DELFT UNIVERSITY OF TECHNOLOGY

MASTERS THESIS

Learning Analytics of Beginner Programming Assignments for Teachers

Author:
Erik OUDSEN
Student number:
4573706

Thesis Supervisor:
Prof. Dr. M.M. SPECHT
Second Supervisor:
Prof. Dr. E. VISSER
Daily Supervision:
MSc. Xiaoling ZHANG

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Web Information Systems Group
Software Technology

July 02, 2021

TU Delft

Declaration of Authorship

I, Erik OUDSEN, declare that this thesis titled, "Learning Analytics of Beginner Programming Assignments for Teachers" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 2021-06-24

DELFT UNIVERSITY OF TECHNOLOGY

Abstract

Electrical Engineering, Mathematics and Computer Science
Software Technology

Master of Science

Learning Analytics of Beginner Programming Assignments for Teachers

by Erik OUDSEN

Due to the increase in student numbers, the amount of time teachers have for each individual decreases. To keep the quality of the program the same, learning analytics can be utilized to assist teachers to give a better overview of problems which the students are struggling the most with. This work makes use of WebLab data of the Introduction to Python course. To provide more useful information for teachers in introductory programming assignments, this work applies learning analytics to revision history of the student submissions of this course. The revision history provides more insight into the progression of the students throughout the assignment compared to only the final solutions. With these data, this work clusters and analyzes student submissions to provide instructors with information about the types of submitted solutions, students who struggled with the assignment, and the biggest hurdles for the students. Results show that the clustering of student submissions is 90% accurate in comparison to manually clustered solutions, however the identification of students who struggled with the assignment should take student behaviour more into account. Because, student behaviour has been discussed in the focus group as being a major factor in whether data can indicate whether a student is struggling or not. Further research should be directed into expanding the proposed algorithms to more complex programming assignments and improving the way the data is visualized to the teachers.

Acknowledgements

In the process of doing my thesis I have received support from a variety of people.

First of all I would like to thank Professor Dr. M.M. Specht for providing invaluable insight on the topic and gave feedback at precisely the right moments to help me bring this thesis to a satisfactory conclusion.
I would like to acknowledge Professor Dr. E. Visser for being part of my thesis committee and making it possible to work with WebLab for this project.
A special thank you to MSc. Xiaoling Zhang for the daily supervision that she provided and the many live lessons that were shared during the project.
I would also like to thank MSc. Frank Mulder, who as the teacher of the course I used for my thesis gave necessary insight in the way the course was structured and the data could be interpreted.
I would also like to acknowledge MSc. Elmer van Chastelet for making it possible to retrieve the data from WebLab in a customized format on very short notice.
Finally, I would like to thank the participants of the focus group for their time and knowledge they provided to the project.

Contents

List of Figures

List of Tables

List of Abbreviations

AST	**A**bstract **S**yntax **T**ree
CPD	**C**ode **P**lagiarism **D**etection
CSD	**C**ode **S**imilarity **D**etection
LA	**L**earning **A**nalytics
LSH	**L**ocality-**S**ensitive **H**ashing
PLA	**P**redictive **L**earning **A**nalytics

Chapter 1

Introduction

This thesis focuses on the field of Learning Analytics (LA), specifically for teachers that provide courses in introductory programming. LA is collecting and analysing the data of students, which in turn would be used for the improvement the learning outcome of the students [1]. The main focus of previous research [2] in the field of LA has been on using LA for students, whereas the use of it has been limited for teachers. Where there has been made use of LA for teachers it is mostly for basic insights, such as passing rates of the students [3]. In this paper the use of LA for teachers is taken a step further by including the revision history of students into the analytics provided to the teacher. LA exist from the simple insights as mentioned before, to clustering similar solutions of students together, to more Predictive Learning Analytics (PLA) [4], by trying to predict which students have a higher likelihood of not passing the final examination. This thesis will try to find an answer to both of these stated use cases.

1.1 Research question

The main research question of the thesis is:
What kind of learning analytics can be leveraged to provide teachers with information about the students' progress and problems in their introductory programming course?
To formulate an answer for this question, multiple sub research questions have been formulated.

- What method is appropriate for finding the most common created solutions of students? This question aims to find an approach to group together similar submitted solutions of students. The approach found should not only allow teachers to spot students with similar solutions, but also those who are stuck with the same problem. Therefore, finding similarities among solutions and calculated the difference between these solutions and cluster them efficiently are keys in addressing this question.

- What insights can be given to the teacher about the problem areas students encountered? This question aims to find insights that can be given to teachers about the problem areas students encountered. Finding these problem areas gives an indication to the teacher what parts of the curriculum are not yet clear to the students. Use can be made of the revision history to get a better understanding where the students had the most trouble.

- Can students that had greater difficulty with the assignments be identified? This questions aims to find if students that had greater difficulty with the assignments can be identified. Identifying these students early can prevent them

from failing to complete the final examination. To identify these students use can be made of the revision history.

1.2 Overview

The report has the following structure. Firstly in Chapter 2 the state of the art is discussed regarding the research questions stated above. The most recent literature will be discussed to investigate what the current solutions are to the stated problems and how the current solution builds upon them. Chapter 3 explains the architecture of the program and the reasoning behind the decisions made in the process of creating the program. This will ensure the reproducibility of the thesis as will it contain the justifications for the decisions made in the design process. Chapter 4 contains the results of the thesis. In this chapter an evaluation is made of the program by comparing the data points of the program by manually acquired data points. Chapter 5 will contain the discussion of the results of the thesis as well as propose further research directions with regards to the solutions found in this thesis. Chapter 6 contains the conclusion of the thesis.

Chapter 2

State of the Art

In this chapter the state of the art is discussed with regards to the aforementioned research questions. Recent literature is discussed to evaluate the current solutions to the problems stated in the research questions. An analysis is made as how to progress from these solutions to create a solution that takes a different approach for the current problem. In the first section related work is discussed compared to this work. In the second section work is discussed with respect to the first sub research question. In the third section work is discussed with respect to the third sub research question. The work related to the second research question is discussed in both in the second and third section.

2.1 Related work

There has been research on the topic of LA in the past, In this section, previous work will be investigated and compared to demonstrate the rationalization of the approach used in this work.
The first paper is by Glassman et al.[5], which looks into the clustering of student solutions for a MOOC on Python programming. It makes use of static and dynamic analysis to determine clusters which teachers can specify how they want the clusters generated. This to give the teacher feedback on the most commonly created solutions, as to provide the feedback that will affect the most students.
The second paper is by Fu et al. [6], which focuses on giving teachers real-time feedback on beginner C programming assignments. It focuses on the errors that students make while completing the assignment to give feedback to the teacher on what students are struggling with the most instantly.
The third paper is by Ihantola et al. [7], which is a literature review on LA in programming. It states that most of the LA systems in place at the time of writing are based on simplistic metric analysis.
The fourth paper is by Worsley et al. [8], which tries to predict grades of the students based on their assignments using Artificial Intelligence. It states that simple clustering and similarity-finding techniques reveal some patterns of students. The more complex patterns were only revealed using computationally expensive Artificial Intelligence algorithms, which would not be viable to use everyday by teachers.

Each of these papers have their own focal point, the first and the fourth paper both look at clustering solutions, but they use different techniques for accomplishing this. The second paper is more focused on giving immediate feedback about the assignments in progress compared to the other papers which try to give information in hindsight. Where this work differs most comparing to the aforementioned papers is the use of the revision history of the students. The second paper comes closest by giving real-time information, thus making use of the latest solution of the student.

However it does not look at the full revision history afterwards to give additional insights to the teacher.

2.2 Finding similar student solutions

In this section, the literature regarding finding similar student solutions is discussed. This ranges from finding code similarities, to distance metrics, and clustering techniques. Each part will be separately discussed.

2.2.1 Code similarity

The first question that needs to be asked is what level of code similarity is required to find similar solutions of students. The definition of what makes a solution different from others can affect the classification of the solutions. In this case, would two different solutions that follow the same structure but use different variable names be classified as different kind of solutions? This is not what constitutes as different kind of solutions in this problem instance. Therefore, a different kind of similarity between solutions needs to be found; a more structurally similar solution. This is where Abstract Syntax Trees (AST) [9] could be very beneficial. Another way would be to compile the student solutions to bytecode and compare them on that level. This does not mean that no attention should be directed towards general similarity algorithms as these are often more developed due to their wider application in mainly plagiarism detection. If the two solutions are exactly the same, this would also mean that they are structurally the same, thus general similarity detection could still be very beneficial.

First, the more general similarity algorithms. The most basic version of a similarity algorithm is the 'diff' function [10]. This function looks at the character based differences in two files and returns the differences between them. This functionality is currently implemented and available in most Linux machines. A more complex example is the paper by Xu et al. [11] in which they used a neural network to compare code on a binary level, this also works across languages. These examples are however both aimed at detecting plagiarism in code on syntactically. Besides that this research is not interested in looking into plagiarism detection in its entirety, a more in depth analysis of the code is required than matching characters or binary one to one as the solutions can also be computationally similar. A paper that makes a first step at approaching the problem that is trying to be solved with this research is a by Shuai Wang and Dinghao Wu [12]. Although the work still focuses on finding plagiarism between source code files, the work does brings to light a solution to the problem of different variable names for the same code functionality, fuzzing. By providing this obfuscation into the source files, the naming conventions are no longer an issue with regards to differences in code and the sole focus is on the differences in code functionality, which is of interest in this research.

After a look at the general similarity algorithms, the following content will focus more on algorithms on structural similarity. There are two papers that stand out in this regard which are Durić et al. [13] and Huang et al. [14]. They both make use of the structure within the code to detect similarities. Both have struck a balance between lexical and structural differences between the source code. The main difference is the purpose between the two tools, Durić et al. [13] is specifically looking

for plagiarism, whereas Huang et al. [14] is just looking at the similarity of the code (Code Plagiarism Detection (CPD) VS. Code Similarity Detection (CSD)). In both of these fields there has been done research in what the best practices are, for the plagiarism detection these can be found in a paper by Novak [15] and for similarity detection these can be found in Novak et al. [16] and Ragkhitwetsagul et al. [17].

2.2.2 Distance metrics

When the structure of the solutions is clearly defined there needs to be a way to define how similar the solutions are to each other. This can be accomplished by using distance metrics. The metrics that will be discussed here are: Levenshtein distance [18], Manhattan distance [19], Hamming distance [20], Jaro-Winkler distance [21], and Cosine distance [22]. These distances will be compared to each other regarding its impact on the accuracy and computational cost for finding similar solutions.

First, take a look at the Levenshtein distance. This distance metric is part of a larger family of distance metrics of which the Jaro-Winkler distance is also a part. This group of distance metrics is also known as the edit distances. The way edit distance metrics work is by altering the two strings until they are similar. The amount of changes that are needed is equal to the length with regard to that distance metric. How these changes are made is what makes the difference between the distance metrics in this class. The Levenshtein metric allows for deletion, substitution, and insertion to calculate the distance between two strings. In contrast, the Jaro-Winkler metric only allows transposition to obtain similar results. For the stated problem the Levenshtein distance is thus a more accurate metric in this case to calculate the distance between two student solutions. Because a simple insertion or deletion of one character would make the solutions fairly similar and would thus warrant a distance of 1, whereas a transposition would require more alterations, thus generating a larger distance. Both distance metrics have similar run times despite their differences. The runtime of these two distance metrics is $O(n + d^2)$ where 'n' is the length of the longer string of the two input strings and 'd' is the edit distance between the two strings. The Hamming distance is also a part of the distance metrics group, but this metric is infeasible for the problem as it requires that the two strings are of the same length. This is a constraint that is not to be guaranteed by all the submissions of the students. This is however unfortunate, because the runtime is more efficient in comparison to the Levenshtein distance or the Jaro-Winkler distance as the runtime is equal to $O(n)$.

The next look will be taken at the Manhattan distance. At first glance this algorithm is useful as the runtime is the same as the Hamming distance from before. However the Manhattan distance is also unusable for the current problem. This is because the Manhattan distance measures the actual difference in a space between two objects. However because the solutions of the students are not in a space defined by inherent features of the submissions this metric can not be used.

The last algorithm that will be looked at is the Cosine distance. The runtime of this metric is slightly worse than the other metrics discussed previously as it has a runtime of $O(nlogn)$. The Cosine distance metric creates vectors of the strings and calculates the distance between these two vectors. This thus needs additional computing power to calculate the distance between two solutions, making it unsuitable

in its current form to calculate all the distances between every student solution.

A possible solution to the computationally slower distance metrics is to make use of hashing techniques. The general use case of these hashing techniques is to minimize the amount of collisions between hashed inputs. However there is also a method which maximizes the amount of collisions while preserving the locality of the submissions. This method of hashing is Locality-Sensitive Hashing (LSH) [23]. What this accomplishes is that a certain amount of buckets is set up and the results that match each other the best will end up in the same buckets. This will give preliminary clusters of solutions. Further distinction between solutions can be made within buckets by applying the previously discussed distance metrics on them. The amount of calculations that needs to be made is drastically reduced by this process.

2.2.3 Clustering

With algorithms for similarity detection and distance metrics the last part to review is clustering techniques of similar solutions. There are a multitude of different clustering techniques, however not all of them are suitable for the current problem instance. As the student solutions have no clearly classified features, no euclidean distances can be calculated between the solutions. Therefore clustering techniques which rely on these, such as K-means clustering, are inviable for this particular case.

One of the most viable clustering techniques for the current problem is hierarchical clustering [24]. Hierarchical clustering is a technique that results in a hierarchy of solutions based on the similarity they have to each other. The result from the clustering can be represented using a dendrogram, an example dendrogram can be viewed in Figure 2.1. The amount of clusters can then be determined by either selecting a fixed number of clusters and cutting the dendrogram accordingly, or by selecting the largest gap in the cluster distances. This has value for the current problem as it is roughly known how many types of solutions are expected, so a fixed value can be chosen to split the clusters on. It also allows for a quick adjustment if an additional cluster is required by the teacher, as all the data is stored in the dendrogram. Research has been done on applying hierarchical clustering techniques on document datasets, in this research they also make the distinction between partitional and agglomerative algorithms, which entails from which direction the clusters are created, from top down or bottom up [25].

