American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-10, Issue-11, pp-01-09
www.ajer.org
Research Paper Open Access

Students' Group Formation Using K-Means Clustering in Combination with a Heterogeneous Grouping Algorithm.

Maudlyn I. Victor-Ikoh[1], Ogunmodimu Dupe Catherine[2]
[1]*Department of Computer Science and informatics, Federal University Otuoke, Otuoke, Nigeria, maudiraq@gmail.com*
[2]*Department of Computer Science and informatics, Federal University Otuoke, Otuoke, Nigeria, dupefatoba@yahoo.com*

ABSTRACT
Many students consider programming to be difficult. This view impedes students' ability to study programming courses and results in a lopsided class, with a small fraction of the class performing well and a larger percentage trailing behind. Student group formation, for collaborative and peer-peer learning, is a vital tool used to improve the process of teaching and learning in a way that is most beneficial to students taking programming courses, hence the mechanism used to define group composition is crucial.Although there have been numerous studies and mechanisms on grouping students for collaborative learning, there has been very little study on grouping students for programming courses. Hence,in this study, we demonstrate how K-Means clustering algorithm can be used to cluster students based on their pre-requisiteknowledge level required to learn a programming course. A complementaryheterogeneous grouping algorithm implemented using the functional programming paradigm in Python, is applied to the resulting clusters to form heterogeneous groups with members of varying knowledge levels.Evolutionary prototyping system design methodology was deployed, with each version incrementally refined to improve the system's grouping performance.After implementation on a programming class of 63 students, the system achieved a satisfactory level of heterogeneity in student grouping. Also,student learning outcome generally improved; collaborative learning and peer-peer learning was achieved for80.95% and 85.75% of students respectively
KEYWORDS: Heterogeneous Student Grouping, K-Means, collaborative learning

---

Date of Submission: 26-10-2021 Date of acceptance: 10-11-2021

I. INTRODUCTION

Student group formation is a vital tool used to improve the process of teaching and learning in a way that is most beneficial to students; it is a necessary activity that improves collaborative, peer-peer learning. In computer programming class environments, peer-peer learning is a key approach to teaching and learning how to program, hence, the mechanism used to define group composition is crucial. Student group composition can either be homogenous or heterogeneous based on defined criteria. The number of students per group is also fundamental.

Some taught programming courses require pre-requisite knowledge of a programming language for that course, while other programming courses do not require prerequisite knowledge. However,basic knowledge in the programming language used to teacha programming course will serve as an added advantage for a student. That student has the potential to comprehend new concepts quicker and will be of benefit to other students if placed in the same group withstudents with little or no mastery level of programming.

Programming courses are perceived as difficult for a lot of students. This perception hinders students' learning ability of programming courses and creates a lopsided class; with a few percentage of the class performing a lot better, and a greater percentage of the class lagging. Studies have shown that collaborative, peer-peer learning improves student learning outcomes [1].

There have been a lot of approaches embarked upon by researchers to group students for peer-peer learning, many of which are supported by machine learning algorithms in combination with intelligently self-defined grouping algorithms [11][13][15][16]. Each approach is aimed at addressing a problem in a specific domain, hence the need for this study.

1.1. Problem Statement

Although, there are sundry studies on grouping students for collaborating learning, however, a number of them focus on online and e-learning platforms. The few studies that are conducted in a classroom setting have been generic, not considering the peculiarity of grouping students for a programming course.

1.2. Aim and Objectives

This study aims at enhancing the best relevant existing model for grouping students into a heterogeneous group for collaborative, peer-peer learning and adapt it into a programming class environment. The specific objectives are to:

i) Cluster studentsof the same pre-requisite knowledge profile using K-Means clustering algorithm

ii) Definea heterogeneous student group formation algorithm using Python Programming Language that is applied to the cluster, which then assigns students of varying pre-requisite knowledge profiles to a group.

iii) Apply the heterogeneous grouping algorithm to a programming course class and examine if the system was able to group students of varying prerequisite knowledge profiles into a group.

iv) Evaluate the effect the grouping had on Students' overall learning outcomes.

