Measuring Students' Academic Performance through Educational Data Mining

Jeff Chak Fu Wong and Tony Chun Yin Yip

Abstract—**Based on a mix of real world data and a simulated dataset for predicting the students' academic performance, we study/compare various decision tree (DT) based algorithms (which include ID3, C4.5 and CART) with different choices of information entropy metrics (which include Shannon, Quadratic, Havrda and Charvát, Rényi, Taneja, Trigonometric and R − norm entropies) to build a decision tree in order to provide appropriate counseling/advise at an earlier stage. DT is one such important technique in educational data mining (EDM) which creates hierarchical structures of classification rules "If ⋯, Then ⋯" building a tree structure by incrementally breaking down the datasets in smaller subsets. The results suggest that basic training of the students has no significant predictive power on performance, while information about their abilities, diligence, motivation and activity in the learning process can predict their grades. As such, the resulting forecasts can be used by the instructor in optimizing the learning process and designing the course content and schedule.**

Index Terms—**Decision tree algorithms, educational data mining, entropy metrics and students' academic performance.**

I. Background

The prediction of students' academic performance (e.g., [1]), which is one of the well-studied educational data mining problems, can be accomplished by the three stages below:

Descriptive analytics - we get students' historical data through an on-line survey and introduce seven attributes (or features) (e.g., [2]-[4]) to help us predict and make decisions related to their academic status.

Predictive analytics - we use various DT algorithms (e.g., [5], [6]) to get a predictive final grade for a student, namely, a student's learning outcome and pattern based on his/her past data.

Prescriptive analytics - we show how we used different decision tree algorithms, e.g., ID3, C4.5 and CART, with various feature selection methods for further/deeper analysis, e.g., we display the If-Then figure and show the precision and accuracy of each method. Based on their final learning outcomes, we can create a better strategy for designing various MATH related course content and schedules at the Chinese University of Hong Kong (CUHK).

There is a significant number of articles (e.g., [1], [7]-[9]) that relate to different choices of information entropy methods used in this study. Despite a comparative study of various information entropy methods with eight real world datasets discussed in [1], analysis and prediction of student's learning performance from an algorithmic perspective is not fully investigated yet.

II. Purpose of Study

In this paper, we address the following issues:

A. *Handling Large Data Samples*

We test our homemade codes on a set of students' learning data samples and show how effective DT rule-based algorithms are. As the size of the required data increases, will the obtained results of student's learning patterns/features be more relevant and have the same result using different DT and entropy measures?

B. *Understanding Data Mining Actionable Trends*

To generate multiple sets of the student's learning data, inspired from [10], the random nested sampling for evolving data streams is used. Hence, many of the original data are repeated in the resulting simulated date set. As the size of the required data increases, for each evolution stage we added 10% more noisy data from the previous data samples. A random selection procedure was used to obtain the noisy data from the normal distribution function. Therefore, we randomly generated seven feature values that were significantly related to students' end of semester marks. In other words, these data are drawn from the random sampling, but are not obtained from the students' survey. Hence, we called it noisy data. How will the noisy data in the synthetic dataset affect the students' prediction results? This process acts like real-time instances of overfitting. We used the so-called back-track pruned (BT-pruned) algorithm [11] to reorganize the nodes of the constructed tree in order to overcome this drawback when large data samples were used.

C. *Gaining Perceptive Knowledge in Teaching Purposes*

In order to examine the training and testing datasets, the ten-fold cross validation model will be used. Will our predictions match the testing dataset?

III. Sources of Evidence

A. *Problem Description: Studying Students' Learning Activities*

The dataset used here is collected from three sources:

Sample A

We used an observed dataset from [12] as a seed; the size

Manuscript received April 19, 2020; revised July 2, 2020.

Jeff Chak Fu Wong and Tony Chun Yin Yip are with Department of Mathematics, the Chinese University of Hong Kong, Hong Kong (e-mail: cuhk.jeff@gmail.com, yipcy@math.cuhk.edu.hk).

doi: 10.18178/ijiet.2020.10.11.1461

of the required data is 50 (see Table I). Then, we combined this seed data set with our real data set, where the size of our required data is 25 (see Table II). Then, we added 10% of the noisy data we mentioned earlier through the normal distributed function into the mix of the two combined data sets (75). Using the concept of random nested sampling, we generated different sets of repeated multiple data depending on the designed size of the data set $n = 200{,}400{,}800{,}1600{,}3200{,}6400$.