Another clustering method, briefly mentioned in the distance metrics section, is LSH. LSH is a method which hashes the input files in such a way that the similar solutions have a high likelihood of having the same digest. These digests are used to place the input files into buckets, the resulting buckets are the clusters that were being sought after. Multiple different algorithms can be used within LSH, most noteably MinHash, which estimates the similarity between two sets. These sets are created by taking permutations of the initial input sets.

To determine if two clusters of solutions can be merged based on their distances between each other, use can be made of linkage criteria. This criteria determines the distance between the two clusters. There are a few of these criteria that can be used, two prevalent ones are complete-linkage and single-linkage. The single-linkage criteria looks at the two closest points between the two sets to determine the distance, whereas complete-linkage looks at the two farthest points of each set. Other criteria

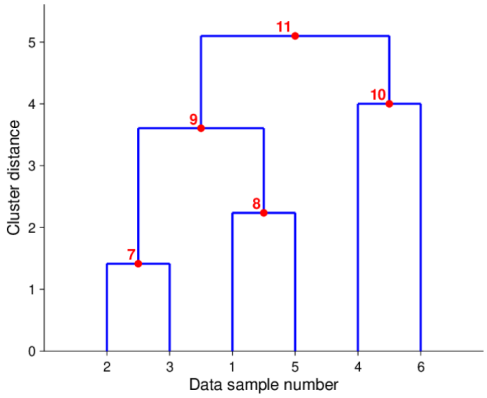

FIGURE 2.1: Example dendrogram result from a hierarchical clustering. Figure from [26]

for example look at a more averaged distance by looking at the distances between all the points in each set towards the other set (Between-groups linkage).

To determine how coherent the resulting clusters are, another metric can be calculated to determine how dense the clusters are. This can be done by calculating the intracluster distance. This distance can be calculated in a couple of ways, but since euclidean distances are unavailable for the current problem the main solution would be Complete Diameter Distance. This is the distance between the two points that are the farthest away from each other. By calculating all the distances in these clusters and between the clusters it can be determined if the clustering was successful or whether or not more or less buckets are needed for the clustering to achieve better accuracy.

2.3 Finding struggling students

In this section there will be an overview of how to find students that have greater difficulties in completing the course in comparison to the rest of the students in the course. First a look is taken at more traditional methods of finding students, in the second section a look is taken at methods which make use of the revisions that the students have while making the assignment.

2.3.1 Traditional methods

A first step for finding students who have difficulties with the material is to look at the final submissions of the assignment. Tests can be run on the submitted code to check for certain mistakes that were made. These tests do have to correlate to specific learning goals of the assignment, in this way they act as a rubric. This is a very basic way of trying to find out who the struggling students are. The flaw of this method is that even if a student has found a way to successfully pass the tests that were created for the assignment, the student might still not fully understand why it works or why it is the correct solution. This flaw is of bigger concern when the student has access to these tests before it has to be handed in, due to the student being able to game the system by continuously altering the code until all the tests have passed.

2.3.2 Revisional methods

Using the revision history of the students assignment gives additional opportunities to retrieve information about the struggles the student faced during the assignment. This could either be done by giving students feedback directly while they are making the assignment, or by data analysis after the students finished the assignment. This is dependent on the type of system used, as a interactive online tool would be able to immediately provide feedback, whereas an offline tool which only stores intermediate results would be more aimed at the data analysis.

The first paper that looks at analyzing the data afterwards is Predicting Performance in an Introductory Programming Course by Logging and Analyzing Student Programming Behaviour by Watson et al. [27]. Their methods made use of a method which took into account how much time the student took to resolve a problem and compared that to the times of the other students. This gave an accurate representation of which students had more problems during the assignment. To make use of this method there needs to be an accurate calculation of the amount of time that the student spend on the assignment and the problems they encountered. This also means that there needs to be a detection system to see when the student was solving a problem with the assignment.

A paper that takes this to the next level is Models for Early Identification of Struggling Novice Programmers by Munson et al. [28]. This paper makes use of keystroke recordings to identify how well students were making the assignment. They determined behavioural patterns that could identify how well the student were actually performing the assignment. This method requires lots of info about the machine that the student is making the assignment on, as the keystrokes need to be recorded. This would therefore not be a valid strategy to take for the current problem.

Another paper describes The normalized Programming State Model by Carter et al. [29]. This model looks both at the syntactic and semantic correctness of the program over the course of the revisions of the students. This does mean that the model needs to understand the submissions on both a syntactic and a semantic level. The paper compared the effectiveness of the model against two previously established measures: The Error Quotient and the Watkin score. It proved to be more effective than these two measures.

Chapter 3

Development of implementation

In this chapter, the design decisions regarding research questions listed in Chapter 1 are laid out. In the first subsection the clustering technique is explained which makes use of LSH and k-means clustering. In the second subsection a look is taken at finding struggling students which mainly makes use of number of submissions and the test scores of the revisions. In the third subsection the visualization is discussed which explains what is shown with regards to what is the most difficult specification test question for each of the created clusters. The chapter ends with an overview of the program's architecture.

3.1 Design decisions

3.1.1 Clustering

Starting with clustering, the initial idea was to use abstract syntax trees to find clusters of student solutions. The creation of the abstract syntax trees was fairly simple with the use of ANTLR [30]. However it turned out to be difficult to actually make use of these trees to achieve viable clusters. Small deviations in the code could create branches which would than be classified as totally different solutions by the clustering algorithm. This is not the goal that was in mind. Therefore a more structural approach was taken to create the clusters. To get a more structural representation of the student solutions, they were first compiled and the bytecode was then used for the clustering algorithm. Not every byte of the bytecode was actually used, as the first few bytes are not part of the actual student solution, but are for instance the version of python that was used to compile the student solution [31], [32].

The clustering algorithm starts with Locality Sensitive Hashing (LSH). LSH splits the bytecode of the student solution into a given amount called stages, each of these stages are then hashed into one of a given amount of buckets. If two student solutions are similar in that stage they will end up in the same bucket as the same hashing function is used for each stage. The amount of stages and buckets that are chosen is of influence as to what results you can expect, therefore tests have been done to determine what values for the stages and buckets work best for this given problem instance.

When all the stages have been hashed into their respective buckets every student solution has a list of bucket numbers to which each of the stages corresponds. If two student solutions have similar buckets on the same position in their list that indicates that they have some similarity. To calculate the distance between these different solutions, these lists of buckets can be used in combination with the hamming distance metric. The hamming distance calculates the number of steps needed for

altering one list to match exactly the other list. If these lists were used to calculate clusters there might be some issues due to the high dimensionality. Dimensionality reduction is necessary to make this more manageable. To accomplish this, the dimensions will be the distances between the student solutions and some reference points. The first reference point is the expert solution provided by the responsible lecturer, the second reference point is the file that the students are given at the start of the exercise. These two reference points were chosen as they represent two different kinds of solutions that students might have, which are either equal to the provided solution or they have not yet started the assignment. These are two clusters which are distinct and provide the basis for the rest of the clusters. The other two reference points are files which have no links to the assignment that the students tried to make. One is a relatively long file compared to the student solutions (500+ lines) and the other is a rather short file (10 lines). The long file is used to determine the non compiling solutions. If a student solution does not compile they will receive the bytecode of that file so that every non compiling solution will have the exact same distances to every other solution, making them a cluster by default. The reason for choosing four reference points is to limit the the amount of ambiguity from the results. If you have two reference points and you only have the distances towards these two points, there are multiple positions in space from which this is possible. Thus If two solutions have the same distances towards the two referent data points, this does not indicate the solutions are similar solutions. If two solutions have the same distances towards the two referent data points, this does not indicate the solutions are similar solutions. Examples for two and three reference points are shown in figure 3.1 and 3.2.

The fifth reference point that is used is the length of the student solution, this gives the final dimension to create enough diversity between the features if one or more of the reference points in some fashion overlap. With the features completed the student solutions need to be clustered together in such a manner that there are a sensible amount of clusters with regards to the assignment and in comparison with the features gained. The first idea was to use regression to achieve this, however for regression you would either need examples of the clusters you would want to create or you do not have any control over the amount of clusters that are created. Having control over the amount of clusters that are created is however something that is necessary given the context of the problem. As the amount of different solutions that students will differ from assignment to assignment. Therefore a more viable alternative would be k means clustering, as it can be stated beforehand how many clusters you want before you start clustering. To make sure that the right amount of clusters is taken the algorithm is run multiple times on different amount of clusters each time. This is tested for the range of 2 to 15 clusters, this range is chosen as there are always two clusters due to the expert solution and the starting file for the students and a maximum of 15 as there are no more than 15 significantly different solutions to warrant its own cluster. However, once you have all these clusters a method needs to be defined to determine how many clusters is actually the right amount. The Davies-Bouldin index [33] was used to determine the number of clusters needed. This is a metric that gives an indication about the cluster density and the separation between the clusters, the lower the score the better it is. Using this metric the an amount of clusters can be determined which mathematically is closest to the optimal amount of clusters.

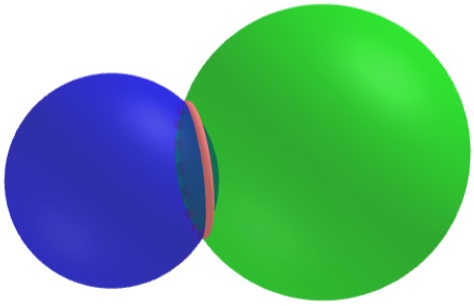

FIGURE 3.1: Intersection between two spheres, there are multiple intersection points between the two spheres

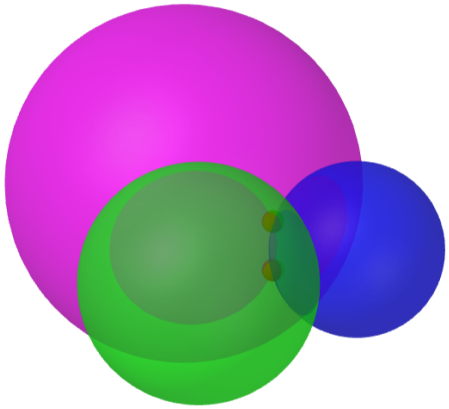

FIGURE 3.2: Intersection between three spheres, there are two intersection points between the three spheres

3.1.2 Finding struggling students

The second part is finding students who have a higher likelihood of having struggled with the assignment compared to the rest of the class in that year. For this, indicators need to be determined to find out which students might have struggled with the assignment. To accomplish this there were two main points that were taken into consideration given the available data. The first point is the number of submissions that each students has provided, which are the revisions in the dataset. Using the number of submissions as indicator of whether a student has struggled more compared to the rest of the class follows the assumption that students who performed more submissions are struggling with the assignment since they were not able to complete the assignment as quick as their peers. To hopefully counteract this a bit only the students who have a significant difference compared to the rest of students of that year are taken into account. These are the students who have a submission count twice the standard deviation over the mean amount of submissions of the students of that year. The second method is to look at the final submission score, this looks at the students who in contrast to the previous group did not persist in trying to solve the problem, but stopped while only having achieved a partial score. Therefore the students who might have struggled more with the assignment are determined somewhat similar for the submission score as they were for the number of submissions. This is by trying to figure out which are the outliers, in this case by looking at the standard deviation on the lower side of the mean score of the students. To avoid bias caused by students who did not make attempts to answer the questions, the scores of students who did not start the assignment were not considered in calculating the mean score.

3.1.3 Visualization

The third part is about visualizing the gathered information into an easy to navigate result screen. This result screen should be able to provide the teacher with all the information in an easy to understand way to prevent an information overload. The information that should be presented comes in three ways. The first are the clusters of similar student solutions that have been described in a previous section. The second are the students who have a higher likelihood of having struggled with the assignment. The third part is the main problem the students faced in each cluster, this could indicate that students who ended up with a certain solution also ran into the same problems. The main part of the visualization are the clusters of the similar solutions, these are presented in columns. For each cluster there is a separate column with the student numbers for each of the students that matched with the rest of the students in the column. The student numbers are also links to WebLab to provide an insight into what kind of solution the students came up with. Inside these clusters the students with a higher likelihood of having struggled are highlighted in red. The letter 'T' is placed behind the student number if there is a higher likelihood of the student having struggled with the testcases, or the letter 'S' is placed behind the student number if there is a higher likelihood of the student having struggled based on the number of submissions.

3.2 Architecture Overview

In Figure 3.3 there is an overview of the architecture as described in the previous section. The flow of the program starts with the submission of the necessary files.

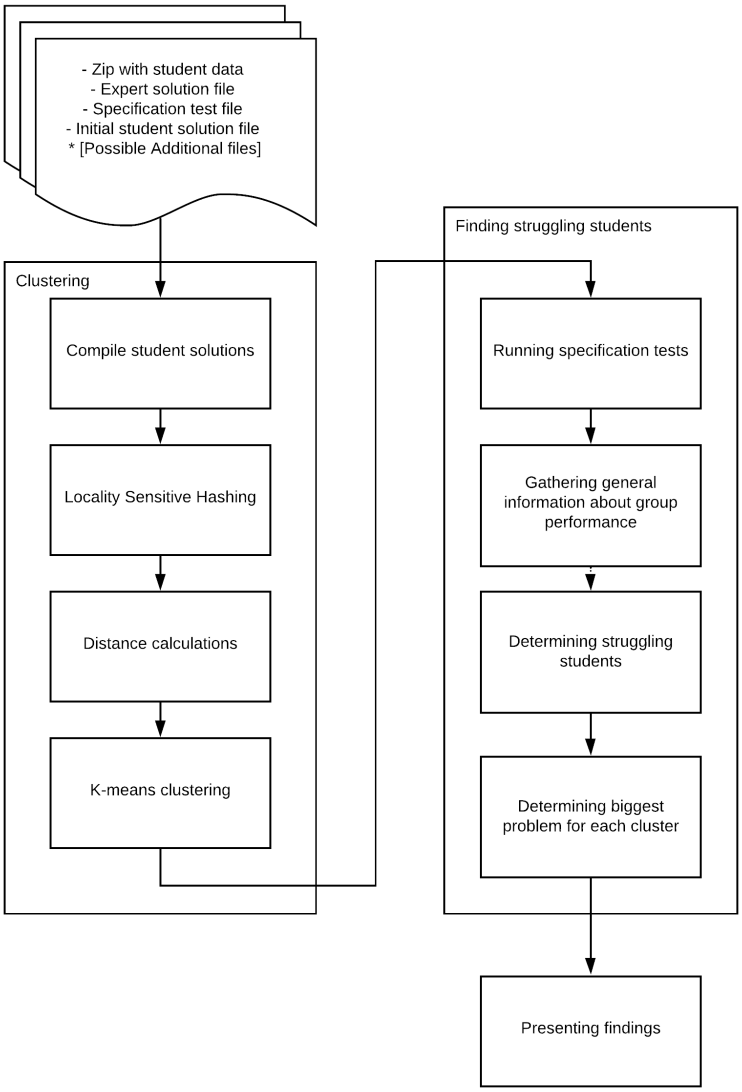

FIGURE 3.3: Architecture overview

These files are the zip file containing all the submissions of the students as exported directly from WebLab, the expert solution file, the file with the specification test, and the initial file that the students receive when they first open the assignment. The input from these files are then used to first go through the clustering algorithm and afterwards through the struggling students algorithm, finishing up with the visualization.

