Semiconductor Optoelectronics, *Vol. 42 No. 1 (2023), 177-190*
https://bdtgd.cn/

Semiconductor Optoelectronics
ISSN: 1001-5868

A FRAMEWORK FOR A RECOMMENDATION SYSTEM TO INCREASE THE ATTAINMENT OF LEARNING OUTCOMES IN PROGRAMMING LANGUAGE COURSES

[1]Jayamary. A, [2]Thaddeus. S
[1]Assistant Professor
[2]Assistant Professor
Don Bosco College(Co-Ed), Affiliated to Thiruvalluvar University,
Yelagiri Hills, Tirupattur Dt, TN

Abstract
Learning outcomes are assertions that depict the knowledge or skills students must acquire on the completion of a course. This paper proposes a three-tier framework for a Recommendation system towards the attainment of learning outcomes in Programming language courses. The first tier consists of the students' log module intertwined with a preliminary generalized clustering module. The second tier consists of a common repository of knowledge with a specific classification Module that works on the Weight-Enhanced Iterative Machine Learning Approach. The third tier consists of a customized resource module with Learning outcomes defined and a rubric system for evaluation. At every level, a blend of machine learning algorithms is used to categorize the learners based on the scores acquired by them during the performance tests and are followed up with dynamic customized resources until they acquire the required score in the tests. The implementation of the framework has proved to be successful as it has increased the level of learning outcome obtained by 77% of the undergraduate students of computer applications.

Keywords
Learning outcomes, Recommendation systems, Learning Performance capability, Students' Learning capacity, Learning outcome attainment,

Introduction

The system of education has undergone a radical shift over the years. During the earlier times, the students were admitted to a Gurukul, where the teacher became a Guru and the student was placed under the Guru for learning and gaining knowledge. With the introduction of modern education in India, students learnt different subjects with the classroom ambience changing to enclosed walls with the students and the teacher. Today the educational system is not just confined to the four walls or textbooks but has become ubiquitous and is extended to virtual classrooms.

Whatever be the method or place of study, today's education aims at achieving the learning outcomes defined for every course. Learning outcomes are declarations that portray the knowledge or skills students must acquire on the completion of the course or program. The learning outcomes are uniform to all students, but all students are not of the same level of comprehending. To enable the students to achieve the learning outcomes, several techniques are used. Teaching techniques or methods refer to the behavior or action that the faculty and the learner exhibit in the learning exchange [1]). Every learner is unique and the learners' deeds and performance deliver insight into the way concepts and the environment are comprehended by them. Similarly every teacher is unique and defines his/her own set of techniques mastered over the years and apply it on different levels of learners [2]. Several techniques are used to teach programming language courses and thereby help students to attain the learning outcomes. The techniques focus on three domains namely Cognitive, Affective and Psychomotor as suggested by Bloom in his Taxonomy[3]. The techniques used can be broadly classified into two categories namely Tutoring systems that use Lecture or software tools and technologies that are action or task-oriented while recommendation systems use blended methods of inculcating the concepts. Several studies reveal that the Strategies adopted for attaining Cognitive skills are Lecture, one-to-one recommendations and computer-based Tutoring. For Affective skills, Role play and simulation are used while for psychomotor skills, demonstration and practice are used[4].

This article begins with the related works on Recommendation system and its framework. The second section deals with the proposed framework of the recommendation system towards the attainment of the learning outcomes in programming language courses, followed by the experiment and methodology. The fifth section deals with the result and discussion followed by conclusion

1. Related Works

Recommendation system is a software system that enables users to make decisions based on the information of items presented to them [5]. The objective of a recommender system is to support the users with the required information by providing individualized recommendations, content and services [6-7]. The Recommendation systems are categorized into three types namely collaborative, content-based or knowledge-based and hybrid. A Collaborative recommender system refers to a category of algorithms where there are diversified ways to locate identical users or items and multiple ways to calculate rating based on ratings of similar users. A knowledge-based recommender system only deals with the simple representation of knowledge. A recommender system is said to be intelligent and hybrid if it has a combination of skill sets, namely representation of knowledge, learning capabilities, and logical reasoning.