TABLE I: THE TRAINING DATASET

ID	OSM	CT	SP	AP	PP	ATT	LC	ESM
1	First	Good	Good	Yes	Yes	Good	Yes	First
2	First	Good	Average	Yes	No	Good	Yes	First
3	First	Good	Average	No	No	Average	No	First
4	First	Average	Good	No	No	Good	Yes	First
5	First	Average	Average	No	Yes	Good	Yes	First
6	First	Poor	Average	No	No	Average	Yes	First
7	First	Poor	Average	No	No	Poor	Yes	Second
8	First	Average	Poor	Yes	Yes	Average	No	First
9	First	Poor	Poor	No	No	Poor	No	Third
10	First	Average	Average	Yes	Yes	Good	No	First
11	Second	Good	Good	Yes	Yes	Good	Yes	First
12	Second	Good	Average	Yes	Yes	Good	Yes	First
13	Second	Good	Average	Yes	No	Good	No	First
14	Second	Average	Good	Yes	Yes	Good	No	First
15	Second	Good	Average	Yes	Yes	Average	Yes	First
16	Second	Good	Average	Yes	Yes	Poor	Yes	Second
17	Second	Average	Average	Yes	Yes	Good	Yes	Second
18	Second	Average	Average	Yes	Yes	Poor	Yes	Second
19	Second	Poor	Average	No	Yes	Good	Yes	Second
20	Second	Average	Poor	Yes	No	Average	Yes	Second
21	Second	Poor	Average	No	Yes	Poor	No	Third
22	Second	Poor	Poor	Yes	Yes	Average	Yes	Third
23	Second	Poor	Poor	No	No	Average	Yes	Third
24	Second	Poor	Poor	Yes	Yes	Good	Yes	Second
25	Second	Poor	Poor	Yes	Yes	Poor	Yes	Third
26	Second	Poor	Poor	No	No	Poor	Yes	Fail
27	Third	Good	Good	Yes	Yes	Good	Yes	First
28	Third	Average	Good	Yes	Yes	Good	Yes	Second
29	Third	Good	Average	Yes	Yes	Good	Yes	Second
30	Third	Good	Good	Yes	Yes	Average	Yes	Second
31	Third	Good	Good	No	No	Good	Yes	Second
32	Third	Average	Average	Yes	Yes	Good	Yes	Second
33	Third	Average	Average	No	Yes	Average	Yes	Third
34	Third	Average	Good	No	No	Good	Yes	Third
35	Third	Good	Average	No	Yes	Average	Yes	Third
36	Third	Average	Poor	No	No	Average	Yes	Third
37	Third	Poor	Average	Yes	No	Average	Yes	Third
38	Third	Poor	Average	No	Yes	Poor	Yes	Fail
39	Third	Average	Average	No	Yes	Poor	Yes	Third
40	Third	Poor	Poor	No	No	Good	No	Third
41	Third	Poor	Poor	No	Yes	Poor	Yes	Fail
42	Third	Poor	Poor	No	No	Poor	No	Fail
43	Fail	Good	Good	Yes	Yes	Good	Yes	Second
44	Fail	Good	Good	Yes	Yes	Average	Yes	Second
45	Fail	Average	Good	Yes	Yes	Average	Yes	Third
46	Fail	Poor	Poor	Yes	Yes	Average	No	Fail
47	Fail	Good	Poor	No	Yes	Poor	Yes	Fail
48	Fail	Poor	Poor	No	No	Poor	Yes	Fail
49	Fail	Average	Average	Yes	Yes	Good	Yes	Second
50	Fail	Poor	Good	No	No	Poor	No	Fail

TABLE II: THE TESTING DATASET

ID	OSM	CT	SP	AP	PP	ATT	LC	ESM
1	Second	Good	Average	Yes	Yes	Average	Yes	First
2	First	Good	Good	Yes	Yes	Average	Yes	First
3	Second	Good	Good	Yes	Yes	Good	Yes	First
4	First	Good	Good	Yes	No	Good	No	Second
5	Third	Average	Average	Yes	No	Good	Yes	Third
6	Third	Average	Average	Yes	Yes	Good	Yes	Second
7	Third	Average	Average	Yes	No	Average	Yes	Third
8	First	Good	Average	Yes	Yes	Average	Yes	First
9	First	Good	Average	Yes	Yes	Good	Yes	First
10	Third	Average	Good	No	No	Good	Yes	Third
11	First	Good	Average	Yes	No	Good	Yes	First
12	Second	Average	Average	Yes	Yes	Poor	Yes	Second
13	First	Good	Good	No	No	Good	Yes	First
14	Second	Good	Good	Yes	Yes	Good	No	First
15	Second	Average	Good	Yes	Yes	Good	Yes	Second
16	First	Average	Good	Yes	Yes	Good	Yes	First
17	Second	Average	Average	Yes	Yes	Good	Yes	Second
18	Second	Average	Average	Yes	Yes	Good	Yes	Second
19	Second	Good	Average	Yes	Yes	Good	Yes	Second
20	Second	Good	Good	Yes	No	Good	Yes	First
21	First	Good	Good	Yes	Yes	Good	Yes	First
22	First	Good	Good	Yes	Yes	Average	Yes	First
23	First	Average	Average	Yes	Yes	Good	Yes	First
24	First	Average	Average	Yes	Yes	Good	No	Second
25	First	Good	Average	Yes	No	Average	No	Second

Sample B

As a real pilot study, we collected a dataset from the SAYT1510 course offered by the Chinese University of Hong Kong. We conducted an on-line survey via Studying Students' Learning Activities. Students who took SAYT1510 were international and local high school students.

Sample C

To validate our models, we combined two data samples sets, **Sample A** and **Sample B**, called **Sample C**, and repeated the same random nested sampling procedure as above.

