Mining Educational Data to Enhance Intelligent Tutoring Systems Using Students Performance: A case Study

AE.Madurapperumage, JB. Ekanayake

Abstract—Effective course designing is a key challenge, which requires expert knowledge and a clear understanding of students' behaviors. Intelligent tutoring systems can serve as a human tutor with excellent knowledge and less response time while understanding the knowledge level of the current cohort, which use to navigate them with most appropriate paths. In this study, we show how students' data can use to improve course design by analyzing them in determining the comprehensive level of assignment questions. The outcome of this research can be used to enhance the capabilities of the intelligent tutoring systems, which will be able to create clusters in a given question set by considering the past experiences of student responses. Teachers and course designers are beneficiaries who would be used the outcomes to improve the quality of the course.

Index Terms— Classification Techniques, Clustering, Educational Data Mining, Intelligent Tutoring System, Prediction, Multilayer Perceptron, Students Performance.

1 Introduction

Many educational institutes are engaging with providing high-quality education to their students. As results, curriculum revision, upgrading the existing course materials, redesigning courses, practicing novel teaching strategies has taken place. Further, the popularity of Massive Online Open Courses (MOOCs) has created a recent trend on how to teach effectively, while delivering expected knowledge to a cohort in the distance. Various strategies have been using over the past few decades with the purpose of fulfilling these requirements. However, one of the fundamentals of these strategies is to structure the program in a way, which is most appropriate for the engaged cohort. Intelligent Tutoring systems (ITS), Intelligent Teaching Assistant Systems (ITAS), personalized education systems, etc have emerged from this idea. ITSs are computer-based instructional systems with models of instructional content that specify what to teach, and teaching strategies that specify how to teach [1]. The fundamental of ITS is applying different types of data mining techniques on educational data repositories. This is known as Educational Data Mining (EDM), which is an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings and using those methods to understand students and the environments which they learn in[2]. A better understanding of students' learning behavior towards different tasks of a course is vital in many situations such as exam preparation, provide early warning, course redesigning, maintain retention rate and etc. Although, student learning behavior analyzing is popular on many other subject-streams the analyzing of student behavior in computer programming is comparatively rare [4]. Since the code writing is a core competency for every computer scientists, it deserves more attention.

The process of modeling a student's ability as they interact with a learning application is known as knowledge tracing. Here the information from previous students' interactions was used to build a model, which can predict the interactions of future students. Intelligent Teaching Assistant for Programming (ITAP) is a data-driven tutoring system introduced by Kelly Rivers, Kenneth R.Koedinger [9]. It combines algorithms for state abstraction, path construction, and state reification to fully automate the process of hint generation. ITAP makes it possible to generate a full chain of hints from any new code state to the closest goal state [9].

In this research, we have analyzed a dataset, which collected from the students who used ITAP ITS for their non-compulsory Python programming course. The data used belongs to 89 total students who worked on the first 19 problems over time. Course duration is seven weeks. No restrictions applied to the number of attempts where students can work on each problem in any order. Feedback has given at each test case. Moreover, the students can request hints from ITAP. The dataset itself contains a record for each attempt and hint request that students made while working [7]. The dataset for this research was collected form DataShop [7]. The extracted list of attributes from the dataset is as follows,

SubjectId, ProblemId, StartOrder, FirstCorrect, EverCorrect, UsedHints, Attempts.

In this research, our primary goal is to analyze student behavior on a computer programming course. Since we realized, the first attempt and the number of attempts can reveal vital information of the question set, those two factors are considered to answer the research questions b) and c).
Following are the research questions.

a) How to determine the comprehensiveness of the questions by considering student behaviors?
b) Will the students be able to code a given question in their first attempt?
c) What would be the number of attempts that one may try on a given question?

2 RELATED WORKS

The field of educational data mining has emerged into multi-disciplinary areas where the researchers are capable of researching various aspects of education. Enhancement of teaching and learning, course designing, curriculum revision, evaluation of teaching and learning methods are a few examples of them. Further, the advancement of accessibility of the Internet has leveraged education among the community. Massive Online Open Courses (MOOC), Learning Management Systems (LMS), Open Social Student Modeling (OSSM), ITS, Learning and Teaching Assistant Systems, and many more ideas have appeared with this trending usage of the Internet. The fundamental of implementing these systems is knowledge extraction from educational information through EDM. The methods of EDM are drawn from a variety of literature, including data mining and machine learning, psychometrics and other areas of statistics, information visualization, and computational modeling [5]. Discovery with models has become an increasingly popular method in EDM research. It has become the second most common category of EDM research, representing 19% of papers [5] in 2008-2009. This includes creating intelligent tutoring systems, providing course navigation, giving recommendations on related concepts, studying students' behavior to extract the learning patterns, and many other prolific research areas.