Today Recommendation systems are being widely used in all domains. The framework of recommendation systems vary depending on the domain used in. One framework proposed by Z. Constantinescu et al., interlocks recommendations based on enriched quality cases with collaborative annotations of users, in order to suggest open courses and educational resources[8]. Another framework for teaching learning process was proposed by F. Zhu [9] describes a learning based on efforts to mature in quality, taking into consideration the context variables such as the user, content, and time. The framework proposed by N. Golovin and E. Rahm [10] is based on multiple hierarchical intelligent agents for modeling of static and dynamic users, generate and alteration of learning plans, individualized recommendation, and real-time learning evaluation progress. H. Park, J. Lee presents an architectural framework of a recommendation system, that uses a middleware for the contextual mining by discovering, coalescing, deducing and learning the context information[11], so as to offer different user services in the changing environments. Another multimedia content recommendation system was developed and implemented by Y.H. Cho et al.,[12]. This system was peer-oriented based on mutual filtering, in order to achieve efficiency of peer search. The framework designed for E-commerce by B. Papasratorn, S. Sukrat, [13], uses artificial intelligence technologies and big data to design the architecture of a Consumer-to-Consumer' E-commerce recommendation system, so as to enhance efficiency in recommendations for online business. The authors of "Personalized Online Sales Using Web Usage Data Mining" [14] illustrate the use of web-based data mining tools for ascertaining the navigation patterns, in order to specify individualized product for online sales. This recommender system describes a neural network architecture supported by off-line training of user actions, for the recommendation of sets of individual navigation options by products. Another framework proposed by L.M.R. Tarouco in "Mitigating elephant flows in sdn-based ixp networks"15] uses a multi-agent client server model application to recommend activities of different characteristics pertaining to the exchange of knowledge among online learner communities.

In "Data mining in personalizing distance education courses" [16], a recommendation system is used to analyze the study records of two programming courses in a distance education curriculum of Computer Science. Techniques like the linear regression and probabilistic models, were applied to describe and predict student performance. The results indicate that a Data Mining System can help a distance education staff, even in courses with few students, to intrude in a learning process at several levels: improving exercises, scheduling the course, and identifying potential dropouts at an early phase.

The article "A general framework for intelligent recommender systems" [5] proposes a framework for an intelligent recommender system that learns, uncovers new information and determines preferences. Another architecture for an intelligent tutoring system given by Jing Qiu et al., [17], uses an autonomous agent to trace the web browsing behavior of users, thereby predicting their interests, and constructing their profiles dynamically. A similar architecture PrivBox, a decentralized reputation system was proposed by M.A. Azad, S. Bag, F. Hao [18], which takes into account the users' comments and protects the privacy and reputation of the service providers. The architecture of Recommendation systems used in healthcare define acquiring of quality information and its analysis, early prediction, prevention and detection of diseases and at the same time protecting the privacy of the patients. The authors of "Content-boosted collaborative filtering for improved recommendations" use a blend of both "collaborative filtering and content-based" methods to boost the user profiles and then propose a course[19]. Another framework for MOOC courses is recommended by the authors of "Collaborative Filtering Recommendation System: A Framework in Massive Open Online Courses"[20], where a collaborative filtering method is used to assess factors such as the history of the learners, their profiles and the ranking choices of participants and then recommends courses for learners. Another framework for E-learning proposed by the authors of "An Efficient framework for E-Learning Recommendation system using fuzzy Logic and Ontology", uses M-tree hierarchy with Ontology for semantic relationship and fuzzy logic[21]. These recommendation systems use recommendation techniques such as Collaborative filtering, Rated Recommendation System, Association rule mining and knowledge –based recommender systems[20].

.

