

A Comprehensive Analysis of Student Behaviour in Open.uom.lk: A Large-scale Asynchronous Open Online Platform

Eshana Ranasinghe, Vishaka Nanayakkara, Buddhika Karunarathne, Malik Ranasinghe, K.A.B.J.W Gunarathna, Sumudu Jayathissa and Tiloka de Silva

The Centre for Open and Distance Learning, Moratuwa, Sri Lanka

enranasinghe1@gmail.com

vishaka@cse.mrt.ac.lk

buddhika@cse.mrt.ac.lk

malikr.uom@gmail.com

buddhinig@uom.lk

chathurani.jayathissa8@gmail.com

tiloka.desilva@gmail.com

Abstract: The open learning platform (open.uom.lk) of the University of Moratuwa, Sri Lanka has attracted over 180,000 registered students in just over one year of its launch. This platform offers the Trainee Full Stack Developer (TFSD) programme, enabling the participants to enter the Information Technology (IT) industry to address the much needed human resources for the growing IT industry of the country. The programme consists of six courses related to IT covering Python Programming, Web Development and Professional Practice in Software Development. The platform operates with minimal restrictions for registrants and has students from all parts of the country with some foreign students with an equitable gender distribution. The steady growth of the registration numbers shows a high level of enthusiasm from the community to explore the potential opportunities in the IT industry. While the platform is being used actively by thousands of participants and new users are registered on a daily basis, it is also observed that some of the participants have shown slow progress at different stages. This study presents analyses performed at different stages of the programme to study the student behaviour and identify the possible causes for the participants not being able to achieve steady progress. The findings of the study indicate that the participants generally find it difficult to get through the programming exercises and assignments. Correlating learning patterns of the students help understanding the overall learning strategies which can be adopted by developers of similar asynchronous learning programmes. Furthermore, the study goes on to suggest and discuss possible solutions to clear the bottlenecks identified at different stages of the programme. The subsequent analyses allow the prediction of completions by participants leading to a machine learning based model for predictive analytics.

Keywords: E-Learning, Online Asynchronous Learning, Student Behaviour, Predictive Analytics

1. Introduction

During the Covid-19 pandemic online education was widely utilised and has proven to be both effective and essential, allowing students to gain knowledge and skills in a more accessible manner. A study by Hayashi et al. (2020) found that over 90% of state and non-state higher education institutes in Sri Lanka had carried out remote learning as of June 2020 which was mostly online. During the pandemic Internet service providers in Sri Lanka provided free access to university servers until August 2020 to boost online learning exemplifying Sri Lankan society's attitude towards education and how important they think access to education is. Online education, especially asynchronous online courses can be used as an effective way to provide education to a large number of people with little need for maintenance after the course is created by using automated grading and not limiting the number of registered students at a time.

The Sri Lankan IT industry faces a severe shortage of skilled workers. While previously the industry recruited IT degree holders, the supply of graduates produced each year has been insufficient to meet its needs ("National IT-BPM Workforce Survey 2019 - ICTA."). The Centre for Open and Distance Learning (CODL) of the University of Moratuwa (UoM), in consultation with members of the IT industry determined that the people entering the IT industry needed to possess adequate coding knowledge and the correct attitude to work in the industry and not necessarily a university degree.

During 2021, the CODL in collaboration with the Faculties of Engineering and IT, created Trainee Full-Stack Developer (TFSD) course. Trainee Full-Stack Developer is an asynchronous online course which covers python programming, web development and professional practices. Its primary goal was to provide IT education to students who completed their Advanced Level examination but have yet to or were unsuccessful in enrolling in university, to increase number of people who could enter the IT industry. It also aims to increase the coding and computer skills of the Sri Lankan population at large, given that the course is open to anyone and is completely free-of-charge.

The Trainee Full-Stack Developer course is made up of six courses; two courses in Python programming, three courses in web development and one course which include lectures in professional practices and the capstone

project. When a student registers they are automatically enrolled into introductory courses in Python programming and web development. Once a student completes the first course on a particular topic they are automatically enrolled in the subsequent course. Students are only permitted to attempt the capstone project once all the courses in Python programming and web development are completed.

The rigor of the course was determined with the goal of creating skilled workers who could be employable in the IT industry. This implies that there will be students for whom the course will be difficult to complete. From the open.uom.lk usage data it was immediately evident that most students were not able to advance past the first two courses. Python for Beginners and Web Design for Beginners had a completion rate of 8% and 5% respectfully. Moreover, even six months after the capstone project module was released only 0.03% of registered users completed the course. However, given the industry-based requirement for initiating the course, there is a need to predict with a reasonable degree of accuracy how many trainees would successfully complete the programme within a certain time period as well the likelihood of a given student to complete the programme.

This paper will cover detail analysis of the students' progress through the six courses and identify where and what is causing these low completion rates. Once identified steps will be taken to improve the completion rate and the quality of the courses overall. Demographic will be gathered and analysed which in the future will be used in combination with the user data to create machine learning based algorithms to make accurate predictions of course completions in future.

