Journal of Information Technology Education: Research

An Official Publication of the Informing Science Institute
InformingScience.org

JITEResearch.org

Volume 16, 2017

Learning Management System with Prediction Model and Course-Content Recommendation Module

Digna S. Evale | Bulacan State University, Philippines | digna.evale@bulsu.edu.ph

Abstract

Aim/Purpose	This study is an attempt to enhance the existing learning management systems today through the integration of technology, particularly with educational data mining and recommendation systems.
Background	It utilized five-year historical data to find patterns for predicting student performance in Java Programming to generate appropriate course-content recommendations for the students based on their predicted performance.
Methodology	The author used two models for the system development: these are the Fayyad knowledge discovery in databases (KDD) process model for the data mining phase and the evolutionary prototyping for system development. WEKKA and SPSS were used to find meaningful patterns in the historical data, while Ruby on Rails platform was used to develop the software.
Contribution	The contribution of this study is the development of an LMS architecture that can be used to augment the capabilities of the existing systems by integrating a data mining technique for modelling the leaners profile; developing of an algorithm for generating predictions; and making the most appropriate recommendations for the learners based on prior knowledge and learning styles.
Findings	The result shows that J48 was the best data mining algorithm to be implemented for finding patterns in the data sets used in this study. Attributes such as age, gender, class schedule, and grades in other programming subjects were found relevant in predicting student performance in Java.
Recommendations for Practitioners	It is recommended that collaboration between the academe and IT industry be strengthened to develop a more advanced LMS which could enhance classroom teaching and improve the learning process.
Recommendation for Researchers	Combination of multiple algorithm in classifying data set is recommended to further improve the algorithm and rule sets of prediction. Inclusion of intrinsic attributes as part of data set aside from personal and academic records is also recommended.

Accepted by Editor Beth Thomsett-Scott │ Received: August 12, 2016 │ Revised: November 28, 2016, May 15, August 1, 2017 │ Accepted: October 31, 2017.

Cite as: Evale, D. S. (2017). Learning management system with prediction model and course-content recommendation module. *Journal of Information Technology Education: Research, 16,* 437-457. https://doi.org/10.28945/3883

Impact on Society	This LMS can be used to produce independent learners.
Future Research	Study about the impact of implementing this LMS in classroom environment will be conducted on the second phase.
Keywords	learning management systems, educational data mining, prediction model, performance prediction, attribute selection, course-content recommendation, index of learning styles

INTRODUCTION

With the advent of electronic learning and the diversity of learners, using learning management systems, educational data mining, prediction model, performance prediction, attribute selection, course-content recommendation, index of learning styles in higher education, an application that can provide a more personal approach in teaching and learning process should be studied. Since the application of Recommender Systems (RS) in the field of education is still a new area to be explored, this study aims to create a fusion of a learning management system (LMS) and RS to develop a web-based learning environment particularly designed for a JAVA SE Programming course. The resultant product will provide personalized recommendations of specific topics and learning activities that are well-matched for individual learners, based on their personal profile, learning style, prior knowledge, and expertise level.

Students or learners, just like customers or product consumers, are also in need of personalized recommendations for learning resources and activities that will match-up with their personal needs, preferences, prior knowledge, and current situation to facilitate an individualized learning or to offer a more personalized learning environment that will promote learning outcomes (Drachsler, Hummel & Koper, 2007; Verpoorten, Glahn, Kravcik, Ternier & Specht, 2009). This need is especially true in subjects that require higher levels of analytical and logical thinking skills, such as computer programming. Being proficient in programming has several advantages, especially if the student is well-versed in a programming language widely used in the field. On July 2015, the Institute of Electronics and Electrical Engineers Spectrum – the flagship publication of IEEE- published a list of top ten programming languages in the world and Java achieved the top position (Cass, 2015). Java is a robust, platform-independent, distributed, and object-oriented programming language. Java is a best choice for system development which requires object-oriented programming concepts and for internet programming (Kafura, 2000). There are many career paths or job opportunities available for Java programmers, since we are living in a technology-driven world where almost every field needs computerized systems and applications (Nelson, 2013).

It is common knowledge that student aptitude or ability to carry-out logical analysis and competence in doing logic formulation is a major factor affecting their performance in programming subjects (Milne & Rowe, 2002). Several studies were conducted regarding the challenges in teaching programming and Java technologies (Ala-Mutka, 2004; Carter & Jenkins, 2010; Clark, MacNish, & Royle, 1998 Mehic & Hasan, 2001; Pendergast, 2006). Teaching programming requires that the methodologies and strategies be appropriate to the learning styles of the target learners. Determining students learning styles, whether they are global or sequential, active or reflective, sensing or intuitive, visual or verbal learners, greatly affects the way they are motivated to learn. This could also help the teacher since there are students who perceive programming as a boring subject (Jenkins, 2002). Gomes and Mendes (2007) recognized that the teachers' inability to support all types of students preferential learning styles is a contributing factor to as why students are having difficulties in learning programming subjects. An assessment of prior knowledge about the subject is also vital in order to determine the strengths and weaknesses of students. Ignoring these factors might cause serious difficulties in learning (Adair & Jaeger, 2011; Jenkins, 2002). The vast scope and complexity of Java, which if not given proper means of introduction and appropriate presentation of materials, might also posit learning problems (Lister, 2004).

For these reasons, the researcher created an LMS with a course content recommendation system for Java programming based on learners' predicted performance that will benefit the learners, especially those who will be evaluated to be in need of learning assistance. Its first specific objective is to integrate educational data mining in the LMS for the purpose of learner classification (also known as prediction) which can be of help in improving learning process (Romero, Ventura, Espejo, & Hervás, 2008) and even in improving institutional effectiveness (Huebner, 2013). Its second objective is to integrate an assessment module that can evaluate the students' index of learning styles and prior knowledge in Java for the purpose of generating recommendations. This way, the students will be given a chance to assess their strengths and weaknesses as learners and to identify the topics with which they could possibly encounter difficulties and later might cause them to fail. The result can be used to serve as an immediate preventive remedy to improve students' performance.

LITERATURE REVIEW

Learning can virtually take place everywhere today. With the Internet, electronic learning (e-learning) is becoming an integral part of education. In fact, Pinner (2014) reported that higher education ranks first in the list of top industry utilizing e-learning. Research has been done regarding theories and practices of online learning which highlight its significant benefits and contributions for the learners. Thus, more educational institutions are gearing up to take advantage of this technology to enhance their mode of delivering classroom instructions (Anderson, 2008). This attempt to provide classroom instruction and learning materials in a new medium gave birth to the era of virtual learning environments, distance learning, learning management systems, content management systems, and learning content management systems. Though these learning alternatives may sound confusing and possess similarities in terms of tools or platforms being used, they all have distinct features that define their differences in terms of functionalities and intended users or target clients. Among these, learning management systems (LMS) are the most suitable and appropriate to cater to the needs in higher education settings (Dubowy, 2013). Al-Busaida and Al-Shihi (2010) stated that LMS have great potential for being in demand worldwide as an alternative learning pathway both in the realm of education and in the field of industry. These are now becoming one of the most important tools and ventures in augmenting the teaching and learning process (Learning Management System Evaluation 2011-2012, 2011).

Among the top functions and purposes of an LMS are the development, storage, sharing, and management of learning objects, such as student activities, reading materials, portfolios, and performance records. It may also serve as a data bank of examinations and other evaluation tools. LMSs can also be used to generate various records and reports that may aid the schools or organizations to become more effective in delivering of learning process (Yasar & Adiguzel, 2010). With the advent of advanced network technology, the power of Web 2.0 may now be integrated into LMS to promote active interaction not only between students and teachers but also between and among students themselves as well. The vast use of interactive discussion tools such as forums, blogs, wikis, and chat rooms promote collaboration among students via different modalities. LMS can further improve the students' capacity in constructing new sets of knowledge on their own making them independent learners (Lonn, 2009). In fact, a study conducted by Firat (2016) regarding the effect of LMS learning behavior in the students' academic performance, stated that "Learning management systems (LMS) have been proven to encourage a constructive approach to knowledge acquisition and to support active learning". LMS can richly augment the traditional face-to-face classroom set-up by providing a boundless communication between the teachers and the students while allowing the students learn at their own pace (Shulamit &Yossi, 2011).

In general, LMS can make classroom management easier since they provide tools that can aid the facilitators of learning (usually the trainers or classroom instructors) in various tasks from the uploading and management of lectures, activities, and other learning resources up to the assessment and evaluation of learning outcomes (Educause Learning Initiative, 2011). A modified traditional LMS applied in education is usually composed of different management modules as shown in Figure 1 (adopted from the work of Gadhavi, Patel, and Patel, 2013).