2. Framework for Learning Outcomes of the programming language courses

In this section, we propose a framework of RS towards the attainment of learning outcomes of PL courses. The general framework of the Recommender system(figure1) consists of the following components namely

1. Students' Profile
   a. Students Log
   b. Students Profile
   c. Clustering of students using Clustering ML Algorithms
   d. Generalized Resources
2. Performance-based clustering
   a. Performance Test1
   b. Classification of learners using Classification ML Algorithms
   c. Customized resources
   d. Defining learning outcomes
3. Performance-based classification
   a. Rubrics-based Evaluation
      i. Outcome Attainment
         1. Reinforcement ML Algorithms

1. Students' Profile

1.1. Students Log

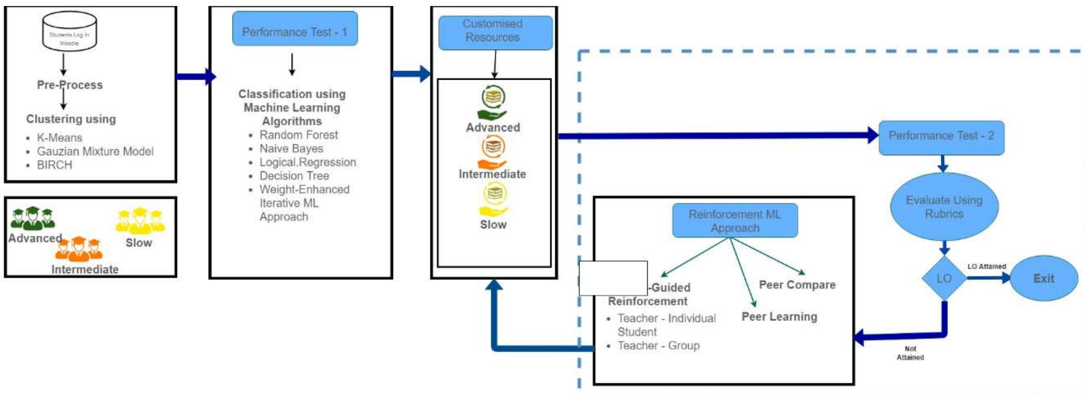
The basic data of the students such as Parents education, their job profile, scores in mathematics at secondary and higher secondary level, aptitude test score and personal support provided for the completion of their homework at home are collected and stored as Students Log.

1.2.Students' Profile

The basic data of the students is preprocessed and all unnecessary elements are removed. This information is then entered onto the database in MOODLE, thereby creating the students profile.

1.3. Clustering of students using Clustering ML Algorithms

Getting to know the level of students is helpful in modifying the pedagogy so as to ensure the learning outcomes are achieved. Felder & Silverman Index of Learning Styles suggests the method of using questions and getting answers for prediction [22] ie, firstly, learners complete the questionnaire and then a calculation technique is used to identify their learning style. The learning style and level is created based on avai.lable information [23]. Collecting information on students' profile using a questionnaire, and applying the clustering algorithms K-Means, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Gauzian Mixture Model, students are categorized into three levels namely slow learners, intermediate learners and advanced learners. Clustering algorithms are applied as these are unsupervised algorithms that extract concealed patterns in data sets and brings in proximity in data points belonging to the same cluster.[24] Taking into account the learning levels, the staff define and refine pedagogies for teaching the programming concepts.



1.4.Generalized Knowledge-base

A knowledge base is created by the faculty of every course. The base includes resources such as presentations, pdf documents, web pages, glossary etc. These are prepared well in advance and are uploaded on MOODLE. After the physical tutoring of topics by the faculty, the students go through the generalized knowledge-base. All the students irrespective of the levels of learning, go through these resources and participate in all online and offline activities such as online quiz, online lessons, glossary, workshop, webpages, group discussion, assignments etc

2. Performance-based Clustering

2.1.Performance Test I

When the students complete going through the resource base, first level performance test is conducted. Based on the scores of the students in this test, a second level categorization is made using Random Forest, Logical regression, Decision Tree and Naïve Bayes and the results of these are compared.

2.2.Specific Classification of learners using ML Algorithms

Then again, taking the same scores, a Weight-Enhanced Iterative Machine learning algorithm designed using a combination of Decision Tree, Random Forest and Logical regression, is applied to bifurcate the students into three categories as Slow, Intermediate and advanced learners. This transparent classification gives the faculty an insight on the actual level of the learners.

2.3.Customized resources

The resource-base is updated with dynamic customized resources such as simple notes for a slow learner where as a pdf for an advanced learner and video tutorial for an intermediate learner is prepared and made available. Based on the second level of classification, students are tutored using the customized resources for every topic.