B. *Training Dataset*

Our goal here is to study/predict students' learning activities based on a set of attributes described in [12]. Similar works are found in [13] and [14]. The training datasets that are combined from **Sample A** are used to build the model as shown in Table I (from one to fifty samples). We set the size of the data samples, $n = 200{,}400{,}800{,}1600{,}3200{,}6400$, where the training set for our example is defined by the target class "students' learning activities" using the end of the semester marks with four modalities:

We assume:$\text{End of semester marks} = \text{First}, = \text{Second}, = \text{Third}, \text{or} = \text{Fail}$

The seven attributes describing the observations are: "Overall Semester Marks (OSM)", "Class Test (CT)", "Seminar Performance (SP)", "Assignment Performance (AP)", "Paper Presentations (PP)", "Attendance (ATT)", and "Laboratory Classes (LC)", which can take the following values:

In terms of the set notations, we have:

$$\text{OSM} = \{\text{First, Second, Third, Fail}\} \qquad \text{CT} = \{\text{Good, Average, Poor}\}$$
$$\text{SP} = \{\text{Good, Average, Poor}\} \qquad \text{AP} = \{\text{Yes, No}\}$$
$$\text{PP} = \{\text{Yes, No}\} \qquad \text{ATT} = \{\text{Good, Average, Poor}\}$$
$$\text{LC} = \{\text{Yes, No}\}$$

a) Overall Semester Marks (OSM): OSM are obtained from the secondary school programme. It is divided into four values: First: $> 60\%$; Second: $> 45\%$ and $\leq 60\%$; Third: $> 36\%$ and $\leq 45\%$; Fail: $\leq 36\%$

b) Class Test (CT): Each semester two class tests are conducted. CT is split into three classes: Good: $> 60\%$; Average: $> 40\%$ and $\leq 60\%$; Poor: $\leq 40\%$

c) Seminar Performance (SP): SP is divided into three classes: Poor: Presentation and confidence are low; Average: Either presentation is good or confidence is good, but not both; Good: Both presentation and confidence are good

d) Assignment Performance (AP): Each semester two assignments are given to students by each teacher. AP is divided into two classes: Yes: Student submitted the assignment; No: Student did not submit the assignment

e) Paper Presentations (PP): At the end of the year a Paper Presentation must be done by the student. PP is divided into two classes: Yes: Student participated in the Presentation; No: Student did not participate in the Presentation

f) Attendance (ATT): Attendance is compulsory for the Students. A minimum of 75% attendance is needed to participate in the End Semester Examination. ATT is divided into three classes: Poor: $\leq 60\%$; Average: $> 60\%$

and $\leq 80\%$; Good: $> 80\%$

g) Laboratory Classes (LC): LC is divided into two classes: Yes: Student completed the Practical lab; No: Student did not complete the Practical lab

h) End Semester Marks (ESM): ESM is obtained in the secondary school programme and can be predicted based on the above seven attributes. It is divided into four values: First:$> 60\%$; Second: $> 49\%$ and $\leq 60\%$; Third: $> 34\%$ and $\leq 49\%$; Fail: $\leq 34\%$.

C. Testing Dataset

The testing dataset was composed of the data from 25 students' in the SAYT1510 course, as shown in Table II.

D. Methodology: Decision Tree Algorithms

Decision tree classifiers are used to predict students' final marks. A decision tree is a flowchart like structure where the leaf nodes represent class labels and the non-leaf nodes represent attributes.

1) ID3

ID3 developed by Quinlan in 1979 [15] constructs a decision tree by employing a top-down, greedy search through the given sets of training data to test each attribute at every node, where the greedy search is based on the concept of heuristic problem solving by making an optimal local choice at each node. By making these local optimal choices, we reach the approximate optimal solution globally.

The ID3 algorithm can be summarized as:

a) At each level (or stage or node), select out the best feature as the test condition (in this paper, seven features are considered).

b) Now split the node into the possible outcomes (internal nodes).

c) Repeat the above steps until all the test conditions have been exhausted into leaf nodes.

In (a), to make that decision, we need to have some knowledge about entropy and information gain. Based on the computed values of entropy and information gain, we choose the best attribute at any particular step.

To be more precise, the ID3 algorithm selects the attribute to be split based on two metrics:

a) Measuring Impurity: An entropy metric measures the amount of information in an attribute. Entropy is calculated for all the remaining attributes. Split occurs at the attribute that has smallest entropy. Given probabilities $p_1, p_2, \cdots, p_s$, where $p_i \geq 0,\ i = 1, \cdots, s,$, where s is the total number of attributes. $\sum_{i=1}^{s} p_i = 1$. Entropy is defined as

$$H(p_1, p_2, \cdots, p_s) = \sum_{i=1}^{s} -(p_i \log p_i).$$

Shannon entropy finds the amount of order in a given database state. A value of $H = 0$ identifies a perfectly classified set. In other words, the higher the entropy, the higher the potential to improve the classification process.

b) Splitting Criteria: An information gain is a statistical property which measures how well a given attribute separates training examples into targeted classes. The one with the highest information (information being the most useful for classification) is selected based on entropy. The information gain, $\mathrm{Gain}(D, A)$ of an attribute A, relative to a collection of examples D, is defined as

$$\mathrm{Gain}(D, A) = H(D) - \sum_{i \in \mathrm{Value}(A)} \frac{|D_i|}{|D|} H(D_i)$$

where $\mathrm{Value}(A)$ is the set of all possible values for attribute A, and D_i is the subset of D for which attribute A has value i (i.e., $D_i = \{s \in D | A(s) = i\}$).