1.3. Programming Teaching Techniques

Researchers have undergone reviews on new approaches to teaching Programming courses. Peer-peer Learning, pair programming, and collaborative learning approaches have emerged [3]. Peer-peer learning is the use of teaching and learning strategies in which students learn with and from each other without any direct intervention from a teacher. Examples of peerlearning are student-led workshops, team projects, study groups, or student-to-student learning partnerships. When one student guides another through a task or subject, it is referred to as peer-to-peer learning [1].Pair programmingis when two programmers work together at onecomputer.It is usually made up of small groups of two (2) or three(3) students taking an active role in solving the task. One student is the driver while the other is the navigator and both switch roles [7]. According to [6] collaborative learning is a situation in which two or more students learn or attempt to learn something jointly. Collaborative learning happens when students work together in groups to discuss ideas and solve issues. Collaborative learning motivates students to share their experiences and learn from others.

1.4. Grouping techniques

Well-structured learning groups benefit the student and fulfill the objective of grouping. The optimal group size as indicated by [4] is to be between two to six, and the composition of the learning groups should take into account different grouping criteria, such as personal interests, motivational orientations, learning successes, and sex. Different grouping criteria may influence students' learning performance; nevertheless, it is nearly impossible to compose learning groups that account for numerous grouping criteria [5]. Group formation varies from Randomly-Assigned Groups, Student-Selected Groups, Homogeneously-Assigned Groups, and Heterogeneously-Assigned Groups [2].

Random assignment is a method of randomly assigning students to different groups. Every student has an equal probability of being placed in a learning group due to random grouping. This straightforward technique is often used in schools and does not necessitate any prerequisites or grouping criteria. However, according to [8], randomly assignedgroups produce poor results, less positive attitudes toward the group experience, and lower group result ratings when compared to other group techniques. In student-selected groups, students choose their partners, with little to no intervention from instructors. Instructors typically announce the required number of students in a single group and then let the students construct the group accordingly [2]. Some students advocate strongly for the formation of their groups, with the claim that knowing each other means that they will be able to work well together. [8] Ascribes this to familiarity among members of self-selected groups that may make communication simpler. Also, due to compatible schedules, the high frequency of meetings for discussion gives lots of opportunities for information sharing, resulting in better performance.

Homogeneouslyand heterogeneouslyassigned groups are both initiated by the instructor using a systematic allocation approach. Homogeneous groups are constructed with the deliberate objective of forming groups in which each member is similar in ability, skills, gender,or other qualities such as academic achievement, grades, marks, and results. Heterogeneously assigned groups, on the other hand, are constructed to form balanced teams comprised of persons representing a variety of abilities, skills, and academic performance.According to Graf and Bekele [11], a reasonably heterogeneous group refers to a group where

studentscores reveal a combinationof low, average, and high studentscores. A study according to [9], high and average-achieving students, in particular, benefit from homogeneous grouping, although poor achievers performed as well whether their groups were constituted homogeneously or heterogeneously. Heterogeneous group increases the potential for peer support but has the drawback of workload sometimes resting squarely on students with better academic performance [9].

1.5. Clustering Algorithm

Clustering is the process of finding out a group of objects which have similar characteristics and assigning them to a cluster/group such that objects in the same cluster are similar in some sense [10]. Clustering maximizes similarity across elements in the same cluster while decreasing disparities between them. Clustering facilitates grouping students based on set objectives. There are several clustering algorithms availablethat cannot be discussed within the scope of this work. Howbeit, each method takes a unique approach to identify clusters in data. There is no one optimal clustering method for all datasets; controlled tests on a dataset are one way to determine which algorithm is suitable for that dataset. In the context of this study, the chosen clustering algorithm was based on three criteria according to a study on the effectiveness of the Application of Clustering algorithms in Student Grouping[12]. The criteria are: (i) the clustering algorithm must be able to take as a parameter the number of specified clusters since with grouping students, the instructor dictates the number of groups to be formed; (ii) the algorithm should be easy to understand and use by the instructor; (iii)the algorithm should be good at working with small datasets since programming class sizes are usually not large. K-Means clustering algorithm fits these criteria. Hence was selected as the clustering algorithm for this study.K-Means Clustering is an unsupervised machine learning process that divides n-observations into k- clusters, with each observation belonging to the cluster with the closest mean (cluster centers or cluster centroid), which serves as the cluster's prototype, hence can form several clusters as defined. K-Means Scales well to huge data sets and is efficient with small datasets; Implementation is relatively easy; consistency is assured and adapts well to a variety of shapes and sizes of clusters. Based on these qualities, K-Means is a popular clustering tool.