2.4.Defining Learning outcomes

As recommended by the School of Information Systems, Singapore Management

University [25] and [26], the key elements of the Learning Outcomes and Competency Framework (LOCF) include three major aspects namely learning outcomes, competencies and assessments. The faculty fixes a set of learning outcomes to be achieved by the students. These outcomes are further subdivided into learning outcomes for every lesson/topic. At times multiple topics could be linked to a single learning outcome. In that case, the faculty fixes weightage for every topic. For eg: For the first LO, concepts 1, 2,3 are related. Then we assume and assign a weightage to it as Concept1 - 40, concept 2 - 40 and concept 3 = 20.

3. Performance-Based classification

3.1. Performance Test 2

After going through the customized resources, another level of performance test is conducted for all students. The tests are conducted using Quiz, programming assignment, programming exercises, debugging code and discussion forum. Every type of test checks for the acquisition of the linked competency. For eg: to test the coding competency, debugging, conversion of algorithm to program and output prediction exercises are given. Similarly for communication skills, problem documentation and discussion forums are used.

3.2.Rubrics based evaluation

Rubrics are defined for evaluating the programming competencies. For instance, to evaluate the data types in a programming language, the faculty checks the programs for the definition of data types. If students have defined variable with the correct data type with the right syntax and without errors, then they are categorized as Level 3(Advanced) learner. However if the learner just knows the data types, but does not know how to use it for defining the variables, he is categorized as Level2(Intermediate) learners. If the student has difficulty in defining or even knowing the data types, he is categorized as Level 1(Slow) learner. The students' programming assignments are evaluated using the rubrics and the scores are generated.

3.3. Outcome Attainment

If the students have reached the specific threshold scores, then it is assumed that the learning outcome of the topic has been attained. If not attained, the students go through the customized resources and participate in the test iteratively until the learning outcome is achieved. At this stage, students go through Teacher-guided reinforcement learning methods where the teacher interacts with individuals and assigns programming assignments separately. The teacher also organizes group discussion and assigns programming exercises separately for each group. The reinforcement techniques also include Peer-to-Peer learning methods.

3. Experiment - Methodology and Validation

3.1 Datasets

The datasets of students belonging to the first year undergraduate computer application were taken up. The students were divided into two groups. One group went through the framework while the other group went through regular teaching alone. No special feedback or follow-up was given to the second group. The groups were a combination of advanced, intermediate and slow learners. Table 1 shows the students' cohort selected for the experiment

Class	Boys	Girls	Total
Slow	6	6	12
Intermediate	5	4	09
Advanced	5	4	09
Total	20	10	30

3.2 Metrics used

Chi-Square test is used to assess the individual student's relationship that exists between the performance of the students and the defined framework.

3.3. Methodology

On the first day of the arrival of students to the college, a preliminary test was conducted in basic maths and logical reasoning. Along with this, the students' basic data such as Secondary and Higher-secondary Mathematics marks, their parents' education profile, area of domicile, support for learning at home etc were collected and then entered onto the MOODLE database. The data was then preprocessed to remove unnecessary elements or missed out data, after which the students profile was created and updated on the database. Based on the scores of Secondary and Higher-secondary basic mathematics and logical reasoning scores, using the unsupervised clustered algorithms K-Means, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Gaussian Mixture model, the students were clustered using into Slow, Intermediate and Advanced learners. This clustering gave the faculty a better understanding of the students. The faculty established there were more number of slow and intermediate learners when compared to the advanced learners. Hence the pedagogy was modified so as to meet the learning needs of all students.