2) C4.5

C4.5 known as J48 in WEKA (Waikato Environment for Knowledge Analysis) is a successor of ID3 developed by Quinlan in 1992 [15] that is also based on Hunt's algorithm. C4.5, not only handles both categorical and continuous attributes to build a decision tree, but also makes use of the Gain $\mathrm{Ratio}(D, A)$ which is computed as follows:

$$\mathrm{Gain\ Ratio}(D, A) = \frac{\mathrm{Gain}(D, A)}{\mathrm{Split\ Information}(D, A)}$$

where $\mathrm{Split\ Information}(D, A)$ represents the information generated by splitting the training set D into i partitions which correspond to the i results of a test on attribute A. Attribute A with the highest Gain Ratio is selected as the splitting attribute. Compared to the ID3 algorithm, the expected entropy described by this second term is simply the sum of the entropies of each subset, weighted by the fraction of examples $\frac{|D_i|}{|D|}$ that belong to $\mathrm{Gain}(D, A)$, and is therefore the expected reduction in entropy caused by knowing the value of attribute A:

$$\mathrm{Split\ Information}(D, A) = -\sum_{i=1}^{s} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}.$$

3) Classification and regression tree (CART)

CART was introduced by Breiman et al. in 1984 [16] and is also based on Hunt's algorithm. CART handles both categorical and continuous attributes to build a decision tree. It handles missing values. CART uses the Gini Index as an attribute selection measure to build a decision tree. Unlike the ID3 and C4.5 algorithms, CART constructs binary splits. Hence, it constructs binary trees. The Gini Index measure does not use probabilistic assumptions like ID3, and C4.5. CART uses cost complexity pruning to remove the unreliable branches from the decision tree to improve the accuracy. Gini impurity is defined as 1 minus the sum of the squares of the class probabilities in a dataset:

$\mathrm{Gini\ Impurity} = 1 - \sum_{i=1}^{s} p_i^2$.

The Gini index is then defined as the weighted sum of the Gini impurity of the different subsets after a split:

$$\mathrm{Gini\ Index} = \sum_{i \in \mathrm{Value}(A)} \frac{|D_i|}{|D|} \mathrm{Gini\ Impurity}(D_i, A).$$

The Gini Index of a pure table which consists of a single class is zero because the probability is 1. Similar to Entropy, the Gini Index also reaches maximum value when all classes in the table have equal probability.

4) Entropy metrics

In what follows, we examine various entropy metrics in order to relax the complexity of the DT constructions with respect to the increasing number of nodes and leaves:

- Quadratic entropy ([17]) -

$$HQ(p_1, p_2, \cdots, p_s) = \sum_{i=1}^{s} p_i\,(1 - p_i)$$

- Havrda and Charvát entropy ([18]) -

$$HC_{\alpha}(p_1, p_2, \cdots, p_s) = \frac{1}{1-\alpha}\left(\sum_{i=1}^{s} p_i^{\alpha} - 1\right)$$

where α is a parameter adjusted by the user, i.e., $\lim_{\alpha\to 1} HC_{\alpha} = H$.

- Rényi entropy ([19]) -

$$R_{\alpha}(p_1, p_2, \cdots, p_s) = \frac{1}{1-\alpha}\log\sum_{i=1}^{s} p_i^{\alpha}$$

where α is a parameter adjusted by the user. Rényi entropy tends to H as $\alpha \to 1$, i.e., $\lim_{\alpha\to 1} R_{\alpha} = H$.

- Taneja entropy ([20]) -

$$T_{\alpha,\beta}(p_1, p_2, \cdots, p_s) = \frac{1}{1-\alpha}\log\sum_{i=1}^{s}\frac{p_i^{\beta+\alpha-1}}{p_i^{\beta}}$$

where α and β are constant inherent parameters. Taneja entropy tends to H as $\alpha \to 1,\ \beta \to 1$, i.e., $\lim_{\alpha\to 1,\ \beta\to 1} T_{\alpha,\beta} = H$.

- Trigonometric entropy ([21]) -

$$C_{\gamma}(p_1, p_2, \cdots, p_s) = \frac{1}{\pi(\gamma-1)}\cos\left(\frac{\pi}{2}\sum_{i=1}^{s} p_i^{\gamma}\right)$$

where γ is a parameter adjusted by the user. Trigonometric entropy tends to H as $\gamma \to 1$, i.e., $\lim_{\gamma\to 1} C_{\gamma} = H$.

- R −norm entropy ([21]) -

$$\mathrm{RNorm}_{\mathrm{R}}(p_1, p_2, \cdots, p_s) = \frac{\mathrm{R}}{\mathrm{R}-1}\left(1 - \sum_{i=1}^{s}\left(p_i^{\mathrm{R}}\right)^{1/\mathrm{R}}\right)$$

where γ is a parameter adjusted by the user. R − norm entropy tends to H as $\mathrm{R} \to 1$, i.e., $\lim_{\mathrm{R}\to 1}\mathrm{RNorm}_{\mathrm{R}} = H$.