In recent years there has been growing interest in using extensive collections of logged programming data to understand how students learn, what they struggle with, and what we can do to improve computer science education [4]. The ability to write code accurately and fluently is a core competency for every computer scientist. However, learning to program is prohibitively difficult for many students, and this limits diversity in students that pursue computer science or gain programming skills to enhance their other careers [9]. Further, knowledge tracing is another strong area, which became a topic for educational researchers who interest in analyzing previous students' learning process to build the prediction models. There are some previous works, which focused on predicting future performance by recognizing the knowledge level of students through the mining of the current cohort's data. The Bayesian Knowledge Tracing algorithm is one of a popular estimating algorithm that has been applied in several studies to model the student learning process [11].

When studying about students' learning process, researchers have realized that the way of providing knowledge does matter to the student performances. Knowledge sequencing and task sequencing have been studied to support this matter. How to sequence the knowledge effectively to deliver the required skills, what would be the correct size of tasks in knowledge delivery, is fine-grained tasks are better than course- grained tasks are some problems that the researchers have been trying to resolve in this context. Considering the learning goal and the current state of students' knowledge reflected by the student model, various sequencing approaches are able to determine which task is currently the most appropriate [3].

It is vital to determine the precise time to teach new programming concepts to the cohort. Learning curve analysis is an approach from broader EDM research that focuses on estimating learners' performance over time [4]. The probability of making mistakes in assignments should lessen with the increasing time where the student can master with skills. Traditionally, skills in this context are formalized as Knowledge Components (KCs), with a given task exercising one or more KCs. Students who possess a mastery of those KCs are more likely to perform the task Correctly [4].

With the rapid growth of Web-based education, educational data repositories overflowed with educational data. Researches began to investigate how to automate human tutors who can help students in-distance by mining collected data from various online educational systems such as MOOCs, LMS, and OSSMs. In the study of the online learning community, many investigators have turned attention to automatically logged web data [13]. Studies showed that students who receive help while programming does better in their courses [9]. But due to practical issues, the assistance they get during the programming is considerably less than required. Easy-To access, automated help source would be a great solution. Intelligent tutoring has been a prolific area of research due to these reasons. ELM Adaptive Remote Tutor (ELM-ART), Logic ITA and ITAP are some examples for ITSs.

ELM-ART is a web-based ITS to support learning Lisp programming [10]. This supports example-based programming, intelligent analysis of problem solutions, and advanced testing and debugging facilities. Logic- ITA is also a web-based tool used in University of Sydney since 2001. This helps to extract useful pedagogical information about the current cohort. Further, this system uses clustering to reveals the patterns of failing students. However, traditional ITSs have a major drawback: they take much time and expert knowledge to create. A survey of several different ITS builders found that the ratio of tutor construction time to student interaction time is very high [9]. Rather than relying on expert knowledge, the mined knowledge through students' data can use in ITSs. Hint generation based data-driven tutoring has resolved this issue. Here the students' data analyzed and used for further course enhancement.

Some researches use data from e-learning environments to make predictions about the learning community. But they focused on clustering the community of learning, classifying the cohort to predict their performance, etc. Moreover, clustering students based on cognitive styles and their overall performance should enable better adaption of the learning materials, with respect to their learning styles [14]. However, there is a significant lack of researches in analyzing student behavior to redesign the course.

The core of every adaptive educational system is a student model, which is also known as the learner model, which represents the current state of the student's domain Knowledge [12]. Using the student model, an adaptive system can support a range of adaptive learning interventions such as mastery learning, scaffolding, adaptive sequencing, or adaptive navigation support [15]. The idea of OSSM is to enhance the cognitive aspects of OSM with social aspects by allowing students to explore each other's models or the aggregated model of the class [12].

