Published online on the journal's webpage: **http://journal.ittelkom-pwt.ac.id/index.php/dinda**

Journal of Dinda

Data Science, Information Technology, and Data Analytics

Vol. 3 No. 2 (2023) 50 - 58 E-ISSN: 2809-8064

Dominant Requirements for Student Graduation in the Faculty of Informatics using the C4.5 Algorithm

Alvina Tahta Indal Karim[1], Sudianto Sudianto[2*]

[1]Department of Information Systems, Institut Teknologi Telkom Purwokerto

[2*]Department of Informatics, Institut Teknologi Telkom Purwokerto

[1]19103075@ittelkom-pwt.ac.id, [2*]sudianto@ittelkom-pwt.ac.id

Abstract

Graduating on time is one of the indicators in the achievement and ranking of educational institutions. The achievement of graduating on time in educational institutions is essential to balance incoming and graduating students. The problem that occurs, the attributes for graduating on time have varying weightings, so the determinants of the attributes for passing on time need to be known so that the anticipation of achieving graduation on time can be met. The purpose of this study is to find out the dominant attributes in the prediction of graduating on time for students. The attributes used are credit scores (Semester Credit Units), GPA scores (Grade Point Average), and English scores (TOEFL). The method used is the C4.5 Algorithm which is one of the classification methods in data mining. The data used was 262 data, split randomly with a composition of training and testing data of 80:20. Data is processed using the data mining process by creating decision trees. The decision tree results using the C4.5 Algorithm show that the GPA value is the most influential attribute in predicting a student's graduation time. In addition, predictions based on the decision tree of the C4.5 Algorithm with criterion = 'gini' and max_depth = 5 showed an accuracy result of 77%.

Keywords: C4.5 Algorithm, Data Mining, Graduation, Prediction, Root

1. Introduction

Timely graduation is one of the indicators in the ranking of educational institutions [1]. The achievement of graduates in educational institutions is essential to balance incoming and graduating students. To achieve graduation on time, Faculty of Informatics, Institut Teknologi Telkom Purwokerto has regulations that must be met so that students are declared graduates. One of the attributes is that students must meet the scores of credits (Semester Credit Units)/SKS, GPA/IPK (Grade Point Average), and English language scores (TOEFL).

Graduating on time provides students benefits and the opportunity to do what they like and achieve achievements by starting early experience in the world of work and saving on education costs. The advantage is also felt by universities, namely that they can be helped in the accreditation assessment process to increase student graduation on time.

The challenge in achieving graduation on time is to identify early, which has challenges and impacts the occurrence of delays in graduate students [2]. So, from the existing problems, predictions are needed so that graduation in students on time can be achieved. In addition, information and notifications about student graduation predictions are critical to know, so they can be identified early for students who do not graduate on time. Therefore, the solution offered is to predict graduation using a data mining approach. Data mining techniques help process the information on predictions of graduates on time. The technique of extracting extensive data to find helpful information for users is the meaning of data mining. Data mining groupings are description, estimate, prediction, classification, and clustering [3] from various fields such as health [4], [5], agriculture [6]–[9], and current phenomena [10]–[13].

In previous studies in predicting student graduation, many of these studies have been carried out with the C4.5 algorithm [13]–[16]. Some of these studies used data mining software such as Rapid Miner and WeKa. Meanwhile, in contrast to this research, the Machine Learning approach with Python programming was chosen to predict students' timely graduation. The

Received: 09-04-2023 | Revised: 26-07-2023 | Published: 01-08-2023

attributes used to focus on the main requirements for students' on-time graduation: credits, GPA, and TOEFL.

The C4.5 algorithm was chosen as the prediction algorithm because it functions as a classification algorithm. The C4.5 algorithm has been widely used to classify numerical and categorial attribute data. The C4.5 algorithm is an enhanced algorithm of the Iterative Dichotomizer 3 or Induction of Decision 3 (ID3) Algorithm, first introduced and developed by J. Ross Quinlan in 1979. The C4.5 algorithm has several advantages over ID3, including attributes that are discrete or continuous type and missing values that can be handled using the C4.5 algorithm, and also this algorithm can trim trees. The main advantages of the C4.5 algorithm are that the resulting model can be a tree or rules that are easy to interpret and convert to Structure Query Language (SQL) rules, are still acceptable for their degree of accuracy, discrete and numeric type attributes can be handled, and handle discrete type attributes efficiently [14].