To the very best of our knowledge, using Trigonometric entropy and R − norm entropy for analyzing student performance has not been done yet.

IV. Main Argument

We have established a list of the R scripts used for our predictions and to find interesting patterns in different educational data mining models. First, we used the ID3 algorithm with the Shannon entropy and compared our results with the WEKA tool. Both results, in terms of the decision tree figures, have no significant differences. Hence, the results using our source codes were reported as follows. Fig. 1 provides the decision tree based on the ranking of seven attributes the C4.5 algorithm with the Shannon entropy and a 70%:30% split, where we took 75 data samples from Table I and Table II.

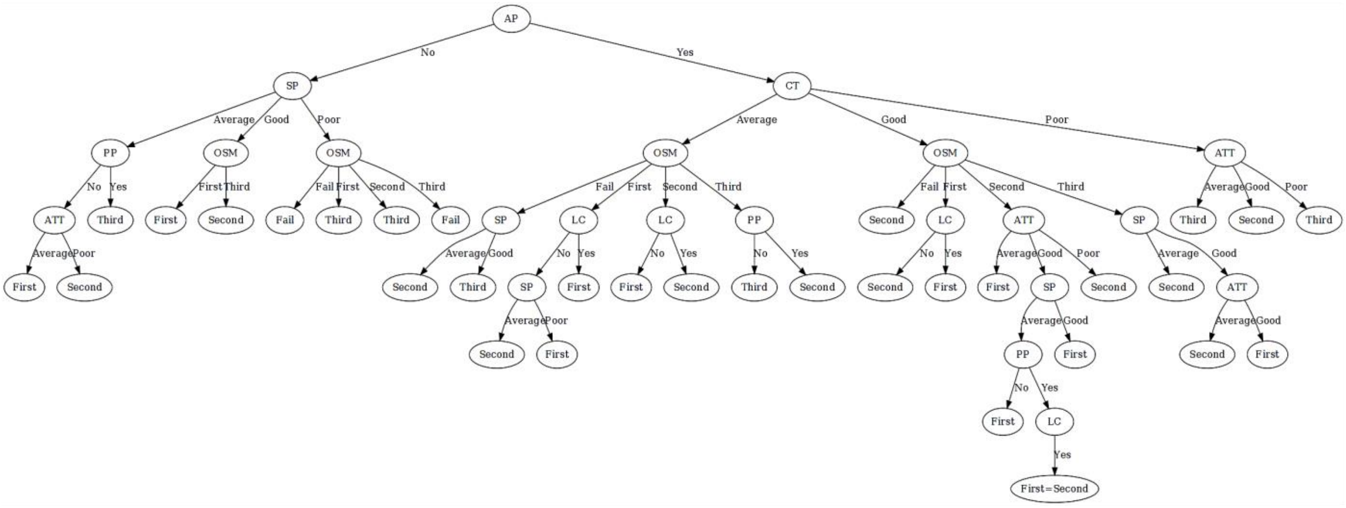


Fig. 1. Decision tree of a 70%: 30% split.

The knowledge represented by the decision tree can be extracted and represented in the form of If-Then rules. We only list some of easier If-Then rules.

- If

– AP - No → SP - Poor → OSM - Fail

- Then

– Fail

- If

– AP - Yes → CT - Poor → ATT - Average

- Then

– Third

- If

– AP - Yes → CT - Good → OSM - Third → SP - Good → ATT - Average

- Then

– Second

- If

– AP - Yes → CT - Good → OSM - Third → SP - Good → ATT - Good

- Then

– First

The confusion matrix is an expression that shows the reliability of an algorithm, i.e., how accurate it is, in terms of containing information on actual values and predictions on classification, as illustrated in Table III:

TABLE III: Confusion Matrix

		Prediction outcome	
		Yes	No
Actual value	Yes	True Positive	False Negative
	No	False Positive	True Negative

where **True Positive** (TP) is the amount of positive data that is correctly classified, **True Negative** (TN) is the amount of negative data that is correctly classified, **False**

Negative (FN) is the amount of negative data and incorrectly classified, **False Positive** (FP) is the number of positive data and incorrectly classified.

By expressing values as a percentages, we have the following:

- **Precision** is the fraction of retrieved instances that are relevant, which is given by

$$\textbf{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \times 100\%.$$

- It is calculated as the total number of true positives divided by the sum of the total number of true positives and the total number of false positives.
- **Recall** is the fraction of relevant instances that are retrieved.

$$\textbf{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \times 100\%.$$

- It is calculated as the total number of true positives divided by the sum of the total number of true positives and the total number of false negatives.
- **Accuracy** is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications, i.e.,

$$\text{Accuracy} = \frac{\text{True positives} + \text{True negatives}}{\text{True positives} + \text{False positives} + \text{True negatives} + \text{False negatives}} \times 100\%$$

A. Accuracy Measures

1) Comparison of different entropy measures

Let us examine the mix of **Sample A**, **Sample B** and **Sample C** datasets. The accuracy comparison of three different DT algorithms with and without using the BT-pruned algorithm are summarized in Table IV when the synthetic training and testing data ratio using the Shannon entropy for each n is 70%:30%. Increasing the size of the synthetic samples showed the accuracy and the convergence of each algorithm using both the BT-pruned and unpruned algorithms. Table IV shows that using the **Sample A** dataset, the BT-pruned algorithm not only has a similar accuracy trend as the unpruned algorithm but also in terms of the number of nodes greatly reduced significance when large data samples were used. For different BT-pruned DT algorithms performance at $n = 6400$, we have

$$\text{CART} > \text{ID3} > \text{C4.5}$$

While for different unpruned DT algorithms performance, we have

$$\text{CART} > \text{C4.5} > \text{ID3}.$$

For further validation testing, we compared our C4.5 decision tree with the one generated by WEKA, where both figures for each sample size n are the same. Until otherwise stated, we used our homemade codes to test different entropy results using the BT-pruned algorithm.