2. Literature Review

In a study by Hayashi et al. (2020) of Sri Lankan university students it was found that while students overall preferred face-to-face classes and practical training, students from architecture, computer science/ Information technology, law, education and arts preferred to have 50% of their classes online rather than have 100% face-to-face classes. Computer science and Information technology courses have a distinct advantage over other fields as their practicals are performed on computers making the switch to online learning easier than for other courses which demand in-person practical training.

Despite the benefits and many advantages of online courses they have a high dropout rate especially compared to traditional in-person courses. According to studies a total of 40%-80% of students drop out of fully online classes (Smith, 2010; Jordan, 2016). According to a study by Levy (2007) students who are at an early point in their program are more likely to drop out. Students who have put in more time and effort into the course are more likely to complete the course, while those who are unprepared for how difficult the course is and are unwilling to put in more effort have a greater chance of dropping out. According to a study by Willging and Johnson (2009) students were less likely to drop out of a course if they invested more time in it however there is still a chance of dropping out at any point in the course. While there is no main cause of a student dropping out, it can vary from personal reasons, job-related reasons or reasons related to the course itself. It is important for institutions to take continuous steps to reduce the dropout rate.

While there is a lot of focus on motivation and behaviour of students, studies suggest that the greatest indicator of online course completion is their prior educational experience. A study by Emanuel (2013) found that most of the participants in MOOCs are already well educated, many having post-secondary education or already being employed. Emanuel posits a better basic education and overall greater access to technology would be required to fully meet the goals of online courses to provide education to those who would otherwise be unable to access it. In a study by Koller & Ng (2013) it was found that the demographics of the students who participated in Coursera were that over 75% had bachelors' degrees and most students were between the years of 20-39. These studies suggest online courses are being mainly used by those who already have an education but want further knowledge and qualifications in a new field or their chosen field.

Online courses are also largely self-driven and depend on the student's ability to manage their time and be dedicated to their academic pursuits. There are fewer supports than in-person classrooms and students will have to face solving complex problems alone. If students do not have academic discipline or are not used to self-learning they will likely become de-motivated and quit. (Bawa, 2016). The access to technology and internet is one of the greatest limiting factors when it comes to widespread adoption of online learning. According to the World Bank (2023) 67% of the Sri Lankan population uses or has access to the internet. This is doubled from 2017 where only 32% of the Sri Lankan population used the internet. The dramatic growth in internet usage by the population suggests an increase in the availability and adoption of internet in rural areas in the future. At the same time, statistics from the Sri Lanka Labour Force Survey suggest that only 34.4% of the Sri Lankan

population was computer literate, of which 40.3% were fluent in Sinhala, 37.5% were fluent in Tamil and 76.3% were fluent in English. This shows a correlation between English fluency and computer literacy, suggesting a lack of English fluency may be limiting who can gain computer literacy and IT education.

Completion rates (defined as the percentage of enrolled students who completed the course) vary from 0.7% to 52.1%, with a median value of 12.6% (Jordan,2015). Since their inception, enrolments on MOOCs have fallen while completion rates have increased. Completion rates vary significantly according to course length (longer courses having lower completion rates), start date (more recent courses having higher percentage completion) and assessment type (courses using auto grading only having higher completion rates).

3. Methodology

3.1 Data

Open.uom.lk is built on Modular Object-Oriented Dynamic Learning Environment (MOODLE) which is an open source Learning Management System (LMS) which records user data and records it in logs and reports. For the purpose of this study, the primary reports used were activity logs for individual activities (lecture videos, quizzes, assessments) and activity completion (overview of an entire course). The activity completion report shows the completion of activities by each student and includes information such as date and time of activity completion. Data were downloaded and analysed using Excel. The period covers from February 2022 to April 2023, and over 190,000 students.

From the activity completion report it is possible to identify the students who successfully completed each course and the entire Trainee Full stack Developer program. As such, surveys were administered on three categories of students: students who started but did not finish course 1 Python for Beginners; those who finished course 1; and those who finished the entire program. The survey included questions about demographic background such as age, sex, and education, as well as information relevant to how likely they were to complete the course, such as their prior coding experience and English language proficiency.

3.2 Identifying Bottlenecks in Course

The total number of students who are enrolled for a course, number of students who completed each lesson, and number of students who completed the entire course was determined. Completing a lesson or course automatically allowed students access to the next lesson/course, so the number of qualified students for each lesson can be presumed to be the number of students who successfully completed the prior lesson. The size of the difference between the number of students who qualified for each lesson and the number of students, who completed each lesson, is chosen as a measure of how difficult a lesson or assessment was. Specifically, the drop out was calculated as:

$$\text{DropoutRate} = \frac{(\text{Qualified} - \text{Completed})}{(\text{Qualified})} \times 100$$

High dropout rates correspond to more significant bottlenecks in the course as the course is structured such that a student cannot progress to the next lesson until they have completed the assessment of the previous lesson.

Once these bottle necks were identified remedial action was taken to encourage more students to overcome these difficult lessons.

3.3 Language Support

Steps were taken in order to make the course more accessible to those who spoke English as a second language and were not very fluent or not confident in their English. The lectures videos transcribed into English scripts. These transcripts were then translated into Sinhala and Tamil. The transcripts in all three languages were added beneath the lecture video so that they could be referenced by students to make it easier to follow the lecture video.