Therefore, this study aims to predict student graduation and find the most dominant attribute in determining student graduation through the decision tree pattern at Faculty of Informatics, Institut Teknologi Telkom Purwokerto.

2. Research Methods

2.1. *Knowledge Discovery in Database (KDD)*

The term KDD, or seeking knowledge in a database, emphasizes applying specific data mining methods, including searching for knowledge in extensive data. Methods of knowledge acquisition in databases containing interconnected tables [15]. Knowledge Discovery in Database (KDD) results can be used for decision-making. The KDD process can be seen in Figure 1 as follows:

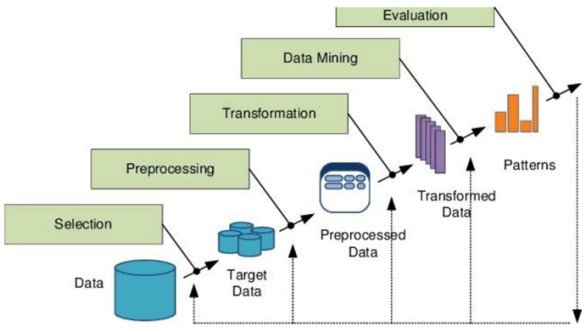

Figure 1. The Process of Knowledge Discovery in Database

The Knowledge Discovery in Database (KDD) process consists of six processes, namely [3]:

a. Data selection
The data is selected from the overall data for use in the data mining process.
b. Preprocessing
Double data dumping, checking for inconsistent data, and correcting errors are preprocessing processes.
c. Transformation
Double data dumping, checking for inconsistent data, and correcting errors are preprocessing processes.
d. Data mining
The search for interesting information on the data that has been selected using a specific algorithm.
e. Evaluation
The evaluation is carried out by checking whether the patterns of information found contradict the facts or hypotheses present in previous studies.
f. Visualization
Visualization is used to present information in the form of decision trees or the form of rules.

2.2. *Decision Tree*

Data Mining is a technique of extracting extensive data to find helpful information for users. Data mining is helpful in many fields of science to find knowledge and information in extensive data [16]. A decision Tree changes the shape of the data, which was initially in the form of a table, namely attributes, and records, into a tree shape so that the tree can represent rules. Trees consist of knots and ribs. An example of a decision tree in Figure 2.

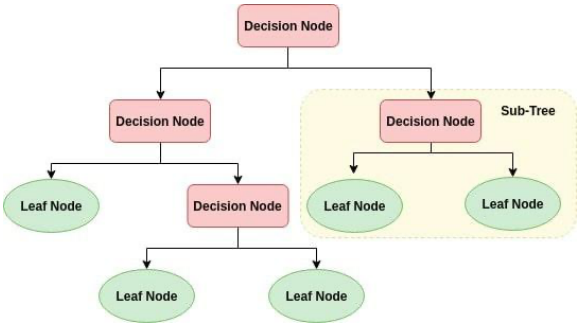

Figure 2. Decision Tree

2.3. *C4.5 Algorithm*

A popular algorithm for building decision trees that can be understood, flexible, and engaging because they can be displayed in the form of images. The steps in building a decision tree using the C4.5 Algorithm are as follows [17], [18]:

a. Prepare training data
Training data is obtained by splitting the overall data, one of which is with a percentage of 80% training data and 20% testing data.

b. Calculating entropy values
The entropy value of each attribute is using the following equation:

$$Entropy(S) = -\sum_{i=1}^{n} pi * log_2\ pi \quad (1)$$

S: case set
n: number of partitions S
pi: the proportion of Si to S

c. Determining the highest gain

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (2)$$

S: case set
A: attribute
n: number of attribute partition A
|Si|: number of cases on partition i
|S|: number of cases in S

d. Repeat steps b and c for each branch until all cases in the branch have the same class.

2.4. *Scikit Learn Decision Tree*

Scikit-learn, or Sklearn, is a machine learning software library for the Python programming language. It features a variety of classification, regression, and grouping algorithms, including vector support engines, random forests, gradient enhancements, k-means, and DBSCAN. It is designed to operate with the numerical and scientific libraries of Python NumPy and SciPy [19].