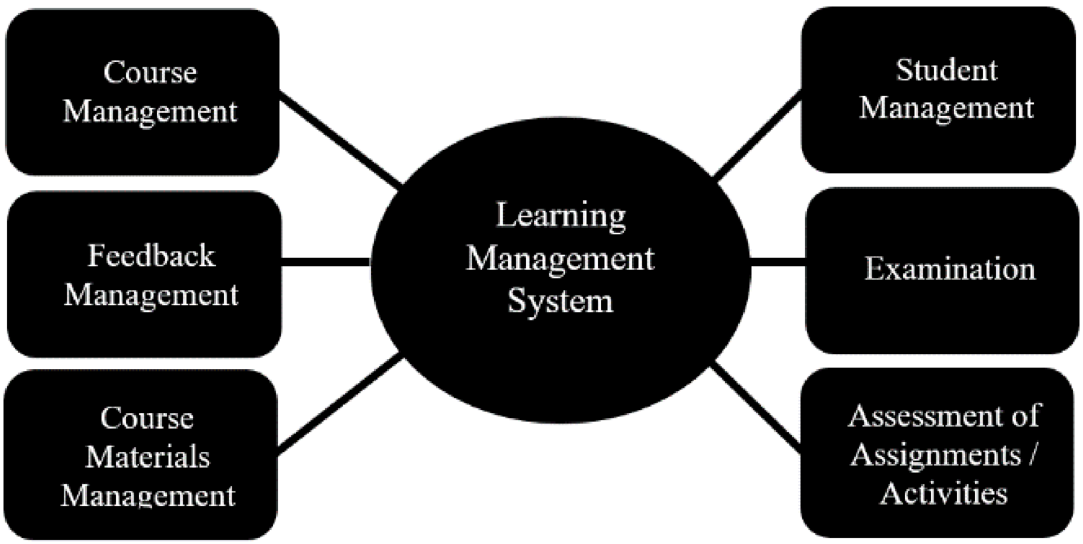

Figure 1: Model of traditional learning management system

In the conventional LMS model, the course management module is a tool that provides an interface to enlist new courses or new classes together with the assigned instructor. From those courses, instructors can register their students and manage their learners' profiles in the student management module. The course materials management module will let instructors upload learning resources in various forms such as tutorials, text, audio and video format. Examination and assessment modules provide an automated way for the instructors to test the learning output of their students. Further, feedback management allows the students to give private comments or opinions about the instructors or the course as a whole (Gadhavi et al., 2013).

The kind of LMS model described above is being used today to augment the typical teaching and learning process inside the classroom. The mechanism of this model is good enough to perform the LMS functions cited in various studies (Al-Busaida et al., 2010; Arh, Matija, Srdjevic, & Srdjevic, 2012), which is to provide a platform for a learning environment that will enhance the way instructors manage and share their learning resources towards their learners, as well as how they will assess various learning outcomes. However, that kind of model, though effective as to its designed purpose, cannot suggest a course of action after evaluation and might put the electronic learning process into an impasse. Thus, a model which could suggest an appropriate piece of learning material or personalized course content best fitted to the individual learner is highly necessary. Such models known to us as Personal Learning Environments (PLE) focus on providing various tools, services, and artifacts so that the system can adapt to students learning needs on the fly (Mödritscher, 2010). The system is a dynamic learning environment that may vary depending on various factors which might affect the students' interest. PLE are used to facilitate student-centered learning approach (Arrufat & Sanchez, 2012) where the learner can have control on his or her own learning environment based on his or her personal learning preferences and choices best suited for learners needs (Hicks & Sinkinson, 2014). PLE are basically founded on the framework of learning management systems coupled with data mining to perform functions for personalized recommendations. This model is also called adaptive LMS and offers a unique learning experience to the user by providing a customized environment (based on personal interest, personality, prior knowledge, and skills) fitted for achieving the learning goals of the student (Yaghmaie & Bahreininejad, 2011).

Sunil and Saini (2013) defined recommender systems (RS) as software that can operate as a personalized information agent. As the system reacts to a user's request, it adjusts its response in a manner that will fit in to the specific need of the user, either based on their profile or history of access. Sikka, Dhankhar, and Rana (2012) stated that e-commerce is now widely implementing the concept of using recommendation systems to attract customers. RS are not fully implemented yet in the discipline of e-learning, although there are frameworks which are already conceptualized and presented. In fact, the success and stability of PLEs, which mainly operate on digital space platforms, are still yet to be seen (Archee, 2012).

A general proposal structure of RS by Imatzi and Megias (2008), shown in Figure 2, and the framework architecture proposed by Imran, Hoang, Chang, and Graf (2014) in Figure 3 serve as the guiding paradigms for this study. Both of the frameworks have their own strengths; however, there are underlying limitations which exist either in the types of the objects that can be recommended by the LMS or in the process and results of its integrated RS. The framework in Figure 2 implemented a very good combination of the hybrid recommendation system as it made use of the content-based RS, collaborative filtering, demographic-based system, and rule-based filtering. However, it has no concrete learner modelling system to capture the learners' profile and the LMS is limited in recommending a course that might be taken by the students. There is no other object that can be recommended by the system.

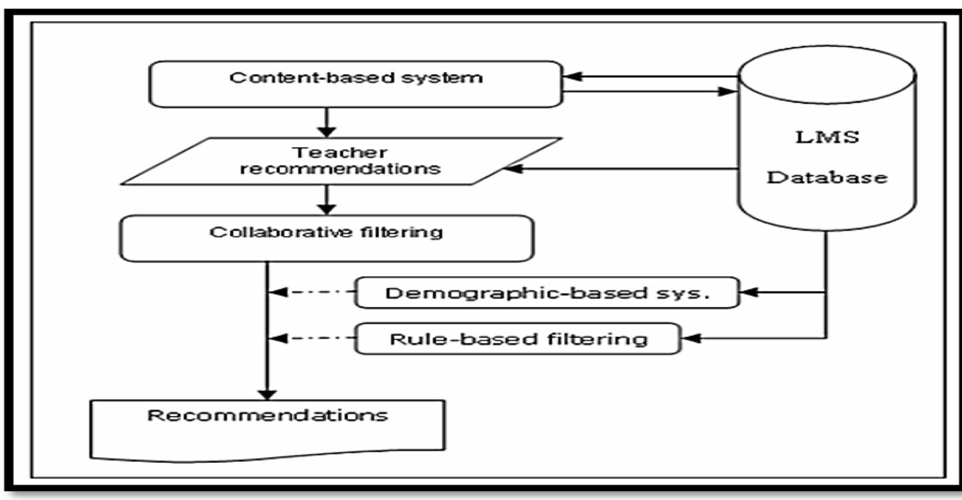

Figure 2: A general proposal structure of RS in LMS by Imatzi and Megias (2008)

Figure 3 shows the framework of an LMS which recommends personalized topics to the learner. Its strength can be credited to the presence of a modelling module which can automatically generate a learner's model. This learner model will store the different information about the learner beyond the personal and demographic profile in a database. This should increase the chance that the recommendation be made by the RS is closely fitted to the needs of the learner. However, the drawback is that this framework uses only neighborhood formation and a rule-based algorithm. In using neighborhood formation, an assumption has to be made that a learner has already taken or studied a particular learning object or topic in the LMS and that he or she has successfully performed the learning activities associated with it; otherwise, the system cannot make recommendations to the next user, called "neighbor". As well known, the success of a learner in a learning object is not always guaranteed.

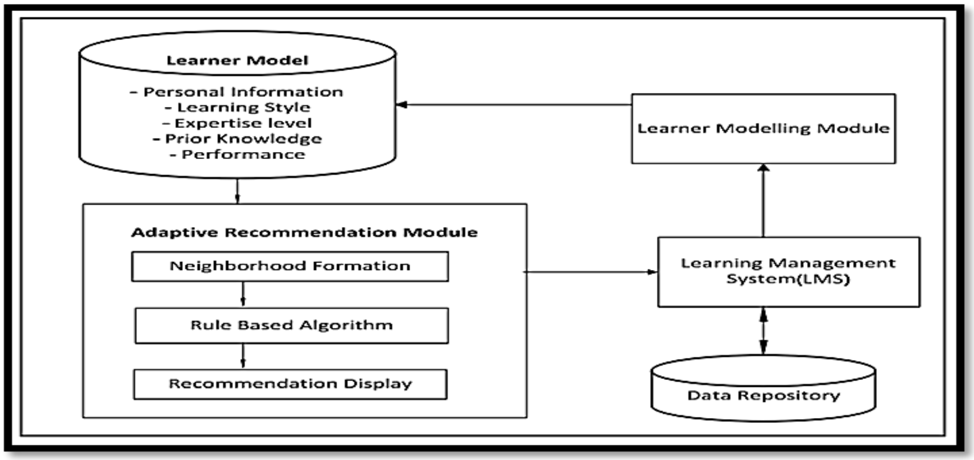

Figure 3: Architecture of proposed framework by Imran et al. (2014)

The focal point of this study is thus to fill the gap found in literature through the development of an LMS architecture that can be used to augment the capabilities of the existing systems by integrating a data mining technique for modelling the leaners profile, developing of an algorithm for generating predictions, and making the most appropriate recommendations for the learners based on prior knowledge and learning styles.