II. RELATED WORK

Christodoulopoulos et al [13] presented a web-based student group building tool that allows instructors to generate homogeneous and heterogeneous groups automatically based on three criteria. The usage of the Fuzzy C-Means method for homogenous grouping is a unique feature of this instrument. Preliminary tests on grouping 18 learners revealed that the resultant groups had a satisfactory level of homogeneity and heterogeneity. However, they could not evaluate the tool in a real context.

Maina et al [10] proposed an approach for grouping students in an online learning group task based on individual learners' collaboration competence level using Skmeans and the Expectation-Maximization (EM) clustering algorithm to first establish homogenous groups. The researchers then formed heterogeneous groupings using an intelligent grouping algorithm. They showed how it can be used in a real context by implementing it in a Learning Management System (LMS) such as Moodle using 36 students with a satisfactory outcome. However, their criteria for grouping students was not based on student knowledge level.

Paguio et al[14] study was aimed at developing a web-based system that used the K-Means Algorithm to group students' appeals in terms of retakes, make-ups, and appeals against exam results. Students' data were clustered using the K-means algorithm into three categories: retake, makeup, and exam outcome. Although the researchers produced considerable results, the k-means algorithm was used in the context of grouping students' appeals rather than for collaborative, peer-peer learning.

Freitas et al [15] presented a tool for student grouping in a classroom, which was based on their backward knowledge of a particular subject. The tracking information from students' answers in an enhanced multiple-choice question is used in the categorizing process. The researchers presented a method for identifying each student's profile on a specific subject and then grouping all of the students' similar profiles. The method identified two groups of students: those who required assistance and those who did not. This allowed teachers to tailor their classes to a specific group of students. The system also shows how much the students learned about the knowledge domain's topics. It, on the other hand, does not use a machine learning method to group students in heterogeneous groups for collaborative, peer-to-peer learning.