Step Decision Tree in Scikit Learn [20] as shown in Figure 3:

a. Splitting data, data is divided into training data and testing data.
b. Choose the best attribute based on the highest entropy and gain values.
c. The construction of a decision tree or rule with the repetition of the calculation of the entropy value and gain.
d. Evaluate the model based on the decision tree that has been built.
e. Performance Evaluation Measures are accuracy, precision, and recall.
f. The evaluation is generated from a confusion matrix that shows the compatibility between actual decision and decision prediction according to the results of the Algorithm [21]. Precision is the precision between actual decisions and decisions predicted in the system. Recall is the success rate of the system in rediscovering information.

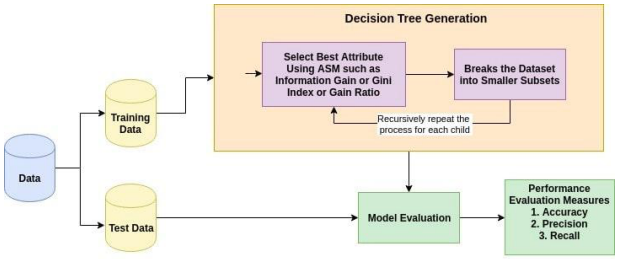

Figure 3. Flow modeling predictions with decision trees

Decision tree building with Scikit Learn python programming:

a. Importing Required Libraries
Import the library to be used, such as pandas, numpy, matplotlib.
b. Loading Data
Input data with file .csv format and adjust the data type of each attribute.
c. Feature Selection
Choose features that will be the feature variable and the target variable.
d. Splitting Data
Perform data splitting using the train_test_split() function. The required parameters are feature variables, target variables, and testing data sizes.
e. Building Decision Tree Model
Build a decision tree model using the DecisionTreeClassifier() function. The required parameters are the feature and target variables in the training data. Creation of a subsequent decision tree model for the prediction of testing data. Performance optimization by determining the criterion to be used, whether 'gini' or 'entropy' is the best, and determining the tree's depth.
f. Evaluating Model
Confusion matrix calculations for model evaluation are accuracy, precision, and recall.
g. Visualizing Decision Trees
Visualize the decision tree by installing graphviz and pydotplus to display the decision tree graph that has been built.

3. Results and Discussion

3.1. *Decision Tree Calculation*

This study used 262 data that were split in the 80:20 random. So that 209-training data were obtained, the data distribution as in Table 1.

Table 1. Training data distribution

SKS	IPK	TOEFL	Decision
144	3.48	610	No
144	3.02	507	No
144	3.15	473	No
…	…	…	…

146	3.51	503	No
144	3.57	560	No
149	3.54	507	No

3.2. *Calculating Entropy Values*

The attributes of SKS, IPK, TOEFL of each class are calculated using the entropy Formula 1. Results from calculations as in Table 3 to Table 9.

3.3 Determining the highest gain

Calculation of Gain (S, A) based on Formula 2 of each attribute, as shown in Table 2:

Gain (Sum, SKS)

$$0{,}89200271 - ((\frac{178}{209} \times 0{,}91507296) + (\frac{80}{31} \times 0{,}993234))$$

Gain (Sum, IPK)

$$0{,}89200271 - ((\frac{80}{209} \times 0{,}91507296) + (\frac{129}{31} \times 0{,}993234))$$

Gain (Sum, TOEFL)

$$0{,}89200271 - ((\frac{6}{209} \times 0{,}91507296) + (\frac{203}{31} \times 0{,}993234))$$

Table 2. Determination of highest gain

Node	Attributes	Number of Cases	Gain
1		209	
	SKS	178	0.00805511
		31	
	IPK	80	0.04195 1736
		129	
	TOEFL	6	0.00289 2353
		203	

Based on the calculations of Table 3 on node 1, it is known that the GPA attribute has the highest gain value with a value of 0.041951736. GPA attributes with classes 3-3.49 and >=3.5 will be the root of the decision tree. The calculation of node 1 will continue to create the following node until all attributes have the same class. From the calculation process, an overview of the decision tree is obtained, as shown in Figure 4.