4. Results and Discussion

4.1 Identifying Bottlenecks in Course

Figures 1, 2 and 3 shows how the students progress through the six courses. The figures illustrate the dropout rate of students at various stages in the course and allows for the identification of bottlenecks in the course. Each data point in the bar charts corresponds to a lesson in a course. The courses are structured such that a student cannot progress to the next lesson until they have watched the lecture video and completed the assessment in the previous lesson. Lessons or assessments which are especially difficult would have a larger discrepancy between the number of students qualified and the number of students who completed the lesson.

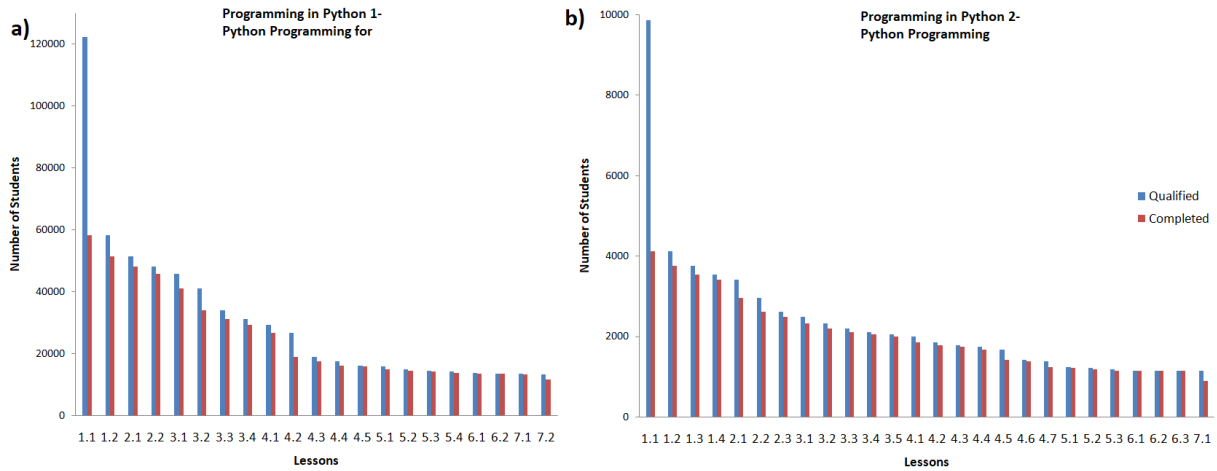


Figure 1 Number of students who qualified and completed each lesson in the Python Programming courses a) Python for Beginners, b) Python Programming

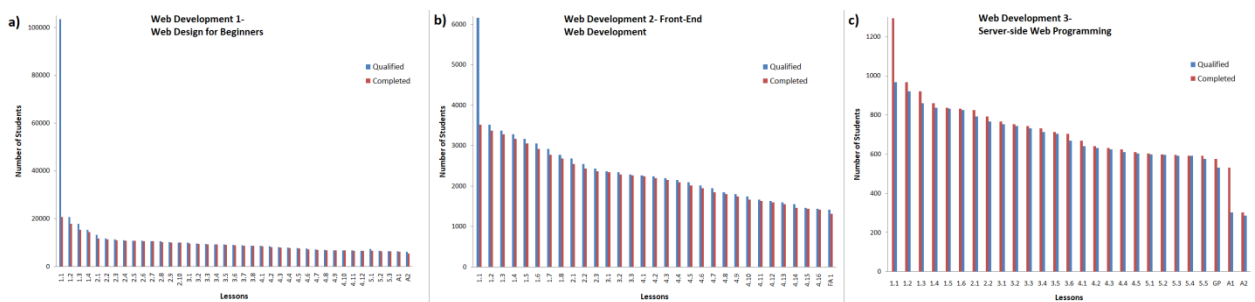


Figure 2 Number of students who qualified and completed each lesson in the Web Development Courses a) Web Design for Beginners, b) Front-End Web Development, c) Server-side Web Programming

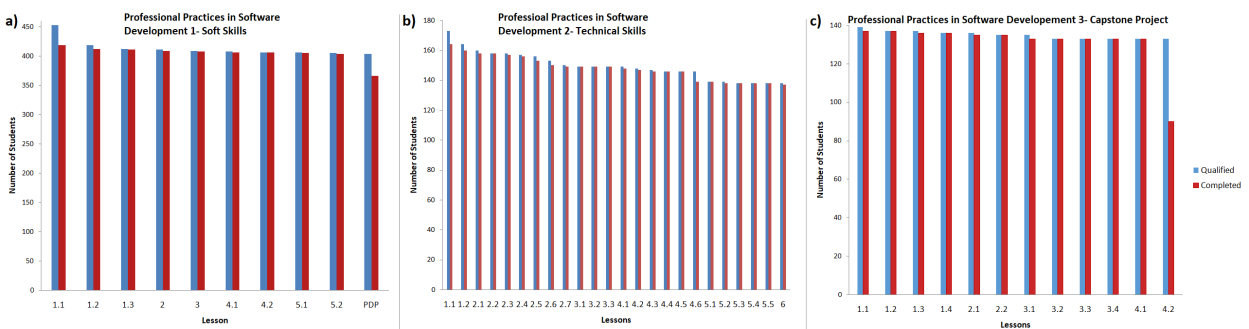


Figure 3 Number of students who qualified and completed each lesson in the Professional Practices in Software Development Courses a) Soft Skills, b) Technical Skills, and c) Capstone Project