Chapter 4

Evaluation

This chapter presents the results and findings of this work. This is done in two different ways. The first method is by comparing the data from the algorithm to the data provided by the teacher. For the clustering of the student solutions, this is done by comparing the resulting clusters to the clusters the teacher had manually created. For the section about struggling students, this is done by comparing the results of the algorithm to the results of the mid term. The second method is through the focus session group. The points that were made during this session are categorised and discussed in this chapter.

4.1 Algorithm evaluation

4.1.1 Clustering

Name of assignment	Number of Clusters	Total number of submissions checked	Wrongly clustered solutions	Percentage correct
Smart home	4	56	7	87,5
Calculate BMI	7	62	5	91,9
Calculate Yahtzee scores	9	62	6	90,3
Bubble sort	6	54	8	85,2
Points and shapes within shapes	5	58	4	93,1
Vehicles	7	50	1	98

TABLE 4.1: Results of clustering the last student submissions

The table consists of five columns, the leftmost column gives the name of the lab assignment. The second column states the number of clusters the algorithm determined to be optimal by the Davies-Bouldin index. The third column states the number of student solutions that were checked for that assignment. The fourth column states the number of wrongly clustered solutions out of the total number of submissions checked in the third column. The fifth column states the percentage of correctly clustered student solutions.

4.1.2 Struggling students

Each table has the name of the assignment at the top left, the results are shown in the bottom right four squares. These depict what the algorithm predicted would be

Smart home		Actual	
		Struggler	Non-struggler
Predicted	Struggler	20	1
	Non-struggler	66	152

TABLE 4.2: Result of identifying struggling students for the smart home assignment

Calculate BMI		Actual	
		Struggler	Non-struggler
Predicted	Struggler	22	4
	Non-struggler	65	137

TABLE 4.3: Result of identifying struggling students for the calculate BMI assignment

Calculate Yahtzee scores		Actual	
		Struggler	Non-struggler
Predicted	Struggler	3	2
	Non-struggler	84	113

TABLE 4.4: Result of identifying struggling students for the calculate Yahtzee scores assignment

Bubble sort		Actual	
		Struggler	Non-struggler
Predicted	Struggler	9	3
	Non-struggler	81	114

TABLE 4.5: Result of identifying struggling students for the calculate bubble sort assignment

Points and shapes within shapes		Actual	
		Struggler	Non-struggler
Predicted	Struggler	30	10
	Non-struggler	57	101

TABLE 4.6: Result of identifying struggling students for the calculate points and shapes within shapes assignment

Vehicles		Actual	
		Struggler	Non-struggler
Predicted	Struggler	25	12
	Non-struggler	31	123

TABLE 4.7: Result of identifying struggling students for the calculate vehicles assignment

the struggling students and how many of these were correct compared to the actual struggling students.

4.2 Focus group evaluation

To validate the designed algorithms a focus group session was held with experts from different fields such as teachers, education, and learning analytics to gain feedback. This feedback has been split up into three different categories, critical remarks, pros and cons, and improvements. Each of these categories are laid out below and will be discussed individually. The anonymized transcription of the focus group session can be found in Appendix C.

4.2.1 Critical remarks

The first category are critical remarks, these are the statements made by the participants of the focus group that criticize the design of the algorithm directly.

- Length of solution creates problem for LSH
  This statement is made with regards to the clustering algorithm, which makes use of LSH to differentiate between the different types of solutions. However inherent to LSH is that if there is a shift in the solution the locality on which the hash is based is no longer valid. Therefore the hashes do not end up in the same bucket. Therefore near similar solutions do not end up in the same cluster. Even though it seems to not interfere with the results this is a theoretical problem that might occur when the scope of this program is taken beyond the current use case of the introduction to Python programming course.

- Student behaviour might interfere with statistics for struggling students
  Currently struggling students are being determined by the number of submissions and the final test score of each of these students. However student behaviour might influence these numbers significantly, making the results unreliable. An example of this would be if a student continuously saves the work while doing the assignment. The system recognizes every save as an additional submission thus increasing the number of submissions for this student, making it more likely that this student is marked as struggling based on the amount of submissions. Even when this might not be the case.

- One testcase for each learning point not always viable, for example if assignment is single algorithm.
  For the visualization to work the best to show which cluster had the most trouble with which testcase it would be ideal if there was only one testcase for each learning point. As the testcase shown as the most problematic for the cluster would be the learning point that the students had the most trouble with. However the focus group pointed out that creating a single testcase for each learning point is not always possible, for example when the assignment is to recreate a single algorithm there is most likely only one correct functional approach. Therefore you would only be left with a single testcase.

- No hard cutoff point for struggling students
  The final critical remark that the focus group mentioned was that there should not be a hard cutoff point to determine struggling students. There should always be a spectrum on which the students can be placed. As a student might be struggling a bit which would then not show up, or a student might have

the behavioural trades specified before and therefore show up when this is not meant to. By just showing the gradient the teacher has the ability to select where the struggling students might be.

4.2.2 Pros and cons

The second category are statements of pros or cons of the proposed algorithms, these were statements made on individual parts of the algorithm which were either positive or negative.

- Compiling to bytecode removes differences of variable naming
  A positive aspect about compiling to bytecode instead of comparing on a syntax level is that a difference of variable naming between students does not impede the similarity between the two students.

- Non static amount of clusters as the amount of clusters is unknown
  Another positive point is the dynamically initialized number of clusters, as when an assignment is received it is not clear how many different types of solutions the students have come up with.

- Could both identify students struggling and students that try to game the system
  Students that are struggling with the assignment and students who are trying to game the system would showcase the same type of behaviour. Therefore the algorithm that works for one would also help to identify the other students, therefore this algorithm could be multi functional depending on the needs of the teacher. This however also means that the algorithm can not distinguish between these two groups.

- Have the testcase for each individual testcase
  Due to not receiving the test scores for each revision from WebLab every testcase is rerun for every submission while analyzing the student solutions. Therefore there is testcase specific data for each student, which is more useful than to reduce everything to a single number.

- Only considers syntax but not the functionality
  A negative point for the clustering algorithm is that it only looks at syntax but not the functionality.

4.2.3 Improvements

The third and final category are statements about possible improvements to the algorithms.

- Look at outliers from clusters to create separate cluster
  An improvement for the clustering algorithm is to look at the created K clusters and check which solutions have a higher distance to the center of its cluster compared to the rest. These are the submission which do not actually fit a single cluster but are outliers. These outliers should be put together in a single

cluster to signify that these are special submissions which should get an additional look by the teacher as there is most likely something wrong in these solutions.

- Look at ratio, and number of submissions until highest score
  Be cautious when looking at the number of submissions that were needed to achieve the maximum score of the assignment. As any submissions beyond that point might just be attempts to learn more.

- Use timestamps to look at sessions
  To further find out if students have been struggling, the time that students have taken to solve the assignment can also be taken into account. As every revision receives its own timestamps, sessions can be determined that students have been working on the assignment. If you add these sessions together you get the total amount of time that students have been working on the assignment.

- Look at fluctuating test score to find struggling students
  Another way to determine whether a student has been struggling is to look at the progression of the test score throughout the revision history. The expectation would be a steady increase in test score with small fluctuations.

- Preview of each cluster
  For a better visualization of the clusters, a preview can be given for each cluster to immediately get an idea what each cluster contains in type of solution.

- Show (in line chart) progression of student
  It might also benefit the visualization to allow a line chart to show the progression of the students test score. By having this chart the teacher can analyze per individual student whether they have struggled with the assignment or not.

- Individual pages for each part
  To avoid cluttering of the results it will be beneficial if all the different result parts would be shown on separate screens.

- Allow naming of clusters
  By allowing the clusters to be manually named the clusters can be more easily identified.

- Color parts in solution which are similar to the rest of the cluster
  This is a more advanced improvement request, as this would entail to identify the parts that are similar in every solution in each cluster and highlight this for each solution.

Chapter 5

Discussion

In this chapter, the results of this work are discussed and future research possibilities are explored. The results are discussed in two parts, first the comparison between the output of the algorithm and the provided results by the teacher and second the discussion with the focus group. In the next section some ethical considerations for this thesis project are laid out, with regards to the use of student data and the anonymity of the focus group participants. The chapter ends with future research possibilities.

5.1 Discussion of the Results

5.1.1 The Effectiveness of the Algorithm

Regarding the results of the clustering algorithm, the overall percentages of correctly clustered solutions are very promising. In general, around 90% of the solutions are clustered correctly according to the manually created clusters. There does not seem to be a clear correlation between the number of clusters that the algorithm perceives to be the optimal amount of clusters and the percentage of correctly clustered student solutions. This would mean that the percentage of correctly clustered solutions does not simply increase because the number of created clusters is higher. Thus having more detailed clusters does not imply better clustering results only because there are more clusters to divide the solutions over. The clustering however is not perfect, as it does not reflect the manually clustered. There might be improvements, as pointed out by the focus group, in finding the solutions which are outliers by themselves as this would improve the overall clustering score.

The second part is the prediction of which students are more likely to have struggled with the assignment. These results are less optimistic compared to the clustering part. For all the assignments that the algorithm looked at it missed more than half of all the students who were struggling. However over 75% of the identified struggling students were indeed students who turned out to be struggling with the assignments. From these results it can be concluded that the current method for determining which students might be struggling with solving the assignments is inadequate. two possible solutions would be to use different indicators or the results of multiple assignments should be combined. Because when assessing the algorithm a lot of the same students kept reappearing from assignment to assignment who turned out to be struggling, while the students that only appeared once as a struggling student for that assignment would most likely not be a struggling student in the midterm.

5.1.2 Discussion of the Focus Group

The final part is the focus group session, where critical remarks were discussed.
The first critical remark is about the length of the solutions which might cause problems for LSH. During the design process this was noticed, however during the testing phase it turned out that the results were not skewed by this point because the students receive an initial file to start their solution from. This file already states the order of the classes or functions the solution should have. Therefore, if a solution is significantly longer or shorter than the other solution, it is most likely a completely different solution, thus not being clustered in the same cluster.
The second critical remark was about student behaviour interfering with the results for clustering students. This is a valid concern as the results of the struggling students algorithm are not a great success. The behaviour played a larger part than was originally thought and should probably be taken more into account in future research.

The third critical remark is about the one test case per learning objective of the assignment. This work theorized that the revision history can be used optimally if each test case are linked back to a single learning objective for the students. This way if they failed a test case it was immediately clear what learning point was not evident to the students. However the focus group pointed out that this could not always be possible, because if an assignment is for example a single algorithm the test cases could not be designed in such a way to fulfill this requirement. Therefore it is impossible to always take maximum usage out of the revision history by looking at the passed or failed tests.
The fourth critical remark is about having a hard cutoff point for where students are struggling. This is also a very valid remark as currently the results for the struggling students are not great. If the hard cutoff was let go and a listing was shown with a likelihood of the students to struggle with the midterm some of the noise from having the hard cutoff could be removed.

5.2 Ethics

In this section the ethics of this work are briefly touched upon and clarified.

5.2.1 Use of Student Data

This thesis used the student data of the introduction to Python programming course. This data was used to improve the course for future years, which therefore constitutes as fair use under the GDPR rules.

5.2.2 Privacy of the Focus Group

The focus group session that was held to validate the results of this work was recorded to have an easier way to analyze the session afterwards. Every participant was informed that the session was audio recorded and that they would never be individually identified for any statements they made during the session. In the transcription of the session every participant has been anonymized.

5.3 Future research

This section presents future research possibilities regarding learning analytics for teachers in beginner programming assignments.

5.3.1 Similarity

One possible future research opportunity would be to look into a different way of finding similar student solutions other than using LSH. This was done to eliminate the theoretical problem of shifting the solution with a few lines making the rest of the hashes in the LSH different. Thus making the solutions differ in the final clustering, even when they might be similar in all other aspects.

5.3.2 K clustering distance

Currently there are K clusters created and these would be presented to the user based on the best index score. However this would leave the outliers in clusters which they do not actually belong to. Therefore a way may be determined to remove these outliers from the clusters. This makes sure that the centers of these clusters are re-centered across the actual clusters.

5.3.3 Indicators for Struggling Students

As the current algorithm for finding struggling students has been relatively ineffective at finding these students, future research should focus on finding more or different indicators which take a closer look at student behaviour. Examples of these different indicators would be to look at the duration of the session and number of sessions, to establish the actual time that students have spent working on the assignment rather than the number of revisions that the students produced. Another indicator would be the progression of the student score over the number of submissions as a pattern might be discovered there which would indicate that a student is struggling. Such a pattern might be that the student is stuck for a very long time on the same test score, indicating that they are stuck with a particular problem.

5.3.4 Visualization

Improvements can be made on the visualization front. This can be done by for example giving a preview for each of the clusters that are presented, to give the teacher an immediate insight in the types of solutions the students have created without having to go to an external website to view them. An other option would be to spread the results over multiple screens, this to avoid the aforementioned information overload.