A knowledge base pertaining to every topic was created and uploaded on MOODLE. The knowledge base consisted of different types of resources such as simplified notes, presentations, PDFs, videos, URLs etc. After the tutoring of the topics by the faculty, follow-up exercises / assignments were given and the first level performance test was conducted. Based on the scores of students, the first level of classification of students was done using Random Forest, Logic regression, Decision tree and Naïve Bayes algorithms. Each of these algorithms gave a different prediction rate. For example, the prediction rate when Random Forest was applied was 97.38, while the Decision tree was 94.5. To acquire a better prediction rate, a Weight-Enhanced Iterative ML Algorithm that blended Decision Tree, Random Forest and Logical Regression was used to classify the students again into Slow, Intermediate and Advanced learners. Each category of learners were then supported with customized resources. For example, a slow learner was given simple notes to learn the concept of variables and data types whereas an advanced learner was given a web page link to learn the concept, while an intermediate learner was provided a PPT to go through the topic. On completing the learning of every concept, a second level performance test was conducted. As suggested by the authors of [27] and [28] each topic was mapped to the course learning outcome, which in turn is mapped to the program outcome and was assigned a threshold score. The LO was also mapped to the type of assessment on every topic[28]. When the learner participated in the second level test and acquired the required threshold value, for example – if the student had acquired 5 on 10 in debugging on Arrays, then he was considered to have attained the learning outcome. However if he scored less than 5, applying reinforcement algorithm, he was redirected to the lesson on the same topic but this time, he could choose another type of resource to learn the concept. For example, an intermediate learner was provided a PDF to go through the same topic while an advanced learner was provided link to a video, while the slow learner was given one-to-one coaching with the faculty. After going through the resource on the concept again, he was allowed to take up the test once again. This process was repeated until he acquired the threshold score.

4. Result and Interpretation

4.1 Calculation of the learning outcome Attainment level

Learning performance capability of the student refers to the score a student acquires in every test on each topic. Students learning capacity (SLC) refers to the sum of all learning performance capability (LPC) in 'n' tests as shown below

SLC=$\sum_{i=1}^{n} \quad = Sum(LPC)/n$ i.e SLC = $\sum_{i=1}^{6} \quad = Sum(LPC)/n$ (6 refers to the number of tests)

To calculate the (SLC-P) $'$ = $\sum_{p=1}^{6} \quad = Sum(LPC)'/n$, where SLC-P$'$ refers to the probability value that is missing in the SLC.

To increase the SLC and reduce the (SLC-P)$'$, the students were put through the defined framework, where after every test, the students were classified into advanced, intermediate and slow learners based on the test scores. Appropriate learning resources were made available based on the learning levels of the students. After going through the resources, students' skills were tested again through various other tests and this loop continued until the student has acquired the required pass percentage.

4.2 Analysis of scores

Chi-Square test was applied to analyze the relationship between the test scores and students going through the defined framework.

Chi-Square test before

Chi-Square test was applied to the test scores before and after the students went through the framework to check if there was any dependence of the customized resources on the performance of the students.

The Hypothesis to be proved was

1. H_0 - Students' performance does not depend on the framework
2. H_1 – Students' performance is dependent on the framework

The following tables show the Chi-square test applied on the scores of every student before going through the framework

time	GENDER	CATEGORY	OBS Datatypes	OBS I/O Statements	OBS Loops/conditions	OBS Classes and objects	OBS Functions	OBS Arrays	OBS Pointers	EXPECTED VALUES ON ALL TOPICS	CHI-TEST	Average
1	M	ADV	7	5	6	6	5	6	7	8	0.677	6.000
2	M	ADV	6	6	7	5	5	5	8	8	0.609	6.000
3	F	MOD	6	5	3	3	5	6	5	6	0.744	4.714
4	F	MOD	6	4	3	5	5	6	6	6	0.868	5.000
5	M	MOD	5	4	4	6	5	5	5	6	0.920	4.857
6	F	SLOW	2	3.5	4	4	4	4	2.5	4	0.951	3.429
7	F	SLOW	3	4	1.5	4	5	5	3	4	0.861	3.643
8	M	SLOW	4	3	4	3	2	1	3	4	0.677	2.857

