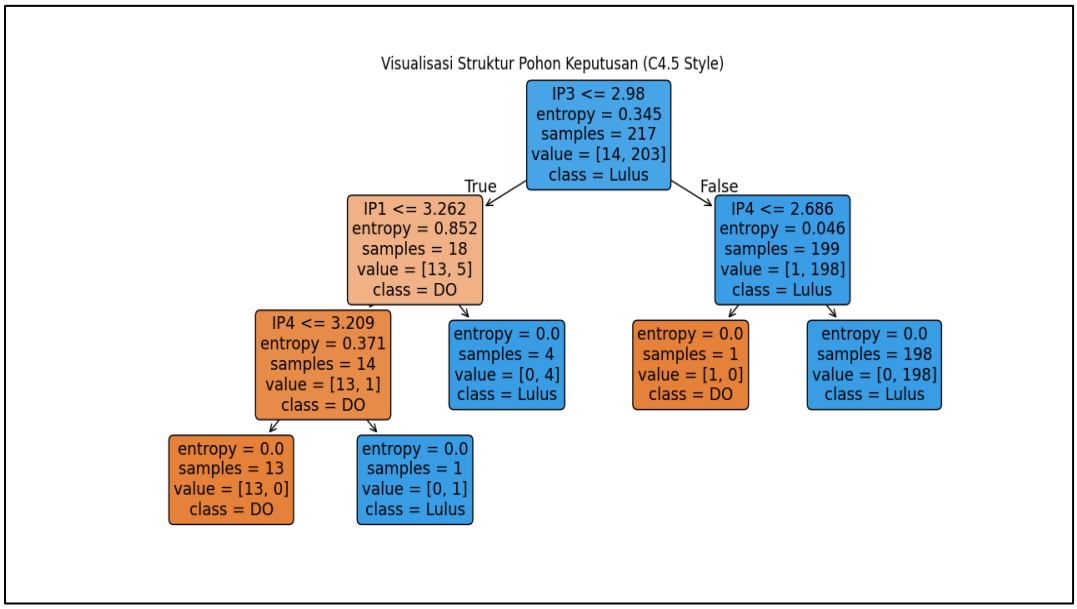


Figure 4.3. Visualization of Decision Tree Structure (C4.5) Model Training Results

The right branch of the root node (IP3 > 2.98) points to the node that checks the 4th semester IP (IP4 ≤ 2.686). Here, 199 samples were very dominantly distributed to the "Pass" class (198 true, 1 false), with a very low entropy value (0.046), indicating a high homogeneity of the data. The two leaf nodes below indicate perfect classification: one "DO" data is identified precisely, and 198 "Pass" data are also classified without errors.

The structure of this tree shows that the IP3 and IP4 attributes are the most powerful academic indicators in predicting graduation, while IP1 makes an important contribution to identifying dropout risks early. This pattern also suggests that college students with low IP3 scores tend to be at high risk for DO, especially if IP1 is also below a certain threshold. In contrast, a high IP4 rating is consistently associated with a "Pass" status. Overall, the resulting decision tree is highly efficient with low depth and a small number of nodes, but it is still able to provide an accurate and logically explainable classification. This model is not only useful for predicting graduation status, but can also be used by campuses or guardian lecturers to develop data-driven intervention strategies to prevent students at risk of dropouts. The high interpretability of the tree structure makes the C4.5 algorithm a suitable choice in educational decision support systems.

3.5 Evaluation Analysis

This study aims to compare the performance of two supervision classification algorithms, namely *K-Nearest Neighbor* (KNN) and *C4.5 Decision Tree*, in predicting student graduation status based on the Achievement Index (IP) score for semesters 1 to 4. Based on the results of the latest experiments, both algorithms show very high classification performance against two target classes ("Pass" and "Drop Out"), with an accuracy value of above 98%. However, there are important characteristic differences between the two that need to be comprehensively considered in the context of model application.

Table 4.9 Interpretation of the results of the performance of the two algorithms

Metric	KNN	C4.5
Precision (DO)	1,00	0,94
Recall (DO)	0,85	0,85

F1-Score (DO)	0,92	0,89
Precision (Lulus)	0,99	0,99
Recall (Pass)	1,00	1,00
F1-Score (Lulus)	0,99	0,99
Macro Precision	0,99	0,97
Macro Recall	0,93	0,92
Macro F1-Score	0,96	0,94
Weighted Precision	0,99	0,99
Weighted Recall	0,99	0,99
Weighted F1-Score	0,99	0,99

The KNN model with optimal parameters K=3, obtained through 10-fold *cross-validation*, produces an accuracy of 99.03%, with a *recall value* of 0.85 for the "DO" class. This means that of the 20 students who were really dropouts, as many as 17 were accurately recognized by the KNN model. On the other hand, the C4.5 model shows an accuracy of 98.71% with an equally large recall value (0.85) for the DO class. Both models were able to identify the majority of dropout students, although there were still some minor misclassifications.