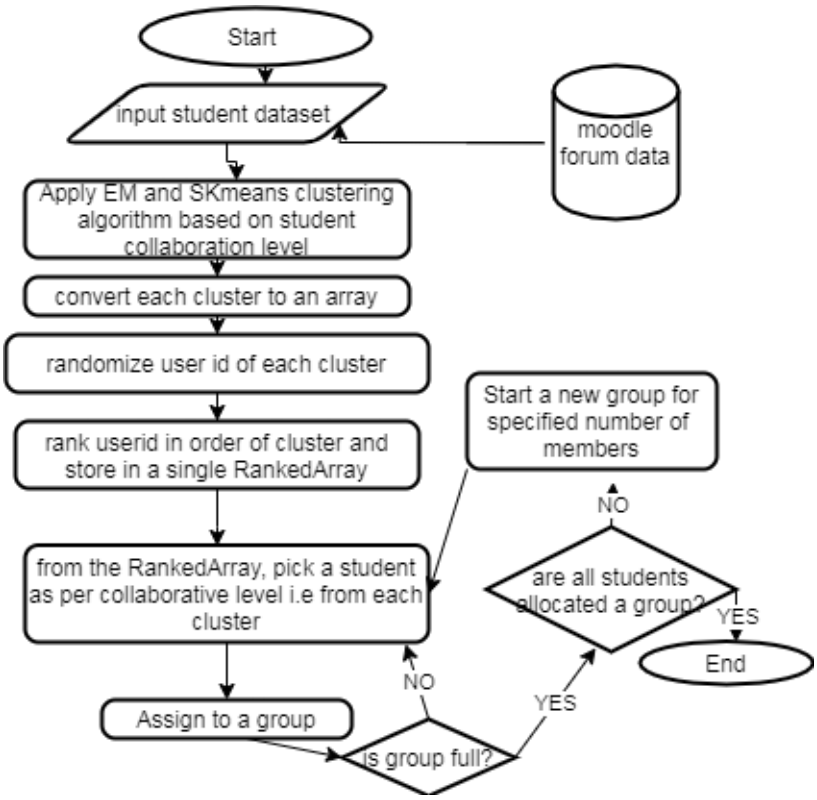
III. METHODOLOGY

The system design methodology applied to this study was the evolutionary prototype methodology. The goal for using the evolutionary prototyping process was to create an early prototype of the proposed system.Each prototype version allows for functionality to be evaluated, with later versions incrementally refined to improve the system's student grouping outcome.

3.1 Existing System Analysis

The existing system was proposed by Maina et al [10]. They proposed a novel approach for grouping students in an online learning group task based on individual learners' collaboration competence level. To create the collaboration competence levels, two machine learning algorithms for clustering namely Skmeans and Expectation-Maximization (EM) were applied to cluster data and generate clusters based on learners' collaboration competence that created three collaborative competence levels (Cluster 0, Cluster 1, and Cluster 2). They developed an intelligent grouping algorithm that utilizes these generated clusters to form heterogeneous groups.

3.1.1 Flowchart of the existing system



3.1.2 Advantages of the existing system

i. The existing systems approach has the advantage of dynamically changing the group membership based on learners' collaboration competence level.

ii. The existing system distributed the students in such a way that each group is assigned four members of different collaboration competence level clusters hence, creating a heterogeneous group.

iii. In addition, students who are in Cluster 0 (highly collaborative cluster) are assigned a mentor role in their group membership as this cluster constitutes highly collaborative members.

3.1.3 Disadvantages of the existing system

i. Since grouping is done based on student level of collaboration with online activities, this criteria does not adequately portray the knowledge level of the student and as a result, may not facilitate collaborative and peer-peer learning.

ii. The system uses more computational time by performing randomization that adds no significant value to the system.

3.2 Proposed System Analysis

The proposed system is an enhancement and adoption of the existing system proposed by Maina et al [10]. The proposed system enhances the existing systemin two ways: (i) by modifying the criteria used for creating clusters, that is, not based on individual learners' collaboration competence level, but based onstudent knowledge profile judged by prerequisite knowledge criteria of the course; (ii) The proposed systemmakes use ofa single clustering algorithmto reduce computing time;(iii) The proposed system is adoptedinto a programming classroom environment.

The proposed system is in a cycle of three phases, with each cycle releasing a prototype version that allows parameters to be recalibrated for optimal performance. At the first phase, the system creates a student knowledge profile based on pre-requisite knowledge assessedby a linear scale question in three subjects' areas S1, S2, S3 which allows students to provide numeric responses to questions asked.

Before applying K-Means clustering, because K-Means is an unsupervised machine learning that learns without any prior knowledge, clustering results may be less accurate since there are no known labels to guide the optimization process. As a result of this limitation, at the first phase of profiling student knowledge level, we provided student performance labels, which will be used as the little amount of labeled data required for semi-supervision. That was achieved by applying *if function* (fig1 and fig2)to our dataset (fig2), to rate students' performance into 5 performance levels: Poor, Okay, Good, Average, and Great. Although this label was not part of the features used as parameters for clustering, it was only a label used to check the performance of our unsupervised clustering algorithm.

```
=IF (AVERAGE (S1, S2, S3) <=2,"Poor",
IF (AVERAGE (S1, S2, S3) <=4,"Okay",
IF (AVERAGE (S1, S2, S3) <=5,"Good",
IF (AVERAGE (S1, S2, S3) <=7,"Average","Great"))))
```

Fig 1 showsthe if function formula for rating students' performance

Student Performance	Criteria	Characteristics
Poor	<=2	A student's performance is rated ***very poor*** if his average score is less than or equal to two.
Okay	<=4	A student is performance rated ***Okay*** if his average score is less than or equal to four.
Average	<=5	A student's performance is rated ***average*** if his average score is less than or equal to five.
Good	<=7	A student's performance is rated as ***good*** if his average score is less than or equal to seven.
Great	>7	A student's performance is rated as ***great*** if his average score is greater than seven.