Figure 1 illustrates the progress of students through the courses Python for Beginners and Python Programming. The graphs have similar structure with large difference between qualified and completed numbers which gradually reduces further into the course. The largest difference was in the first lesson of both courses, this is probably due to students who were enrolled in the course but never accessed it. According to the platform records, as of the 13th of April 2023, of the 122291 students enrolled in Python for Beginners 19178 students never accessed it. There is also a chance the students started lesson 1.1 but did not finish it, not because it was

difficult but because they were disinterested. Python for Beginners has a completion rate of 8% and Python programming has a completion rate of 9%.

Similar trends can be seen in Figure 2 for the graphs of Web Development Courses: Web Design for Beginners, Front-End Web Development, and Server-side Web Programming. This is likely also due to student being automatically enrolled once they completed registration or the previous course but decide not to start the course for various reasons. The pass rates of Web Design for Beginners, Front-End Web Development, and Server-side Web Programming are 5%, 21% and 22% respectively.

This behaviour, to some extent, is expected given free online courses typically have high attrition rates (Jordan, 2016) with most of the attrition occurring in the beginning of the course (Levy, 2007). However the Professional Practices in Software courses do not display these trends as students are only enrolled in the Professional Practices course after completing both of the Python Programming courses and all three Web Development courses (figure 3). The students who are enrolled in this course have already put in a considerable amount of time and effort into the programme and are likely motivated to finish it. This behaviour is consistent with studies done by Willging and Johnson’s (2009) which suggest students were less likely to drop out of a course if they invested more time in it. The pass rates of Soft Skills, Technical Skills, and the Capstone Project are 80%, 79% and 64% respectively. The high pass rates of the Soft skills and Technical Skills component of course 6 can be attributed to the assessments all being the form of multiple choice questions as well as the course material being comparatively easier to the previous courses. The Capstone project is very challenging as it contains a lot of coding and is a culmination of all the technical knowledge the student has gained over the course, so it is likely the students who make it so far are the ones who are dedicated and are motivated to finish the course as it is nearly completed.

Figures 4-6 show the dropout rates of the students of each lesson over the six courses. The dropout rates reflect the number of students who could not complete the lesson as a proportion of the students who completed the previous lesson. The graph for Python for Beginners in Figure 4 shows that aside from the initial dropout from enrolled students not participating in the course, the greatest dropout rates were at lessons 3.1 (10%), 3.2 (17%) and 4.2 (29%) which corresponds to the lessons which had assessments in the form of coding exercises.

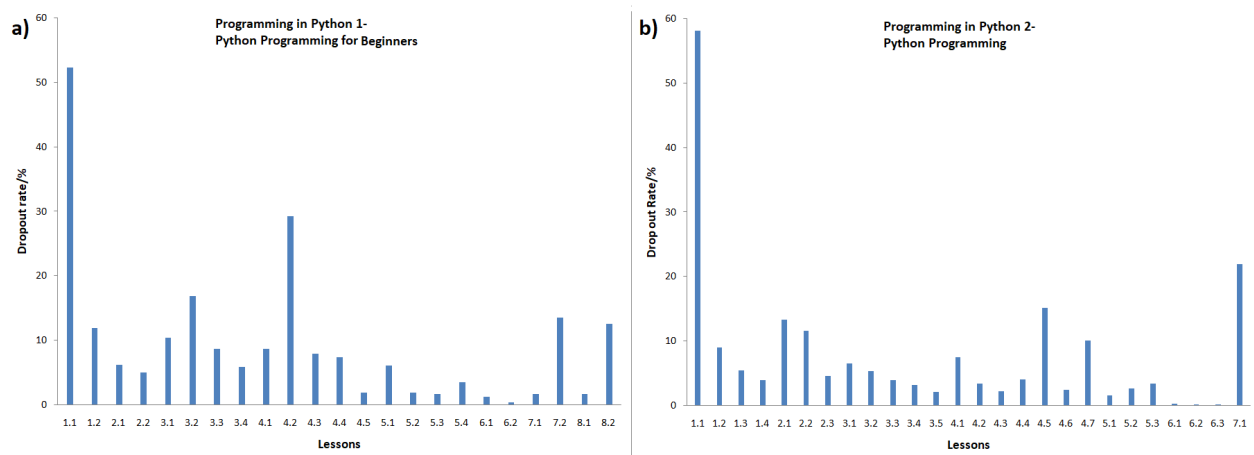


Figure 4: Dropout rate for each lesson in the Programming in Python Courses a) Python for Beginners, b) Python Programming