Chapter 6

Conclusion

The expectation would be a steady increase in test score with small fluctuations. By looking at the code on a structural level and comparing it with LSH, features can be created for each of the solutions which can be used as input for a clustering algorithm. Improvements can be made by trying to identify the student solutions which differ a lot from the rest of the group to avoid placing them into one of the existing clusters.
The identification of students who appeared to have had greater difficulty with the assignment turned out to be more difficult. The current results are not complete, as many of the students who were struggling are not identified by the current algorithm. The current algorithm only makes use of the number of submissions and the final test score. Student behaviour should be taken more into account when trying to identify these students.
The way the teacher can be given insight into the problem areas the student encountered is foremost by showing the results of the clustering results and the students that struggled the most during the assignment. Further insight can be given by looking into the individual test cases that the students failed the most if these test cases are setup in such a way that they represent singular learning points.

Overall, learning analytics can play a big role in providing teachers with more insight into the progress of the students and the problems they encountered during their introductory programming course. This could either be by looking at the solutions the student created or by looking into the problems the students encountered while solving the assignments by taking a look at the revision history.

Appendix A

Thesis Proposal

Erik Oudsen
4573706
e.m.oudsen@student.tudelft.nl

Learning analytics for programming assignments
Thesis Proposal

2021-02-16

1 Introduction

Learning analytics in higher education looks at the usage of learning systems and tries to find ways to make this beneficial for the students and teachers [1]. When applying this to programming education, the purpose is to find out what types of analytics would be useful to look for and how to use that data to make improvements in such a way that students and possibly teachers benefit from it. Learning analytics for programming assignments is an interesting topic as it is able to optimize the way programming courses are given and enhance the information that is retained for the students that participate in them. Due to the increasing number of new students that enroll for computer science related courses it is important that the time that teachers spend is as efficiently as possible. If the learning systems could take away part of the tasks of the teachers this would be beneficial for the time management of these teachers. However, implementing learning analytics into learning systems is not as straightforward as might be hoped. This is due to the difficulty for learning systems to provide the right information that is needed or providing information that could be seen in multiple different ways [2]. Therefore a more complex system needs to be developed to deal with this open context in order to provide the best benefits for the students and teachers. Many attempts have been made to provide useful feedback to students in learning systems based on their programming submissions [3]. These systems however mostly focus on static analysis of the code submissions. The systems do, for example, not take into account the progress the student makes from submission to submission. This is a key element of where improvements can be made with regards to the currently available learning systems. The insights that can be gained by looking at the revision history of the students can be very useful for teachers. These insights could for example provide them with information about how well the taught topics were understood or whether particular students had more trouble solving the assignment than was expected. The solution proposed here is to make use of the history of student submissions to provide feedback to the teacher about how well the students are performing for their course.

2 Thesis Objective

The objective of the thesis is to provide useful feedback to teachers about how well the students are performing in the course based on the revision history of students. This feedback will be in such a form that the teacher has a faster and better overview into which solutions the students came up with and what problems they encountered. This feedback would aid the teacher to prepare for the next lecture, as to explain the parts that were most difficult for the students. The way this would be done is to make use of all the solutions that the students made over the course of the assignment. This feedback will consist of grouping the similar solutions together. This way the teacher can quickly determine if a large group of students made the same mistakes while solving the assignment. The second part of the feedback will be an indicator, visible for the teacher, for certain students of which is suspected that they had a more difficult time while solving the assignment. This will also give the teacher an indication of the difficulty of the assignment. If a certain student gets highlighted multiple times for different assignments this could also indicate that the student is having a difficult time with the course and actions could be taken before exam time.

2.1 Concrete Research Questions

The main research question that this thesis project tries to answer is:
How can learning analytics be leveraged to provide teachers with information about the students progress and problems in their introductory programming course?
To answer this question the following sub questions will be answered:

1. How can the most common created solutions of students be found?
2. How can insight be given to the teacher about the problem areas students encountered?
3. How can students that appeared to have had greater difficulty with the assignments be identified?

The first question is aimed at trying to find out how similar submissions of students can be grouped in such a way that a human would also do it. This requires finding out which differences in code would lead to a significantly different solution than that of another student.
The second question is figuring out where in the history of the student solving the assignment they had the most trouble with understanding the problem. At what point in time did the student figure it out, and what was the part that the student figured out, and finally how is this information represented to the user in such a way to make it clear.
The third question is focused on finding students that have significant problems with completing the assignments. The main goal of this is to find the ”trial-and-error-programmers”that continually make small changes to the program until the maximum score that is obtainable for the assignment is achieved. During this, the student is not learning the actual problem, but gaming the system. The system will try to pinpoint these students to give an overview to the teacher who these students are and what could be done to help them focus on solving the problem at hand instead of trying to game the assignment.

2.2 Contributions

Below are the contributions that this thesis project hopes to have accomplished when it is finished. Programmatic contributions:

1. Provide an overview for the teacher about the different types of solutions that the students came up with.
2. Provide an overview for the teacher about the hurdles the students encountered.

Research contributions:

1. Find potential indicators for teachers that provide the most useful information about the students.
2. Find an algorithm that can group student submissions based on similarity similar to how a teacher would group the students.

3 State of the Art

In this section the state of the art is laid out with regards to the above stated research questions. The most recent literature will be reviewed to find out what the current solutions are to these problems.
The first part that will be looked at will be papers regarding clustering of similar pieces of code. Although there has not been much research done in clustering of similar pieces of code, there has been research in finding similar pieces of code within your own codebase (duplication) [4], or more potently finding duplications outside of your own codebase or more commonly known as plagiarism [5], [6], [7], [8]. The goal of this thesis is not to find plagiarism but to find similar solutions. Thus if the algorithms provide a similarity index between two pieces of code this index can be used to cluster the results together using for instance hierarchical clustering [9]. These different algorithms for code comparison should be looked at to determine which would best be fit to provide a code similarity metric.
The second part is more focused on how the available learning analytics can best be presented to the teacher. This focuses more on what indicators a teacher might look at to see how the students are currently working with the learning material and what visualisations are best used to convey the data correctly to the teacher. This is currently done by creating models such as the ”triadic model of teaching analytics”[10] or by aggregating and presenting the information in such a way that the teacher is confident that the information provided is sound [11].
The last part this thesis will focus on is how students can be identified that have greater difficulty solving the assignments than their peers. This goes in the direction of predictive learning analytics which aims to identify learners who may not complete a course [12]. In the field of predictive learning analytics there are more studies that aim to take predictive learning analytics to a larger scale than single course test trials [13], [14], [15]. These efforts have created greater insight into what are good indicators that a student has greater difficulty with the assignment.

4 Method

In this section a plan will be laid out on how this thesis project will be approached. The first step in the process would be to look at the available literature in the form of a literature review to get the first generic requirements. These will form the foundation of the final requirements to bring this project to a successful conclusion. The next step would be to hold informal interviews with the stakeholders of the project. During these interviews more concrete requirements will be formed in conjunction with the general requirements laid

out during the literature review. Together these requirements will determine the guidelines for the project.
After these requirements have been determined, a first prototype will be developed. This prototype will incorporate the requirements and thus fulfill them as the first prototype will be demonstrated.
The first prototype will be demonstrated to the stakeholders to determine what parts of the prototype should be enhanced or altered to better satisfy the stakeholders needs. These alterations would be incorporated into the second prototype. This version will then be used to evaluate its efficacy with regards to the setup requirements.
This evaluation will be done by comparing the results that the prototype provides with manually generated results by stakeholders and/or other experts that are associated with the course in such a way that they can provide reliable manually produced results.

5 Milestones and Deliverables

below are the milestones and deliverables for the project. At each date is specified what would be required to be presented.

1. Literature review (15 Feb - 01 Mar)
   Deliverable: Literature review chapter of the report.
2. Informal interviews with stakeholders (01 Mar - 08 Mar)
   Deliverable: Final requirements for the project.
3. Create first prototype (08 Mar - 12 Apr)
   Deliverable: Prototype that incorporates the requirements.
4. Create second prototype (12 Apr - 03 May)
   Deliverable: Prototype with the suggested improvements from the first prototype.
5. Evaluate prototype (03 May - 17 May)
   Deliverable: Evaluation of the prototype.
6. Create report (17 May - 14 Jun)
   Deliverable: First draft of the report.
7. Rewrite report based on feedback (14 Jun - 28 Jun)
   Deliverable: Rewritten report based on feedback.
8. Final presentation (05 Jul)
   Deliverable: Final presentation

References

[1] G. Siemens und R. S. J. d. Baker, „Learning Analytics and Educational Data Mining: Towards Communication and Collaboration," in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, Ser. LAK '12, Vancouver, British Columbia, Canada: Association for Computing Machinery, 2012, S. 252–254, ISBN: 9781450311113. DOI: 10.1145/2330601.2330661.

[2] P. Prinsloo, S. Slade und F. Galpin, „Learning Analytics: Challenges, Paradoxes and Opportunities for Mega Open Distance Learning Institutions," in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, Ser. LAK '12, Vancouver, British Columbia, Canada: Association for Computing Machinery, 2012, S. 130–133, ISBN: 9781450311113. DOI: 10.1145/2330601.2330635.

[3] H. Keuning, J. Jeuring und B. Heeren, „A Systematic Literature Review of Automated Feedback Generation for Programming Exercises," *ACM Trans. Comput. Educ.*, Jg. 19, Nr. 1, Sep. 2018. DOI: 10.1145/3231711.

[4] O. Maqbool und H. A. Babri, „The weighted combined algorithm: a linkage algorithm for software clustering," in *Eighth European Conference on Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings.*, 2004, S. 15–24. DOI: 10.1109/CSMR.2004.1281402.

[5] L. Luo, J. Ming, D. Wu, P. Liu und S. Zhu, „Semantics-Based Obfuscation-Resilient Binary Code Similarity Comparison with Applications to Software and Algorithm Plagiarism Detection," *IEEE Transactions on Software Engineering*, Jg. 43, Nr. 12, S. 1157–1177, 2017. DOI: 10.1109/TSE.2017.2655046.

[6] M. Chilowicz, E. Duris und G. Roussel, „Syntax tree fingerprinting for source code similarity detection," in *2009 IEEE 17th International Conference on Program Comprehension*, 2009, S. 243–247. DOI: 10.1109/ICPC.2009.5090050.

[7] L. Jiang, G. Misherghi, Z. Su und S. Glondu, „DECKARD: Scalable and Accurate Tree-Based Detection of Code Clones," in *29th International Conference on Software Engineering (ICSE'07)*, 2007, S. 96–105. DOI: 10.1109/ICSE.2007.30.

[8] M. Ďuračík, E. Kršák und P. Hrkút, „Scalable Source Code Plagiarism Detection Using Source Code Vectors Clustering," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, 2018, S. 499–502. DOI: 10.1109/ICSESS.2018.8663708.

[9] S. C. Johnson, „Hierarchical clustering schemes," *Psychometrika*, Jg. 32, Nr. 3, S. 241–254, Sep. 1967, ISSN: 1860-0980. DOI: 10.1007/BF02289588. Adresse: https://doi.org/10.1007/BF02289588.

[10] R. Vatrapu, C. Teplovs, N. Fujita und S. Bull, „Towards Visual Analytics for Teachers' Dynamic Diagnostic Pedagogical Decision-Making," in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, Ser. LAK '11, Banff, Alberta, Canada: Association for Computing Machinery, 2011, S. 93–98, ISBN: 9781450309448. DOI: 10.1145/2090116.2090129. Adresse: https://doi.org/10.1145/2090116.2090129.

[11] A. Van Leeuwen, „Learning analytics to support teachers during synchronous CSCL: balancing between overview and overload," *Journal of Learning Analytics*, Jg. 2, Nr. 2, S. 138–162, Dez. 2015. DOI: 10.18608/jla.2015.22.11. Adresse: http://dx.doi.org/10.18608/jla.2015.22.11.

[12] C. Herodotou, B. Rienties, A. Boroowa, Z. Zdrahal und M. Hlosta, „A large-scale implementation of predictive learning analytics in higher education: the teachers' role and perspective," *Educational Technology Research and Development*, Jg. 67, Nr. 5, S. 1273–1306, Okt. 2019, ISSN: 1556-6501. DOI: 10.1007/s11423-019-09685-0. Adresse: https://doi.org/10.1007/s11423-019-09685-0.

[13] C. Herodotou, B. Rienties, B. Verdin und A. Boroowa, „Predictive Learning Analytics 'At Scale': Guidelines to Successful Implementation in Higher Education," *Journal of Learning Analytics*, Jg. 6, Nr. 1, Apr. 2019. DOI: 10.18608/jla.2019.61.5. Adresse: https://doi.org/10.18608/jla.2019.61.5.

[14] C. Herodotou, B. Rienties, M. Hlosta, A. Boroowa, C. Mangafa und Z. Zdrahal, „The scalable implementation of predictive learning analytics at a distance learning university: Insights from a longitudinal case study," *The Internet and Higher Education*, Jg. 45, S. 100 725, Apr. 2020. DOI: 10.1016/j.iheduc.2020.100725. Adresse: https://doi.org/10.1016/j.iheduc.2020.100725.

[15] C. Herodotou, M. Hlosta, A. Boroowa, B. Rienties, Z. Zdrahal und C. Mangafa, „Empowering online teachers through predictive learning analytics," *British Journal of Educational Technology*, Jg. 50, Nr. 6, S. 3064–3079, Juli 2019. DOI: 10.1111/bjet.12853. Adresse: https://doi.org/10.1111/bjet.12853.