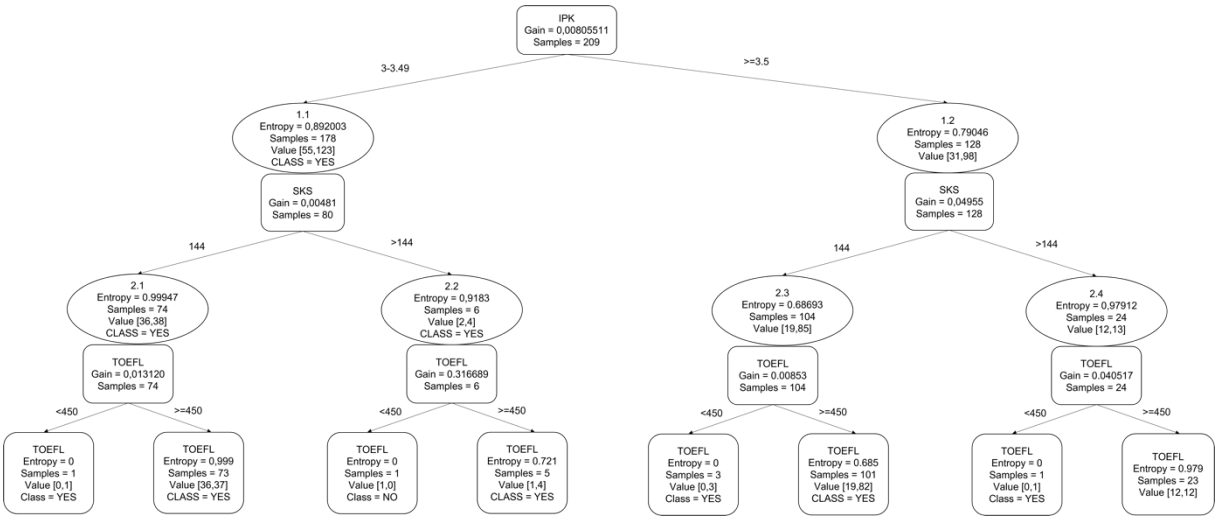

Figure 4. Decision tree with calculations

Table 3. Entropy calculation node 1, LTW (Pass on Time) and LTTW is the opposite

Node	Attributes	Class	Number of Cases (s)	LTTW	LTW	Entropy	Gain
1			209	69	140	0.91507	
	SKS	144	178	55	123	0.892	0.00805511
		>144	31	14	17	0.99323	
	IPK	3-3.49	80	38	42	0.9982	0.041951736
		>=3.5	129	31	98	0.79556	
	TOEFL	<450	6	1	5	0.65002	0.002892353
		>=450	203	68	135	0.91993	

Table 4. Entropy node 2.1 calculation

Node	Attributes	Class	Number of Cases (s)	LTTW	LTW	Entropy	Gain
2.1	IPK	3-3.49	80	38	42	0.9982	
	SKS	144	74	36	38	0.99947	0.004811148
		>144	6	2	4	0.9183	
	TOEFL	<450	2	1	1	1	0.0000525
		>=450	78	37	41	0.9981	

Table 5. Entropy node 3.1 calculation

Node	Attributes	Class	Number of Cases (s)	LTTW	LTW	Entropy	Gain
3.1	SKS	144	74	36	38	0.99947	
	TOEFL	<450	1	0	1		0.013120071
		>=450	73	36	37	0.99986	

Table 6. Entropy node 3.2 calculation

Node	Attributes	Class	Number of Cases (s)	LTTW	LTW	Entropy	Gain
3.2	SKS	>144	6	2	4	0.9183	
	TOEFL	<450	1	1	0		0.316689088
		>=450	5	1	4	0.72193	

Table 7. Entropy node 2.2 calculation

Node	Attributes	Class	Number of Cases (s)	LTTW	LTW	Entropy	Gain
2.2	IPK	>=3.5	128	31	98	0.79046	
	SKS	144	104	19	85	0.68593	0.049552149
		>144	24	12	13	0.97912	
	TOEFL	<450	4	0	4	0	0.012619129
		>=450	124	31	94	0.80293	

Table 8. Entropy node 3.3 calculation

Node	Attributes	Class	Number of Cases (s)	LTTW	LTW	Entropy	Gain
3.3	SKS	>144	24	12	13	0.97912	
	TOEFL	<450	1	0	1		0.040517052
		>=450	23	12	12	0.97941	

Table 9. Entropy node 3.4 calculation

Node	Attributes	Class	Number of Cases (s)	LTTW	LTW	Entropy	Gain
3.4	SKS	144	104	19	85	0.68593	
	TOEFL	<450	3	0	3		0.008532747
		>=450	101	19	82	0.69752	

Rules made of manual calculation decision trees are:

1. IF IPK 3-3.49 AND SKS 144 AND TOEFL < 450, THEN LTW
2. IF IPK 3-3.49 AND SKS 144 AND TOEFL >= 450, THEN LTW
3. IF IPK 3-3.49 AND SKS > 144 AND TOEFL < 450, THEN LTTW
4. IF IPK 3-3.49 AND SKS >144 AND TOEFL >= 450, THEN LTW
5. IF IPK >= 3.5 AND SKS 144 AND TOEFL < 450, THEN LTW
6. IF IPK >= 3.5 AND SKS 144 AND TOEFL >= 450, THEN LTW
7. IF IPK >= 3.5 AND SKS > 144 AND TOEFL < 450, THEN LTW
8. IF IPK >= 3.5 AND SKS >144 AND TOEFL >= 450, THEN LTW

3.4 Decision Tree with C4.5 algorithm

Decision tree building with Scikit Learn Python programming:

a. Importing Required Libraries
   Import libraries to be used, such as pandas, numpy, and matplotlib.
b. Loading Data
   Student data will be used to become training data, and testing data is 262. The attributes used are SKS, GPA/IPK, and TOEFL. The data is entered into the system for processing, as shown in Table 10.
c. Feature Selection
   The selection of features that will be feature variables is SKS, GPA, and TOEFL, and the target variable is Decision.

Table 10. Overall Dataset of 262 data

SKS	IPK	TOEFL	Decision
144	3.63	400	Yes
145	3.67	400	Yes
144	3.53	404	Yes
…	…	…	…
144	3.04	660	No
144	3.39	674	Yes
144	3.48	674	Yes

d. Splitting Data

Data splitting uses 80% for training and 20% for testing data. The training data taken was 80% of 262 data, as many as 209 data, while the testing data took 20% was 53 data, as shown in Tables 11 and 12.

Data splitting is done to avoid overfitting. Overfitting is when the model overfits its training data and fails to fit the additional data reliably. Random data sampling aims to keep the modeling process from partial data towards the possibility of different data characteristics.

Table 11. Data Training 209 Data

SKS	IPK	TOEFL	Decision
144	3.37	570	No
144	3.35	524	Yes
144	3.65	510	Yes
…	…	…	…
144	3.8	517	Yes
145	3.71	514	Yes

Table 12. Data Testing 53 data

SKS	IPK	TOEFL
144	3.48	630
144	3.53	600
…	…	…
144	3.61	627
144	3.81	564
144	3.86	520

Table 13. Results of the evaluation of the model with criterion "gini"

Parameters	Value = 0,442			
	Accuracy	Precision	Recall	F1
Criterion “gini” Max_depth=3	71%	63	58	58.5
Criterion “gini” Max_depth=5	77%	74.5	64	65.5
Criterion “gini” Max_depth=10	75%	74.5	64	65.5

Table 14. Hasil evaluasi model dengan criterion “entropy”

Parameters	Value = 0,915			
	Accuracy	Precision	Recall	F1
Criterion “entropy” Max_depth=3	73%	66.5	59.5	59.5
Criterion “entropy” Max_depth=5	75%	70	62.5	63.5
Criterion “entropy” Max_depth=10	75%	66.5	61	62.5

This criterion with max_depth 5 has a higher accuracy value than other parameters, so this parameter will be used in constructing decision trees, as shown in Tables 13 and 14.

The criteria parameter in the DecisionTreeClassifier function has two criteria: "gini" and "entropy". The Gini index has a value in the interval [0, 0.5] while the Entropy interval is [0, 1]. As seen in Table 13, the value of the "gini" criterion is 0.442 with the root of the decision tree, namely the GPA attribute. Similar to table 14, which uses entropy criteria, the root attribute of the decision tree is GPA but with a different entropy value from the "gini" value, which is 0.915 because the value in the entropy interval is [0.1]. In terms of processing, entropy is more complex because it uses logarithms so that the calculation of the Gini Index will be faster.

In addition to the parameter criteria, the DecisionTreeClassifier() function determines the maximum depth of the tree, namely max_depth. The max_depth experiments used were 3, 5, and 10. The resulting accuracy with the "gini" criterion was higher than the entropy criterion. The best max_depth consideration was the decision of a tree with a maximum tree depth of 5, so the best accuracy was obtained, namely 77%.

e. Evaluating Model

Confusion matrix calculations for model evaluation are accuracy, precision, and recall.