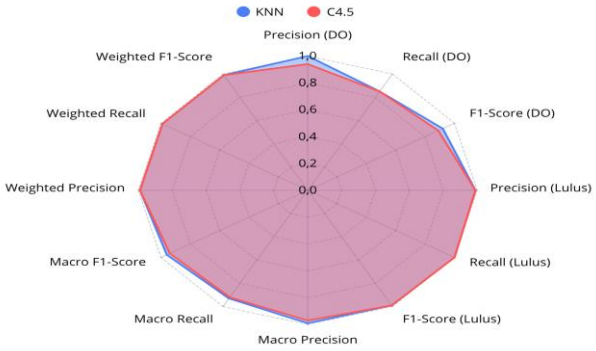


Figure 4.4 comparison of KNN and C4.5 performance

Figure 4.4 shows a comparative visualization of the performance of two classification algorithms, namely K-Nearest Neighbor (KNN) and C4.5, based on the evaluation metrics obtained from the model test results. The metrics compared include Precision, Recall, and F1-Score in each class (DO and Pass), as well as Macro Average and Weighted Average values. Each section of the diagram shows the contribution of the metric values of both algorithms, making it easier to identify the model that performs better.

Based on the image, it can be seen that both algorithms have excellent performance in classifying student data. In the Pass class, both KNN and C4.5 showed almost identical results with a Precision value of 0.99, Recall of 1.00, and F1-Score of 0.99. This shows that both models are able to identify graduating students with a very high degree of accuracy.

The performance difference is more noticeable in the DO class, which is a minority class in the dataset. The KNN algorithm obtained a Precision value of 1.00, higher than C4.5 which obtained a value of 0.94. In addition, the F1-Score KNN value of 0.92 is also higher than C4.5 of 0.89, while the Recall value of both algorithms reaches 0.85. These results show that KNN is better at identifying students with DO status with a lower rate of positive prediction errors.

In the overall evaluation using Macro Average, KNN again showed a slightly superior performance with a Precision value of 0.99, Recall of 0.93, and F1-Score of 0.96. Meanwhile, the C4.5 algorithm obtained a Precision value of 0.97, Recall of 0.92, and F1-Score of 0.94. This difference indicates that on average in each class, KNN has better classification ability than C4.5.

However, on the Weighted Average metric, both algorithms obtained the same score, which was 0.99 for Precision, Recall, and F1-Score. This shows that overall both models have a very high level of performance against the datasets used. However, because KNN is able to produce better scores in the DO class and has a higher Macro Average value, the KNN algorithm can be considered as a superior model to C4.5 in this study, especially in handling classification in minority classes.

However, although the KNN model recorded slightly higher numerical accuracy than C4.5 (0.32 accuracy difference), the C4.5 model was still more recommended in the context of this study. This is due to the structural and interpretive advantages offered by C4.5. This model is able to generate an explicit decision tree structure, consisting of classification rules that can be traced and explained. In the context of higher education, especially for academic decision support systems, the ability to explain why a student is predicted as "Pass" or "DO" becomes critical to support targeted interventions.

This advantage of C4.5 is in line with the findings in the study (Marcolino *et al*. 2025), which affirm the importance of interpretability in classification models applied to sensitive domains such as education. In addition, the Gain Ratio-based attribute selection approach in C4.5 allows for data splitting that is more adaptive to class imbalances and information variations between features, compared to KNN that relies solely on the distance between points in the feature space.

The macro performance of the model also supports the advantages of C4.5 comprehensively. The *f1-score macro average* value of 0.94 indicates that this model provides balanced predictions for both classes, including minority classes. In contrast, although KNN has a slightly higher *f1-score* macro (0.96), it does not provide any further explanatory insights or decision bases. This makes KNN suitable for rapid numeric-based predictions, but less than ideal if the model is used as a decision-making tool at the institutional level.