Appendix B

Clustering variables figures

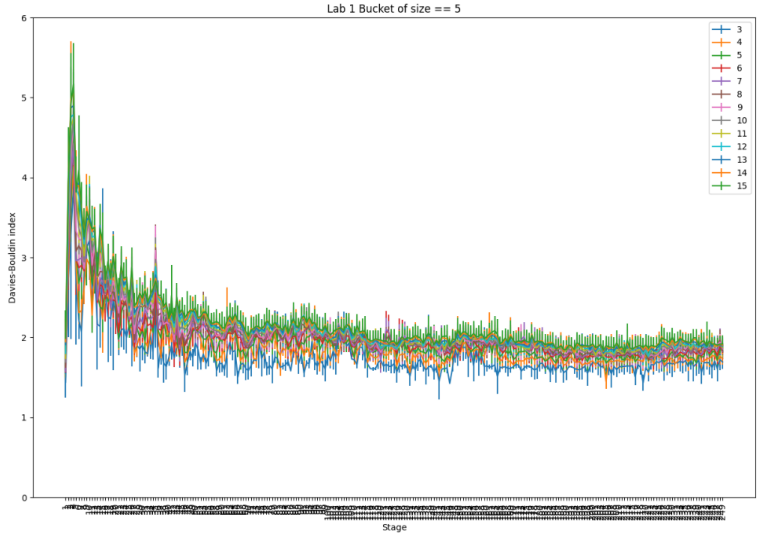

FIGURE B.1: Results from running with 5 buckets, stages versus the Davies-Bouldin index on lab assignment 1

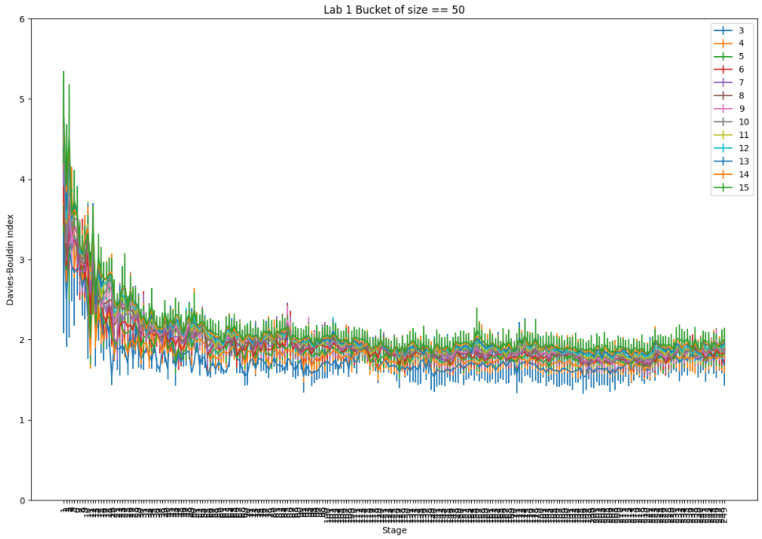

FIGURE B.2: Results from running with 50 buckets, stages versus the Davies-Bouldin index on lab assignment 1

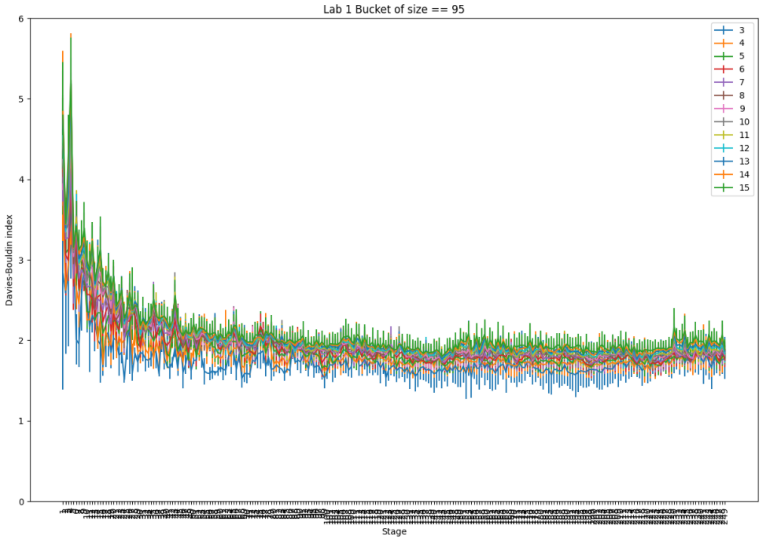

FIGURE B.3: Results from running with 95 buckets, stages versus the Davies-Bouldin index on lab assignment 1

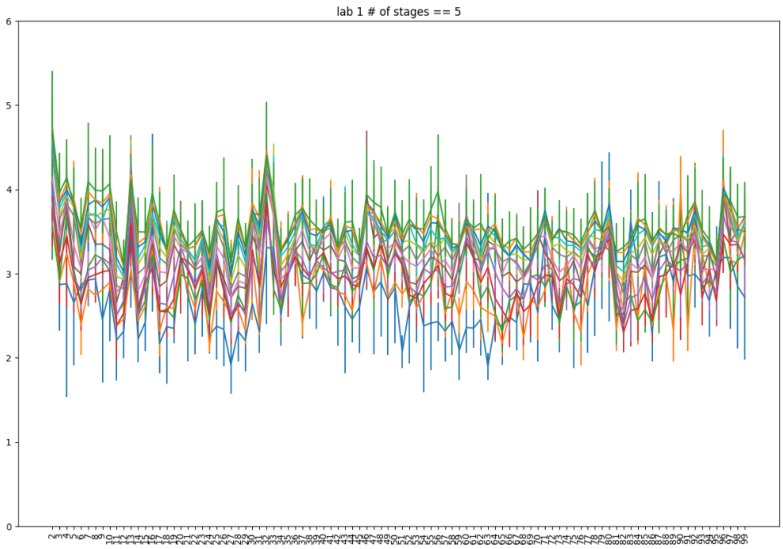

FIGURE B.4: Results from running with 5 stages, buckets versus the Davies-Bouldin index on lab assignment 1

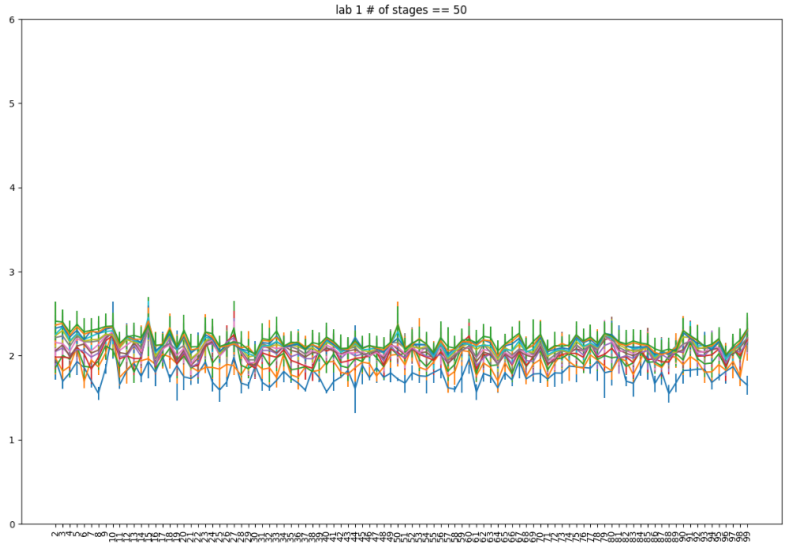

FIGURE B.5: Results from running with 50 stages, buckets versus the Davies-Bouldin index on lab assignment 1

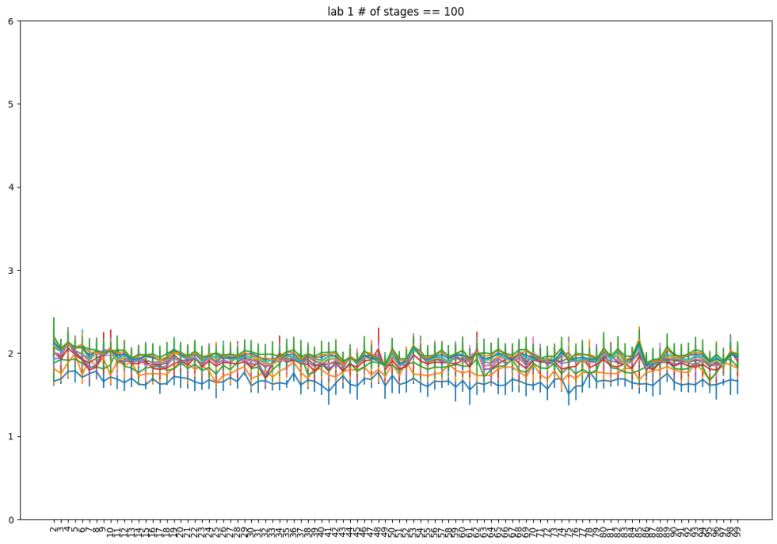

FIGURE B.6: Results from running with 100 stages, buckets versus the Davies-Boulding index on lab assignment 1

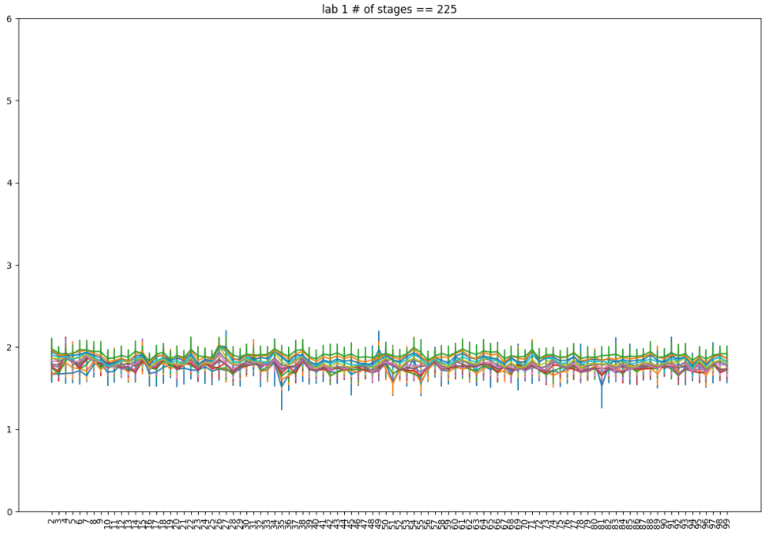

FIGURE B.7: Results from running with 225 stages, buckets versus the Davies-Bouldin index on lab assignment 1

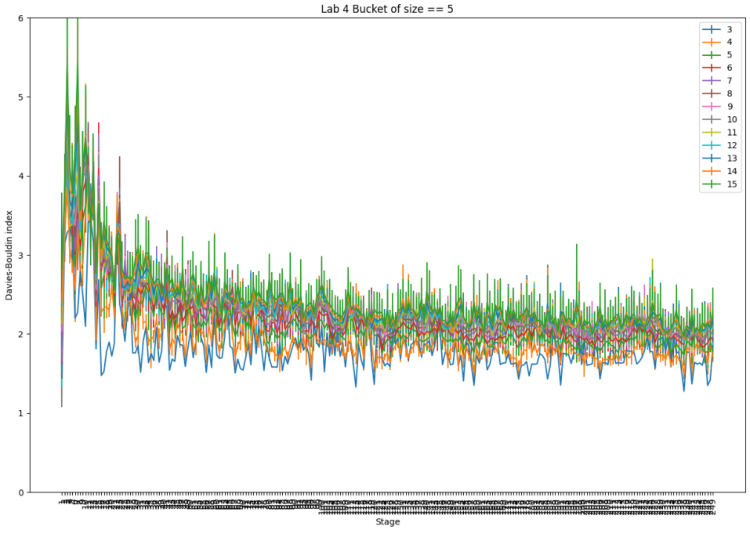

FIGURE B.8: Results from running with 5 buckets, stages versus the Davies-Bouldin index on lab assignment 4

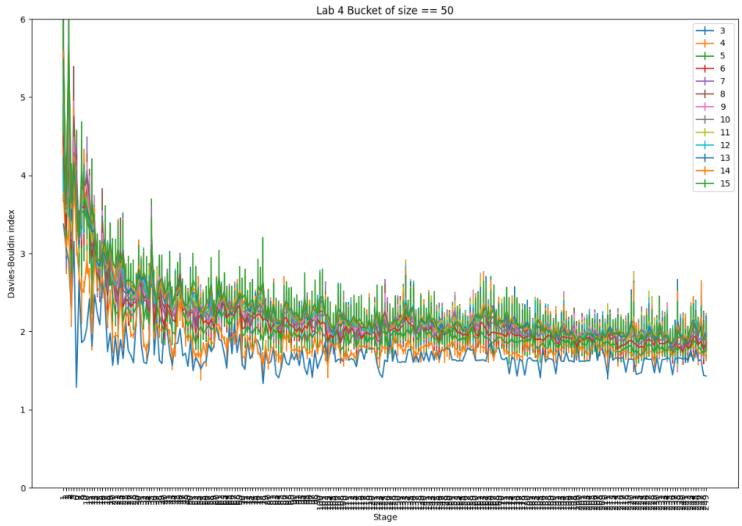

FIGURE B.9: Results from running with 50 buckets, stages versus the Davies-Bouldin index on lab assignment 4

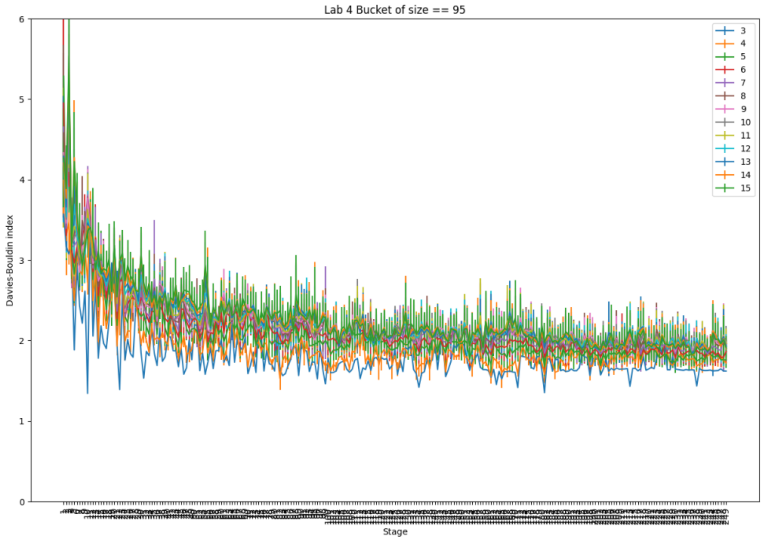

FIGURE B.10: Results from running with 95 buckets, stages versus the Davies-Bouldin index on lab assignment 4