METHODOLOGIES

DEVELOPMENT

To carry out this study, the author used two models for the system development: these are the Fayyad knowledge discovery in databases (KDD) process model for the data mining phase and the evolutionary prototyping for system development. KDD process model (Figure 4) is considered to be of the best academic research model for data mining in education (Fayyad, Piatetsky-Shapiro, Smith, & Uthurusamy, 1996) while evolutionary prototyping is well-suited for the system development which requires building prototype software in an incremental manner (Overmyer, 1991).

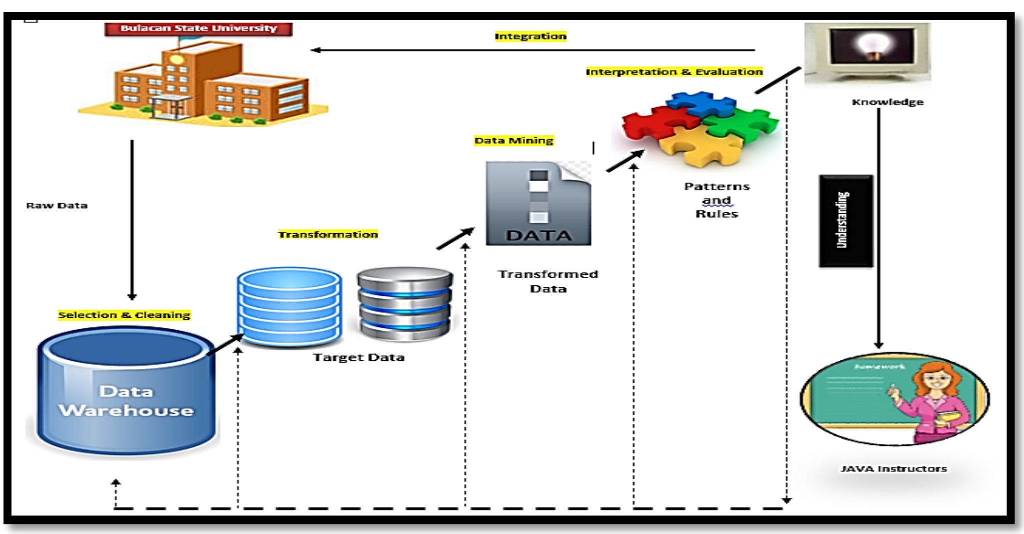

Figure 4: Knowledge discovery in database process model

There are five stages for KDD: selection, pre-processing, transformation, mining, and interpretation. In selection, possible attributes are collected for the data set while pre-processing is the filtering and removing of irrelevant data. Transformation is determining the most suited data mining technique to provide the best prediction algorithm. Mining is discovering the pattern captured through classification rules, regression models, or decision tree. Interpretation and evaluation is the process of visualization or finding the meaning of data patterns extracted from the mining stage.

Table 1 shows the different attributes relevant for modeling a learner profile. The attribute was evaluated through the Waikato Environment for Knowledge Analysis (WEKA) data mining tool and the IBM Statistical Package for the Social Science (SPSS). There were eight attributes: namely, gender, age, course, section, schedule, and three academic performance measures for programming languages.

Table 1: Possible attributes for building a learner's model

Attributes	Values	Description
gender	{female = 0; male = 1}	gender of the learner
age	{ less than or equal 16, 17, 18, 19, 20, 21 above 21}	age of the learner
course	{ BS Computer Science = 0, BIT Computer Technology = 1, BS Information Technology = 2 }	course of the learner
section	{A, B, C, D, E, F, G, H, I, J}	section where the learner belongs
schedule	{morning = 0; afternoon=1; evening=2}	Java class schedule 7am-12nn = am; 12:nn-5pm = pm; 5pmm-8pm = eve
*Prog1	{1.0,1.25,1.5,2.0,2.25,2.5,3.0,4.0,5.0}	grade of the learner in Turbo C programming
*Prog2	{1.0,1.25,1.5,2.0,2.25,2.5,3.0,4.0,5.0}	grade of the learner in C ++ programming
*Prog3	{1.0,1.25,1.5,2.0,2.25,2.5,3.0,4.0,5.0}	grade of the learner in Visual Basic programming
* 1.0 as the highest and 5.0 as the lowest		

The LMS system was developed using evolutionary prototyping (Figure 5) and contains two sets of modules – the first one is for prediction and the other is an examination module for the evaluation of learning style and assessment of prior knowledge on Java programming.

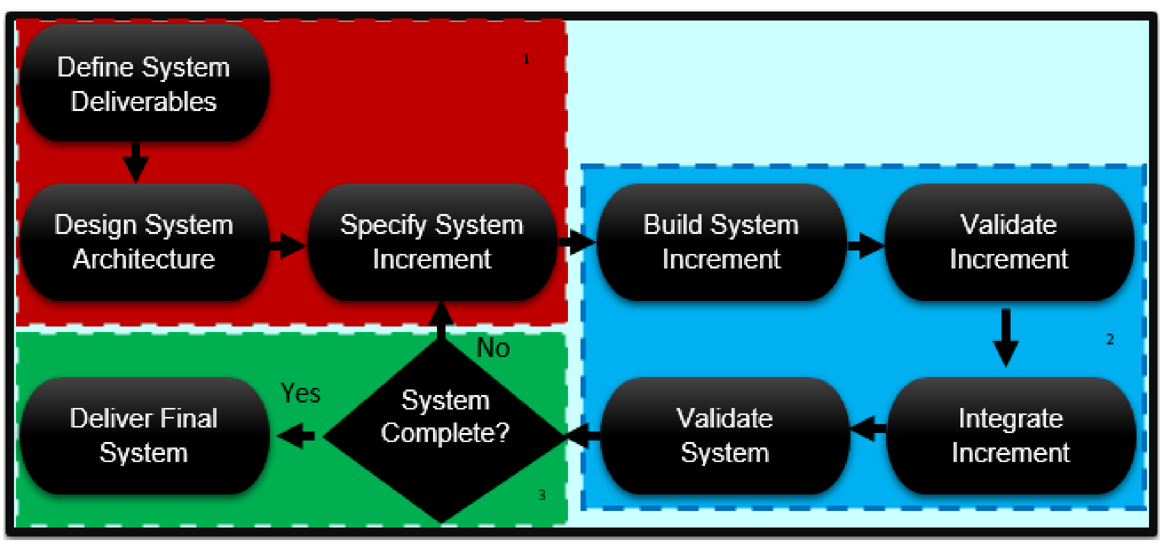

Figure 5: Processes in incremental prototyping (Butter, 2014)

Prediction of whether the student will pass or fail in Java programming will be automatically generated by the system. If predicted as 'failed', the learner will be required to proceed with the examination module. The first exam is a forty-one-item test which can classify the student as global or sequential, active or reflective, sensing or intuitive, and visual or verbal learner. The next exam is a one hundred forty–item questionnaire which will test the level of understanding of the students in Java programming. Questions are from concepts which are already included in their prior programming subjects, but the implementation is in the Java language. At the end of the test, reports that can be used to improve the teaching and learning process will be generated for the students and teachers. The prototype was created using Ruby on Rails for the front-end and SQL for the back-end. The system can be configured in a local area network environment or can be accessed via a web-browser for greater accessibility to students.

Validity Testing

To determine the validity of the predictions generated by the system regarding the learner's performance in Java programming as either "Passed" or "Failed", Cohen's kappa (Cohen,1960) will be used to compare the predicted grade of the system and the actual final grade of the learner.

Cohen's kappa (K) is a measure of the agreement between two raters who each classify N items into C mutually exclusive categories. The two raters either agree or disagree in their rating; there are no degrees of disagreement. The equation for K is:

$$\mathbf{K} = \frac{\mathbf{P(A)} - \boldsymbol{P(E)}}{\mathbf{1} - \boldsymbol{P(E)}}$$ *Equation No. 1*

where: P(A) = number of agreements
P(E) = number of agreements expected by chance

After computing for the value of K, the magnitude guidelines shown in Table 2, as suggested by Landis and Koch (1977) can then be used to interpret this kappa value.

Table 2: Interpretation of the magnitude of kappa values as suggested by Landis and Koch

Kappa Statistic	Strength of Agreement
< 0.00	Poor
0.00 - 0.20	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.00	Almost perfect

To illustrate how Cohen's kappa is used, consider a scenario where two raters are tasked to classify N items into two possible categories (e.g., Passed (P) or Failed (F)), the format of contingency table that will be created is shown in Table 3.

