

# Analyzing Student Performance in Programming Education Using Classification Techniques

<https://doi.org/10.3991/ijet.v15i02.11527>

Kissinger Sunday (✉)

Usmanu Danfodiyo University, Sokoto, Nigeria  
kissinger.sunday@udusok.edu.ng

Patrick Ocheja

Kyoto University, Kyoto, Japan

Sadiq Hussain

Dibrugarh University, Assam, India

Solomon Sunday Oyelere, Oluwafemi Samson Balogun,

Friday Joseph Agbo

University of Eastern Finland, Kuopio Finland

**Abstract**—In this research, we aggregated students log data such as Class Test Score (CTS), Assignment Completed (ASC), Class Lab Work (CLW) and Class Attendance (CATT) from the Department of Mathematics, Computer Science Unit, Usmanu Danfodiyo University, Sokoto, Nigeria. Similarly, we employed data mining techniques such as ID3 & J48 Decision tree algorithms to analyze the data. We compared these algorithms on 239 classification instances. The experimental results show that the J48 algorithm has higher accuracy in the classification task compared to the ID3 algorithm. The important feature attributes such as Information Gain and Gain Ratio feature evaluators were also compared. Both the methods applied were able to rank search methods. The experimental results confirmed that the two methods derived the same set of attributes with a slight deviation in the ranking. From the results analyzed, we discovered that 67.36 percent failed the course titled “Introduction to Computer Programming”, while 32.64 percent passed the course. Since the CATT has the highest gain value from our analysis; we concluded that it is largely responsible for the success or failure of the students. Recommendations were given on how to improve the failure rates in the future.

**Keywords**—Learning Analytics, Educational data mining, Programming education, Classification algorithms.

## 1 Introduction

Over the years, there has been an increase in the use of technology to capture digital data on learner’s interest. According to [1] Learning Analytics (LA) is an egressing

research area that tends to improve the learning outcome of students by developing methods to analyze and detect patterns and to infer changes with an ultimate goal- to improve learning. The application of data mining (DM) in educational settings gave rise to the field of Learning Analytics [2]. Recently, there has been an exponential research growth for harnessing and utilizing data mining techniques for scientific research in educational settings giving rise to a field known as educational data mining [3]. Educational data mining simply refers to the process by which new techniques are been developed to discover data emanating from educational settings which are then used to understand the behavior of students and the environment they learn in. [4]. The different techniques used for data mining can be employed for educational data mining (see Figure 1). Similarly, various classification algorithms are explored in data mining. Figure 2 depicts the various classification algorithms used in data mining. Educational data mining also explored the same algorithms to find hidden information from the datasets and also may be applied in predicting at-risk students and preventing dropouts.

Studying computer programming as a course is both challenging and daunting and the few privileged students who studied the course found it uninteresting after some time. The success rate in average for the first introductory programming course, denoted CS1, has been estimated to be 67% worldwide [5]. To motivate students to succeed and master a programming course, several researchers have become interested in looking for factors that can make the teaching and learning of computer programming interesting. In particular, computer scientist has been researching for ways to explore various features, behavior and performance of students for the sole purpose of identifying weak students. For instance, the use of mobile learning to support students in programming education has been explored by Oyelere et al. [6].

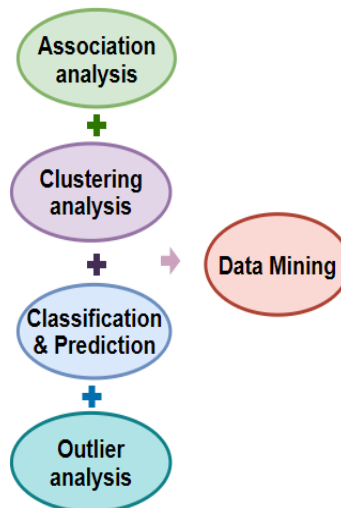
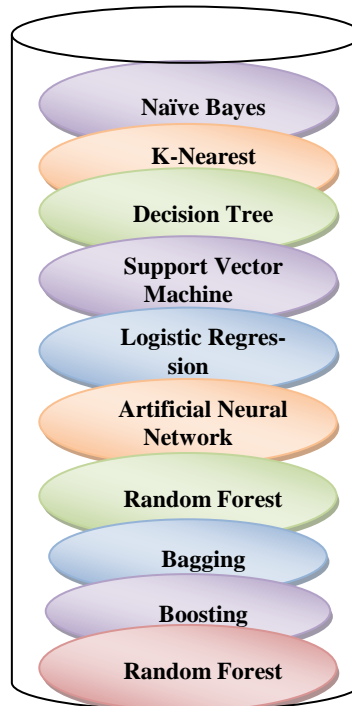


Fig. 1. Techniques used in data mining



**Fig. 2.** Classification Algorithms used with Data Mining Technology

Similarly, the use of context-aware and adaptive system, called smart learning environment for programming education [7] was proposed to enhance students learning experience. The smart learning environment introduces a blend of formal and informal learning in which the learner can learn from any location, based on learning preference and context. Recently, researchers have adopted a paradigm shift to a more data-driven approach by studying and analyzing programming patterns and behavior of students. This includes patterns in programming and compilation states thus, making it more efficient and effective at reflecting the effort and progress of the students for the entire course duration.