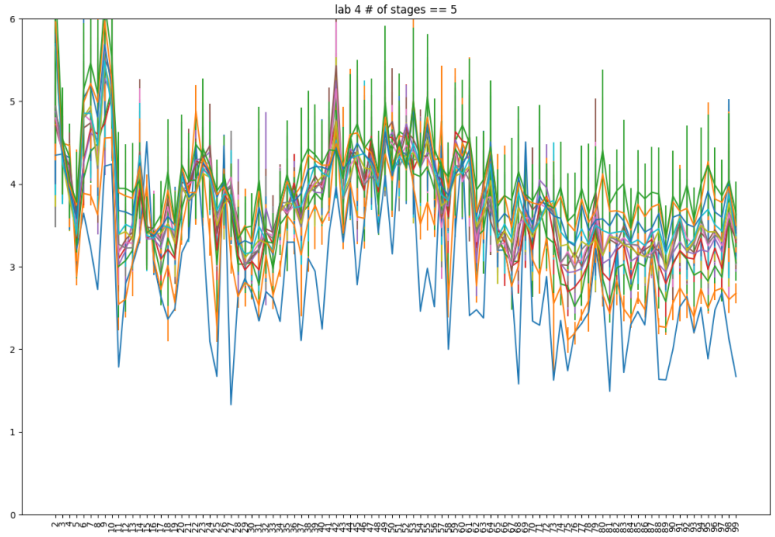

FIGURE B.11: Results from running with 5 stages, buckets versus the Davies-Bouldin index on lab assignment 4

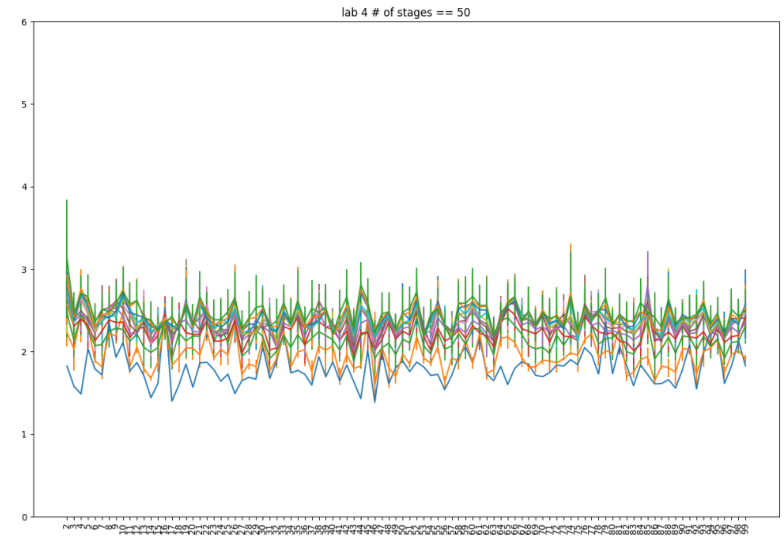

FIGURE B.12: Results from running with 50 stages, buckets versus the Davies-Bouldin index on lab assignment 4

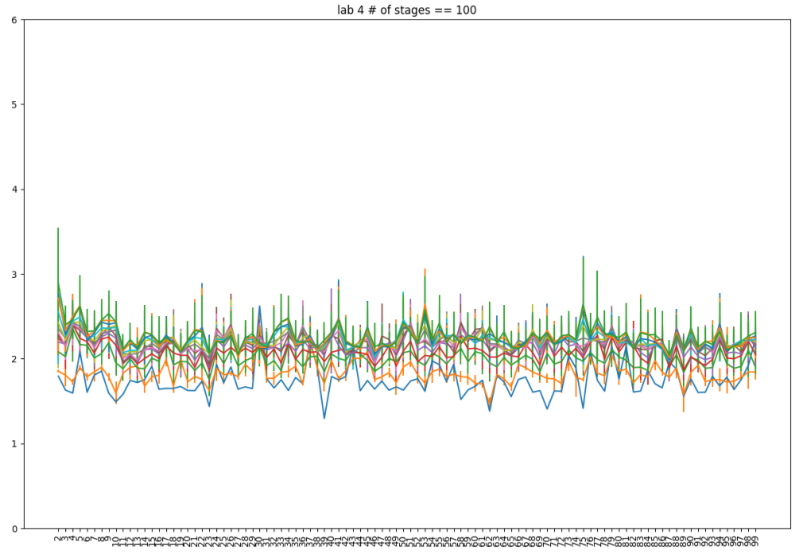

FIGURE B.13: Results from running with 100 stages, buckets versus the Davies-Bouldin index on lab assignment 4

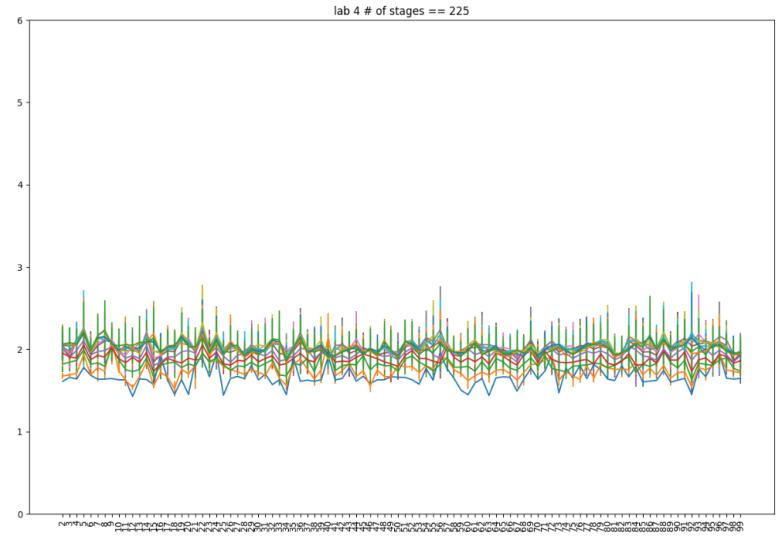

FIGURE B.14: Results from running with 225 stages, buckets versus the Davies-Bouldin index on lab assignment 4

Appendix C

Focus group transcription

C.1 Introduction

Par1: Alright, so the topic for this study was the learning analytics for teachers and I made use of the WebLab data from the intro to Python course from Par3 Mulder. The idea of the WebLab data was to make use of the revision history as it keeps track of that. And hopefully with this data we can gain additional insight into the performance of students in the solving of the assignments. During this session there will be three main parts, for each part I will give a quick explanation of what I have done. And afterwards there will be a question for which you have about 2-3 minutes to think about your opinion of it. And afterwards I would like to have a discussion about the entire part. It does not necessarily have to be about the specific question. Because the goal of this meeting is to have as many opinions as possible. So either remarks about the methods I used or improvements suggestion or ideas in general about how you would solve it. Good morning Par2. Just so you are aware, this meeting is audio recorded, so if you have any problems with that. No problems with that?

Par2: No that is okay, sorry that I am a little late

Par1: No problem.

Par3: Par1, are the questions mostly focused on like the software side of things, like we would like to see different features, something like that, or more the research side like have you validated your data or your conclusion or whatever.

Par1: Anything is valid, I have some on the methods I used. So what kind of indicators you would like, but also on what you would actually perform to have in the system.

Par3 : Alright

Par1: So I would like to have as many discussions as possible preferably between everyone present as I hope the different backgrounds work together to gain different viewpoints. So are there any questions up to this point, or can we get started. I see no further questions, so let's start with the first part.

C.2 Clustering

2:30 approx
Which is clustering, so the first part that I would like to focus on. Is to actually gather clusters of students who have come up with the same solution to the assignment or to the problem. To do this I followed the following steps. We first started with Locality sensitive hashing, so essentially splitting the solution of the student into very small parts. And hashing those into a set amount of bins. And if two of these solutions have similar bucket distributions well they are more likely to be similar. But with these bucket distributions you can also calculate the distances between those solutions which can be useful but it can also be useful to calculate the distances between a specific point, well to static points. So in this case I calculated the distances to for example the expert solution file. So the actual solution that Par3 has provided. But also the initial file that the students received before they start the assignment. When you have these solutions you can calculate distances between them and the distances between those points I used as features for the k means clustering algorithm. Well as for k means clustering you have to provide a k for how many clusters you want. But beforehand you are not sure how many different kind of solutions the students came up with. So therefore I used a k from 3-15 to make sure there is some flexibility on that end. And to get a theoretical point of which k would be the best I made use of the Davies-Bouldin index which essentially gives a score based on the inter and intra cluster distance. So, that is the first cluster sizes that will be displayed on the results. Which brings us to the question section.
What do you think of the proposed clustering solution methods or do you think it is appropriate or would you like to propose alternatives for it. For that I would like to give you about 2-3 minutes to think about... form an opinion about it.

Par4: can I ask a question right now?

Par1: Of course of course

Par4: So, if you are talking about you are splitting the solution into small parts and hashing the small parts right?

Par1: Yes

Par4: and then calculating similarity. So what is the granularity of this, is that a single line is that, what is the granularity

Par1: What I do is I compile it to bytecode and the sizes of it are also dependent on the size of the solution of the students. Because I ran some tests on the size that are needed, and about each stage, well each part is called a stage, about 200 seems to be optimal. Which splits the solution up into 200 parts.

Par4: Okay, so if you compile it to bytecode you abstract over things like identifiers right. If I name my variable X and you name it Y then, yeah okay.

Par5: And just to make sure the goal here is to determine clusters right? Not for the purpose of fraud detection. or both?

Par1: Yeah, it is for giving feedback, no fraud detection.

Par5: Okay okay,

Par2: You mentioned that the size of the chunk than depends on the size of the solution. Wouldn't this give problems in the comparison?

Par1: I would think so, in the results it turns out not to give any problems. Which I am very thankful for.

Par2: I find that very surprising, beacuse if you compare for instance, my solution contains the same parts as your solution except I wrote the last bit 50 lines longer. Then my chunks are also going to be bigger. And your hashes are going to be different right?

Par1: True true, but I believe that if you wrote your last part 50 lines bigger you have definitely a different solution than the other student.

Par2: Okay yeah.

Par2: So you say that works well according to your result. By what metric does this work well? So did you evaluate various solutions that your algorithm said are similar and said, yeah okay these are indeed similar. Like the similar idea or the thought process behind it. Or how did you evaluate this?

Par1: For the first 4 lab assignments I got the types of solutions from Par3. Then I looked at clusters that my algorithm gave and compared these solutions to the template solutions that Par3 gave and they matched, well near perfect.

Par2: So Par3 gave you multiple template solutions?

Par1: Yes

Par2: Okay. And these were based on what Par3 would expect students to come up with in alternative ways.

Par1: I believe they were the actual solutions the students came up with.

Par3: So, I am not sure what we are talking about now, but if you talk about the reference solution that I give on WebLab. That is just solutions that I think are correct. And these are mostly like, okay this is a solution with loop and this is a solution with list comprehensions for instance. And I did not actually consider student solutions in there. But all these solution were available to Par1. But as for this subjective reference thing. I am not sure if my reference solutions are the right thing. to calibrate the algorithm, maybe I missed some details in what you said.

Par1: Okay so we are about into the two minutes. Does anybody have any opinions that they would like to discuss?

Par5: So yeah, I wondered, for the k means clustering you mentioned distances, perhaps you mentioned it. How do you define these distances?

Par1: That is a very good point, so when we split it into these 200 stages we also have an x amount of buckets and if the two solutions you are to compare are in the same bucket you just place a 1 and otherwise a 0 and you get a string of length x and you just use the hamming distance to calculate the distances.

Par5: Right, right right, and they go into the same bucket again based on the bytecode?

Par1: Yes, so if the hashes are the same

Par5: Ah okay, still hash based. I see, okay okay. yeah and I was wondering if this is for the purpose of giving feedback and not fraud detection. I mean everything you are doing now. Probably also works very well for detecting fraud cases. For the purposes of giving feedback, should we not also look at the functionality of the code rather than only the syntax of the code or however you want to put it right WebLab has spectests and regular tests. Would the scores that the code get also be relevant in clustering similar solutions and therefor similar feedback together?

Par1: I believe they would, yes.

Par5: Do you think that could be fit in the current clustering that you have?

Par1: Well, maybe I could use, well to take these solutions so these test scores, and based on that try to take them together. So to merge the clusters, otherwise I don't see a way to actually place it into the current clustering algorithm

Par5: Right right. Okay

Par1: Anybody else? Alright, then maybe do you believe about a fixed amount of clusters do you believe the teachers should be able to give it beforehand so the amount of clusters you would like to see. Or do you believe more flexibility is more useful?

Par2: So I think if you look at an assignment, there is very different assignments so the assignments we have in the CPL course they take students much longer and there are much more individual parts then some of the practice assignments that we have in ADS which focus really on one algorithm and are also much shorter in their implementation. So I think for those it would make sense to give more different types of feedback for the bigger assignment. So I think it makes sense then to also increase the amount of clusters. Otherwise you would group people together without giving them all the right feedback.

Par3: And I would think, if I were to do this as a human I think I would not really start with a fixed set of clusters or a fixed number of clusters. I would just get the solution and say, okay this is my first one, and then consider the second one whether it is similar to the first one if it is than merge them together etc. And then I mean I might end up with 10 clusters, but I might also end up with 3. And I do not choose this beforehand. As I do not know what stuff I get that is similar. I am not saying

that your way is bad by the way. This could still be useful, if I say give me 5 clusters I could still get some kind of idea of the types of solutions that students have, but as a human I would do this differently.

Par5: I agree with Par3, and I would also wonder about the, at least for me for the courses that I grade in WebLab specifically, there are a handful of solutions that are just completely unique. Like they match nothing else they are just very out there, usually completely incorrect. That do not match any of the other well, and I wonder if you try to force those into a cluster whether that upsets the other results. So I am also not in favour of setting a fixed amount of clusters, but I would also be in favour of creating one group of which to say: Sorry but these solutions go nowhere, these are individual cases you would need to look at. Trying to force them into others would make my results worse.

Par3: I have seen a commercial solution for this, and there you were actually able to change the clusters afterwards. So they would be presented and they would have indeed some leftover category as Par5 suggested. And you would just be able to say, okay this belongs to that category, and then use that to train the algorithm. I mean that might be a bridge too far for your research, but you know that is what it did. There was a machine learning algorithm behind it that would be trained by the clustering that was suggested by the lecturer.