3 Methodology

Variety of data mining methods can apply to educational data repositories. According to Baker's taxonomy, there are the main five categories of data mining methods: Prediction, clustering, relationship mining, the distillation of data for human judgment and discovery with models [5]. Additionally, the aim of data exploration and visualization is to display data along with certain attributes and make extreme points, trends and clusters obvious to human eye [6]. In this research, we use clustering, classification, and visualization techniques implemented in WEKA data mining tool to address the research questions.

We use K-means and EM clustering techniques to address the first research question; First, we calculate the following features from the question set.
FirstCorrectTruePercentage (FCTP)
EverCorrectTruePercentage (ECTP)
UsedHintTruePercentage (UHTP)
AttemptsMean (AM)

Then, the Pearson correlation test on each pair of attributes is conducted and the result is shown in Table-1.
When clustering questions based on above mentioned four features, using K-means and EM clustering techniques, we observed a clear separation of the question set.
To address the second and third research questions, we use J48 decision learner, Multilayer perceptron, and Naïve Bayes classification algorithms. The number of attempts is converted to nominal values by applying Weka filters due to the requirement of the classification algorithms.

4 Results and Discussion

4.1 Data Exploration

The Pearson correlation test was conducted among the values of the attributes of the dataset. The test result is shown in Table 1.

The FCTP and ECTP have positively correlated and while FCTP has negatively correlated with UHTP and AM. If the question can solve at their first attempt, they will not use hints which provide for the question and vice versa. Furthermore, UHTP and AM have positively correlated. If the given question needs more than one attempt to solve, the students may use hints.
The calculated attribute values for each problemID are shown in Table 2.
The Maximum value of FirstCorrectTrue Percentage and the minimum value of AttemptsMean belong to the question DoubleX. The minimum value of FirstCorrectTrue-Percentage, the maximum value of UsedHintTruePercentage and the minimum value of EverCorrectTruePercentage belong to the ProblemID IsPunctuation. By considering basic statistics values, we can deduce idea about the questions.
The problemIDs graphed against the attributes are shown in the following figures. Each attribute is sorted in ascending order and graphed with problemID.

4.2.1 How to determine the comprehensiveness of the questions by considering student behaviors?

To address the above research question the clustering algorithms K-means, EM and Farthest First algorithms are used with the attributes AM, FCTP, UHTP, and ECTP. Since the last three attributes are binary the TRUE values have been considered as impact variables. The number of clusters has been determined with the purpose of clustering them as Hard, Moderate and Easy. 66% of the dataset was used as the training dataset and rest used as testing dataset. The results of each clustering algorithms are shown in Table 3 and 4.

The percentages of instances in each cluster formed by the aforementioned algorithms are shown in Table 5.
K-means algorithm simply assigns the instances to clusters based on the mean value of cluster. EM algorithm determines the probabilities of cluster membership based on one or more probability distribution.
The clusters formed using the K-Means algorithm can be labeled considering the means of attribute values of the instances assigned to each cluster. The labeled clusters are shown in Table 6 and example clusters are shown in Figure 5 by considering the attribute FirstCorrectTruePercentage.
Summarizing, the question set can be categorized into three as easy, moderate and hard based on the given attributes and hence, the comprehensiveness of the questions can be identified using clustering techniques.

4.2.2 Will the students be able to code a given question in their first attempt?

To address the above issue we trained three prediction models; Decision tree, Naïve Bayes and multilayer perceptron to predicting the attribute FirstCorrect. The number of attempts is ignored as the attempt number does not have an impact on the FirstCorrect value. The performances of the models are shown in Table 7.

According to the results, the MLP model performs better than

the other classifiers. However, the time taken to build the MLP model is considerably higher. The NB performs as same as MLP but with the relatively smaller time period and hence, the NB can be considered as the best classifier in terms of the prediction quality and the speed.
Summarizing, a prediction model can be trained to predict whether a student can code a given problem at his or her first attempt. To that end, we trained three prediction models; J48, NB, and MLP. The NB outperforms the other two models.

4.2.3 What would be the number of attempts one may try on a given question?

The attempt number is another way to measure the comprehensiveness of the problem. The higher number of attempts showed the challenge of the question and vice-versa. To predict the attempt number the same aforementioned classification techniques have been used. The prediction quality of each classifier is shown in Table 8.
The J48 model performed better than the other two models in terms of accuracy and speed.
Summarizing, No-linear prediction models such as J48, NB, and MLP can be trained to predict the number of attempts a student might try to provide a correct answer for a given question. The J48 decision tree provides a decent accuracy compare to the other two models.