TABLE IV: ACCURACY OF DIFFERENT DT METHODS WITH THE BT-PRUNED ALGORITHM AGAINST DIFFERENT SAMPLE SIZES FOR A 70%: 30% SPLIT

	BT-pruned DT algorithms					
	ID3 (Shannon Entropy)		C4.5 (Shannon Entropy)		CART (Gini Index)	
	Accuracy	# of Nodes	Accuracy	# of Nodes	Accuracy	# of Nodes
n = 200	25.00%	32	31.67%	51	28.33%	55
n = 400	20.83%	123	21.67%	112	15.83%	124
n = 800	19.17%	167	20.00%	159	18.33%	160
n = 1600	47.71%	205	48.96%	260	47.08%	204
n = 3200	69.48%	286	68.23%	263	69.27%	283
n = 6400	77.66%	305	77.55%	276	78.70%	301
	Unpruned DT algorithms					
	ID3 (Shannon Entropy)		C4.5 (Shannon Entropy)		CART (Gini Index)	
	Accuracy	# of Nodes	Accuracy	# of Nodes	Accuracy	# of Nodes
n = 200	28.33%	79	33.33%	76	28.33%	79
n = 400	15.00%	149	17.50%	148	16.67%	149
n = 800	15.83%	235	17.92%	249	15.83%	235
n = 1600	46.88%	323	48.12%	332	46.46%	322
n = 3200	68.96%	329	68.65%	330	68.96%	329
n = 6400	78.07%	352	78.23%	351	78.44%	350

TABLE V: DETAILED ACCURACY OF CLASSIFIERS USING DIFFERENT BT-PRUNED DT METHODS WITH *N* = 6400 FOR THE SHANNON ENTROPY AND A 70%: 30% SPLIT

		Detailed Accuracy of Classifiers			
		True Positive	True Negative	False Positive	False Negative
		(TP)	(TN)	(FP)	(FN)
ID3	First	433	1268	87	132
	Second	412	1291	97	120
	Third	380	1355	128	57
	Fail	266	1476	117	6
C4.5	First	430	1266	90	134
	Second	410	1277	99	134
	Third	393	1335	115	77
	Fail	256	1473	127	64
CART	First	437	1265	83	135
	Second	415	1289	94	122
	Third	381	1360	127	52
	Fail	278	1460	105	77

Concerning the choice of different parameters against different choices of entropy methods, the results of the application of two BT-pruned ID3 and C4.5 DT algorithms with regard to the TP, TN, FP and FN are summarized in

Table V. The asterisk placed in the table represents the highest number of true positives from the testing data. Here is a list of interesting findings:

- The CART DT algorithm with the Gini Index attribute split is the best TP measure, as shown in Table V.
- Table VI summarizes the accuracy of each entropy method and reveals that, with a specified choice of parameters, the Havrda and Charvát entropy is the best one. We used the Naíve Bayes classifier and studied the same problem using WEKA, where its accuracy was 59.22.
- Table VI shows that using Quadratic, Havrda and Charvát, Rényi, Taneja, Trigonometric and $\mathrm{R}-\mathrm{norm}$ of entropies, results in better accuracy rates than the Shannon entropy. Here are a few observations:

– We agreed with [17] that using the Quadratic entropy, the accuracy of the BT-pruned ID3 algorithm is better than that of BT-pruned C4.5 one.

– We agreed with [19] that using the Rényi entropy is better than using the Shannon one in terms of the accuracy measurement.

– We agreed with [20] that if large values are selected for a pair of parameters (α, β), the accuracy of using Taneja entropy will close to that of the Shannon entropy.

– Our findings show that using the Trigonometric and $\mathrm{R}-\mathrm{norm}$ entropies with a specified choice of parameters will maintain good accuracy.

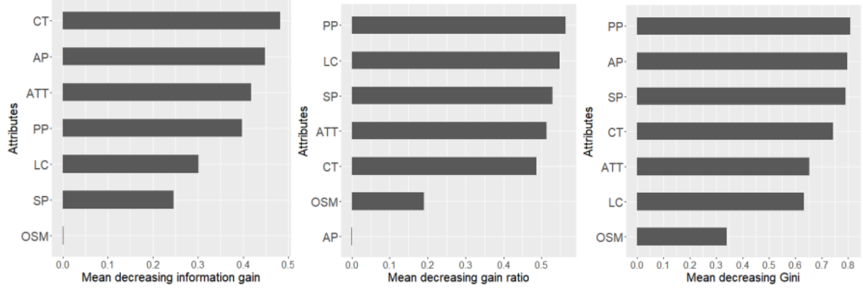
For illustrative purposes, using the BT-pruned DT algorithm with the Rényi entropy, we set $n = 6400$ and calculated the mean decreasing gain (or the mean decreasing information gain), the mean decreasing gain ratio and the mean decreasing Gini for ID3, C4.5 and CART respectively against a set of attributes. Hence, we extracted the rank of attributes based on a set of the mean decreasing values, as shown in Fig. 2. Inspection of Table VII indicates that the lowest mean decreasing value is the parent node, e.g., OSM for ID3.