Table 3: Format of the contingency table for computing Cohen's kappa

	RATER 1			
RATER 2		PASSED	FAILED	TOTAL
	PASSED	A	B	G=A+B
	FAILED	C	D	H=C+D
	TOTAL	E=A+C	F=B+D	N=A+B+C+D

In Table 3, A is the number of items that both raters classified into Passed category, while B is the number of items that Rater 1 classified into the Passed category but Rater 2 classified into the Failed category, etc. With this contingency table, computing for P(A) and P(E) are as follows:

$$\mathbf{P(A)} = \frac{\mathbf{A} + \boldsymbol{D}}{\boldsymbol{N}} \qquad \textit{Equation No. 2}$$

and

$$\mathbf{P(E)} = \frac{\mathbf{E}}{\boldsymbol{N}} * \frac{\mathbf{G}}{\boldsymbol{N}} + \frac{\mathbf{F}}{\boldsymbol{N}} * \frac{\mathbf{H}}{\boldsymbol{N}} \qquad \textit{Equation No. 3}$$

To illustrate how kappa statistics work, consider the contingency table described in Table 4 that describes the classification for two raters on a one hundred-item set.

Table 4: Sample contingency table for computing Cohen's kappa

	RATER 1			
RATER 2		PASSED	FAILED	TOTAL
	PASSED	52	13	65
	FAILED	10	25	35
	TOTAL	62	38	100

Computing for the kappa of the two raters in the contingency table, we get

from Equation No. 1

$$\mathbf{P(A)} = \frac{\mathbf{52 + 25}}{\mathbf{100}} = \mathbf{0.770}$$

from Equation No. 2

$$\mathbf{P(E)} = \frac{\mathbf{62}}{\mathbf{100}} * \frac{\mathbf{65}}{\mathbf{100}} + \frac{\mathbf{38}}{\mathbf{100}} * \frac{\mathbf{35}}{\mathbf{100}} = \mathbf{0.536}$$

from Equation No. 3

$$\mathbf{K} = \frac{\mathbf{0.777 - 0.536}}{\mathbf{1 - 0.536}} = \mathbf{0.50431}$$

Grove et al. (1981) indicated that the level of acceptable interrater reliability as being around K>0.6 or even K>0.5.

Further, to get the percentage of accuracy or correct prediction, this formula can be used:

$$Percentage\ correct = \frac{A + D}{A + B + C + D} \qquad \textit{Equation No. 4}$$

RESULTS

LMS ARCHITECTURE

The LMS architecture shown in Figure 6 depicts some of the prominent details significant for the successful development of a learning management system for Java programming with prediction model and course-content recommendation module.

The first phase of this study focused on the data mining process needed by the system, especially the attribute selection and classification that are essential in classifying data correctly. The output of the first phase was the prediction model for the learners' performance in Java programming.

The second phase dealt with the development of the prototype system, which includes the incremental development of three subsystems: prediction system, ILS evaluation, and the Java programming examination. At the end of this phase, the output is a course-content recommendation for the learner containing the programming topics recommended based on the result of ILS and Java examination.

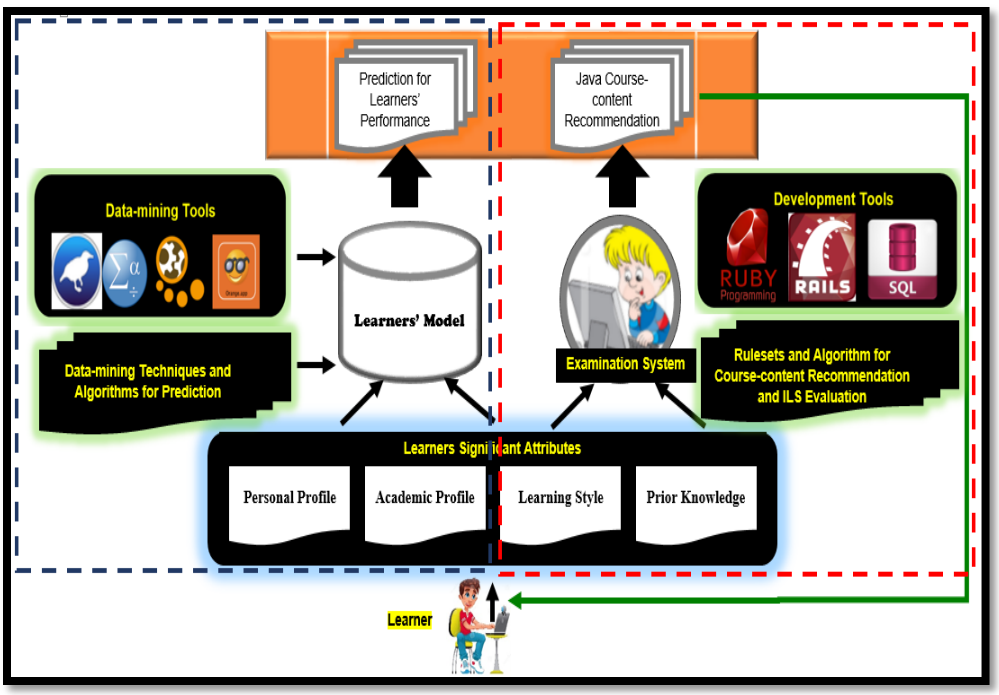

Figure 6: Architecture of LMS

ATTRIBUTE SELECTION

In order to identify the major attributes relevant in developing a data model and rule sets for predicting the performance of Java learners, five-year historical data (2010-2015) of students with Java programming experience in the Bulacan State University was filtered and analyzed using different algorithms. Attribute selection was done using standard regression analysis, forward and backward conditional regression, likelihood ratio, and WALD test using SPSS. WEKA was also used to conduct preprocessing through filtering by *AttributeSelection* before the data was subjected to an attribute evaluator. Table 5 shows the summary of results from the attribute selection process.

From the eight original attributes, there are two variables which were found to be insignificant. With a critical *p value* of .05 (significant predictors should have smaller critical p value), binary logistic regression using SPSS found *section* and *course* as highly insignificant with .747 and .221 p value respectively. *Gender* can be interpreted as not highly significant since its p value is .016 both in standard and forward regression and .053 in backward regression.

Table 5: Summary of attribute selection result

Tool	Method		Attribute w/p-value >0.000	p-value
IBM SPSS 20	Standard Regression		Gender	.016
			Section	.747
	Forward Regression	Conditional Likelihood Ratio WALD	Gender	.016
	Backward Regression	Conditional Likelihood Ratio WALD	Course	.221
			Gender	.053
*Other attributes p-value =0.000 * Predicted Percentage Correct of Classification Table = 89.9%				
Tool	**Method**		**Attribute Removed**	**merit of best subset**
Weka	CfsSubsetEval(BestFirst)		course, section	.239
	* Predicted Percentage Correct of Classification Table = 76.1%			
	Method		**Attribute with no. of folds !=10**	**No. of folds**
	CfsSubsetEval (GreedyStepWise)		Gender	70%
			Course	0%
			Section	0%
	* Other attributes appeared 10 times (100%) in 10-fold validation			

Attributes were further analyzed using WEKA. In the pre-processing step, filtering through *Attribute-Selection* was done, and the result was parallel with the SPSS data – *course* and *section* were automatically removed, meaning they are found as highly insignificant.

Figure 7 shows the visualization of each attribute of experimental data. To further verify the significance of the attribute gender, CfsSubsetEvaluation was performed. In BestFirst method, gender was found significant with 0.239 value of merit of best subset found (value is from 0 to 1 representing the incorrectly classified instances) which means that there is 76.1 percent of correctly classified instances. In GreedyStepWise search method selected through Cross Validation, course and section are not found in any of the ten folds while gender appeared in 7 out of 10 folds (70%). With these data, the researcher came up with following attributes as significant predictors: age, gender, schedule, grade in Programming 1, grade in Programming 2, and grade in Programming 3.

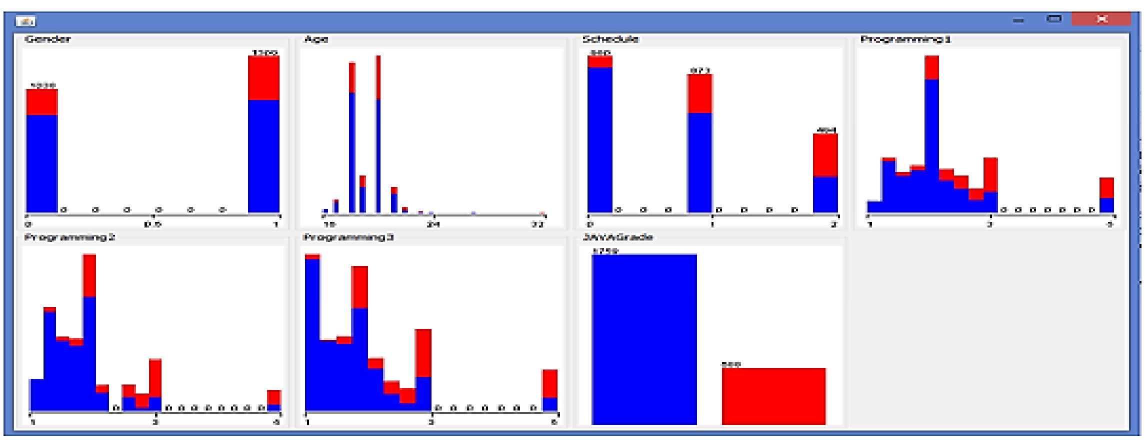

Figure 7: Visualization of different attributes

DETERMINING THE BEST DATA MINING ALGORITHM

The six significant attributes were used in determining the best model and rule sets for prediction. Classification was done using several data mining algorithms, and the one which gave the highest percentage of correct prediction was used. Table 6 summarizes the results.