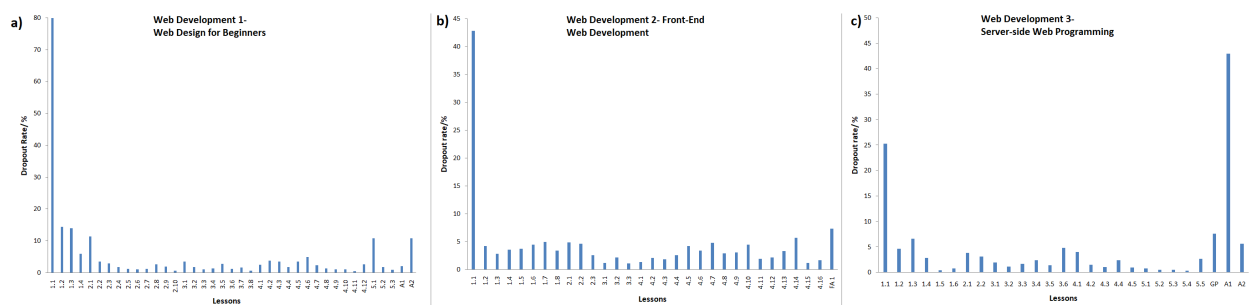


Figure 5: Dropout rate for each lesson in the Web Development Courses a) Web Design for Beginners, b) Front-End Web Development, c) Server-side Web Programming

The graph of dropout rates in Web Design for Beginners as shown in figure 5 shows that aside from the initial dropout there is no particular lesson that the students found especially difficult, the dropout out rate remaining at around 10%. Rather than trying to find a way to make it easier for students to complete this course it is more important for students to participate in it.

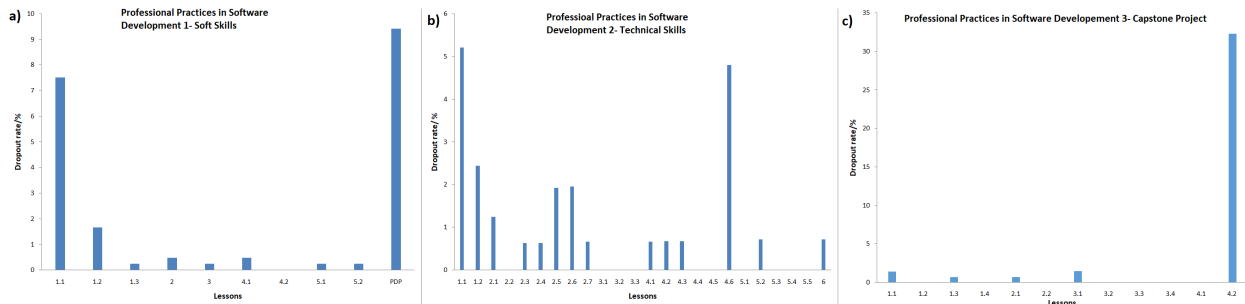


Figure 6: Dropout rate for each lesson in the Professional Practices in Software Courses a) Soft Skills, b) Technical Skills, c) Capstone Project

It was evident that the further the students progressed through the courses the lower the dropout rate was. This is likely due to the more skilled and more motivated students being filtered out through the course. The most significant attrition occurred in Python for Beginners and Web Design for Beginners, therefore moving forward these courses were focused on to reduce the dropout rates.

In Web Development for Beginners it was hypothesised that the low participation rate was due to confusion in the course name. Initially the Python for Beginners was titled course 1 and Web Design for Beginners was titled course 2. Although students could start either of them once they were successfully registered due to the naming of them as course 1 and 2 it might have led students to assume they must complete Python for Beginners before attempting Web Development for Beginners. To amend the confusion the courses were renamed as Programming in Python - 1 Python for Beginners and Web Development - 1 Web Design for Beginners.

4.2 Remedial Actions Taken to Increase Completion Rates

Initially there were no coding exercises in Python for Beginners until lesson 4.2. However after several months, the completion data of Python for Beginners was analysed and the dropout rate chart was plot as shown in Figure 7. From this it was clear there was a bottleneck in lesson 4.2 and lesson 4.3 which have a dropout rate of 22.7% and 23.9% respectively. These bottlenecks were attributed to the coding assessment in these lessons which is the first instance in the course where the students had a coding exercise as an assessment as opposed to purely multiple choice questions for assessments. It was hypothesised that introducing easier coding exercises earlier in the course will give students more coding practice and they would therefore find the coding exercises in lesson 4.2 and 4.3 easier to complete.

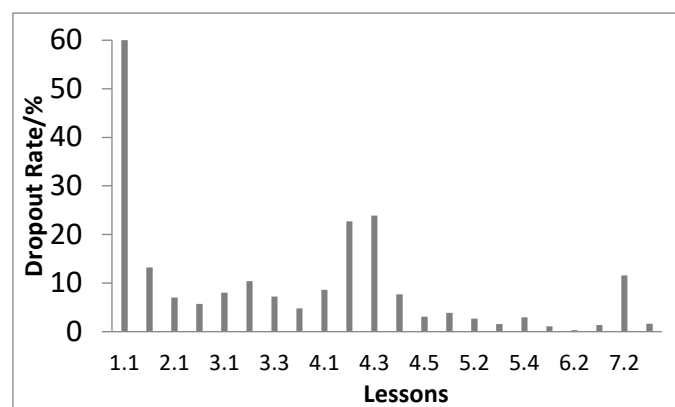


Figure 7: Dropout rate of each lesson in Python for Beginners prior to August 27

Coding exercises were added to lesson 3.1, 3.2, 3.3 and 3.4 on August 27 2023. To analyse whether the hypothesis was correct another dropout rate graph was plot comparing the dropout rates and progress through the course of students who did the new coding exercises in lesson 3 against the students who completed lesson 3 before the coding exercises were added as shown in Figure 8.