Par1: Either Par7 or Par6 have any opinions on this part?

Par7 : I would have just a comment, how would you expect teachers to know the number of clusters

Par1: Well in case of Par3 this would be possible as he has experience with previous years, how many he would expect in that sense. In that instance it might make sense to set a certain number of clusters. It might be useful in that case.

Par7: So I think this would be an argument against having the teacher set a fixed number of clusters.

Par2: So have you considered something where instead of setting the number of clusters we set the maximum difference between items in a cluster instead. So then well then you have some metric of similarity within your cluster algorithm. And if we say a cluster can only contain items of a certain distance apart then you automatically get some number of clusters and it is just well one number to tune which determines how unique and out there your items will be. Which also automatically creates the outlier cluster. And put them in separate clusters as these would not be able to fit within a cluster with other solutions.

Par1: That is indeed an excellent solution, I had not thought of that. I like that one.
Alright so then maybe we will move on to the second part.

C.3 Struggling students

17:13
Par1: The second part is about identifying students who might have more trouble with the assignment compared to the rest of the class. The 'trial-and-error programmers' as Par3 Mulder named them if I remember correctly. These are the students who are not trying to solve the assignment, but are trying to game WebLab to gain the full score.

To find these students I made use of the revision history of the students, firstly I looked at the amount of submission that the students made. And secondly I looked at the test score that the students got. For both of these I looked at the outliers and marked those students as potentially having had more trouble with the assignment.

The question for you than becomes, what indicators would you use to identify these struggling students? Take 2-3 minutes to think about this please.

Par4: But is it about finding struggling students or about finding students that are trying to game the system by just trying as much as possible?

Par1: Well basically the students that try to game the system, but those are the students who were struggling more, it is not about that they tried to game the system, it is about that they did not learn anything. Because that is the biggest problem. If they want to game the system just for their own sake, well that is their own loss, it should not be in that sense of course.

Par3: So from my perspective, because I more or less requested this feature . If you talk about intention, not many students have the intention of gaming the system. I mean that might be what they do, but it is a natural thing. Like you see a number on the screen, like 4/10 and you change something and it says 6/10 and you say. Okay I got a higher score so that is good, and you keep doing that until you get 10/10 and you move on to the next one. And well that is obviously detrimental, so from my point of view, you could either say they are not learning what they should learn or this behaviour is a predictor for their exam results. So if they do the formative assignments in the quarter in this way, than they will probably fail the exam, and I want to pick those studetns so that I can help them change their strategy so that they will learn something and pass their exam. But I saw Par5 disagreeing with my first point.

Par5: Well yes and no, so I do think that your course might be slightly different from some others for example in CPl the work is graded right. So is the student really trying to understand and therefore struggling to understand the material or are they only interested in getting that higher grade and therefore trying whatever they can throw at it. So I think there is a slight difference there, and therefore gaming the sytem I think yes, this will also identify students trying to game the system depending on the game setup.

Par3: So it actual has a dual purpose. Both of them, it could have a dual purpose, I am not entirely sure that this system is designed in such a way that you can reliabily identify students in both categories and at least both outcomes could be useful.

But you wanted us to think about how to identify these right?

Par1: Yes correct.

Par1: Most of you seem to have something in mind by the looks of it. Does anybody want to start?

Par6: If I may start here, so first of all I was wondering if you have calculated the ratio of the submissions and their points that they have got. So to that I would say, if a student has submitted like 10 times but then gains very very slow progress then be rated as a struggling student. And then, I mean, regarding the total points that we can also say: Okay if the students really can get them until their last submission they got a very low score this can also be regarded as a struggling student, but then indeed there will be some students who will try a lot of things out so, via the system. And I guess in that scenario, the ratio, so how much of the score they have obtained by the tryout that would be one indicator. And also I wonder like the number of submission the place that they got the highest score. From my own experience when you are playing with the online system. Even when you got the highest score then you might try a lot of things after that. And then you go back to the original solution which will give you the highest score.

Par1: I had not really thought about the second part, I was basically part of gaming the system, I just tried to gain the highest score and then quit the program as fast as possible. Yes I think it will be very useful to also look at the ratio in that sense.

Par7: I have an extra comment here, I was thinking something similar as to what Par6 was thinking about students that develop their solutions incrementally, I do not know if the system really works that way or you just submit your final solution. But I can imagine that it is a longer assignment, and I understand that is in the later stages of the course. Then you might build parts of the solution and just submit that and then building on top of that. But I am not sure if they get feedback directly or not.

Par1: They get the amount of specification tests that passed for each time they submit.

Par7: Okay, so this would be a case. And I think to look at this. And I think you also looked at final grade. or you mentioned that you also looked at the final grade and the number of revisions that they made. So I think that it might be interesting to look at the past data right if this hypothesis has some evidence for it.

Par1: For that I could make use of the midterm data that Par3 provided. So the students that I would try to identify then look at if they actually got a passing grade for the midterm or not.
For your first mention if students incrementally finish the assignment, so especially for the later assignments, it does not really matter if they do it incrementally. Because if it is such a large assignment there will not be many students who will finish it in one submission and immediately get the highest score. Then you just have a larger amount of submission for your mean point. So it is really about the deviation from the rest of the class.

Par7: Okay I see yeah.

Par3: But different students might have different strategies, I mean. I grew up in a time when software was super unreliable, so you had to press save every minute to make sure you kept your data. And I still do that, so I press save all the time. And if that creates a new revision in WebLab then in my case I would have a lot of revisions before I have a correct one. And someone else might just press save if they have written 20 lines of code. I mean, if you are looking at the deviation of the average, what average then? Is the average then a student who is between those two? Do they even exist?

Par1: That is a very good point so to that end I would also like to have multiple metrics, to make sure that the students that are identified are identified for a valid reason and not for a behavioural trade, such as saving every few seconds. If there would be more indicators that would be very welcome.

Par4: Did you look at something like time spend, like do you have the timestamps?

Par1: I have the timestamps, but the problem is if students took more than a day to solve the assignment. Than I have a bit of a problem

Par4: So I also worked on implementing this into WebLab for the future, but it is still in development. About calculating the amount of time that students spent on solving the exercises, the way that I do it now without extra logging is by looking at sessions essentially. So if submissions are within 5 minutes of one another you could say that they are within the same session, but if there is more than an hour in between or so, then you could say: Okay the student probably closed the browser or whatever and start it again an hour later. This way you can calculate how many sessions and how much time the student has spent each session and you can sum that up to have a total time spent.

Par1: That is also a very valid strategy.

Par2: Something coming back to the point that Par3 made, you could consider looking also at which submissions also have a spectest run submitted to it. Because even when I save every 5 seconds I will not click the run spec test button every 5 seconds. Because I will have typed code which I know would not compile. So well, it is not a perfect solution of course, because the same problem because some students will click the run spec test button all the time while others write the whole piece and then click the spec test button, but maybe by combining these two points you get a more accurate revision history which like is more similar between students.

Par1: Actually I do not receive the information if the students actually press the specification test. This is basically what I receive, A list of all the students and in that there is just the final solution their current test cases and per revision just every time they saved, not just the times they ran the specification test or not. That is a shame, but maybe it can be added in WebLab that it actually outputs that. Because I believe the person who helped me get this data can also get more data in it.

Par2: So it is shown in the UI so it defenitely has this data, so I think it will be better off adjusting the export.

Par4: Jeah, we have that data I can ask Elmer, was it Elmer you went to for the data?

Par1: Jeah

Par4: Okay, I can ask Elmer to include, because now you also do not have the amount of spec test that passed for each revision.

Par1: That is correct, I rerun every specification test for each revision.

Par4: Aha, I see, okay.

Par7: Okay if you are on that topic, I also have a bit of a crazy idea, I do not know if it will work or not. But thinking about student behaviour and the fact that they have this trial and error thing. It makes me think that they are probably not writing their code intentionally to pass some spec tests, so I think what does it tell us when in one revision the spec test pass and in the following revision the same spec test does not pass anymore, because the student change the code in such a way, not to destroy their solution. Somehow this will tell me that they are not aware, probably where in their code or which piece of their code might be connected to some spec test. again, might depend on how or what type feedback they get when they run their feedback.

Par1: Well the first thing I think of when I such statement as they go from not passing to passing to not passing again is fraud, but I promised Par3 that I would not look into that.

Par7: Why would you say it is fraud?

Par1: Let's just say there are some app groups which share solutions that then are placed into the editor. And then they are like, well this passes and then they go back to their own solution to try and figure out which part actually did not work.

Par2: Well, but I think you are also ignoring a large group of typical behaviour. which is, let's say, we write a solution, then we run spec test and we get 0. Because there was some mistake, then we remove the mistake and run it again and we get 30 points then we try to implement another part and we run it again but we broke something else so we get 19 then after that we fix our mistake in breaking the other thing and then we get 31. After that a compile error again, so 0. 10, 32, so you see this up and down on a normal student which is normal behaviour, but there is a slowly increasing line for a student. Which might suddenly jump up quite a bit, but there is an increase for the most part and these students are not necessarily committing fraud.

Par1: True true, which is why I promised Par3 not to look into it, as I do not have a valid way to determine whether it actually is. It is just a hunch I have.

Par3: I do think Par7 is onto something, because, it also depends on the quality of the spec tests ofcourse, but let's assume the spec tests are good enough. Then I think, a student that works with intention, at least with the type of exercise they have in my course. Then usually their spec test will strictly increase. or at least stay the same or increase and usually not decrease. I might be wrong, but at least it is something to look into, so see what, just look into the solutions where you see a decrease in spectest score halfway, and see whether those students are part of the group that we are looking for.

Par7: i actually really liked Par2's description how the score actually fluctuates but overall increases, so maybe that is one indicator, well it is not an indicator, jeah maybe it is an indicator that you could look at if really there is something there.

Par3: But what are you looking for then, what kind of indicator, like do you calculate a slope or does it increase fast enough.

Par7: yeah that would be, I would not start with calculating slope or anything like that, but building these graphs maybe and explore this as a human maybe, instead of reducing it to one number.

Par2: i think that is also one of the potential problems with the data currently reported by WebLab. What you really would like to know would be the particular test that the students passes or fails right in each iteration, you may have these as you rerun every solution on every test. on every revision. But at the moment we have this whole grading system which reduces everything to a single number, which is the grade, and that is what you apply your algorithms on. But I think it would be potentially more valuable if you apply your machine learning directly on the individual tests and see how these change over time.

Par1: Alright then we will move on the last part.

C.4 Visualization

36:31

Par1: Which is a visualization based approach, well because if you have your information you also want to show it in such a way that the teacher can get some use out of it. So there are three things that we would like to visualize, the first one is of course the clusters which we talked about first and secondly is the students with a higher likelihood of having struggled with the assignment, so either based on the test score or the submission count or any of the other indicators that we just discussed. I also would like to have the most difficult problem that the students encountered, so what is the core problem the student struggled with per cluster. And my question would actually be, what would your visualization look like before I show you mine, to prevent ideas blocking. So I give you about two minutes to think about that one.

Par5: I am a bit confused now I think, I thought the clustering was on a per exercise basis, so for different exercises we would have different clusters. So what is the

last bullet point than here?

Par1: I have the clusters of each student and I want to know what the problem was within the assignment that the student struggled with the most, which test question basically.

Par5: right right, yeah so maybe the answer could actually be like the for loop iterating over the items something like this.

Par3: But is that what you identify? like the problem in the code or are you just saying spectest number 5. fails the most.

Par1: Currently the spec test but is also part which I think the specification test could adress, so, this is a bit of a problem between CPL and the python introduction course. So in CPL you would actually want to assign a grade so when you successfully complete the core part of the assignment you actually want a passing grade and the edge cases are single points so to say. But for the Python part if you actually want feedback it might actually be more useful to have one spectest for each learning point so to say. So when a point then keeps failing, so each testcase know which part the student is struggling with the most. but that is a thing of designing the specification test.

Par2: Also not always possible, cause test are just input output samples, which may not actually catch a particular problem.

Par1: Alright, does anybody have any ideas that they would like to share?

Par2: So one of the main things that I would be interested in, so lets say that your clusters are good representations of types of solutions then I would like to see a representative of each cluster, that is a solution that shows me, or part of a solution that shows me relevant part for this particular cluster.

Par1: Very good point indeed, yes.

Par3: And with regard to the second point, I am not sure what idea you have right now, but I would prefer not to have a hard cutoff point where someone is either a struggler or she is not. Instead I would like to have a likelihood, maybe I would be able to see that likelihood. And then order by the likelihood or something, and then I can decide what to do with those students, maybe send an email to the top ten or something.

Par6: in addition to Par3s remark, I would also like to see the history of the submission, I mean with the line chart or a trend visualization. That you can show, with the likelihood, for example the number of specification tests passed for each submission, than you have the dynamics for each submission, and then you will see for like the course or the instructor than can see how is the progression line for the students.

Par1: Very good points, it makes me a bit scared of actually showing my visualization. Any other ideas, or should I show my own visualization?
That is this one. So, for each column I have created the clusters, with each student number being the actual student. I have highlighted the struggling student, so it is

not a gradient so to say, which a T if it is testscore based, or an s if it is submission count based. So, what I immediately see, well here in the top I presented what is the most failed testcase for each cluster. there can be multiple, but in this case it is singular. If it states that there is not a clear test question it means that there are at least 8 testcases with the same amount of failures. So if for the left most columns it means that the students have not tried to solve the problem they are the initial submission, this column here these are the students who passed the course last year they just pasted the solution from last year in it and they only have passing test cases. So in that sense there is no failing test cases. And I think that I can show you 1, every student number is a link to the WebLab page directly, so I do not have a preview of the solution. But I can show you, for example, what Par3 has done in his submission, he basically pasted the solution in it. So that is basically, oh yeah, I can change the cluster sizes from 3 to 15 as described earlier.

Par7: Can you please explain again the T and the S.

Par1: In the second part i tried to differentiate students who might have struggled based on test score and on submission count, so the T is when students have very below average test scores, in this case it is mainly 0 but also some three to fours, as there are 18 test cases. And the S is then for the submission count.