Table 6: Summary of the accuracy result of different algorithm

Technique	Algorithm	Accuracy	Kappa / *std. error
Tree Classifiers	CHAID	87.5	.007
	Exhaustive CHAID		
	CRT	89.8	.006
	QUEST	89.4	.006
	J48	94.74	.8464
Bayes Classifiers	BayesNet	92.84	*.7938
	Naïve Bayes	84.63	*.5642
Rule Classifier	JRip	94.21	*.8302

C4.5, which is one of the most widely-used classification algorithms (Smith & Bull, 2003), gave birth to J48 - an improved implementation of a decision tree classifier that can be used for predicting performance. J48 gained popularity because of its high percentage of correct prediction, optimized decision tree diagram, and straightforward rule sets which do not need complicated interpretation (Rajput, Aharwal, Dubey, Saxena, & Raghuvanshi, 2011).

The result in Table 6 shows that in predicting the performance of students in Java programming, J48 is the best algorithm to be used since it has the highest percentage of accuracy in making predictions and at the same time has the highest Cohen's kappa. Kappa value of 0.8464 means that the prediction is strongly reliable with sixty-four to eighty-one percent reliability based on the suggested Cohen's kappa interpretation of McHugh (2012).

J48 generated decision tree and rules for classification which was translated into a machine-readable codes using Ruby on Rails. Figure 8 shows an excerpt of the code.

```
if user.prog_2 <= 2.25
  if user.schedule == "am" || user.schedule == "pm"
    if user.prog_3 <= 1.75
      status = "passed"
    else #user.prog_3 > 1.75
      if user.prog_2 <= 1.753
        if user.prog_1 <= 2.5
          status = "passed"
        else# user.prog_1 > 2.5
          if user.age <= 18
            status = "passed"
          else# user.age > 18
            if user.prog_1 <= 3
              status = "passed"
            else# user.prog_1 > 3
              status = "failed"
            end
          end
        end
      else# user.prog_2 > 1.753
        if user.prog_1 <= 2
          if user.schedule == "am"
            status = "passed"
```

Figure 8: Sample of partial code developed to translate the rules generated by J48 classifier

DEVELOPMENT OF WEB-BASED EXAMINATION SYSTEM

The system was developed using Ruby on Rails guided by the principles of the incremental prototyping development model. The LMS contains two sets of modules – the first one is for prediction and the second is an examination system for evaluation of learning style and assessment of prior knowledge on Java programming. The student will be asked to enter all the information found to be significant in predicting his performance, along with some other data for security process, when signing up. Figure 9 shows the student's sign up form for capturing pertinent data.

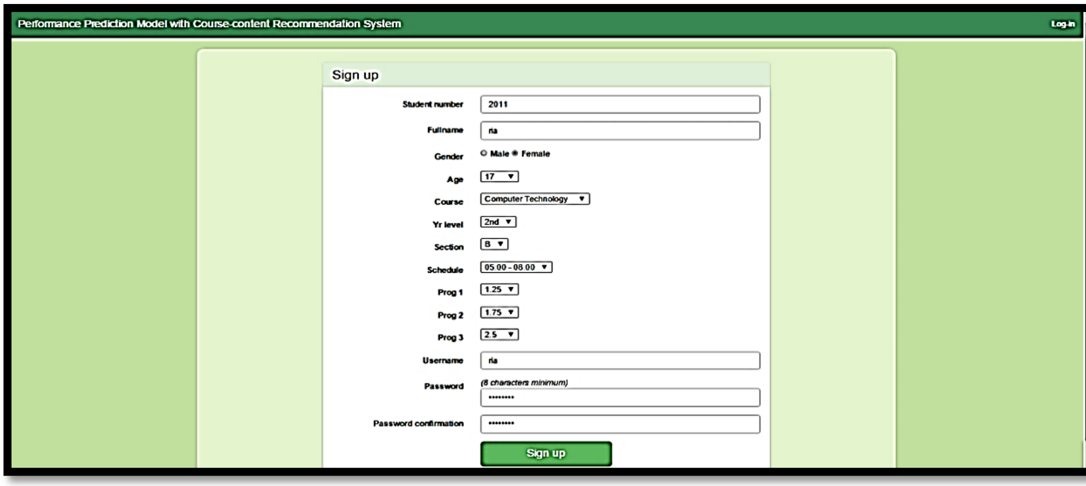

Figure 9: Student sign up form

Students should enter their school identification number, full name, gender, course, year level, section, class schedule, and grades in previous programming languages. They are also required to create a unique username and password for their future log in. Based from those personal attributes of the students, the system can already predict the tendency of success and failure in the subject. This prediction was made possible by the data mining algorithm embedded in the system. For the students who were predicted as "passed", logging in to the next module is optional (Figure 10); however, those who were predicted as "failed" will be required to proceed with the ILS evaluation and Java examination modules.

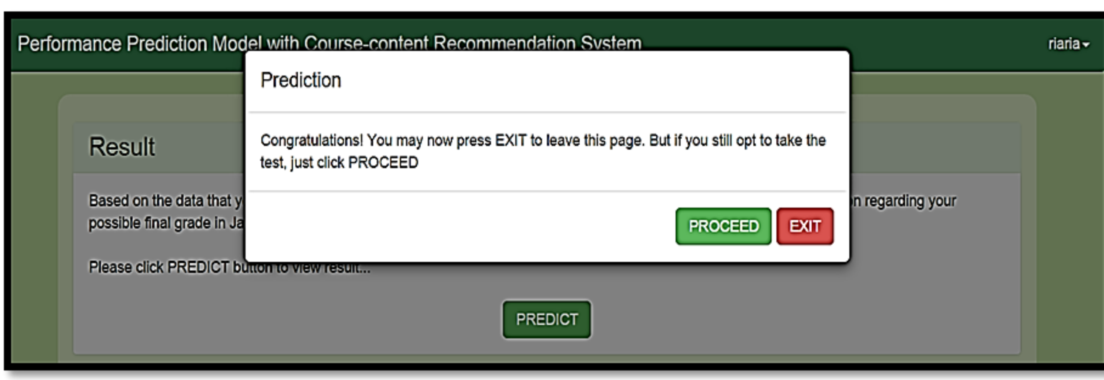

Figure 10: Performance prediction result

Figure 11 shows the web interface for capturing the ILS questions devised by Dr. Richard Felder. The exam is a multiple-choice type with two options for each question. An algorithm that prohibits the selection of more than one option is embedded in the system; thus, the examinee is only allowed to choose exactly one answer per question.

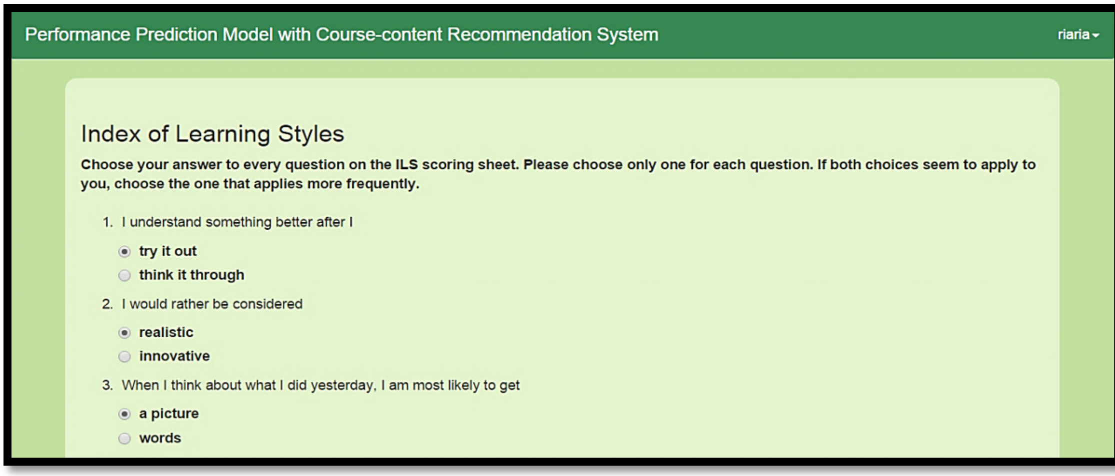

Figure 11: Index of learning style assessment page

Immediately after the evaluation, the result will be automatically generated by the system. An algorithm for evaluating the result was designed and developed based on the scoring scheme given by Dr. Felder. The result will determine the student's learning style and will show the general characteristics of being such type of learner, as shown in Figure 12.