5. Supporting Teachers and designers

5.1 Teachers

- New teaching strategies can be used to explain the questions with a large mean of attempts.
- Since the comprehensiveness of the problems has known, teachers can effectively prepare examinations.
- Using created models for predicting attempt number and firstCorrect value, teachers can trace the knowledge of the current cohort and get necessary actions to uplift the performance of students.

5.2 Course designers

- Can design the course based on the performance of the past students.
- The sequence of the assignment questions can be determined using existing cluster set.
- Can understand the sequence of knowledge grabbing of students.
- Make effective curriculum with the assistance of students' behaviors. Maintain the retention rate less by creating the course works in a more understandable way to students.

6 Conclusion

In this research, we describe the comprehensiveness of a question set by considering student learning behaviors. The information mined through this research can help teachers and course designers to fulfill necessitates with the purpose of edifying knowledge among students. The clustering methods led us to recognize the comprehensiveness of the questions on the basis of students' learning behaviors towards the given question set. Classification techniques led us to predict the students' responses to questions. Consequently, the knowledge of the students in a given problem domain can be evaluated using machine learning models and the appropriate techniques can be applied to enhance the students' knowledge on the problem domain.
The results of this research can be used in various aspects of intelligent tutoring including exam preparation, assignment sequencing, concept sequencing, course modifying, material uplifting and many more. The future extension of this research would be to analyze student performance on different question types. The retrieved information about each individual can be used to predict the future directions of students. Moreover, the usability of the hints of each individual and each question can be analyzed.

7 References

[1] M. K. Wenger E, Artificial Intelligent and Tutoring Systems, Los Altos, 1987.

[2] "Educational Data Mining," [Online]. Available: www.educationaldatamining.org. [Accessed 10 January 2019].

[3] P. B. Roya Hosseini, "JavaParser: A Fine-Grain Concept Indexing Tool for Java Problems," in CEUR Workshop, 2013.

[4] E. H. ,. K. K. Kelly Rivers, "Learning Curve Analysis for programming: Which concepts do Students Struggle with?," in International Computing Education Research Conference, 2016.

[5] K. Y. Ryan S.J.D. Baker, "The State of Educational Data Mining in 2009: A Review and Future Visions," vol. 1, no. Fall 2009, 2009.

[6] K. Y. Agathe Merceron, "Educational Data Mining : A case Study," in AIED, 2005.

[7] "PSLC DataShop," [Online]. Available: https://pslcdatashop.web.cmu.edu/Files?datasetId=2865.. [Accessed 05 January 2019].

[8] "Git Hub," [Online]. Available: https://github.com/thomaswp/CSEDM2019-Data-Challenge. [Accessed 04 January 2019].

[9] K. R. K. Kelly Rivers, "Data-Driven Hint Generation in Vast Solution Spaces : A Self-Improving Python Programming Tutor," Onternational Journal of Artificial Intelligent, vol. 27, pp. 37-64, 2017.

[10] E. S. ,. G. W. Peter Brusilovsky, "ELM- ART : An Intlligent tutoring system on World Wide Web," in International Conference on Intelligent Tutoring System, 1996.

[11] U. N. Jussi Kasurinen, "Estimating programming knowledge with Bayesian knowledge tracing," in ACM SIGCSE Bulletin, 2009.

[12] S. S. J. G. R. H. V. Z. P. Brusilovsky, "Open Social Student Modeling for Personalized Learning," IEEE, 2016.

[13] K. D. J. P. Erik W. Black ⁎, "Data for free: Using LMS activity logs to measure community in online courses," The Internet and Higher Education, vol. 11, no. 2, pp. 65-70, 2008.

[14] M. V. M. M. M. M. Milos Jovanovic, "Using data mining on student behavior and cognitive style data for improving e-learning," International Journal of Computational Intelligence Systems, vol. 5, no. 3, pp. 597-610, 2012.

[15] P. J. Durlach, "Support in a framework for instructional technology," Design recommendations for ITS: Instructional Management, vol. 2, pp. 297-310, 2014.