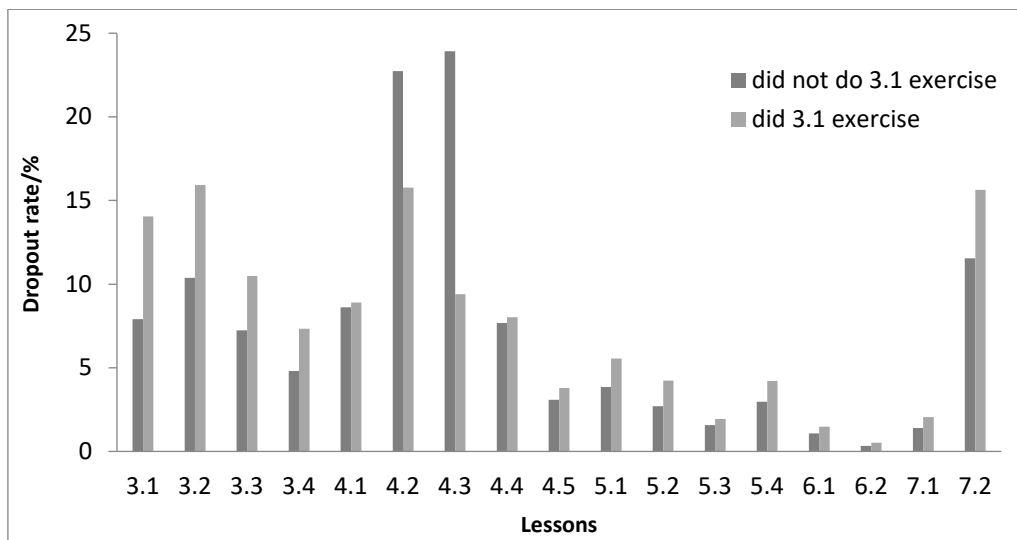


Figure 8: Dropout rate of each lesson in Python for Beginners before and after coding exercise in 3.1 was introduced

From the graph it is evident that intruding coding exercises earlier in the course did reduce the number of dropouts at lesson 4 but the dropout rates at lesson 3 as well as every other lesson increased slightly. When comparing the students who completed lesson 3.1, students who had to do the coding exercises earlier compared to those who did not have to do the coding exercise until later, they had a greater likelihood of completing the entire Python for Beginners course with the pass rate increasing from 26.4% to 28.9%. When assessing the likelihood that students who qualified for lesson 3.1 would finished the entire Python for Beginners course, the pass rate reduced from 26.5% to 24.9%. This suggests that introducing coding exercises earlier meant students got stuck at an earlier point in the course where they either could not complete the exercise or quit the course because it was difficult. In this case it maybe more reasonable to only introduce practical coding exercises later in the course once the students gains more theoretical coding knowledge.

4.3 Language Support

The additions of the translated lecture video transcripts were generally well received. However the rest of the course including lecture notes and assessments were only in English.

The idea to translate the lecture notes and the assessments were rejected after discussing with consultants from the IT industry. It would be detrimental to the students if they were too dependent on the translations as it would be essential for students to be fluent in English for them to be employed in the IT industry. Hopefully with the help of the Sinhala and Tamil transcripts students who are not fluent in English would be able to improve their English proficiency as they follow the course.

4.4 Collecting Demographic Data to be Used in Machine Learning Model

In order to collect demographic data surveys were sent to the students who had started Python for Beginners (Course 1) but had not finished the course, students who completed Python for beginners and students who completed the entire TFSD programme. The response rate of students who had not complete Python for Beginners was 1.8% (1959 of 110938), students who completed Python for Beginners had a response rate of 6.4% (634 of 9879) and students who completed the entire TFSD programme was 47.8% (64 of 134).

In addition to typical demographic data such as age, gender, and occupation, more specific questions such as their coding experience prior to participating in TFSD, their English proficiency, and their level of education, as these would have the greatest impact on their ability to complete the programme.

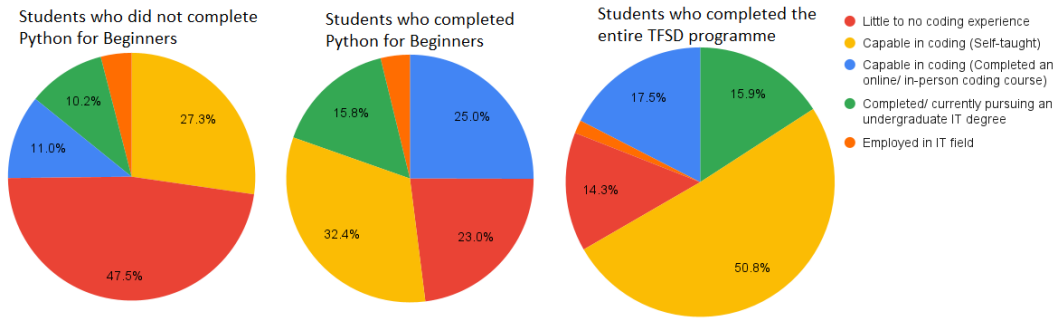


Figure 9: Students self-assessment of their coding proficiency prior to participating in the TFSD Programme