Operationally, both algorithms can be efficiently implemented in the Python programming language and integrated in academic information systems. However, although KNN can provide numerically high performance, C4.5 is more recommended for implementation in academic risk early detection systems, especially when logical explanations and consideration of minority classes are required. This model not only demonstrates highly competitive classification performance, but also provides transparency, interpretability, and flexibility in supporting data-driven interventions for at-risk students. This model provides explicit classification rules that can be used as a reference for guardian lecturers and study program management in designing a student assistance system. This approach supports the principles of algorithmic ethics and academic justice, as emphasized by (Villar & Andrade (2024) in a comparative study of supervised learning algorithms in the context of higher education.

Conclusion

Based on the results of the experiment, evaluative analysis, and in-depth discussion of the performance of the K-Nearest Neighbor (KNN) and C4.5 (Decision Tree) algorithms in

predicting the graduation status of students based on the IP scores of semesters 1 to 4, the following conclusions were concluded:

1. The KNN model with optimal parameters of K=3, obtained through 10-fold cross-validation, showed very high classification performance, with an accuracy of 99.03% and an f1-score value of 0.99 for the "Pass" class and 0.92 for the "DO" class. This model was able to recognize all students who graduated perfectly (*recall* = 1.00), and identify 17 out of 20 students who did (*recall* = 0.85).
2. The C4.5 (*Decision Tree*) model also showed excellent classification performance, with an accuracy of 98.71%, as well as an f1-score value of 0.99 for the "Pass" class and 0.89 for the "DO" class. This model classifies 289 out of 290 college students who graduated and 17 out of 20 college students who DO correctly.
3. A comparison between the two algorithms showed that KNN excelled in accuracy and f1-score metrics for both classes, including the minority class "DO". However, C4.5 remains relevant and competitive, mainly because of its advantages in interpretability through clear decision tree visualization. This is important in the context of decision support systems in educational institutions, where transparency of classification logic is needed by policy makers.
4. Both models showed good ability in handling data imbalances, especially after the application of stratified sampling and *cross-validation* techniques (CV=10). This proves that *the machine learning* approach can be reliably used to build a predictive system based on student academic data.
5. Visualization of the C4.5 model decision tree structure reveals that the IP attribute of semester 3 is the most decisive indicator of graduation status. Students with low GPs in the final semesters tend to be classified as DO, which confirms the importance of academic intervention in the final phases of study.

Refrences

[1] M. Rasyid, Z. Sitorus, R. Farta Wijaya, and M. Iqbal, "MACHINE LEARNING ANALYSIS IN IMPROVING THE EFFICIENCY OF THE STUDENT ADMISSION DECISION MAKING PROCESS NEW AT PANCA BUDI MEDAN DEVELOPMENT UNIVERSITY," *Bulletin of Engineering Science, Technology and Industry*, vol. 2, no. 3, pp. 216–225, 2024, [Online]. Available: https://bestijournal.org

[2] M. Iqbal and S. Efendi, "Data-Driven Approach for Credit Risk Analysis Using C4.5 Algorithm," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 14, no. 1, pp. 11–20, May 2023, doi: 10.21512/comtech.v14i1.8243.

[3] M. Indra and D. Nasution, "Analysis of the Number of Children Owned by Nursing Home Residents Using the C4.5 Algorithm," *Journal of Information Technology, computer science and Electrical Engineering (JITCSE)*, vol. 1, no. 3, pp. 137–141, 2024, doi: 10.30596/jitcse.

[4] M. A. Fawaz, K. Khairul, and A. P. U. Siahaan, "Analysis of Performance Comparison between K-Nearest Neighbor (KNN) Method and Naïve Bayes Method in Reward for Honda Motorcycle Salesman Tour," *sinkron*, vol. 8, no. 3, pp. 1932–1944, Jul. 2024, doi: 10.33395/sinkron.v8i3.13935.

[5] Z. Sitorus and M. Indra, "Analysis of Room Allocation Based on Age in Nursing Homes Using the C4.5 Decision Tree Method," *Journal of Information Technology, computer science and Electrical Engineering (JITCSE)*, vol. 1, no. 2, pp. 153–157, May 2024, doi: 10.61306/jitcse.v1i2.

[6] R. Lukito, "'Compare But Not to Compare': Kajian Perbandingan Hukum di Indonesia," *Undang: Jurnal Hukum*, vol. 5, no. 2, pp. 257–291, Dec. 2022, doi: 10.22437/ujh.5.2.257-291.