Confusion Matrix

N P ← sebagai klasifikasi
[[5 10] | LTTW
[2 36]] | LTW

$$Confusion\ Matrix = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) * 100\%$$

$$= \left(\frac{36+5}{36+5+10+2}\right) * 100\%$$

$$Akurasi\ = 77\%$$

f. Visualizing Decision Trees

Visualization of the decision tree with install graphviz and pydotplus to display the graph of the decision tree that has been built can be seen in Figure 5. The decision tree shows the class names No (LLTW) and Yes (LTW).

Based on the mining that has been done, the rules obtained from the decision tree on the Scikit Learn Decision Tree are as follows:

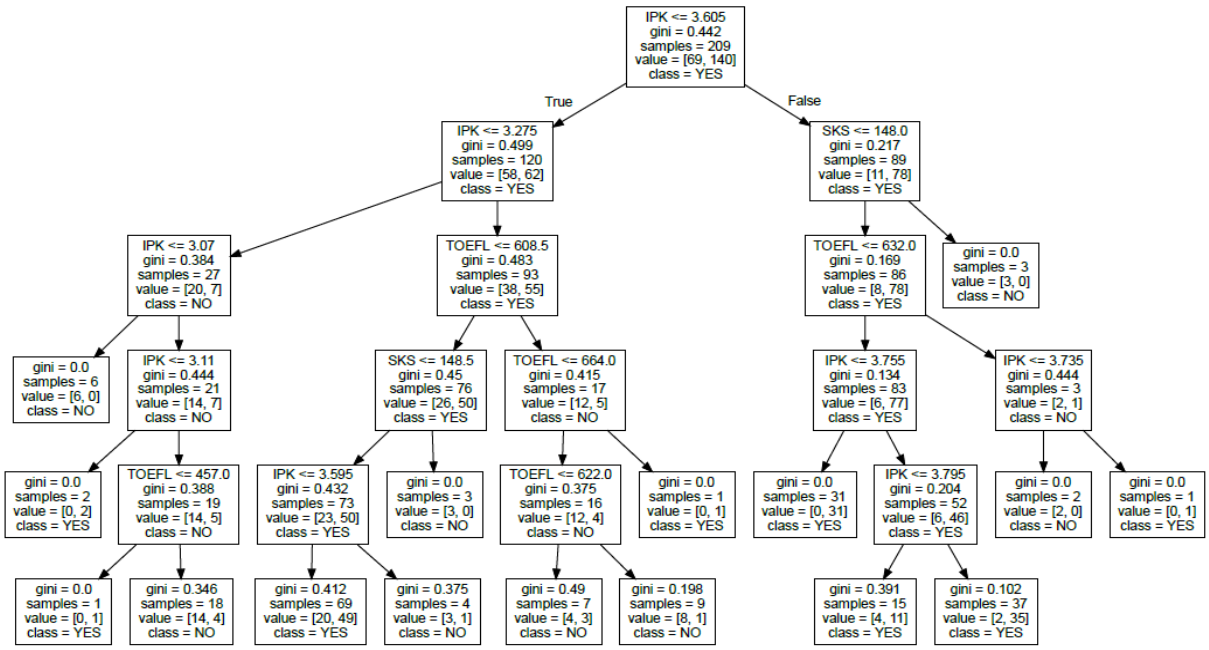

Figure 5. Decision Tress with data mining

1. IF IPK <= 3.60 AND IPK <= 3.27 AND IPK <=3.07 THEN class: NO
2. IF IPK <= 3.60 AND IPK <= 3.27 AND IPK > 3.07 AND IPK <= 3.11 THEN class: YES
3. IF IPK <= 3.60 AND IPK <= 3.27 AND IPK > 3.07 AND IPK > 3.11 AND SKS <= 457 THEN class: YES
4. IF IPK <= 3.60 AND IPK <= 3.27 AND IPK > 3.07 AND IPK > 3.11 AND SKS > 457 THEN class: NO
5. IF IPK > 3.27 AND TOEFL <= 608.50 AND SKS <=148.50 AND IPK <= 3.59 THEN class: YES
6. IF IPK > 3.27 AND TOEFL <= 608.50 AND SKS <=148.50 AND IPK > 3.59 THEN class: YES
7. IF IPK > 3.27 AND TOEFL <= 608.50 AND SKS > 148.50 THEN class: NO
8. IF IPK > 3.27 AND TOEFL > 608.50 AND TOEFL <= 664 AND TOEFL <= 622 THEN class: NO
9. IF IPK > 3.27 AND TOEFL > 608.50 AND TOEFL <= 664 AND TOEFL > 622 THEN class: NO
10. IF IPK > 3.27 AND TOEFL > 608.50 AND TOEFL > 664 AND THEN class: NO
11. IF IPK > 3.60 AND SKS <= 148 AND TOEFL <=632 AND IPK <= 3.75 THEN class: YES
12. IF IPK > 3.60 AND SKS <= 148 AND TOEFL <=632 AND IPK > 3.75 AND IPK <=3.79 THEN class: YES
13. IF IPK > 3.60 AND SKS <= 148 AND TOEFL <=632 AND IPK > 3.75 AND IPK >3.79 THEN class: YES
14. IF IPK > 3.60 AND SKS <= 148 AND TOEFL > 632 AND IPK <= 3.74 THEN class: NO
15. IF IPK > 3.60 AND SKS <= 148 AND TOEFL > 632 AND IPK > 3.74 THEN class: YES
16. IF IPK > 3.60 AND SKS > 148 THEN class: NO

The C4.5 Algorithm process between manual and coding has similarities at the root of the decision tree, namely the GPA attribute. The difference between the two is in the GPA value, where in the manual tree, the GPA value as the root of the decision tree is 3-3.49. At the same time, in the sci-kit learn decision tree, the GPA value is <= 3.65. The difference in value is due to manual calculations making class intervals according to the Institut Teknologi Telkom Purwokerto (ITTP) graduation requirements. The process of testing the C4.5 decision tree rule with 5 test data samples is shown in Table 15.