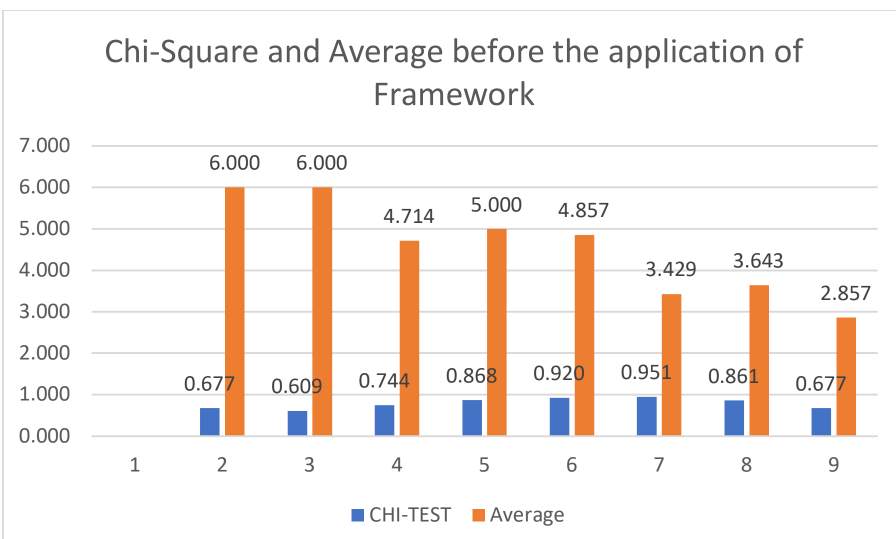
The following table shows the Chi-square test applied on the scores of every student after going through the framework

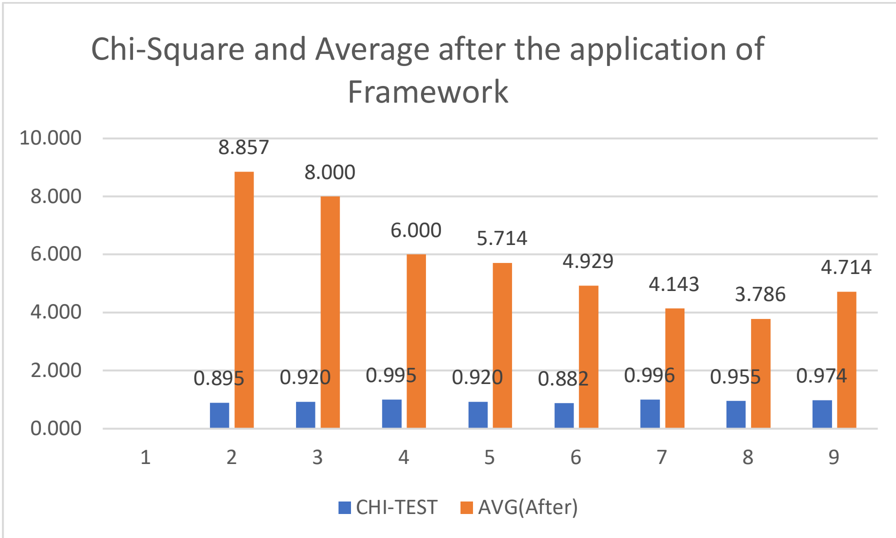
#	GENDER	CATEGORY	OBS Data types	OBS I/O Statements	OBS Loops/Conditions	OBS Classes and objects	OBS Functions	OBS Arrays	OBS Pointers	EXPECTED ON ALL TOPICS	CHI-TEST	AVG(After)
1	M	ADV	10	10	10	6	9	9	8	8	0.895	8.857
2	M	ADV	10	9	6	6	7	9	9	8	0.9	8.000

											20	
3	F	MOD	7	6	6	6	7	5	5	6	0.995	6.000
4	F	MOD	5	5	7	6	8	5	4	6	0.920	5.714
5	M	MOD	6	5	3	5	4.5	6	5	6	0.882	4.929
6	F	SLOW	4	4.5	3	4	4.5	5	4	4	0.996	4.143
7	F	SLOW	5	4	2	4	3	4	4.5	4	0.955	3.786
8	M	SLOW	5	5	5	4	5	4	5	4	0.974	4.714

It was observed that the students' Chi-square values before the application of framework was 0.677, 0.609, 0.744, 0.868, 0.920, 0.951, 0.861, 0.677 while the average score of these students were 6.000, 6.000, 4.714, 5.000, 4.857, 3.429, 3.643, 2.857. However after going through the framework the Chi-square values increased to 0.895,0.920,0.995,0.920,0.882,0.996,0.955,0.974 and average to 8.857, 8.000,6.000,5.714,0.929,4.143,.3.786,4.714 respectively.