Fig 2 shows students' performance rating criteria

StudentID	S1	S2	S3	Performance rate
111	10	9	8	Great
121	4	2	1	Okay
131	2	3	1	Poor
141	7	5	2	Good

Fig 3 Structure of Dataset

In the second phase, K-Means clustering algorithm is used to create knowledge clusters of students based on assessment from students' performance. During clustering, the elbow method was used to determine the optimal n-clusters to be formed. Optimal n-clusters fell between 3, 4, and 5 as seen in fig. and was concluded to be five (5) so that the number of clusters will be equivalent to the number of knowledge levels, and will also correspond to the maximum number of members found in a group as suggested by[4].

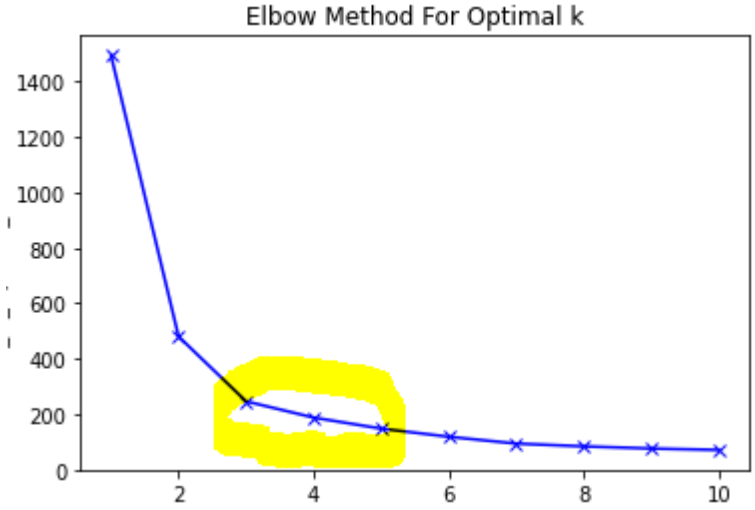


Fig 4 shows optimal n-clusters

At the end of phase2, five (5) knowledge competent clusters were created (fig5) based on student knowledge profile: Poor, Okay, Good, Average, and Great.

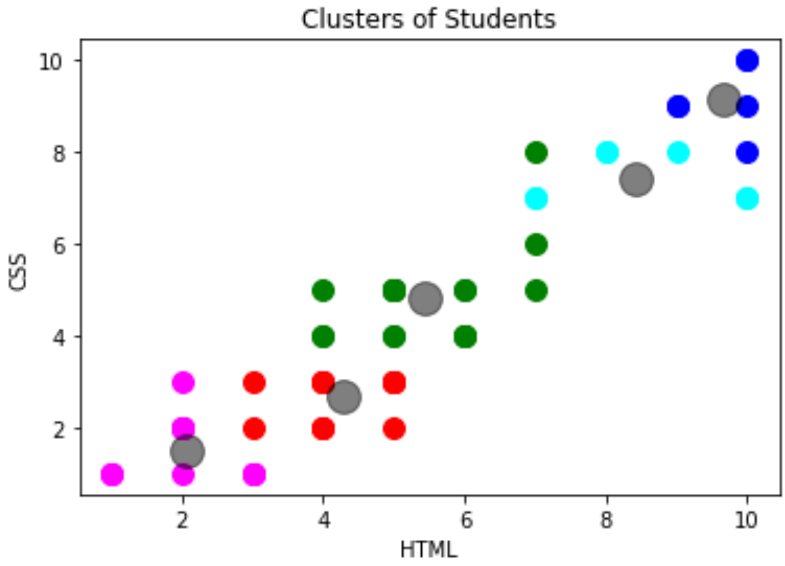


Fig 5 shows five(5) Clusters of Students

During the third phase, we developed a heterogeneous grouping algorithm as illustrated in fig6that creates a group having a maximum of five (5) members with each member selected from each of the five (5) clusters. A sufficiently heterogeneously group is one in which studentscores reflect a mix of varying academic levels. 5-member group became appropriate as made ideal by [4] for the group size of programming groups. Each group has students from the five (5) knowledge profiles.