TABLE VI: ACCURACY OF DIFFERENT ENTROPY METHODS WITH SPECIFIED PARAMETER(S) $N = 6400$ STUDENT SAMPLES

		BT-pruned DT algorithms	
		ID3	C4.5
Entropy Methods	Shannon Entropy	77.66 %	77.55 %
	Quadratic Entropy	78.70 %	77.40 %
	Havrda and Charvát Entropy	78.96 % ($\alpha = 0.01$)	78.18 % ($\alpha = 5$)
	Rényi Entropy	78.91 % ($\alpha = 0.01$)	78.49 % ($\alpha = 20$)
	Taneja Entropy	78.28 % ($\alpha = 2.0, \beta = 0.6$)	78.12 % ($\alpha = 2.0, \beta = 1.0$)
	Trigonometric Entropy	78.28 % ($\gamma = 2.0$)	78.28 % ($\gamma = 2.0$)
	$\mathrm{R}-\mathrm{norm}$ Entropy	78.44 % (R $= 2$)	78.49 % (R $= 0.5$)

TABLE VII: SORTING THE ATTRIBUTES USING THE INFORMATION GAIN, THE GAIN RATIO AND THE GINI INDEX THROUGH THE BT-PRUNED DT ALGORITHMS WITH RÉNYI ENTROPY

	ID3	C4.5	CART
1	OSM	AP	OSM
2	SP	OSM	LC
3	LC	CT	ATT
4	PP	ATT	CT
5	ATT	SP	SP
6	AP	LC	AP
7	CT	PP	PP



a. ID3 b. C4.5 c. CART

Fig. 2. Attributes against mean decreasing information gain, mean decreasing gain ratio and mean decreasing Gini.

TABLE VIII: THE RESULTS OF THE ACCURACY OF THE BT-PRUNED C4.5 DT METHOD

Experiment		10%	20%	30%	40%	50%	60%	70%	80%	90%
Overall Accuracy		72 %	84 %	76 %	92 %	56 %	72 %	68 %	72 %	76 %
Accuracy	First	80 %	84 %	88 %	96 %	76 %	84 %	84 %	88 %	92 %
	Second	92 %	92 %	88 %	96 %	84 %	88 %	88 %	88 %	88 %
	Third	100%	100%	100 %	100 %	96 %	100 %	96 %	96 %	96 %
	Fail	100%	100%	100 %	100 %	100 %	100 %	100 %	100 %	100 %
Precision	First	100%	84.61%	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	Second	100%	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	Third	100%	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	Fail	–	–	–	–	–	–	–	–	–
Recall	First	61.53%	84.61%	76.92 %	92.30%	53.84 %	69.23 %	69.23 %	76.92 %	84.61 %
	Second	77.77%	77.77%	66.67 %	88.89%	55.56 %	66.67 %	66.67 %	66.67 %	66.67 %
	Third	100 %	100 %	100 %	100 %	66.67 %	100 %	66.67 %	66.67 %	66.67 %
	Fail	–	–	–	–	–	–	–	–	–

2) Split validation tests

In what follows, we only report the result of using the BT-pruned C4.5 DT algorithm with Shannon Entropy, the information gain feature selection model and $n = 6400$. By adopting the split validation, a training experiment will be conducted based on a predetermined split ratio: For example, we used 10% of the **Sample C** dataset. For each experiment, we repeated calculations 10 times and took the average of these 10 accuracy values, e.g., their precision and recall values. The overall accuracy of each experiment is summarized in Table VIII, where the accuracy/precision/recall values of each modality are given. Since the **Sample B** testing dataset does not contain any Fail grades, the last row of Table VIII has no value of its own. What we learned here is that the more data we collected, the likelier the predicted outcome will be accurate. In other words, if the training dataset contains a large volume of students' learning performances, then the user will certainly predict their learning outcome with high precision when the suitable EDM algorithm is used.

V. Conclusions

This paper describes a study for predicting student academic performance by using only an online questionnaire survey. Although we only used a limited amount of real data (from 25 students), we used the random nested sampling method to generate a large class of data based on published results. We have implemented different BT-pruned decision tree algorithms with different entropy methods. Using the split validation, we have shown that our homemade codes yield good prediction accuracy even when the size of the training dataset is large and also influenced by noisy data. Hence, the If-Then decision rules provide more accurate results.

Properly used, the BT-pruned decision tree algorithms developed in this study could help us to predict students' learning performances, which could be used to identify students that would benefit from early intervention or to design students' activities according to their skills and knowledge. By following the procedures described in the paper, when facing noisy or contaminated data from the old data, practitioners may use the pruning decision tree algorithm to improve the generalization performance in decision tree induction and get more insight into their students' performances.