Par3: Is every box a cluster

Par1: Yes

Par3: So I mean, it is identified with the test case. So what is the testcase that is identified on top.

Par1: That is the most failed test case for that cluster. The cluster is created beforehand and then afterwards it is looking at which test question failed most likely.

Par3: If I wanted to use this to just see the different solutions, I just click the first of every cluster and get the impression that I want?

Par5: That confuses me because, Par3s solution, which you say is just a copy paste of the correct answer is in a cluster which seems to indicate that it is doing something wrong, or am I misreading this now.

Par1: Because it is in the same cluster as the actual solution, all of these solutions have basically the same solution as the actual solution file. but during the solving of the solution file, most students made a mistake with this test question. Because it is also counting the revision history.

Par5: So in that sense opening the first one might not necessarily give you the right impression immediately, for the first two solutions, what you might see in WebLab now would be identical. It is just that on the road there, there were differences right?

Par1: yes, so the clusters are for the final solution. So the solution they finally came up with. And the testcase is what they struggled the most with while solving the assignment. So you will not see that problem when opening the first of each

cluster indeed.

Par5: right right, so if you were to open the first one of each cluster you would see very different pieces of code, yeah okay.

Par3: My use case for the clustering case would be: The students have done some assignments beforehand and I would like to give them some feedback during the seminar. So there I only care about the final solution and not about the revision history. But I must say, this visualization, anything I can use for my course I am of course happy with, so don't take the criticism too personal, but it seems you tried to combine all the things you found out with your research into one overview. i was actually expecting one overview with the clusters and then not the student numbers that I care about, but one with an example of that cluster. And then for the struggling thing I just get an ordered list of likelihood that students are struggling, the testcase that failed the most. I am not sure how to visualize that. But these are all different cases, and combining them all into one overview confuses me a little.

Par2: I agree with those points.

Par5: I think I agree, and I think the thing that confuses me most is that now the testcase that fails most often now seems to be title of the cluster. I looked more closely now, and I see that two of them also have the same testcase. So perhaps, this would take some confusion away actually, because that is actually not what it is based on. But yeah, but I would at least put that in a different view.

Par2: you could also consider just naming the clusters something, and then at the bottom of the cluster have a more detailed description of why this was a cluster. So you could say, all these students on their way to the final solution failed a particular testcase. Or, stuff like that.

Par3: But that is not what the clustering was based on right?

Par1: no, currently it is only on the final solution.

Par3: I mean in a commercial solution I have seen before you could name the clusters, just give the names yourself as a lecturer. So it showed the code, it was the primary view, you saw pieces of code listed under each other. And then you could give it a name if you want. And otherwise it would just call it cluster 1 cluster 2 etc. I would go through them in order anyway when I discussed them.

Par5: This makes a lot of sense to me, because what you hope to see is when you open examples is that you see that there is one cluster that did it nicely with a for loop and another one that did it nicely with a list comprehension, and there is another one that used this. So hopefully when you see 4 or 5 examples you could see like okay this is this cluster. And giving them a shot descriptive name yourself, yeah I would like that.

Par3: It does remind me that I made a mistake earlier in this session, because I did give Par1 the clusters indeed that was manually picked. So what I did last year, I created an export of all the solutions, and with some scripts I concatenated them all into one file. And then I clustered them manually. And I send those clusters to Par1.

So these clusters were not based on the reference solutions but they were based on manual effort to to create clusters om similar solutions.

Par5: You created a nice training set.

Par2: I was already thinking that based on the description. But Yeah, coming back to that representative, you could for example take, your clusters are made up of multiple features, but outliers in multiple directions and also a central points, because that should give you essentially the variance within a clusters, so the code blocks essentially.

Par3: Yeah you could essentially also do something with the colors right, always be careful with colors, because of colorblind people of course. But you could at least mark the stuff that is similar in almost all the solutions and then maybe some lines of code are different but you still cluster them together because 10 lines are always the same. You could mark those 10 lines, if you wanted to.

Par1: Alright, in terms of time. Does anybody have any final remarks in general, so about the project as a whole or the session itself?

Par7: I have something, overall I think it is quite a ambitious project, so congratulations on all the work you have done so far. I think it is quite cool. A suggestion I had before but forgot to mention was when talking about indicators to identify struggling students probably you have done this before, there might be literature that describes behaviour of struggling students in beginning programming. So you probably can take some inspiration from there to look into what indicators to look for. in this data.

Par1: I have indeed looked at it, but the main point they basically come up with is the amount of submissions not taking into account, sometimes in the literature they say that a submission is when a student sends an email with their submission to the teacher and not online just continuously saving it. That is a bit op discrepancy there.
Any other final remarks?

Par4: Yeah, I would like to say thank you for hosting this session, because we are also looking to implement learning analytics in WebLab and this also helps me do my job better.

Par3: yes, and I am looking forward to using this actually in my next run of my python course.

Par2: Yeah I think it would be very nice, even if it is rough around the edges it is nice to get a start on this to be able to gain these additional insights.

Par3: Yeah, it will save me a lot of time with the manual clustering.

Par2: Will you do that every year Par3 or?

Par3: No this year was the first time I did it. I did not do it for all the weeks, I think I did it for the first 4 weeks. that I gave up. Then I just showed the reference

solution. So I did it once and I thought it was useful. And maybe it was not very precise, but at least I could just scroll through all the solutions and vaguely conclude what the clusters were. But having a system for this will be a huge improvement.

Par1: Then I would like to thank you all for taking time out of your day for this session and I will wish you all a good week. Thank you

Bibliography

[1] J. A. Reyes, "The skinny on big data in education: Learning analytics simplified", *TechTrends*, vol. 59, no. 2, pp. 75–80, Jan. 2015. DOI: 10.1007/s11528-015-0842-1. [Online]. Available: https://doi.org/10.1007/s11528-015-0842-1.

[2] P. Leitner, M. Khalil, and M. Ebner, "Learning analytics in higher education—a literature review", in *Learning Analytics: Fundaments, Applications, and Trends*, Springer International Publishing, 2017, pp. 1–23. DOI: 10.1007/978-3-319-52977-6_1. [Online]. Available: https://doi.org/10.1007/978-3-319-52977-6_1.

[3] K. Mangaroska and M. Giannakos, "Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning", *IEEE Transactions on Learning Technologies*, vol. 12, no. 4, pp. 516–534, 2019. DOI: 10.1109/TLT.2018.2868673.

[4] C. Herodotou, B. Rienties, A. Boroowa, Z. Zdrahal, M. Hlosta, and G. Naydenova, "Implementing predictive learning analytics on a large scale: The teacher's perspective", in *Proceedings of the Seventh International Learning Analytics amp; Knowledge Conference*, ser. LAK '17, Vancouver, British Columbia, Canada: Association for Computing Machinery, 2017, 267–271, ISBN: 9781450348706. DOI: 10.1145/3027385.3027397. [Online]. Available: https://doi.org/10.1145/3027385.3027397.

[5] E. L. Glassman, J. Scott, R. Singh, P. J. Guo, and R. C. Miller, "Overcode: Visualizing variation in student solutions to programming problems at scale", *ACM Trans. Comput.-Hum. Interact.*, vol. 22, no. 2, Mar. 2015, ISSN: 1073-0516. DOI: 10.1145/2699751. [Online]. Available: https://doi.org/10.1145/2699751.

[6] X. Fu, A. Shimada, H. Ogata, Y. Taniguchi, and D. Suehiro, "Real-time learning analytics for c programming language courses", in *Proceedings of the Seventh International Learning Analytics amp; Knowledge Conference*, ser. LAK '17, Vancouver, British Columbia, Canada: Association for Computing Machinery, 2017, 280–288, ISBN: 9781450348706. DOI: 10.1145/3027385.3027407. [Online]. Available: https://doi.org/10.1145/3027385.3027407.

[7] P. Ihantola, A. Vihavainen, A. Ahadi, M. Butler, J. Börstler, S. H. Edwards, E. Isohanni, A. Korhonen, A. Petersen, K. Rivers, M. A. Rubio, J. Sheard, B. Skupas, J. Spacco, C. Szabo, and D. Toll, "Educational data mining and learning analytics in programming: Literature review and case studies", in *Proceedings of the 2015 ITiCSE on Working Group Reports*, ser. ITICSE-WGR '15, Vilnius, Lithuania: Association for Computing Machinery, 2015, 41–63, ISBN: 9781450341462. DOI: 10.1145/2858796.2858798. [Online]. Available: https://doi.org/10.1145/2858796.2858798.

[8] P. Blikstein, M. Worsley, C. Piech, M. Sahami, S. Cooper, and D. Koller, "Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming", *Journal of the Learning Sciences*, vol. 23, no. 4, pp. 561–599, 2014. DOI: 10.1080/10508406.2014.954750. eprint: https://doi.org/10.1080/10508406.2014.954750. [Online]. Available: https://doi.org/10.1080/10508406.2014.954750.

[9] B. Cui, J. Li, T. Guo, J. Wang, and D. Ma, "Code comparison system based on abstract syntax tree", in *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, 2010, pp. 668–673. DOI: 10.1109/ICBNMT.2010.5705174.

[10] H. Thimbleby, "A review of donald c. lindsay's text file difference utility, diff", English, vol. 32, no. 6, pp. 752+, 1989, Article, ISSN: 00010782.

[11] X. Xu, C. Liu, Q. Feng, H. Yin, L. Song, and D. Song, "Neural network-based graph embedding for cross-platform binary code similarity detection", in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17, New York, NY, USA: Association for Computing Machinery, 2017, 363–376, ISBN: 9781450349468. DOI: 10.1145/3133956.3134018. [Online]. Available: https://doi.org/10.1145/3133956.3134018.

[12] S. Wang and D. Wu, "In-memory fuzzing for binary code similarity analysis", in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2017, pp. 319–330. DOI: 10.1109/ASE.2017.8115645.

[13] Z. Đurić and D. Gašević, "A source code similarity system for plagiarism detection", *The Computer Journal*, vol. 56, no. 1, pp. 70–86, Mar. 2012, ISSN: 0010-4620. DOI: 10.1093/comjnl/bxs018. eprint: https://academic.oup.com/comjnl/article-pdf/56/1/70/1419281/bxs018.pdf. [Online]. Available: https://doi.org/10.1093/comjnl/bxs018.

[14] Liuliu Huang, Shumin Shi, and Heyan Huang, "A new method for code similarity detection", in *2010 IEEE International Conference on Progress in Informatics and Computing*, vol. 2, 2010, pp. 1015–1018. DOI: 10.1109/PIC.2010.5687856.

[15] M. Novak, "Review of source-code plagiarism detection in academia", in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2016, pp. 796–801. DOI: 10.1109/MIPRO.2016.7522248.

[16] M. Novak, M. Joy, and D. Kermek, "Source-code similarity detection and detection tools used in academia: A systematic review", *ACM Trans. Comput. Educ.*, vol. 19, no. 3, May 2019. DOI: 10.1145/3313290. [Online]. Available: https://doi.org/10.1145/3313290.

[17] C. Ragkhitwetsagul, J. Krinke, and D. Clark, "A comparison of code similarity analysers", *Empirical Software Engineering*, vol. 23, no. 4, pp. 2464–2519, 2018, ISSN: 1573-7616. DOI: 10.1007/s10664-017-9564-7. [Online]. Available: https://doi.org/10.1007/s10664-017-9564-7.

[18] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", *Soviet Physics Doklady*, vol. 10, p. 707, Feb. 1966.

[19] E. F. KRAUSE, "Taxicab geometry", *The Mathematics Teacher*, vol. 66, no. 8, pp. 695–706, 1973, ISSN: 00255769. [Online]. Available: http://www.jstor.org/stable/27959476.

[20] R. W. Hamming, "Error detecting and error correcting codes", *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[21] M. A. Jaro, "Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida", *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 414–420, 1989. DOI: 10.1080/01621459.1989.10478785.

[22] A. Singhal *et al.*, "Modern information retrieval: A brief overview", *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.

[23] K. Zhao, H. Lu, and J. Mei, "Locality preserving hashing", in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI'14, Québec City, Québec, Canada: AAAI Press, 2014, 2874–2880.

[24] S. C. Johnson, "Hierarchical clustering schemes", *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967, ISSN: 1860-0980. DOI: 10.1007/BF02289588. [Online]. Available: https://doi.org/10.1007/BF02289588.

[25] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets", *Data Mining and Knowledge Discovery*, vol. 10, no. 2, pp. 141–168, 2005, ISSN: 1573-756X. DOI: 10.1007/s10618-005-0361-3. [Online]. Available: https://doi.org/10.1007/s10618-005-0361-3.

[26] M. Embrechts, C. Gatti, J. Linton, and B. Roysam, "Hierarchical clustering for large data sets", in. Jan. 2013, vol. 410, pp. 197–233, ISBN: 978-3-642-28695-7. DOI: 10.1007/978-3-642-28696-4_8.

[27] C. Watson, F. W. B. Li, and J. L. Godwin, "Predicting performance in an introductory programming course by logging and analyzing student programming behavior", in *2013 IEEE 13th International Conference on Advanced Learning Technologies*, 2013, pp. 319–323. DOI: 10.1109/ICALT.2013.99.

[28] J. P. Munson and J. P. Zitovsky, "Models for early identification of struggling novice programmers", in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18, Baltimore, Maryland, USA: Association for Computing Machinery, 2018, 699–704, ISBN: 9781450351034. DOI: 10.1145/3159450.3159476. [Online]. Available: https://doi.org/10.1145/3159450.3159476.

[29] A. S. Carter, C. D. Hundhausen, and O. Adesope, "The normalized programming state model: Predicting student performance in computing courses based on programming behavior", in *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ser. ICER '15, Omaha, Nebraska, USA: Association for Computing Machinery, 2015, 141–150, ISBN: 9781450336307. DOI: 10.1145/2787622.2787710. [Online]. Available: https://doi.org/10.1145/2787622.2787710.

[30] ANTLR. [Online]. Available: https://www.antlr.org/.

[31] N. Batchelder.

[32] K. Struct. [Online]. Available: https://formats.kaitai.io/python_pyc_27/index.html.

[33] D. L. Davies and D. W. Bouldin, "A cluster separation measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979. DOI: 10.1109/TPAMI.1979.4766909.