This paper studies the performance of the student's in a programming course using data mining techniques. Data mining techniques are frequently used to study and subsequently analyze the performance of students in programming by utilizing the rich tasks it provides. We employ the classification task in this research to evaluate and subsequently improve student's performance in programming. We also employ the decision tree method of data mining in this paper. This is as a result of its high accuracy level for predicting student performance [8].

The challenges that students encounter in a programming course has been topical and created concern for educators and researchers in the recent time. Efforts to make the programming easier to learn have been explored at different context. For instance, the use of games and gamification [9], puzzle-based techniques [10], and other peda-

gical approach are explored to support students for a better learning experience. In the context of Nigeria, the use of mobile learning system—MobileEdu-puzzle [11], shows that the learning experience of students in introductory programming class were enhanced. Although, these efforts exist, it is important to further analyze student's performance in a programming course, using some variables as factors that are capable of impacting their grades. Important information such as attendance of student continuous assessment, class quizzes, and marks are useful for such analysis. In other to achieve the objectives of the study we considered two research questions. RQ1: What fine-grained programming logs data should be aggregated to study their effect on student's performance? RQ2: How will the performance of students be analyzed? This research aimed at mining educational data to analyze student's performance in the introductory programming course in Nigeria's context. The following are the objectives of this study:

- Aggregate fine-grain log programming student's data in order to study their effect on the performance of students
- Employ data mining techniques such as Decision tree algorithm to analyze students' performance
- Compare ID3 and J48 algorithms result in the instance classification.

## **2 Related Work**

Data mining can be defined according to [12] as the process of extracting inexplicit, unknown and useful information from data. Mining data is often used in finding structural patterns in data, it forms a strong basis in making predictions. Educational data mining, however, deals with the methods and techniques for extracting knowledge from educational data. Although this research area, however nascent, is beginning to gain popularity by the day. At present, there are many ongoing kinds of research in this area. This is due to its potential in educational institutions. A 1995 and 2005 survey about educational data mining presented by [2], found out the importance of mining educational data and hence, encouraged researchers to explore this nascent field. They concluded by discovering some specific requirements not presented in other domains. A case study presented by [13] demonstrated the importance of mining educational data in higher education especially for the improvement of graduate student results. In their research, the data set from the College of Science and Technology in Khanyounis from 1993-2007 was used. Knowledge discovery was achieved through the application of data mining techniques. In particular, they were able to discover association rules which were then sorted using lift metric. Data mining classification methods such as Naive Bayesian and Rule induction were used to predict the performance of graduate students. A recent survey on mining educational data carried out by [14] proposed a taxonomy of tasks in educational data mining and moved further in their research by grouping similar applications into sub-categories and categories respectively. They concluded their research by reviewing existing surveys and books about educational data mining.

A research carried out by [15] titled Modeling student performance using data mining; they developed software for mining educational data to improve the rate at which students succeed using student profiling. The application made use of the data that was generated from the university domain. According to their findings, the success of the system was slightly distorted because the data set expected to be in some columns were not there. They were, however, able to deduce that increasing the number of variables and the amount of data will lead to better predictions about the success rate of students.

A systematic review of existing literature on predicting student's performance based on the techniques and methods of data mining was conducted by [16]. The researchers highlighted the data mining methodologies used for the prediction of student performance. More importantly, they focused on the algorithm and how it can be used to discover the most important feature in student data. The classification task was used to evaluate the performance of students in research [17]. In particular, they explored and used the decision tree method of data mining majorly due to its popularity and simplicity. By using this method, they were able to extract hidden knowledge that describes the performance of students in the final semester examination. The research helped in discovering students' dropout and more importantly those students in need of special attention and intervention such as advising/counseling.

### **3 Research Design**

In this research, we will employ the data mining processes that include data preparation, preprocessing, data selection, data transformation, data mining and evaluation (Fig. 3).

#### **3.1 Data preparation**

We extracted our data from the first-semester lecture of computer programming course (CSC 201) which spanned three months from September – November 2017 at Usmanu Danfodiyo University, Department of Mathematics, Computer Science Unit. The course, titled “Introduction to computer programming” teaches students the introductory aspect of programming; its syntax and semantics amongst other salient features. Computer Science majors are required to take the course while it is optional for some departments. The course material consists of three parts with each part corresponding to 1 month of the course. Two hours of weekly lectures were always provided by the teaching staff – mainly lecturer II including 20 hours of weekly support in the computer laboratory. The course outline contains for example, an overview of java programming, data types, variables, and arrays operators, control statements java classes, inheritance, packages and interfaces, etc. Two example programming tasks and one version of the solution is presented in Table I. At the end of each semester, the students were graded accordingly. The grading is as follows: Continuous assessment (20% of the total score), Assignments (5% of the total score), student attendance (5% of total score) and exams (70% of total score). For a student to pass a particular

course, such students should possess at least 40% of the total score. The participants who participated in the final exams totaling 239 provided data for this study.