Of the students who completed the entire TFSD programme 85.7% were capable of coding to some degree prior to participating in the programme, this is greater than students who did not finish Python for Beginners and students who completed Python for Beginners of which 52.5% and 77% respectively were capable in coding to some degree prior to participating in the course. Students with prior coding experience are more likely to complete the programme of which those who are self-taught are more likely to complete the course than other students as evident in Figure 9. Only 14.3% of students who completed the TFSD programme had no coding experience prior to participating in the programme, which means while less likely it is still possible to complete the course without any prior coding knowledge. This suggests that students who have in the past taught themselves some coding are more comfortable with self-learning and have the right habits to complete TFSD.

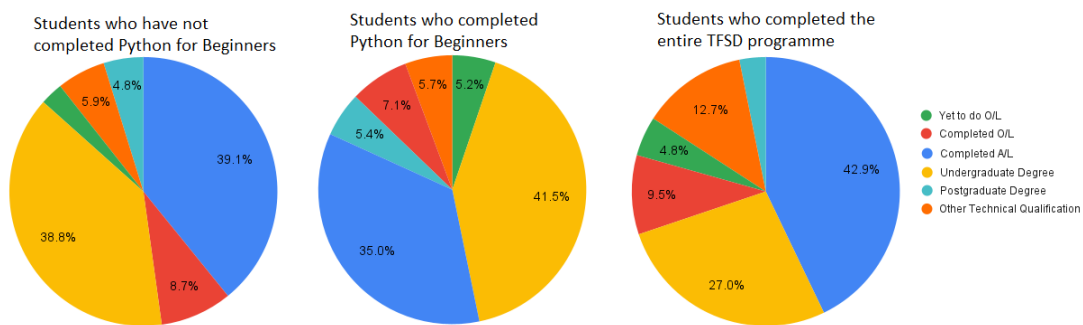


Figure 10: Students highest level of education

Figure 10 shows the highest level of education of the students who have not completed Python for Beginners, students who completed Python for Beginners and students who completed the entire TFSD programme. The largest demographics who participated in TFSD were students who have their A Level qualifications and those who already have an undergraduate degree. Although of the students who completed Python for Beginners 47% had an undergraduate degree, of those who completed the TFSD programme only 30% had undergraduate degrees. This difference may reflect these students lack time rather than interest as participants with undergraduate and postgraduate qualifications have a high degree of participation in MOOCs (Emanuel, 2013). These results show the majority of participants in the programme were school and university students who likely have the interest, ability and time to complete the programme. TFSD was created with aim to teach students who have finished their A-Levels.

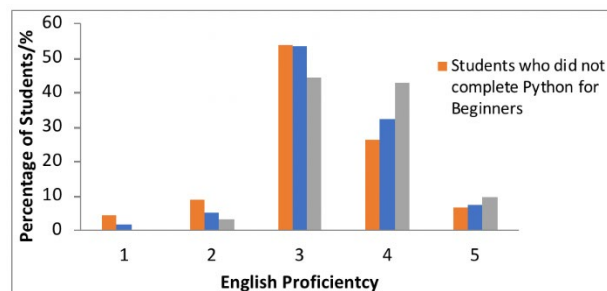


Figure 11: Students self-assessment of their English Proficiency on a scale of 1-5

The number of students was normalised by taking the percentage of the total number of students who responded to the survey. The average the students who did not complete Python for Beginners, students who completed Python for Beginners and students who completed the entire TFSD programme had an average language proficiency rating of 3.2, 3.4 and 3.6 respectfully. The students who completed the entire programme have a relatively higher proficiency in English or at least a greater confidence in their English fluency. This is their perceived language proficiency has a great impact on their completion rate. If the students' confidence in their English ability were improved it may help them complete the course rather than getting de-motivated and giving up.

5. Conclusion

The usage data from the open.uom.lk platform was analysed and successfully used to identify the bottlenecks in the courses which corresponded to the largest dropout rates. It was identified that a majority of the dropouts occurred in the first two courses and steps were taken to amend this. As the dropout rates in Python for Beginners was highest for the lessons which first introduced coding exercises it was hypothesised that by introducing easier coding exercises earlier in the course will give students more coding practice and they would therefore find the coding exercises later in the course easier therefore reducing the dropout rate. However, it is observed that the addition of these coding exercises slightly decreased the pass rate.

6. Future work

An algorithm was created to record the number of registered students and students who completed each course everyday at the same time. This data was used to plot the increase in registered users and the increase in the number of participants who complete the entire course.

Data recorded from the open.uom.lk platform was used to plot figure 12 which shows the number of students who registered on the open.uom.lk platform over time from February 2022 to April 2023 and figure 13 which the number of students who completed the Trainee Full-stack Developer Programme over time. Figure 12 shows there was an initial rapid increase in number of registrations which grows but the rate of growth decreased overtime. Figure 13 show that there is a steady increase in the number of students who complete the TFSD programme and does not necessarily correlate to the increase in registration number.