The experiment showed 77% of students making an improvement in the attainment of learning outcome, while 23% of the students showed very slight improvement.





The experiment showed that in spite of teaching the proper structure of the programs, around 45% of the students still made a mistake in writing the basic structure ie including the header file, writing the main function and return statements. Probing a little into that, the faculty realized that it was because of the learners' usage of online compilers. Since the classes were initially conducted online, the learners were encouraged to use online compilers for executing the programs. The online compilers had the basic structure given by default and all that the students had to do was to write the inline coding. So the students had never learnt the structure until they physically attended the lab session. Again the faculty had to teach and insist on writing the full program in the observation notebook before getting into the lab. Another common error made by the students was in syntax. The learners often erred while ending the statements with a semicolon. They either forgot to end it with a semicolon or ended even the conditions and loops with a semicolon. Another common error that learners made was in the initialization or exiting value in the loops. For example, in a program of finding the sum of n numbers, the initial value of 'i' in 'for loop' was confused or mixed up with the number of iterations or 'n' value. It was realized that students had difficulty in these as there were no physical lab sessions during the Covid'19 season.

Once the learners started attending the classes physically, they were supported by the faculty to write programs using proper structure and at least 7 to 10 programs were executed on every topic. This eventually enabled the students to remember the syntax and structure and also helped them to move from a lower level to the next level of learning.

Conclusion and future work

The authors implemented the framework in a rural college, where the majority 80% of the students are first generation learners and have completed their higher secondary education in vernacular language. With the early prediction of slow learners, the faculty found it easy to know the level of learners and thereby help them with the needed resources for learning. The staff were also able to change their pedagogy based on the need of the hour, which in turn has supported in ensuring that the learning outcomes are attained by all students and thus learn the programming language better and move to the next level **sooner**. The authors are still working on stacking the supervised machine learning algorithms to gain better accuracy and reduce time in prediction (Prediction time).

References

1. Norland & Heimlich J. E., 2002, “Teaching Style: Where Are We Now?. New Directions for Adult and Continuing Education”, Volume 93, Pages 17 to 25.
2. Ruby Jr. R. & Ladd P, 1999, “Learning Style and Adjustment Issues of International Students. Journal of Education for Business”, volume 74, Issue no. 6, pages 363-367
3. https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/ (Data on the site as on 16/12/2021
4. Ghazali Mustapha, Mohd Mohayidin Ghazali, Suandi Turiman, & Mohd. Konting Majid, Kamaruddin Norfaryanti, Nor Azirawani Man, Adam Azura & Norziah Siti Abdullah, 2009, “Implementation of Outcome-Based Education in University Putra Malaysia: A Focus on Students’ Learning Outcomes”, International Education Studies – Volume 1, issue 4, DOI: 10.5539/ies.v1n4p147
5. Aguilar J., Valdiviezo-Díaz P., G. Riofrio, 2017, “A general framework for intelligent recommender systems”, Application of Computer Information, Volume 13, issue 2, Pages147 to 160.
6. Tuzhilin A. and Adomavicius G, 2005, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," in IEEE Transactions on Knowledge and Data Engineering, volume 17, issue no. 6, pages 734 to 749, doi: 10.1109/TKDE.2005.99.
7. H. Ye, (2011), “A Personalized Collaborative Filtering Recommendation Using Association Rules Mining and Self-Organizing Map”, 2011, Journal of Software, volume 6, issue 4, pages 732 to 739, DOI: 10.4304/jsw.6.4.732-739
8. Z. Constantinescu, M. Vladoiu, G. Moise, Qorect, 2013, “A case-based framework for quality-based recommending open courseware and open educational resources,” International Conference on Computational Collective Intelligence, Springer, pages 681 to 690.
9. A.W.P. Fok, F. Zhu, H.H.S. Ip, J. Cao, “PeRES: A personalized recommendation education system based on multi-agents and SCORM”, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume no 4823, pages 31 to 42
10. N. Golovin, E. Rahm, 2004, “Reinforcement learning architecture for web recommendations”, International Conference on Information Technology, Coding and Computing, volume 1, pages. 398 to 402.
11. H. Park, J. Lee, 2005, “A framework of context-awareness for ubiquitous computing middlewares, Proceedings – Fourth Annual ACIS International Conference on Computer and Information Science”, volume 2005, pages 369 to 374.
12. J.K. Kim, H.K. Kim, Y.H. Cho, 2008, “A user-oriented contents recommendation system
in peer-to-peer architecture, Expert Systems Applications, Volume 34, issue , pages 300 to 312.
13. B. Papasratorn, S. Sukrat, 2018, “An architectural framework for developing a recommendation system to enhance vendors’ capability in C2C social commerce, Social Network Analysis and Mining, Volume 8, Issue 1