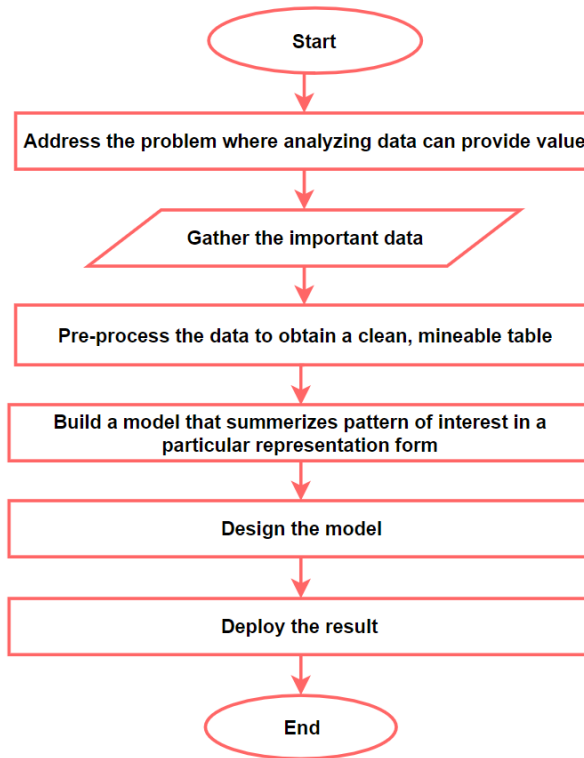


Fig. 3. Data mining steps

Table 1. Two examples of programming tasks and a version of the solution

Question	Question
Write a complete java program to display all the odd numbers between 1 and 20 inclusive.	You have been entrusted with the job of weather forecasting, and you have a device that can only read temperature in Celsius and you are expected to report in Fahrenheit. Write a complete java program that converts a temperature from degrees Celsius to Fahrenheit.
Example solution <pre> public class OddNumbers { public static void main(String args[]){/* The program starts here */ int x=1;// initialization variable while (x&lt;=20){/* loop con- dition until it is less than or equal to 20*/ System.out.println(x); /* </pre>	Example solution <pre> public class CelsiusToFahrenheit{ /* The program starts here */ public static void main(String args[]){ /* declaration of input variables */ int celsius=100; int product; double fahrenheit; /* conversion constants */ double SCALERATE = 9.0/5.0; </pre>

<pre>print the value of x */ x+=2; /* increment the value of x with 2 to */ } } /* The program ends here */</pre>	<pre>int TEMPSCALE = 32; /* Calculate the Fahrenheit equivalent */ fahrenheit = celsius * scaleRate + temp- Scale; /* Print the result (output) of the conver- sion */ System.out.println("The Fahrenheit equiva- lent of 100 degree Celsius is" + fahren- heit); }} /* The program ends here */</pre>
---	--

### 3.2 Selection and transformation of data

Here, we selected the required fields for data mining. Furthermore, we gave an overview of all the response variables as well as the predictor variable for reference purpose (Table II).

**Table 2.** Variables related to student

Variables	Representation	Values
FSG	First Semester Grade	{A, B, C, D, E, F}
CTS	Class Test Score	{Poor, Average, Good}
ASC	Assignment Completed	{Yes, No}
CATT	Class Attendance	{Poor, Average, Good}
CLW	Class Lab Work	{Yes, No}
SSG	Second Semester Grade	{A, B, C, D, E, F}

Key

- A=First Class>=70%
- B= Second Class Upper>=60%
- C=Second Class Lower>=50%
- D=Third Class>=45%
- E=Pass>=40%
- F=Fail>=0%

The domain values are defined below:

- **FSG** – First Semester Grade. We split this into six classes :{A, B, C, D, E, F}
- **CTS** – This signifies Class Test Score. We calculated CTS by conducting only written test. We have three categories of CTS:  
  - < 40%=Poor,
  - >= 40% and < 60%=Average
  - >=60%= Good.
- **ASC** – This stands for Assignment Completed. The lecturer always ensures that two assignments were given to the students at each semester; this is to ensure that the students are always kept busy at all times. We have two classes for the Assignment performance: *Yes* – assignment was submitted by the student, *No* – assignment was not submitted by the student.
- **CATT** – This stands for Class Attendance. As a requirement for participating in Second Semester examination a minimum of 75% attendance is compulsory to all participating students. However, students with a very poor attendance are normally

allowed to participate in second semester examinations based on genuine reasons. We have three categories of CATT:

$< 60\% = \text{Poor}$ ,  
 $\geq 60\% \text{ and } < 75\% = \text{Average}$   
 $\geq 75\% = \text{Good}$ .

- **CLW** –We have the following categorization for class Lab work  
*Yes – class lab work was completed by the student*  
*No – class lab work was not completed by the student.*
- **SSG**–This stands for Second Semester Grade and it is categorized as a response variable. Just like the FSG we categorize this into six class values: {A, B, C, D, E, F}

### 3.3 Using decision tree algorithm

Because of its powerful features, this algorithm is widely used for classification and prediction in both machine learning and data mining. One of the advantages of choosing this method is because the decision tree represents rules in contrast to using neural networks. Humans can readily understand and interpret these rules because of its simplicity and comprehensibility to uncover large or small data structure and predict them [18].

Decision tree usually is a flowchart classifier just like the tree data structure where

- A test on an attribute is denoted by a non-leaf node
- An outcome of the test is represented by the tree branch
- A value of the target attribute indicates a terminal node
- The topmost node in a tree is the root node

We decided to choose the decision tree algorithm because of the following strong features:

- High dimensional data can be easily handled with the decision tree
- Small-sized trees can easily be interpreted
- The steps to be followed to properly classify decision tree induction are fast

**ID3 decision tree:** To build our decision tree, we would use an algorithm developed by [19] known as ID3, which has been the primary algorithm from which decision trees are constructed. This algorithm makes use of a top-down, greedy search method to search through the space of possible branches with no provision of back-tracking. A tree based on the information gain is constructed. This information is obtained from the training instances, which is then used to classify the test data [20].