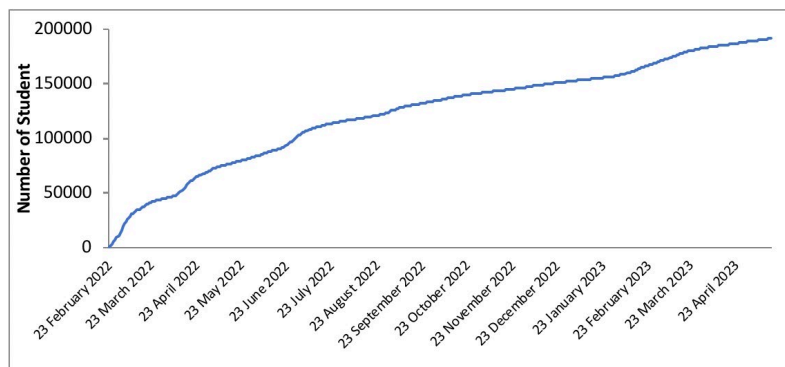


Figure 12: Number of student registered in the Trainee Full-stack Developer Programme over time

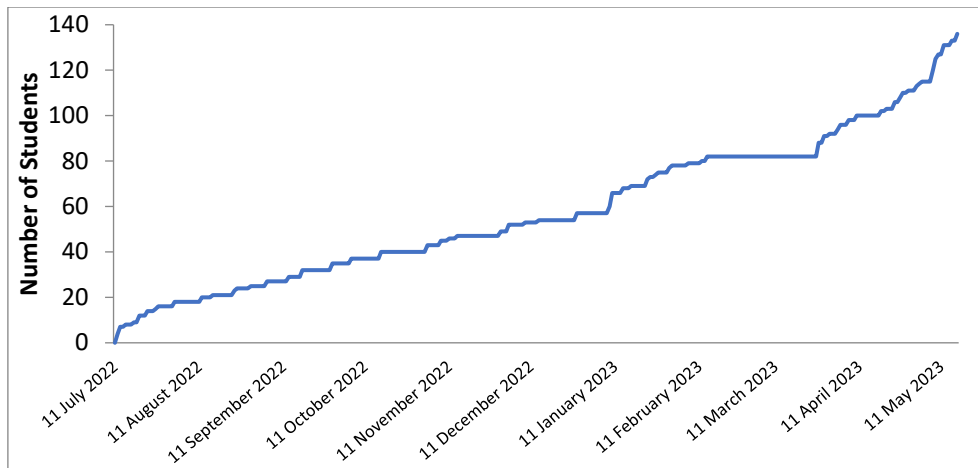


Figure 13: Number of Students who completed the Trainee Full-stack Developer Programme over time

As shown by the results we see a steady progress of completions. However, what is essential for the local IT industry is to have a prediction of the number of candidates who will be available for recruitment after completing all courses. With the usage data derived from the MOODLE platform and the demographic results of the student surveys we plan to use machine learning based algorithms to make accurate predictions of completions in future. This requires a sufficient number of students completing all 6 courses for our models to have accurate ground truth data.

References

- Bawa, P. (2016) "Retention in Online Courses: Exploring Issues and Solutions—A Literature Review" SAGE Open January-March 2016: 1–11
- Emanuel, E. J. (2013). Online education: MOOCs taken by educated few. *Nature*, 503(342).
- Hayashi, R., Garcia, M., Maddawin, A., Hewagamage, K. P. (2020) "Online Learning in Sri Lanka's Higher Education Institutions during the COVID-19 Pandemic" ADB Briefs No.151 <http://dx.doi.org/10.22617/BRF200260-2>
- Jordan, K. (2015) "Massive Open Online Course Completion Rates Revisited: Assessment, Length and Attrition" *International Review of Research in Open and Distributed Learning* Volume 16, Number 3
- Koller, D., & Ng, A. (2013). The online revolution: Education for everyone. Seminar presentation at the Said Business School, Oxford University, 28th January 2013. Retrieved from <http://www.youtube.com/watch?v=mQ-K-sOW4fU&feature=youtu.be>
- Levy, Y. (2007). Comparing dropouts and persistence in e-learning courses. *Computers & Education*, 48, 185-204
- "National IT-BPM Workforce Survey 2019 - ICTA." <https://www.icta.lk/projects/national-it-bpm-workforce-survey-2019/>. Accessed 26 Jun. 2022.
- Sittarage N. P. K. (2018) "English Language Education in Sri Lanka Link with the Learners' Motivational Factors" *Humanising Language Teaching*, August 2018 - Year 20 - Issue 4, ISSN 1755-9715
- Smith, B. (2010) "E-learning technologies: A comparative study of adult learners enrolled on blended and online campuses engaging in a virtual classroom (Doctoral dissertation)" Retrieved from ProQuest Dissertations and Theses database.
- Sri Lanka Labour Force Survey Annual Report – 2021
- Willging, P. A., & Johnson, S. D. (2009). Factors that influence students' decision to dropout of online courses. *Journal of Asynchronous Learning Networks*, 13(3), 115-127
- World Bank Data "Individuals using the Internet (% of population) - Sri Lanka" <https://data.worldbank.org/indicator/IT.NET.USER.ZS?locations=LK> (accessed May 20 2023)