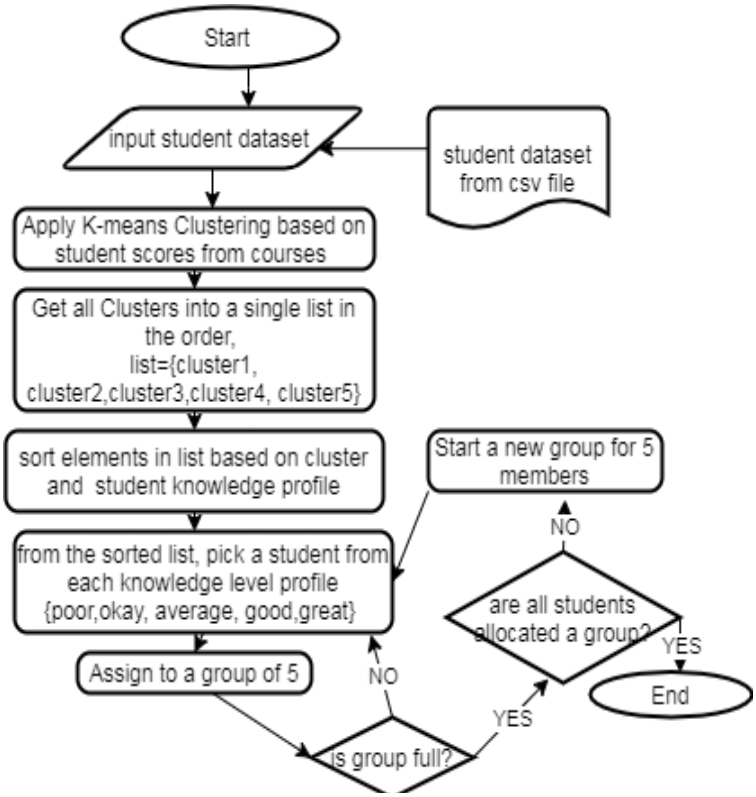


Fig 6 Heterogeneous grouping algorithm

IV. IMPLEMENTATION, RESULTS, AND DISCUSSION

To test our grouping algorithm, our proposed system was implemented on 63 students of computer science background learning web programming. Each student's knowledge profile was determined based on a linear scale question on three pre-requisite subject areas: Html, CSS, and JavaScript,and their performance were rated: Poor, Okay, Good, Average, and Great by the *if function*. K-Means was used to form 5 clusters (cluster0, cluster1, cluster2, cluster3, cluster4).

Total Number of students	**63**
Total number of clusters	5
Clusters	
Number of Student ID in cluster0= 15 **list of Student ID** - [4121 4215 4120 4165 4224 4147 453 4226 4154 4254 4184 4086 4120 4240 4192] **Their Corresponding Knowledge Profile** - ['Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay' 'Okay']	
Number of Student ID in cluster1= 6 **list of Student ID** - [4116 4230 4231 4220 4130 4231] **Their Corresponding Knowledge Profile** - ['Great' 'Great' 'Great' 'Great' 'Great' 'Great']	
Number of Student ID in cluster2= 21 **list of Student ID** - [4197 4135 4110 4133 4193 4203 4177 4131 4228 4222 4251 4210 4205 4129 4112 2000 4204 4158 4219 4110 4132]	

Their Corresponding Knowledge Profile - ['Good' 'Average' 'Okay' 'Good' 'Good' 'Okay' 'Okay' 'Good' 'Okay' 'Okay' 'Average' 'Okay' 'Good' 'Okay' 'Okay' 'Okay' 'Good' 'Okay' 'Okay' 'Okay' 'Good']

Number of Student ID in cluster3= 7
list of Student ID - [4220 4248 4178 4183 4114 4181 4217]
Their Corresponding Knowledge Profile - ['Great' 'Average' 'Great' 'Average' 'Average' 'Great' 'Average']