### 3.4 Attribute selection measures

This measure is responsible for determining the procedure to be followed in splitting the tuples at a given node. It also provides a ranking for every attribute that is



involved in the description of the training tuples. To choose the splitting attributes for particular tuples, we consider the attributes with the highest score.

- Information gain: This attribute selection measure is frequently used for selecting an attribute among the various attributes at each step while building the tree. To calculate the homogeneity of a sample, the ID3 algorithm employs a mechanism called entropy to calculate the homogeneity of a sample. The entropy for a sample that is homogenous is considered to be zero. For an equally divided sample, the sample remains one. We define the binary classification of an entropy of a set S (s contains only positive and negative examples) as:

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

In this case, the proportion of S belonging to class  $i$  is  $P_i$ . The reduction in entropy caused by partitioning the examples according to this attribute is measured using Information gain.

In this paper, we define  $0 \log_2 0$  to be 0 in all the calculations involving entropy.

For example, assuming S consists of 35 examples; this includes 10 negative and 25 positive examples [+25, -10]. Then the entropy of S is defined as:

$$Entropy(S) = -\frac{15}{25} \log_2 \left(\frac{15}{25}\right) - \frac{10}{25} \log_2 \left(\frac{10}{25}\right) = 0.970$$

From the above equation, the entropy of S becomes zero if every member belonging to S belongs to the same category.

$$Entropy(S) = -1 \log_2(1) - 0 \log_2(0) = -1 \times 0 - 0 \times \log_2(0) = 0$$

We present the information gain, Gain (S, A) of an attribute A. This attribute is relative to a collection of examples S:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

= information needed before splitting – information needed after splitting

In this case, the subset of S is  $S_v$  for which attribute A has value v (i.e.,  $S_v = \{s \in S \mid A(s) = v\}$ ) and Values (A) comprises all possible values for attribute A

C4.5 uses gain ratio to split the training data set S into various partitions in order to normalize the information gain using defined value as:

$$SplitInfoA(S, A) = -\sum_{i=1}^n \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \quad (3)$$

The value above represents the information that corresponds to n outcomes of a test on the attribute A. The gain ratio is defined thus;

$$Gain\ Ratio(S, A) = \frac{Gain(S, A)}{SplitInfoA(S, A)} \quad (4)$$

The highest gain ratio value was selected as the splitting attribute [9]. Relevant attributes are those attributes in the decision tree with non-leaf nodes. We therefore stated the algorithm of our decision tree as follows:

- i) An attribute will be selected if it best differentiates the output attribute.
- ii) For every selected attribute, create a separate tree branch.
- iii) Create subgroups from the instances in order to be a reflection of the attribute values of the selected node.
- iv) The attribute selection process should be terminated if:
  - a) There exists a value that is identical for the output attribute for all members of a subgroup, hence the process for selecting attribute for the current path should be terminated and the branch on the current path with the defined value should be properly labelled.
  - b) No further distinguishing in the single node can be determined or there exist a subgroup containing a single node. The branch with the output value that is seen by majority of the remaining instances should be labelled as in (a) above.
- v) Repeat the above process for every subgroup in (iii) that is not a terminal.

## 4 Results

We present the data set containing the results of  $n = 239$  students in programming courses (CSC 201) which were obtained from the Department of Mathematics, Computer Science Unit Usmanu Danfodiyo University (UDUS), Sokoto-Nigeria. The dataset used in this study covers 2016/2017 academic session (Table III presents sample data and Table IV present the frequency of the occurrence of each grade).

**Table 3.** First 20 records of UDUS dataset

S/No	FSG	CTS	ASC	CATT	CLW	SSG
1	F	Average	No	Poor	Yes	F
2	F	Average	No	Poor	Yes	F
3	C	Average	No	Poor	Yes	F
4	F	Average	No	Poor	Yes	F
5	F	Average	No	Poor	Yes	F
6	F	Good	No	Poor	Yes	F
7	F	Average	No	Poor	Yes	F
8	F	Average	No	Poor	Yes	F
9	F	Good	Yes	Poor	Yes	F
10	F	Good	Yes	Poor	Yes	F
11	F	Good	Yes	Poor	Yes	F
12	F	Good	Yes	Average	Yes	E
13	D	Average	No	Poor	Yes	F
14	F	Average	No	Poor	Yes	F
15	D	Average	No	Poor	Yes	F
16	F	Average	No	Poor	Yes	F
17	E	Average	No	Poor	Yes	F
18	E	Good	No	Poor	Yes	F
19	F	Average	No	Poor	Yes	F
20	F	Good	Yes	Average	Yes	D

**Table 4.** Frequency of Grade occurrence

Grade	Frequency	Percentage (%)
F	161	67.36
E	27	11.30
D	15	6.27
C	22	9.20
B	6	2.51
A	8	3.35
Total	239	100

From table IV, 67.36% failed the course while 32.64% passed the course as represented.