14. Xuejun & Edwards, Zhang J. & Harding, Jenny, 2007, “Personalised Online Sales Using Web Usage Data Mining”, Computers in Industry, Volume 58, pages 772 to 782. DOI: 10.1016/j.compind.2007.02.004.
15. R.P. Esteves, L.A.D. Knob, L.Z. Granville, and L.M.R. Tarouco, 2017, “Mitigating elephant flows in sdn-based ixp networks”, IEEE Symposium on Computers and Communications (ISCC), pages 1352 to 1359
16. Sutinen, E., & Toivonen, Hamalainen, W., Suhonen, J., 2004, “Data mining in personalizing distance education courses”, World conference on open learning and distance education, Hong Kong , pages 1 to 11.
17. Xiao-Fang Wang, Min Xu, Yu-Hui Qiu, Jing Qiu, 2002, “Using autonomous agents to make recommendation in a modeling market”, Proceedings, International Conference on Machine Learning and Cybernetics, volume 3, pages 1322 to 1325.
18. M.A. Azad, S. Bag, F. Hao, Privbox: Verifiable decentralized reputation system for online marketplaces, Future Generation Computer Systems, Volume 89, pages 44 to 57
19. Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In AAAI/IAAI (pp. 187-192).
20. Onah, D. F. O., and J. E. Sinclair. "Collaborative filtering recommendation system: a framework in massive open online courses." *INTED2015 Proceedings* (2015): 1249-1257.
21. Tambe, Shubhangi S., and Ganesh V. Kadam. "An Efficient framework for E-Learning Recommendation system using fuzzy Logic and Ontology." *International Research Journal of Engineering and Technology (IRJET)* 3.6 (2016): 2062-2067
22. Felder R.M., and L.K.Silverman, 1988, “Learning and Teaching Styles in Engineering Education”, Engr.Education, volume 78, issue 7, pages 674 to 681
23. Warattha Sukpongthai, Worapat Paireekreng, Duangjai Jitkongchuen, 2016,"Improving Soft Computing Performance with Ant Colony Optimization for Multiclass Classification: The Application for Learning Style Classification", International Conference on Intelligent Systems, Modeling and Simulation, Volume 7, pages 101 to 105
24. A Detailed Study of Clustering Algorithms Kamalpreet Bindra1 , Anuranjan Mishra2, 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Sep. 20-22, 2017, AIIT, Amity University Uttar Pradesh, Noida, India
25. I Baumgartner and Shankararaman Venky, 2013, “Actively linking learning outcomes and competencies to course design and delivery: experiences from an undergraduate Information Systems program in Singapore”, Proceedings of IEEE Global Engineering Education Conference Berlin, Germany, pgs- 238-246, DOI: 10.1109/EduCon.2013.6530112.
26. Venky Shankararaman and Ducrot J, 2015, “Leveraging Competency Framework to Improve Teaching and Learning: A Methodological Approach”, Journal of Education and Information Technologies. Springer, Volume. 20, pages 1-29
27. Design of Student and Course Learning Outcomes Measurement Hans Kristianto1,

Susiana Prasetyo2, Ratna Frida Susanti3, Mariskha Tri Adithia4, JPI, Vol. 10 No. 1, March 2021 p-ISSN: 2303-288X, e-ISSN: 2541-7207 DOI: 10.23887/jpi-undiksha.v10i1.29061

[28] https://www.myklassroom.com/blog/sample-calculation-to-implement-outcome-based-education-model-at-your-institution/