Table 15. Validation of prediction with 5 samples

SKS	IPK	TOEFL	Decision	Predicted
144	3.53	600	Yes	Yes
144	3.61	627	Yes	Yes
145	3.64	437	Yes	No
144	3.32	580	No	Yes
144	3.86	520	Yes	Yes

Based on the predictions according to decision tree rules. It can be seen from the sample test data that with a credit score of 145, GPA of 3.64, and TOEFL 437, the actual decision is LTW, while the prediction is LTTW. This is because the TOEFL score obtained is less than 450, so the prediction results that students who have this data will be LTTW. As for a reason for the actual decision YES, this was because the student had taken the English language test (TOEFL) 3 times. However, his score was still less than 450 on three occasions.

4. Conclusion

This study concludes that predictions using the C4.5 algorithm obtained the GPA attribute as the dominant attribute in predicting time passes. The puck tree between manual and system calculations has similarities in the root as the dominant attribute in predicting student graduation, namely GPA. In addition, the model tests that have been carried out show the best accuracy results of 77% with the parameters in DecisionTreeClassifier(), namely criterion='gini' and max_depth=3. The suggestion for the subsequent research is adding more attributes required for the punctuality of student graduation.

References

[1] M. Pendidikan, D. A. N. Kebudayaan, and R. Indonesia, "Peraturan Menteri Pendidikan Dan Kebudayaan Nomor 03 Tahun 2020 Tentang Standar Nasional Perguruan Tinggi," no. 47, 2020.

[2] L. Y. Lumban Gaol, M. Safii, and D. Suhendro, "Prediksi Kelulusan Mahasiswa Stikom Tunas Bangsa Prodi Sistem Informasi Dengan Menggunakan Algoritma C4.5," *Brahmana : Jurnal Penerapan Kecerdasan Buatan*, vol. 2, no. 2, pp. 97–106, 2021, doi: 10.30645/brahmana.v2i2.71.

[3] Y. Mardi, "Data Mining : Klasifikasi Menggunakan Algoritma C4.5," *Edik Informatika*, vol. 2, no. 2, pp. 213–219, 2017, doi: 10.22202/ei.2016.v2i2.1465.

[4] T. Widodo, S. Maghfiroh, S. H. B. Ginting, A. Aryaputra, and S. Sudianto, "Prediction of Covid-19 Cases in Central Java using the Autoregressive (AR) Method," *Data Science*, no. 1, 2023.

[5] T. K. Putri, M. L. Arnumukti, K. Khatimah, E. Zalsabila, and S. Sudianto, "Diabetes Diagnostic Expert System using Website-Based Forward Chaining Method," *Data Science*, vol. 3, no. 1, 2023.