In order to know which node to use as our tree node; we need to mine the logs data. We therefore calculate the information gain but first we calculate the entropy of the various attributes. The information gain for A relative to S was calculated by first calculating the entropy of S. We presented S here as a set of 239 sets comprising the following:

$$\begin{aligned}
 \text{Entropy} &= -P_{fail}\log_2(P_{fail}) - P_{pass}\log_2(P_{pass}) - P_{thirdClass}\log_2(P_{thirdClass}) \\
 &\quad - P_{secondClassLower}\log_2(P_{secondClassLower}) \\
 &\quad - P_{secondClassUpper}\log_2(P_{secondClassUpper}) \\
 &\quad - P_{firstClass}\log_2(P_{firstClass}) \\
 &= -\frac{161}{239}\log_2\left(\frac{161}{239}\right) - \frac{27}{239}\log_2\left(\frac{27}{239}\right) - \frac{15}{239}\log_2\left(\frac{15}{239}\right) - \frac{22}{239}\log_2\left(\frac{22}{239}\right) - \\
 &\quad \frac{6}{239}\log_2\left(\frac{6}{239}\right) - \frac{8}{239}\log_2\left(\frac{8}{239}\right) = 1.603
 \end{aligned}$$

We determined the attribute that is best for a particular node by using information gain.

$$\begin{aligned}
 \text{Gain (S, FSM)} &= \text{Entropy}(S) - \frac{|S_F|}{|S|}\text{Entropy}(S_F) - \frac{|S_E|}{|S|}\text{Entropy}(S_E) - \\
 &\quad \frac{|S_D|}{|S|}\text{Entropy}(S_D) - \frac{|S_C|}{|S|}\text{Entropy}(S_C) - \frac{|S_B|}{|S|}\text{Entropy}(S_B) - \frac{|S_A|}{|S|}\text{Entropy}(S_A) = \\
 &\quad \frac{|S_f|}{|S|}\text{Entropy}(S_f) = \frac{134}{239}\left\{\frac{-6}{134}\log_2\left(\frac{6}{134}\right) - \frac{2}{134}\log_2\left(\frac{2}{134}\right) - \frac{11}{134}\log_2\left(\frac{11}{134}\right) - \right. \\
 &\quad \left. \frac{6}{134}\log_2\left(\frac{6}{134}\right) - \frac{13}{134}\log_2\left(\frac{13}{134}\right) - \frac{96}{134}\log_2\left(\frac{96}{134}\right)\right\} = 0.81823 \\
 &\quad \frac{|S_E|}{|S|}\text{Entropy}(S_E) = \frac{43}{239}\left\{\frac{-1}{43}\log_2\left(\frac{1}{43}\right) - \frac{2}{43}\log_2\left(\frac{2}{43}\right) - \frac{5}{43}\log_2\left(\frac{5}{43}\right) - \right. \\
 &\quad \left. \frac{2}{43}\log_2\left(\frac{2}{43}\right) - \frac{11}{43}\log_2\left(\frac{11}{43}\right) - \frac{22}{43}\log_2\left(\frac{22}{43}\right)\right\} = 0.3412 \\
 &\quad \frac{|S_D|}{|S|}\text{Entropy}(S_D) = \frac{12}{239}\left\{\frac{-1}{12}\log_2\left(\frac{1}{12}\right) - \frac{1}{12}\log_2\left(\frac{1}{12}\right) - \frac{1}{12}\log_2\left(\frac{1}{12}\right) - \right. \\
 &\quad \left. \frac{1}{12}\log_2\left(\frac{1}{12}\right) - \frac{8}{12}\log_2\left(\frac{8}{12}\right)\right\} = 0.07957
 \end{aligned}$$

$$\frac{|S_C|}{|S|} Entropy(S_C) = \frac{33}{239} \left\{ \frac{-1}{33} \log_2 \left( \frac{1}{33} \right) - \frac{1}{33} \log_2 \left( \frac{1}{33} \right) - \frac{3}{33} \log_2 \left( \frac{3}{33} \right) - \frac{2}{33} \log_2 \left( \frac{2}{33} \right) - \frac{2}{33} \log_2 \left( \frac{2}{33} \right) - \frac{24}{33} \log_2 \left( \frac{24}{33} \right) \right\} = 0.1995$$

$$\frac{|S_B|}{|S|} Entropy(S_B) = \frac{14}{239} \left\{ \frac{-2}{14} \log_2 \left( \frac{2}{14} \right) - \frac{2}{14} \log_2 \left( \frac{2}{14} \right) - \frac{10}{14} \log_2 \left( \frac{10}{14} \right) \right\} = 0.82252$$

$$\frac{|S_A|}{|S|} Entropy(S_A) = \frac{3}{239} \left\{ \frac{-2}{3} \log_2 \left( \frac{2}{3} \right) - \frac{1}{3} \log_2 \left( \frac{1}{3} \right) \right\} = 0.0119$$

Hence, Gain (S, FSM) = 1.603-0.81823-0.3412-0.07957-0.1995-0.82252-0.01193 = -0.66995

Hence performing the above for CTS, CATT, ASC and CLW we have:

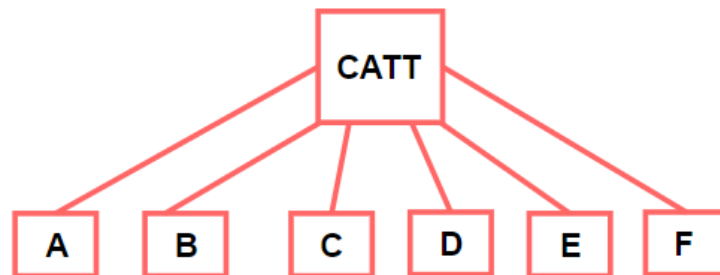
**Table 5.** Subset of FSG

		SSG						
		A	B	C	D	E	F	Total
FSG	A	0	0	0	2	0	1	3
	B	0	0	2	2	0	10	14
	C	1	1	3	2	2	24	33
	D	0	1	1	1	1	8	12
	E	1	2	5	2	11	22	43
F		6	2	11	6	13	96	134
Total		8	6	22	15	27	161	239

**Table 6.** Table VI. Gain value

Gain	Value
(S,FSG)	-0.66995
(S, CTS)	0.11960
(S, ASC)	0.12464
(S, CATT)	0.97955
(S, CLW)	0.01680

We therefore use CATT as the root node due to its highest gain value.



**Fig. 4.** Root Node - CATT

We also select attribute using the gain ratio, after split information must have been calculated. Hence, split information is represented in table VII.

**Table 7.** Table VII. Split Information

Split	Value
(S,FSG)	2.27295
(S, CTS)	1.45340
(S, ASC)	1.47836
(S, CATT)	0.62345
(S, CLW)	1.5862

The gain ratio is presented in Table VIII.

**Table 8.** Gain Ratio

Gain Ratio	Value
(S,FSG)	-0.29475
(S, CTS)	0.10293
(S, ASC)	0.12431
(S, CATT)	1.57118
(S, CLW)	0.01059

If FSG =A AND CLW=Yes or No AND ASC=Yes or No AND CATT=Average AND CTS=Average OR Good THEN SSG=D  
 If FSG =A AND CLW='Yes' OR 'No' AND ASC='Yes' OR 'No' AND CATT=Average AND CTS=Average OR Good THEN SSG=E  
 If FSG =B AND CLW='Yes' OR 'No' AND ASC='Yes' OR 'No' AND CATT=Average AND CTS=Average OR Good THEN SSG=D  
 If FSG =B AND CLW='Yes' OR 'No' AND ASC='Yes' OR 'No' AND CATT=Poor AND CTS=Average OR Good OR Poor THEN SSG=F  
 If FSG =B AND CLW='Yes' OR 'No' AND ASC='Yes' OR 'No' AND CATT=Good AND CTS=Average OR Good THEN SSG=C

**Fig. 5.** Examples of If-Then rules

We continue this process until all the data are classified or all the attributes are exhausted. This knowledge is represented in the form of IF-THEN rules in Fig. 5

**Table 9.** Comparison of important feature selection methods

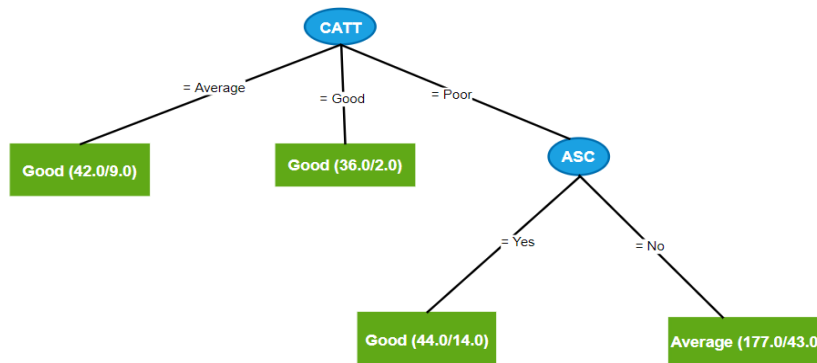
Information Gain Ranking Filter		Gain Ratio feature evaluator	
Ranking	Attributes	Ranking	Attributes
1.2361	CATT	1	CATT
0.1255	ASC	0.1298	ASC
0.1196	CTS	0.1225	CTS
0.0871	FSG	0.0473	FSG
0.0167	CLW	0.0245	CLW

In Table IX, two different feature selection methods were applied. They were Information Gain and Gain Ratio feature evaluators. Both the methods applied rank search method. The most influential features found were CATT, ASC, CTS, FSG and CLW. The experimental results confirmed that the two methods derived the same set of attributes with a slight deviation in ranking.

**Table 10.** Comparison of ID3 and J48 Classification methods

Classifier	Precision	Recall	F-Score	Accuracy	Kappa Statistics
J48	0.801	0.87	0.828	87.02	0.747
ID3	0.848	0.861	0.851	85.355	0.727

Total number of instances considered for the classification task was 239. Table X compares the J48 and ID3 algorithm results. In the J48 classification, the correctly classified instances were 208 and the incorrectly classified instances were 31. The accuracy was found to be 87.02%. The mean absolute error was 0.0563 and the root mean squared error was 0.1779. While the relative absolute error was 32.0058% compared to the root relative squared error was 60.4193%. Whereas in the ID3 classification, the correctly classified instances were 204 and incorrectly classified instances were 35. The accuracy was found to be 85.35%. The mean absolute error was 0.0511 and the root mean squared error was 0.1874. While the relative absolute error was 29.4579% compared to the root relative squared error of 64.3234%.