TABLE 1
PEARSON CORRELATION VALUES OF EACH SELECTED ATTRIBUTES

	FCTP	ECTP	UHTP	AM
FCTP	1.000	0.793	-0.767	-0.737
ECTP	0.793	1.000	-0.662	-0.518
UHTP	-0.767	-0.662	1.000	0.770
AM	-0.737	-0.518	0.770	1.000

TABLE 3
FINAL CLUSTER CENTROIDS OF K-MEANS ALGORITHM

		Clusters		
Attribute	Full Data (12.0)	0 (4.0)	1 (3.0)	2 (5.0)
FCTP	0.4984	0.1479	0.4097	0.8321
AM	2.7619	4.9648	2.319	1.2654
UHTP	0.1276	0.2731	0.0904	0.0334
ECTP	0.8329	0.5948	0.8115	1.0363

TABLE 4
FINAL CLUSTER CENTROIDS OF EM

	Clusters		
Attribute	0 (0.37)	1 (0.32)	2 (0.31)
FCTP			
Mean	0.4063	0.8184	0.172
Std.Dev	0.0601	0.0956	0.0409
AM			
Mean	2.8459	1.36	5.6655
Std.Dev	0.9113	0.2743	2.0196
UHTP			
Mean	0.1324	0.0348	0.2714
Std.Dev	0.0847	0.0305	0.0995
ECTP			
Mean	0.7993	1.0163	0.6193
Std.Dev	0.1063	0.0986	0.1121

TABLE 7
RESULTED CLUSTERS NAMED WITH HARD, EASY AND MODERATE

Classifier	Correctly Classified Instances (%)	Incorrectly Classified Instances (%)	TTime Taken to build model (sec)
J48	76.1	23.9	0.01
NB	79.2	20.8	0
MLP	79.3	20.7	2.02

TABLE 2
BASIC STATISTICS VALUES OF ATTRIBUTES WITH RELATE TO DIFFERENT PROBLEMID

No	Problem Id	FCTP	AM	UHTP	ECTP
1	HelloWorld	0.52	2.58	0.11	0.92
2	DoubleX	0.95	1.06	0.03	1
3	RaiseToPower	0.88	1.22	0.02	1
4	ConvertTwoDegree	0.36	1.73	0.13	0.65
5	LeftOverCandy	0.76	1.52	0	1
6	IntToFloat	0.89	1.07	0.02	0.96
7	FindRoot	0.46	1.71	0.02	0.63
8	HowManyEggCartons	0.24	4.39	0.17	0.66
9	KthDigit	0.16	6.19	0.22	0.68
10	NearestBusStop	0.15	3.76	0.18	0.67
11	HasTwoDigit	0.39	3.23	0.13	0.81
12	OverNineThousand	0.68	1.45	0.10	1.23
13	CanDrinkAlcohol	0.34	2.66	0.03	0.86
14	IsPunctuation	0.11	4.04	0.44	0.37
15	OverToN	0.20	9.68	0.36	0.68
16	BackwardsCombine	0.75	1.83	0.04	0.92
17	IsEvenPositive	0.17	5.88	0.25	0.67
18	FirstAndLast	0.41	4.41	0.23	0.91
19	SinglePigLatin	0.36	3.55	0.27	0.82
Maximum Value		0.95	9.68	0.44	1.23
Minimum Value		0.11	1.06	0	0.37

TABLE 5
PERCENTAGE OF INSTANCES IN EACH CLUSTER FORMED BY K-MEANS AND EM ALGORITHMS.

Clusters	Simple K-Means (%)	EM (%)
0	57	14
1	29	29
2	14	57

TABLE 6
RESULTED CLUSTERS NAMED WITH HARD, EASY AND MODERATE

Cluster Number	Name of the Cluster
Cluster 0	Hard
Cluster 1	Moderate
Cluster 2	Easy

TABLE 8
RESULTED CLUSTERS NAMED WITH HARD, EASY AND MODERATE

Classifier	Correctly classified instances (%)	Incorrectly classified Instances (%)	Time taken to build model (sec)
J48	66.7	33.3	0
NB	64.3	35.7	0
MLP	65.3	34.7	7.9