[7] Moch. R. Yuliansyah, M. B, and A. Franz, "Perbandingan Metode K-Nearest Neighbors dan Naïve Bayes Classifier Pada Klasifikasi Status Gizi Balita di Puskesmas Muara Jawa Kota Samarinda," *Adopsi Teknologi dan Sistem Informasi (ATASI)*, vol. 1, no. 1, pp. 08–20, Jun. 2022, doi: 10.30872/atasi.v1i1.25.

[8] I. Faturrahman and P. Setiawati, "PERANCANGAN CHATBOT BERBASIS ARTIFICIAL INTELLIGENCE DENGAN MENGGUNAKAN PYTHON UNTUK MANAJEMEN HUBUNGAN PELANGGAN DI RUDY SALON & SPA," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 4412–4419, Jun. 2025.

[9] A. Fakhri and Y. C. Winursito, "ANALISIS PENUMPANG KAPAL TITANIC MENGGUNAKAN TITANIC DATASET DENGAN BANTUAN PEMROGRAMAN PYTHON," *Jurnal Sains Student Research*, vol. 2, no. 1, pp. 537–542, Feb. 2024.

[10] R. G. Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, Feb. 2023, doi: 10.47233/jteksis.v5i1.750.

[11] F. Wilyani, Q. N. Arif, and F. Aslimar, "Pengenalan Dasar Pemrograman Python Dengan Google Colaboratory," *Jurnal Pelayanan dan Pengabdian Masyarakat Indonesia*, vol. 3, no. 1, pp. 08–14, Mar. 2024, doi: 10.55606/jppmi.v3i1.1087.

[12] G. D. Nursyafitri, "Tutorial Jalankan Script Python dengan Google Colab," https://dqlab.id/tutorial-jalankan-script-python-dengan-google-colab.

[13] A. Fakih, M. A. Hamzami, M. R. Hadianto, and N. I. S. Alifah, "Perbandingan Akurasi Algoritma C4.5 dan K-NN Untuk Prediksi Kelulusan Mahasiswa Penerima Beasiswa," *Jurnal Komputer Antartika*, vol. 3, no. 1, pp. 18–25, Jan. 2025, doi: 10.70052/jka.v3i1.623.

[14] M. A. A. Lobo and A. C. Talakua, "Klasifikasi Data Penerima Beasiswa Menggunakan Algoritma C4.5," *Jurnal Penelitian Inovatif*, vol. 4, no. 2, pp. 575–582, May 2024, doi: 10.54082/jupin.355.

[15] F. Karepesina and L. Zahrotun, "PENERAPAN DATA MINING UNTUK PENENTUAN PENERIMA BEASISWA DENGAN METODE K-NEAREST NEIGHBOR (K-NN)," *TECHNO*, vol. 24, no. 1, pp. 1–9, Dec. 2023.

[16] S. R. Cholil, T. Handayani, R. Prathiv, and T. Ardianita, "Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa," *IJCIT (Indonesian Journal on Computer and Information Technology)*, vol. 6, no. 2, pp. 118–127, Jul. 2021.

[17] T. V. Sandiva, S. Defit, and G. W. Nurcahyo, "Implementasi Algoritma C4.5 Untuk Prediksi Penerima Beasiswa Program Indonesia Pintar," *Jurnal KomtekInfo*, vol. 11, no. 4, pp. 354–362, Sep. 2024, doi: 10.35134/komtekinfo.v11i4.582.

[18] Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3), 175–185.

[19] Zhang, Z., Zhao, Y., & Wang, X. (2017). An Improved KNN Algorithm Based on Attribute Weight Optimization. *Computational Intelligence and Neuroscience*, 2017.

[20] Bhargava, M., Sharma, R., Bhargava, R., & Mathuria, M. (2013). Decision Tree Analysis on J48 Algorithm for Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), 1114–1119.

[21] Journal of Big Data Editors. (2025). Resampling approaches to handle class imbalance: a review from a data perspective. *Journal of Big Data*, 12, 71.

[22] Marcolino, M. R., Porto, T. R., & Cechinel, C. (2025). Student dropout prediction through machine learning optimization: insights from Moodle log data. *Scientific Reports*, 15:9840.

[23] Villar, A., & Robledo Velini de Andrade, C. (2024). Supervised machine learning algorithms for predicting student dropout and academic success: A comparative study. *Discover Artificial Intelligence*, 4(2).

[24] Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer.
[25] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data mining: Practical machine learning tools and techniques* (4th ed.). Morgan Kaufmann.