**Fig. 6.** Visualization tree of the J48 classification result based on Weka classifier

The decision tree method makes use of attribute selection measure like gain ratio and information gain equations (2) and (4) which were discussed in section 3.4. Having established the parameter for splitting the various attributes, we then employed the use of the Weka classifier [21] to create a visualization tree of the J48 classification result. This classifier was able to reduce the over fitting associated with building a decision tree as well as pruning. For instance, out of five distinct attributes, CATT, and ASC attributes were only shown, the remaining attributes like CTS, FSG and CLW were pruned from the tree based on attribute selection measure of information gain. Attribute is represented by ellipse whereas actual attribute values are represented

in rectangular box with specification of numbers. It is also posited that the top most node represents the root node and is labeled CATT from figure 6 above. This node was automatically selected by the J48 classifier following by the ASC node.

## **5 Discussion**

This section will discuss the analyzed result on student performance. This analysis is based on the highest accuracy of classification methods and also the main important factors that may influence the performance of students. Table X shows the classification accuracy of both ID3 and J48 classification methods. From the table, it is obvious that J48 has higher accuracy as compared with ID3 technique by 1.6644%. By looking at Table IV, a higher percentage of the students failed the course titled Introduction to Computer Programming by 67.36%. This is strongly connected with the avoidance of lectures by the students as represented in Table VI. In this table, the Class Attendance (CATT) attribute has the highest information gain making it a suitable candidate as the root node. The next highest attribute is the Student Assignment (ASC) as shown in the same table with the next highest information gain value. This was further collaborated in Figure 6 in which the Weka classifier was automatically select CATT as the root node in the tree followed by the ASC. One unique feature of this classifier is the ability to automatically perform both pruning and over fitting of the various attributes. We also represented the classification result in the form of If-Then rules as represented in figure 5. One strong feature of the decision tree is its ability to represent information in the form of if-then rules [17]. The result clearly shows that class attendance plays an essential role in enhancing student's performance, as 75% attendance is compulsory in order to write the exams. However, student might also write the exam if the attendance is below 75% on genuine reasons such as sickness, accident, etc. Such reasons however, must be backed by a valid document. Any student found wanting in this aspect is awarded a failed (F) grade. Probably the reasons why student were absent in class might be connected to the difficult nature of programming courses, "novice programmers find it difficult to remember and correctly apply programming language vocabulary, logic, errors, syntax, semantics, and styles" [22], or contextual reasons [23].

## **6 Conclusion and Future Work**

The paper shows the usefulness of data mining particularly in the domain of tertiary education in analyzing the performance of undergraduate students. Data was gathered from Usmanu Dandodiyo University Sokoto, Department of Mathematics, Computer Science Unit, introductory programming course. The various techniques of data mining were applied in discovering hidden knowledge. In particular, we used the decision tree method to analyze the data set in which we discovered that the Class Attendance (CATT) plays a major role in determining the success or failure rate of students. This study will be of benefits to both lecturers and students. In particular, struggling student will be easily identified through his/her attendance rate in class.

Furthermore, intervention techniques such as counseling, advising will come in handy for students who feel they can pass the course without attending the class. This will go a long way in boosting the academic performance of students. The future direction of this research will look at the possible factors that lead to the massive failure of students in programming courses in Usmanu Danfodiyo University, Sokoto, Nigeria with a view to ameliorating the problem. It has however been established from this research that avoidance of lectures by the students is largely responsible for the massive failure of programming courses. We tend to explore this further in subsequent research to know the reasons why student avoid programming class. For example, we pose the following research question for future study: could absence of students in programming class be due to the daunting nature of programming or the way it is been taught? Answering this question in the future will provide a clue or clear picture on how to minimize the failures.

## 7 Acknowledgement

The authors acknowledged Usmanu Dandodiyo University Sokoto, Department of Mathematics, Computer Science Unit for providing the data used in this study and Ms. Zahraa Fadil Muhsen, Iraq for some of the figures used in the paper.

## 8 References

- [1] R. Ferguson, "The state of learning analytics in 2012: A review and future challenges." Knowl.Media Inst. Tech. Rep. KMI, 2012.
- [2] C. Romero and S. Ventura, "Educational data mining: a survey from 1995 to 2005.," *Expert System Application* 33(1), p. 135–146, 2007.
- [3] S. Suhirman, T. Herawan, H. Chiroma, and J. M. Zain. Data Mining for Education Decision Support: A Review. *International Journal of Emerging Technologies in Learning (iJET)*, Vol. 9, No. 6, pp. 4-19, 2014. <https://doi.org/10.3991/ijet.v9i6.3950>
- [4] B. K. Baradwaj and S. Pal, "Mining Educational Data to Analyze Students" Performance," (IJACSA) *International Journal of Advanced Computer Science and Applications*, pp. Vol. 2, No. 6, 2011. <https://doi.org/10.14569/ijacsa.2011.020609>
- [5] J. Bennedsen and M. E. Caspersen., " Failure rates in introductory programming," *ACM SIGCSE Bulletin* 39.2, pp. 32-36, 2007. <https://doi.org/10.1145/1272848.1272879>
- [6] S.S. Oyelere, J. Suhonen, and E. Sutinen. M-Learning: A New Paradigm of Learning ICT in Nigeria. *International Journal of Interactive Mobile Technologies (iJIM)*, Vol. 10, No. 1, pp. 35-44, 2016. <https://doi.org/10.3991/ijim.v10i1.4872>
- [7] F.J. Agbo, and S.S. Oyelere. *Smart Mobile Learning Environment for Programming Education in Nigeria: Adaptivity and Context-Aware Features*. Springer Nature Switzerland AG. pp. 1061–1077, 2019. [https://doi.org/10.1007/978-3-030-22868-2\\_71](https://doi.org/10.1007/978-3-030-22868-2_71)
- [8] Y. Alsultanny, "Selecting a suitable method of data mining for successful forecasting," *Journal of Targeting, Measuring and Analysis for Marketing*, pp. 207-225, 2011. <https://doi.org/10.1057/jt.2011.21>
- [9] J. Melero, H-L. Davinia, and J. Blat, "Towards the Support of Scaffolding in Customizable Puzzle-based Learning Games," in DOI:10.1109/ICCSA, June 2011. <https://doi.org/10.1109/iccsa.2011.64>