Number of Student ID in cluster4= 14
list of Student ID - [4198 1981 4214 4111 4252 4143 4143 4247 4200 4214 4153 4195 4156 4111]
Their Corresponding Knowledge Profile - [' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor' ' Poor']

Groups	
Total number of groups	**12**

Group 1 == [[4121,'Okay', 0], [4116,'Great', 1], [4197,'Good', 2], [4135,'Average', 2], [4198,' Poor', 4]]

Group 2 == [[4215,'Okay', 0], [4230,'Great', 1], [4133,'Good', 2], [4251,'Average', 2], [1981,'Poor', 4]]

Group 3 == [[4120,'Okay', 0], [4231,'Great', 1], [4193,'Good', 2], [4248,'Average', 3], [4214,'Poor', 4]]

Group 4 == [[4165,'Okay', 0], [4220,'Great', 1], [4131,'Good', 2], [4183,'Average', 3], [4111, 'Poor', 4]]

Group 5 == [[4224,'Okay', 0], [4130,'Great', 1], [4205,'Good', 2], [4114,'Average', 3], [4252,'Poor', 4]]

Group 6 == [[4147,'Okay', 0], [4231,'Great', 1], [4204,'Good', 2], [4217,'Average', 3], [4143,'Poor', 4]]

Group 7 == [[453,'Okay', 0], [4220,'Great', 3], [4132,'Good', 2], [4143,' Poor', 4]]

Group 8 == [[4226,'Okay', 0], [4178,'Great', 3], [4247,'Poor', 4]]

Group 9 == [[4154,'Okay', 0], [4181,'Great', 3], [4200, 2, 2, 2, 2.0, ' Poor', 4]]

Group 10 == [[4254,'Okay', 0], [4214,' Poor', 4]]

Group 11 == [[4184,'Okay', 0], [4153,'Poor', 4]]

Group 12 == [[4086,'Okay', 0], [4195,'Poor', 4]]

Fig 7 shows students identity in each cluster as well as their assigned heterogeneous groups

The output from K-Means clustering algorithm shows that clusters 0, 1, and 4 were purely homogenous in terms of student performance while clusters 2 and 3 were a blend. This blend was a result of student performances that fell within borderlines; thus clusters2 and 3comprise of students with the closest mean, although their performance ratings vary. It was for this reason, student performance label was included in the dataset as means of supervising the clusters formed, although, not part of the features used as parameters for clustering, it was only a label used to check the performance of our unsupervised clustering algorithm.

Through the grouping algorithm, heterogeneous groups consisting of varying performance levels were formed. As observed from Fig.7, the attributes of a student in any group include: {student id, knowledge profile,

cluster}where 0,1,2,3,4 represents the cluster a student was selected from.A total of 12 groups were formed with each member from each cluster. Hence our approach was successful in creating heterogeneous collaborating learning groups

Members in each group wereappraised to assess if collaboration and peer-peer learning was achieved and if it had any positive significance in improving learning outcome as a result of the heterogeneous groups. 80.95%of students affirmed that the grouping which brought about collaborative learning improved their learning of programming (fig. 8) and 85.75% asserted to peer-peer learning during the group exercise which also facilitated their learning outcome(fig. 9). A higher percentage could not be achieved because heterogeneous groups improve the possibility for peer support but have the disadvantage of putting the burden of work on students who perform better academically [9].Therefore, it can be concluded that our heterogonous grouping approach produced a significant result in improving the learning outcome of students in programming classes.

Collaboration	*Student Count*
Maybe	5 (7.93%)
No	7 (11.11%)
Yes	51 (80.95%)
Grand Total	**63**

Fig 8 showsstudents appraisal on collaboration

Peer-peer learning	*Student Count*
No	9 (14.28%)
Yes	54(85.75%)
Grand Total	**63**

Fig 9 showing students appraisal on peer-peer learning

V. CONCLUSION

This study has followed an improved approach for grouping students using K-Means clustering algorithm complementing with a heterogeneous grouping algorithm using python. To demonstrate the practicality and efficiency of the grouping approach, the system was implemented on students of computer science background. This grouping approach efficaciously places students in heterogeneous groups to significantly improve learning outcomes for students learning Programming.