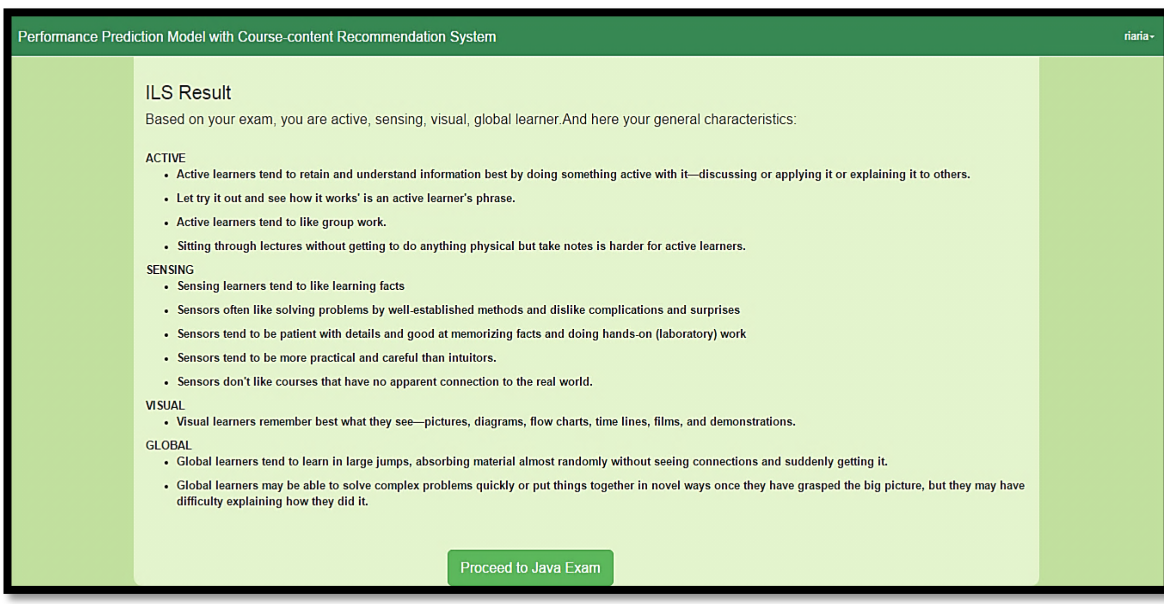

Figure 12: Index of learning style evaluation result

After completing the index of learning style exam, students will be directed to another page for the assessment of their knowledge and skills in Java programming. The questionnaire was validated by certified Java specialists and experts. It is a multiple-choice type exam with 140-test items (fourteen topics with ten questions each) which generally cover almost all of the fundamental competencies in various programming subjects implemented using Java. Questions are given randomly, even the four choices for each question were also randomized to reduce, if not completely avoid, the chance of getting the exact question twice. This is possible because of the implementation of a random test generator in the test bank.

The examinee is not allowed to leave unanswered questions. Answers for all items should be completed so that the examination form can be successfully submitted for checking. As shown in Figure

13, a message prompt asking the user to review all items found unanswered by the system is automatically generated upon the submission of questionnaire with an incomplete response.

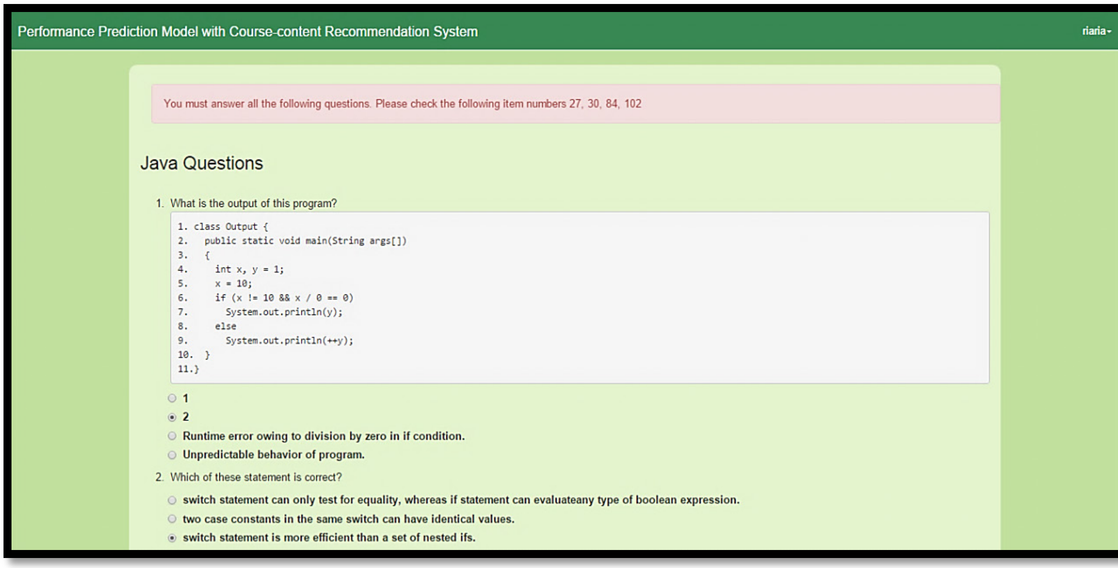

Figure 13: Java programming assessment page

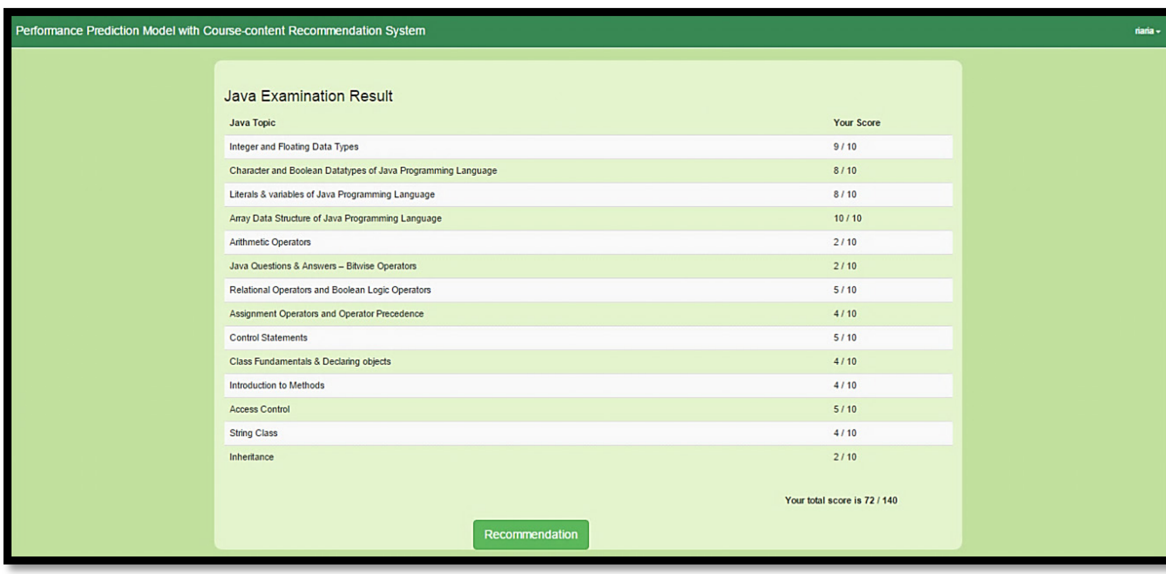

Figure 14: Java programming assessment result page

The system will grade and display the result of the assessment right after the successful submission of the test. Since there is an algorithm which tags and link each question to a certain topic or category where it belongs, aside from the total score, the system can also display the exam result as a summary of each category. This will help the students get an overall impression of what Java programming is all about and the probability of how well they could possibly perform in the subject based on their current level of understanding. Figure 14 shows a sample of the result.

Aside from the automatic checking of student's scores, the system also automatically generates recommendations. The system will compute the percentage of scores for each category, then identify all categories above seventy five percent as the student's strengths. On the categories below seventy five percent, the system will recommend topics which the students should give more focus on to improve future performance.

The system will also generate helpful tips or recommendations concerning how the student should handle learning problems and activities based on the result of his or her ILS. Figure 15 shows an example of this.