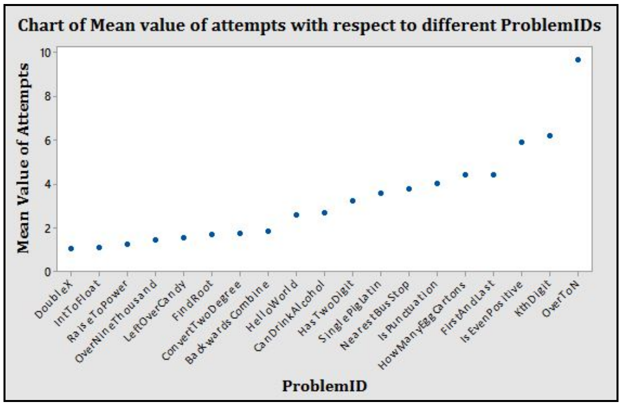

Fig. 1 Mean values of Attempts Vs ProblemID

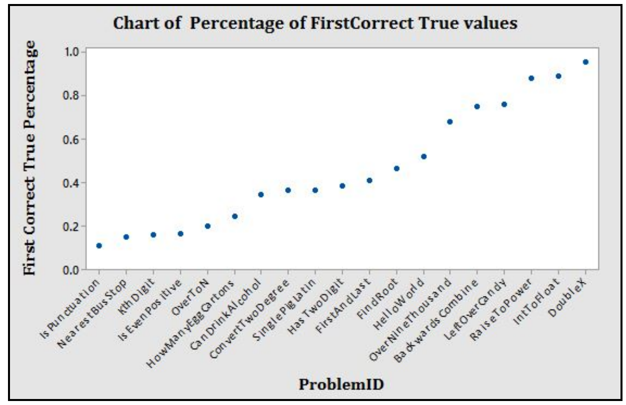

Fig. 2 First Correct True Percentage Vs ProblemID

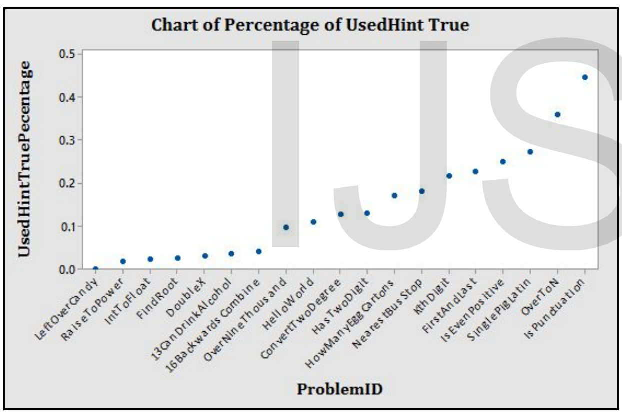

Fig. 3 Used Hints True Percentage Vs ProblemID

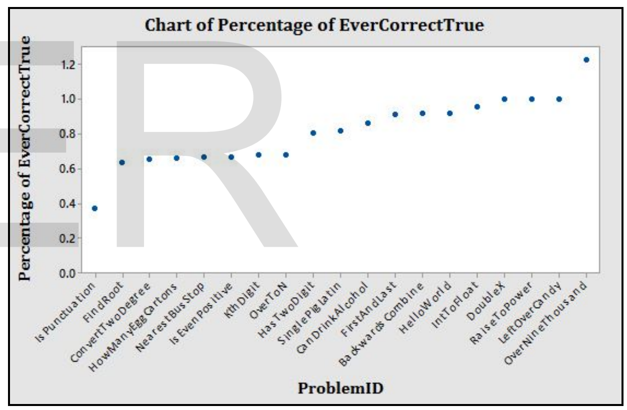

Fig. 4 Percentage of Ever Correct True Percentage Vs Problem-ID

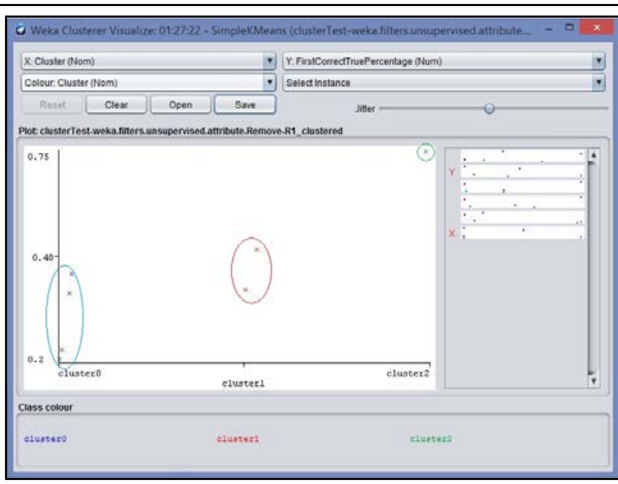

Fig. 5 Cluster set with respect to FirstCorrectTruePercentage