Contribution to knowledge

An approach for allotting students in heterogeneous groups to improve the learning outcome of students learning to program has been developed and implemented.

Suggestions for Future Work.

Still adopting the evolutionary prototyping system design methodology, the future scope of this research work will include an integration of the three phases into a web-based system.

REFERENCES

[1]. C. Krohn and B. Sabitzer , “Peer-learning and Talents Exchange in Programming: Experiences and Challenges ,” *12th International Conference on Computer Supported Education*, 2020.

[2]. H. Nhan and T. A. Nhan, “Different Grouping Strategies for Cooperative Learning in English Majored Seniors and Juniors at Can Tho University, Vietnam ,” *Education Sciences*, vol. 9, no. 59, pp. 1–16, Mar. 2019.

[3]. N. Bubica and I. Boljat, “Strategies for Teaching Programming to Meet New Challenges: State of the Art,” *CIET*, 2014.

[4]. R. E. Slavin, “Research on cooperative learning: consensus and controversy,” *Journal of the Educational Leadership*, vol. 47, no. 4, pp. 52–54, 1989.

[5]. G.-J. Hwang, P.-Y. Yin, C.-W. Hwang, and C.-C. Tsai, “An Enhanced Genetic Approach to Composing Cooperative Learning Groups for Multiple Grouping Criteria,” *Educational Technology & Society*, vol. 11, no. 1, pp. 148–167, 2008.

[6]. C. A. Bagley and C. Chou, “Collaboration and the importance for novices in learning java computer programming,” *ACM SIGCSE Bulletin*, vol. 39, no. 3, pp. 211–215, Jun. 2007.

[7]. K. Han, E. Lee, and Y. Lee, “The effects of pair programming on achievement and motivated strategies in programming,” *J. Korea Assoc. Computer, Education.*, vol. 9, no. 6, pp. 19–28, 2006.

[8]. K. J. Chapman, M. Meuter, D. T. Toy, and L. Wright, "Can't We Pick our Own Groups? The Influence of Group Selection Method on Group Dynamics and Outcomes," *Journal of Management Education*, 2006.
[9]. J. Baer, "Grouping and achievement in cooperative learning," *College Teaching*, vol. 51, no. 4, pp. 169–175, 2003.
[10]. E. M. Maina , R. O. Oboko, and W. Waiganjo , "Using Machine Learning Techniques to Support Group Formation in an Online Collaborative Learning Environment ," *International Journal of Intelligent Systems and Applications*, vol. 9, no. 3, pp. 26–33, 2017.
[11]. S. Graf and R. Bekele, ". Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization," *In Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*, pp. 217–226, 2006.
[12]. C. Xu and L. Zheng, "Effectiveness Analysis of The Application of Clustering in Student Grouping," *Proceedings of the 2013 the International Conference on Education Technology and Information System (ICETIS 2013)*, 2013.
[13]. C. Christodoulopoulos and K. A. Papanikolaou, "A Group Formation Tool in a E-Learning Context ," *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, vol. 2, Oct. 2007.
[14]. M. A. C. Paguio, M. F. Dumlao, S. C. Ambat, A. E. Damian, and D. B. Gonzales , "Grouping of Student's Appeal System using K-Means Algorithm: A Study ," i*nternational Journal of Advanced Research in Computer Science and Software Engineering* , vol. 5, no. 9, pp. 769–774, Sep. 2015.
[15]. S. A. A. Freitas, R. C. Silva, and T. F. R. Lucena, "A tool for students' grouping in classroom," *2016 IEEE Frontiers in Education Conference (FIE)*, 2016.
[16]. S. Graf and R. Bekele, ". Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization," *In Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*, pp. 217–226, 2006.

Maudlyn I. Victor-Ikoh, et. al. "Students' Group Formation Using K-Means Clustering in Combination with a Heterogeneous Grouping Algorithm."*American Journal of Engineering Research (AJER)*, vol. 10(11), 2021, pp. 01-09.