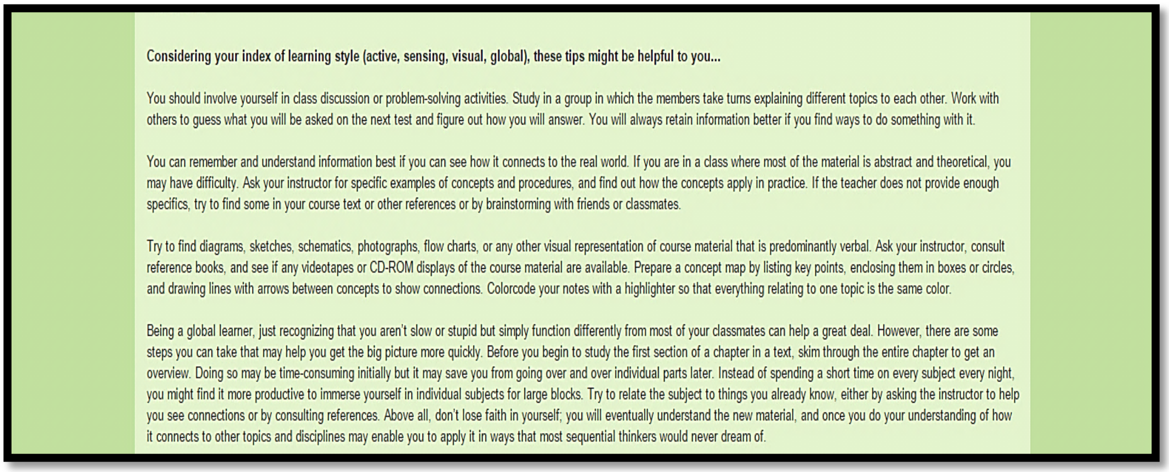

Figure 15: Recommendation page

SYSTEM EVALUATION

To evaluate the accuracy and reliability of the system in predicting the performance of the students, the researcher determined the degree of agreement (kappa value) of the actual grade from the historical data versus the prediction generated by the developed system. Approximately twenty percent (20%) or four hundred seventy (470) students, also called as instances, were randomly selected from the historical data set. These instances were individually entered into the system and let the system generate its prediction. The result of prediction is shown in the contingency table given in Table 7.

Table 7: Contingency table for computing system performance using kappa value

		PREDICTED		
		PASSED	FAILED	
ACTUAL	PASSED	386	18	404
	FAILED	16	50	66
	TOTAL	402	68	470

The contingency table shows that out of 470 instances, three hundred eighty-six (386) were correctly predicted as true positive – meaning, they actually passed the subject and were predicated as “passed” by the system. Fifty (50) were correctly predicted as false negative – those who actuallyfailed and were predicted as “failed” by the system. A total of thirty-four (34) instances were misclassified or predicted incorrectly. There were sixteen (16) who actually failed but were predicted as “passed” (known as false positive) and eighteen (18) who actually passed but were predicted as “failed” (also called true negative). Using the Cohen’s kappa formula given in Equation No. 1, the kappa value of the developed system is computed as 0.7041 which is interpreted as “substantial” considering the interpretation of the magnitude of kappa value in Table 2. This means that the reliability of the system in making predictions whether the student will pass or fail in Java programming is already acceptable.

From the contingency table, the accuracy of correct prediction can be computed to as the sum of correctly classified instances over the total instances, using Equation No. 4, that is:

$$\boldsymbol{Percentage\ correct} = \frac{\mathbf{386 + 50}}{\mathbf{470}}$$

$$\boldsymbol{Percentage\ correct} = \mathbf{93\%}$$

DISCUSSION

This study has successfully integrated educational data mining technique into a traditional LMS. Functionalities for prediction and course-content recommendation were incorporated in addition to the standard features of LMS.

The first was the attribute selection or process of identifying which variables or factors in the learner's profile are significant contributors in predicting performance in Java programming. Initially, eight variables were identified as possible predictors of performance; these are age, gender, course, section, schedule, grade in programming 1, grade in programming 2, and grade in programming 3. It was found that "course" is insignificant which means that there is no performance pattern associated to the course taken by the student, which implies that students taking up computer science, information technology, and computer technology all have the same probability of passing or failing in Java programming. "Section" was also found insignificant. This is somehow expected since students are just randomly assigned to specific section usually based on first-come first-serve basis and not on their academic grades.

After the significant predictors were determined, these predictors were used as parameters for creating a model or ruleset that was used for predicting students' performance. Classification tree algorithms, such as decision tree classifiers, Bayes, and rule classifiers were used because they work best in predicting categorical values (e.g., "passed" or "failed"). They are also good in handling missing data and are easy to interpret. The algorithm generated by J48 was chosen to be integrated in the system because J48 received the highest accuracy and kappa value. In predicting learners' performance, an accuracy level of at least seventy percent is already acceptable; however, in some other critical applications (i.e., in the field of medicine), the desired accuracy level is as high as one hundred percent if possible.

The ruleset generated by J48 was translated into source code, which is necessary for developing a prediction module to be integrated into the LMS. From the rulesets, it can be concluded that in predicting whether the student will pass or fail, the model is starting its evaluation from the grade in programming 2, meaning, that variable is the predictor with the greatest weight in determining the student's performance. Programming 2 is Advanced C language which is closely related or similar to Java in terms of syntax, operators, and statement structures. If the grade is less than or equal 2.25 (1.0 to 2.25), the class schedule is either in the morning or in the afternoon, and the grade in programming 3 is less than 1.75, the student will be predicted as "passed". It is interesting to note that even if the grade in programming 2 is less than or equal 2.25, but the schedule is in the evening, the grade in programming 3 should be 1.0 to 1.5 in order to pass. This means that time when the students are taking the subject is also a factor in performance. It can be interpreted that students in the morning and afternoon session are learning better than those in evening session. Another remarkable thing in the ruleset is that it has a point where, after checking all the grades and considering the schedules and age, the decision on predicting performance will be based on gender – that is, if the gender is "female", predictions is "passed", otherwise it will be predicted as "failed". This means that based on the historical data, on that given condition, more females received passing grades than males, thus, J48 is giving more probability of passing to females than males.

Lots of rules were generated by J48, and these were all translated into a computer algorithm. Afterwards, it was embedded in the assessment module of the LMS. All of the variables found significant during attribute selection process were included as mandatory fields in the design of the assessment module which is why the system can provide a prediction right after the user signs up. To further verify its prediction accuracy and validity, the researcher manually tested 470 records of students from the historical data. All necessary attributes or predictors were manually encoded into the system, and, since these are historical data, it was already known if they passed or failed the courses. After entering the predictors, the system generated its predictions. From there, comparison was made between the actual versus the predicted value to manually compute for the accuracy and kappa of the rulesets after it was translated into computer algorithm. There is an accuracy level of ninety three percent which means that the probability that the system can correctly predict or classify instances is ninety three percent, an acceptable accuracy level in determining students' achievements.

Limitations of the System

This LMS for Java programming with a prediction model and course-content recommendation module was able to meet the ideal requirements for generating predictions and recommendations. There are some limitations in resources and time constraints. Encountered limitations are as follows:

1. Intrinsic attributes of the learners are not included as factors for predicting their performance.
2. Recommendations is limited to offline course-content.
3. Impact of implementing this LMS platform in the teaching and learning process is not yet included on this phase of research.

Possible Improvements

This study has several spaces for further improvements:

1. Other attributes such as students' attendance, economic status and interests might be included, as well as possible predictors in determining the performance of students in Java programming.
2. Numerical value of students' final grade might be considered to provide a more specific and personal prediction better than the "Passed" or "Failed" remarks.
3. Combination of multiple algorithm in classifying data set is also recommended to further improve the algorithm and rule sets of prediction.

Conclusion

The existing frameworks for LMS found in recent literatures and studies can still be improved by applying the concept of educational data mining and recommendation systems. LMS features can go beyond having traditional functionalities for the management of learning materials, courses, student records, and the like. Data mining can be explored to develop feature which can classify students according to their predicted performance. Although a data mining task is a multi-stage process and may take a long time, the benefit and value that it can add to a learning management system is worth all the challenges. Integrating a prediction system, which considers students grades in previous related-subjects, is a great aid to the teachers in conducting a pre-assessment of students' possible performance in the future. A module which can recommends applicable course-content with consideration to the learners' index of learning style, adds power to the functionality of the LMS. Aside from the usual way of simply recommending learning topics, recommendations on how the learner could make the most out of the learning experience are also provided. The framework of this study is a good attempt at improving the current state of LMS technology. There are still many challenges that should be examined to come up with a perfect virtual learning system or personalized learning environment that can really satisfy all the needs and uniqueness of each learner.

References

Adair, D., & Jaeger, M. (2011). Difficulties in teaching and learning the Java programming language. In *Proceedings of the 17th International Conference on Engineering Education* (pp. 21-26).