Direction for further research emerged while this study was being conducted. The most significant direction would be to extract data from the qualitative approach. Based on these data, we will have a better understanding of students' needs. In addition, when the size of data samples increases, the visualization of the decision tree is nearly impossible. Tracing the If-Then rules is also our next research objective. Based on our preliminary work, for further studies, in order to obtain a big picture of Students' Academic Performance in our courses, we will collect more data, examine our proposed algorithms and publish our results elsewhere.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The first author conducted the research, wrote the paper and analyzed the data, while the second author produced all numerical results and plots; both authors approved the final version.

References

[1] S. Sharma, J. Agrawal, and S. Sharma, "Classification through machine learning technique: C4.5 algorithm based on various entropies," *International Journal of Computer Applications*, vol. 82, pp. 28-32, November 2013.

[2] K. S. Priya and A. V. S. Kumar, "Improving the student's performance using educational data mining," *International Journal of Advanced Networking and Applications*, vol. 4, pp. 1680-1685, February 2013.

[3] A. A. Saa, "Educational data mining & students' performance prediction," *International Journal of Advanced Computer Science and Applications*, vol. 7, pp. 212-220, May 2016.

[4] P. V. P. Sundar, "A comparative study for predicting student's academic performance using Bayesian network classifiers," *IOSR Journal of Engineering*, vol. 3, pp. 37-42, February 2013.

[5] V. E. Lee, L. Liu, and R. Jin, "Decision trees: Theory and algorithms," in *Data Classification: Algorithms and Applications*, Charu C. Aggarwal, Ed. Chapman and Hall, 2014, pp. 87-120.

[6] Gupta, A. Rawat, A. Jain, A. Arora, and N. Dhami, "Analysis of various decision tree algorithms for classification in data mining," *International Journal of Computer Applications*, vol. 163, pp. 15-19, April 2017.

[7] A. T. Velmurugan, "A comparative analysis on the evaluation of classification algorithms in the prediction of students performance," *Indian Journal of Science and Technology*, vol. 8, pp. 974-6846, August 2015.

[8] B. K. Baradwaj and S. Pal, "Mining educational data to analyze students' performance," *International Journal of Advanced Computer Science and Applications*, vol. 2, pp. 63-69, October 2011.

[9] K. Pal and S. Pal, "Analysis and mining of educational data for predicting the performance of students," *International Journal of Electronics Communication and Computer Engineering*, vol. 4, pp. 1560-1565, October 2013.

[10] E. Caro, C. González, and J. M. Mira, "Student academic performance stochastic simulator based on the Monte Carlo method," *Computers & Education*, vol. 76, pp. 42–54, 2014.

[11] R. Jothikumar and R. V. Siva Balan, "C4.5 classification algorithm with back-track pruning for accurate prediction of heart disease," *Biomedical Research*, vol. 27, pp. 107-111, January 2016.

[12] A. A. Saa, "Educational data mining & students' performance prediction," *International Journal of Advanced Computer Science and Applications*, vol. 7, pp. 212-220, May 2016.

[13] K. Hamoud, A. S. Hashim, and W. A. Awadh, "Predicting student performance in higher education institutions using decision tree analysis," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, pp. 26-31, February 2018.

[14] J. G. Vasquez and B. E. V. Comendador, "Competency discovery system: Integrating the enhanced ID3 decision tree algorithm to predict the assessment competency of senior high school student," *International Journal on Advanced Science Engineering Information Technology*, vol. 9, pp. 60-65, 2019.

[15] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[16] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[17] A. Philip and U. S. Nnamdi, "The quadratic entropy approach to implement the ID3 decision tree algorithm," *Journal of Computer Science and Information Technology*, vol. 6, pp. 23-29, December 2018.

[18] N. Mathur, S. Kumar, S. Kumar, and R. Jindal, "The base strategy for ID3 algorithm of data mining using Havrda and Charvát Entropy based on decision," *International Journal of Information and Electronics Engineering*, vol. 2, pp. 253-258, December 2012.

[19] R. Pate and K. K. Rana, "Use of Rényi Entropy calculation method for ID3 algorithm for decision tree generation in data mining," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 2, pp. 30-34, May 2014.

[20] H. A. Jeiad, Z. J. Mohammed Ameen, and A. A. Mahmood, "Employee performance assessment using modified decision tree," *Engineering and Technology Journal*, vol. 36, pp. 806-811, December 2018.

[21] V. Majerník, "Entropy - A universal concept in sciences," *Natural Science*, vol. 6, pp. 552-564, April 2014.

Jeff Chak-Fu Wong received his B.S. and M.S. degrees in mathematics and geodesy from the University of New Brunswick, Canada in 1997 and 2001, respectively, and his Ph.D. degree in mathematics from the Chinese University of Hong Kong, Hong Kong in 2004. He is currently a senior lecturer at the Department of Mathematics, Chinese University of Hong Kong. His research interests include computational social networks, data mining and machine learning.

Tony Chun-Yin Yip received his B.S. in mathematics from University of Hong Kong, Hong Kong, in 2019. His present interests include data mining, machine learning and AI related problems.