[6] Sudianto, Y. Herdiyeni, A. Haristu, and M. Hardhienata, "Chilli quality classification using deep learning," in *2020 International Conference on Computer Science and Its Application in Agriculture, ICOSICA 2020*, 2020. doi: 10.1109/ICOSICA49951.2020.9243176.

[7] S. Sudianto and R. D. Wahyuningrum, "Identifikasi Sebaran Nitrogen pada Tanaman Padi Berbasis Pengetahuan Fenologi dan Remote Sensing," *Jurnal Nasional Pendidikan Teknik Informatika (Janapati)*, vol. 11, no. 3, pp. 166–175, 2022.

[8] R. M. S. Adi and S. Sudianto, "Prediksi Harga Komoditas Pangan Menggunakan Algoritma Long Short-Term Memory (LSTM)," vol. 4, no. 2, pp. 1137–1145, 2022, doi: 10.47065/bits.v4i2.2229.

[9] Sudianto, Y. Herdiyeni, and L. B. Prasetyo, "Machine learning for sugarcane mapping based on segmentation in cloud platform," presented at the The 3rd International Conference on Engineering, Technology and Innovative Researches, Purwokerto, Indonesia, Purwokerto, Indonesia, 2023, p. 020001. doi: 10.1063/5.0132180.

[10] S. Sudianto, P. Wahyuningtias, H. W. Utami, U. A. Raihan, and H. N. Hanifah, "Comparison Of Random Forest And Support Vector Machine Methods On Twitter Sentiment Analysis (Case Study : Internet Selebgram Rachel Vennya Escape From Quarantine) Perbandingan Metode Random Forest Dan Support Vector Machine Pada Analisis Sentimen Twitt," *Jutif*, vol. 3, no. 1, pp. 141–145, 2022.

[11] S. Chandra Ayunda Apta, N. Trivetisia, N. A. Winanti, D. P. Martiyaningsih, T. W. Utami, and S. Sudianto, "Analisis Komparasi Algoritma Machine Learning untuk Sentiment Analysis (Studi Kasus: Komentar YouTube 'Kekerasan Seksual')," *Jurnal Pengembangan IT*, vol. 7, no. 2, pp. 80–84, 2022.

[12] S. Sudianto, A. D. Sripamuji, I. R. Ramadhanti, R. R. Amalia, J. Saputra, and B. Prihatnowo, "Penerapan Algoritma Support Vector Machine dan Multi-Layer Perceptron pada Klasifikasi Topik Berita," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 11, no. 2, pp. 84–91, 2022.

[13] S. Sudianto, "Analisis Kinerja Algoritma Machine Learning Untuk Klasifikasi Emosi," vol. 4, no. 2, pp. 1027–1034, 2022, doi: 10.47065/bits.v4i2.2261.

[14] J. Han and M. Kamber, *Data Mining: Concepts and Techniques, Second Edition*. 2006.

[15] D. D. Darmansah and N. W. Wardani, "Analisis Pesebaran Penularan Virus Corona di Provinsi Jawa Tengah Menggunakan Metode K-Means Clustering," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 1, pp. 105–117, 2021, doi: 10.35957/jatisi.v8i1.590.

[16] D. D. Darmansah, "Analisis Penyebaran Penularan Virus Covid-19 di Provinsi Jawa Barat Menggunakan Algoritma K-Means Clustering," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 3, pp. 1188–1199, 2021, doi: 10.35957/jatisi.v8i3.1034.

[17] S. Wibowo, R. Andreswari, and M. A. Hasibuan, "Analysis and Design of Decision Support System Dashboard for Predicting Student Graduation Time," 2018.

[18] Dikriani, Disty, and Alvina Tahta Indal Karim. "Comparison of C4. 5 and Naive Bayes Algorithm Methods in Prediction of Student Graduation on Time (Case Study: Information Systems Study Program)." *Journal of Dinda: Data Science, Information Technology, and Data Analytics* 3.1 (2023): 40-44.

[19] "NUMFOCUS." https://numfocus.org/sponsored-projects (accessed Feb. 04, 2023).

[20] Avinash Navlani, "Decision Tree Classification in Python Tutorial," 2018. https://www.datacamp.com/tutorial/decision-tree-classification-python (accessed Feb. 04, 2023).

[21] G. F. Mandias, "Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Di Universitas Klabat Dengan Metode Klasifikasi," *Konferensi Nasional Sistem & Informatika*, p. 20, 2015.