Al-Busaidi, K., & Al-Shihi, H. (2010). Instructor's acceptance of learning management systems: A theoretical framework. *Communications of the IBIMA*, Vol. 2010, Article ID 862128. https://doi.org/10.5171/2010.862128

Ala-Mutka, K. (2004). Problems in learning and teaching programming–A literature study for developing visualizations in the Codewitz-Minerva project. *Codewitz Needs Analysis*, 1-13.

Anderson, T. (2008). *The theory and practice of online learning*. Athabasca University Press.

Archee, R. (2012). Reflections on personal learning environments: theory and practice. *Procedia-Social and Behavioral Sciences*, 55, 419-428. https://doi.org/10.1016/j.sbspro.2012.09.520

Arh, T., Matija, P., Srdjevic, B., & Srdjevic, Z. (2012). AHP Supported Evaluation of LMS Quality. *CEUR Workshop Proceedings*, *922*, 52-57.

Arrufat, M. J. G., & Sánchez, V. G. (2012). Steps to reflect on the personal learning environment. Improving the learning process? In *PLE Conference Proceedings* (Vol. 1, No. 1).

Butter, M. C., Quintana, M. G. B., & Valenzuela, E. S. (2014). Incremental prototyping model for the development of educational platforms: A process of design and quality standards. *Journal of Universal Computer Science. 20*(10), 1407-1417.

Carter, J., & Jenkins, T. (2010). The problems of teaching programming: Do they change with time? In *11th Annual Conference of the Subject Centre for Information and Computer Sciences* (p. 6).

Cass, B. (2015, July 20). *The 2015 top ten programming languages.* IEEE Spectrum. Retrieved from http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages

Clark, D., MacNish, C., & Royle, G. F. (1998, July). Java as a teaching language—Opportunities, pitfalls and solutions. In *Proceedings of the 3rd Australasian Conference on Computer Science Education* (pp. 173-179). ACM.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37-46. https://doi.org/10.1177/001316446002000104

Drachsler, H., Hummel, H., & Koper, R. (2007). *Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning.* Paper presented at the SIRTEL workshop at the EC-TEL 2007 Conference. September 17-20, 2007, Crete, Greece.

Dubowy, M. (2013, February 19*). LMS vs LCMS vs CMS…Changing one letter makes a big difference* [Web log post]. Retrieved November 22, 2014, from https://www.opensesame.com/blog/lms-vs-lcms-vs-cmschanging-one-letter-makes-big-difference

Educause Learning Initiative. (2011). *7 things you should know about LMS alternatives.* Retrieved from https://library.educause.edu/~/media/files/library/2010/7/eli7062-pdf.pdf

Fayyad, U., Piatetsky-Shapiro, G., Smith, P., &Uthurusamy, R. 1996. *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press.

Firat, M. (2016). Determining the effects of LMS learning behaviors on academic achievement in a learning analytic perspective. *Journal of Information Technology Education: Research*, *15*, 75-87. https://doi.org/10.28945/3405

Gadhavi, M., Patel, A., & Patel, C. (2013). A survey paper on e-learning based Learning Management System. *International Journal of Scientific & Engineering Research*, *4*(6), 171-177.

Gomes, A., & Mendes, A. J. (2007, September). Learning to program-difficulties and solutions. In *International Conference on Engineering Education*–ICEE (Vol. 2007).

Grove, W. M., Andreasen, N. C., McDonald-Scott, P., Keller, M. B., & Shapiro, R. W. (1981). Reliability studies of psychiatric diagnosis: Theory and practice. *Archives of General Psychiatry, 38*(4), 408-413. https://doi.org/10.1001/archpsyc.1981.01780290042004

Hicks, A., & Sinkinson, C. (2014). Critical connections: personal learning environments and information literacy. *Research in Learning Technology*, *23*(1), 21193. https://doi.org/10.3402/rlt.v23.21193

Huebner, R. A. (2013). A survey of educational data-mining research. *Research in Higher Education Journal*, *19*, 1-13.

Imran, H., Hoang, Q., Chang, T. W., & Graf, S. (2014, April). A framework to provide personalization in learning management systems through a recommender system approach. In *Asian Conference on Intelligent Information and Database Systems* (pp. 271-280). Springer International Publishing.

Itmazi, J. A., & Megías, M. G. (2008). Using recommendation systems in course management systems to recommend learning objects. *International Arab Journal of Information Technology (IAJIT)*, 5(3), 234-240.

Jenkins, T. (2002, August). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, pp. 53-58).

Kafura, D. (2000). *Object-oriented software design and construction with Java.* Prentice Hall Professional Technical Reference.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics,* 159-174. https://doi.org/10.2307/2529310

Lister, R. (2004, January). Teaching Java first: Experiments with a pigs-early pedagogy. In *Proceedings of the Sixth Australasian Conference on Computing Education*-Volume 30 (pp. 177-183). Australian Computer Society, Inc.

Learning Management System Evaluation 2011-2012 (2011, July 27). Retrieved from http://blogs.butler.edu/lms/files/2011/08/executive-summary.pdf

Lonn, S. D. (2009). *Student use of a learning management system for group projects: A case study investigating interaction, collaboration, and knowledge construction* (Doctoral dissertation, The University of Michigan).

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, *22*(3), 276-282. https://doi.org/10.11613/BM.2012.031

Mehic, N., & Hasan, Y. (2001). Challenges in teaching java technology. In *Proceedings of the 2001 Informing Science Conference*, Krakow, Poland, 365-371. https://doi.org/10.28945/2386

Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—Views of students and tutors. *Education and Information technologies*, *7*(1), 55-66. https://doi.org/10.1023/A:1015362608943

Mödritscher, F. (2010). Towards a recommender strategy for personal learning environments. *Procedia Computer Science, 1*(2), 2775-2782. https://doi.org/10.1016/j.procs.2010 .08.002

Nelson, S. (2013, August 15). The benefits of learning computer programming. [Web-log post]. Retrieved September 15, 2015 from http://www.collegeamerica.edu/blog/the-benefits-of-learning-computer-programming

Overmyer, S. P. (1991). *Revolutionary vs. evolutionary rapid prototyping: balancing software productivity and HCI design concerns.* Center of Excellence in Command, Control, Communications and Intelligence (C3I), George Mason University.

Pendergast, M. (2006). Teaching introductory programming to IS students: Java problems and pitfalls. *Journal of Information Technology Education: Research, 5*(1), 491-515. https://doi.org/10.28945/261

Pinner, R. (2014, August). What is the difference between an LMS and a VLE? *Learning Management Systems.* Retrieved November 23, 2014 from https://elearningindustry.com

Rajput, A., Aharwal, R. P., Dubey, M., Saxena, S. P., & Raghuvanshi, M. (2011). J48 and JRIP rules for e-governance data. *International Journal of Computer Science and Security (IJCSS), 5*(2), 201.

Romero, C., Ventura, S., Espejo, P. G., & Hervás, C. (2008, June). Data mining algorithms to classify students. In *Educational Data Mining* 2008.

Sikka, R., Dhankhar, A., & Rana, C. (2012). A survey paper on e-learning recommender system. *International Journal of Computer Applications*, *47*(9), 27-30. https://doi.org/10.5120/7218-0024

Smith, M. G., & Bull, L. (2003, April). Feature construction and selection using genetic programming and a genetic algorithm. In *European Conference on Genetic Programming* (pp. 229-237). Springer Berlin Heidelberg.

Sunil, L., & Saini, D. K. (2013). Design of a recommender system for web based learning. In *Lecture Notes in Engineering and Computer Science* (Vol. 1, pp. 363-368).

Shulamit, K., & Yossi, E. (2011). Development of e-learning environments combining learning skills and science and technology content for junior high school. *Procedia-Social and Behavioral Sciences*, *11*, 175-179. https://doi.org/10.1016/j.sbspro.2011.01.056

Verpoorten, D., Glahn, C., Kravcik, M., Ternier, S., & Specht, M. (2009, September). Personalisation of learning in virtual learning environments. In *European Conference on Technology Enhanced Learning* (pp. 52-66). Springer Berlin Heidelberg.

Yaghmaie, M., & Bahreininejad, A. (2011). A context-aware adaptive learning system using agents. *Expert Systems with Applications*, *38*(4), 3280-3286. https://doi.org/10.1016/j.eswa.2010.08.113

Yasar, O., & Adiguzel, T. (2010). A working successor of learning management systems: SLOODLE. *Procedia-Social and Behavioral Sciences*, *2*(2), 5682-5685. https://doi.org/10.1016/j.sbspro.2010.03.928

BIOGRAPHY

Dr. Digna S. Evale is an Associate Professor at the College of Information and Communications Technology and a lecturer at the Graduate School in Bulacan State University. She received her doctorate degree in Information Technology major in data mining from AMA University. She is also interested in pursuing research under the umbrella of business intelligence, forecasting, and optimization applied in education.