- [10] S.S. Oyelere, F.J. Agbo, A.A Yunusa, I.T. Sanusi, K. Sunday (2019). Impact of puzzle-based learning in computer science education: the case of MobileEdu. 18th IEEE International Conference on Advanced Learning Technology (ICALT), Maceio-AL, Brazil 2019. <https://doi.org/10.1109/icalt.2019.00072>
- [11] S.S. Oyelere & J. Suhonen, "Design and implementation of MobileEdu m-learning application for computing education in Nigeria: A design research approach," IEEE, pp. 27-31, 2016. <https://doi.org/10.1109/latice.2016.3>
- [12] B. N. Patel, S. G. Prajapati and K. I. Lakhtaria, "Efficient Classification of Data using Decision Tree," Bonfring International Journal of Data Mining, March 2012.
- [13] M. A. T. Mohammed and M. E. H. Alaa, "Mining Educational Data to Improve Students' Performance: A Case Study," ICT Journal, 2012.
- [14] B. Behdad, R. Z. Osmar, E. Samira and I. Donald, "Educational data mining applications and tasks: A survey of the last 10 years." Springer, pp. 537-553, 2017.
- [15] G. Huseyin and I. Ayhan, "Modeling Student Performance in Higher Education using data mining," Springer International Publishing, 2014.
- [16] A. M. Shahiri, W. Husain and N. A. Rashid, "A Reveiw on Predicting Student Performance using Data mining technique," In Proceedings of the Third information Systems International Conference, 2015.
- [17] J. Han and M. Kamber, "Data Mining: Concepts and Techniques.," KauffmANN Publishers, San Francisco, 2001.
- [18] M. M. Quadri and N. Kalyankar, "Drop out feature of student data for academic performance using decision tree techniques.," Global Journal of Computer Science and Technology, 2014.
- [19] S. Ankur and C. Vijay, "Comparison between ID3 and C4.5 in Contrast to IDS Surbhi Hardikar, Vol. 2 (7).," VSRD-IJCSIT, pp. 659-667, 2012.
- [20] J. Quinlan, "Induction of Decision Trees, Machine Learning 1: pp.81-106, Kluwer Academic Publishers, Boston, (1986).," Kluwer Academic Publishers, pp. pp. 81-106, 1986. <https://doi.org/10.1007/bf00116251>
- [21] <http://www.cs.waikato.ac.nz/~ml/weka/>
- [22] S.S. Oyelere, J. Suhonen, and T.H. Laine. Integrating Parsons programming puzzles into a game-based mobile learning application. Proceedings of the 17th Koli Calling International Conference on Computing Education Research. ACM, New York, pp. 158-162, 2017. <https://doi.org/10.1145/3141880.3141882>
- [23] S.I. Malik, M. Shakir, A. Eldow, and M. W. Ashfaque. Promoting Algorithmic Thinking in an Introductory Programming Course. International Journal of Emerging Technologies in Learning (iJET), Vol. 14, No. 1, pp. 84-94, 2019. <https://doi.org/10.3991/ijet.v14i01.9061>

## 9 Authors

**Kissinger Sunday** is a lecturer at the Department of Mathematics, Computer Science Unit, Usmanu Danfodiyo University, Sokoto, Nigeria. His research interest focuses mainly on computing education, machine learning and educational data mining.

**Patrick Ocheja** is a Ph.D. student at Kyoto University. His research is focused on connecting lifelong learning using blockchain technology.

**Sadiq Hussain** is System Administrator at Dibrugarh University, Assam, India. His research interest includes data mining and machine learning.

**Solomon Sunday Oyelere** is a Postdoctoral researcher at the University of Eastern Finland, School of Computing, Finland. His research interest focus mainly on improving learning environments through smart technology, pedagogy and content.

**Oluwafemi Samson Balogun** is a Postdoctoral researcher at the University of Eastern Finland School of Computing, Finland. His research interest includes data science, categorical data analysis, biostatistics and modeling.

**Friday Joseph Agbo** is a Ph.D. student at the University of Eastern Finland, School of Computing, Finland. His research interest include smart learning, technology-enhanced learning, computational thinking, and programming education.

Article submitted 2019-08-16. Resubmitted 2019-10-03. Final acceptance 2019-10-05. Final version published as submitted by the authors.