# Predictive Learning Analytics

Bachelor End Project 2018-2019

Murtadha Al Nahadi
Robin Faber
Frank Ooijevaar
Wessel Turk

Technische Universiteit Delft

TU Delft
Delft University of Technology

Challenge the future

# Predictive Learning Analytics

## Bachelor End Project 2018-2019

by

**Murtadha Al Nahadi**
**Robin Faber**
**Frank Ooijevaar**
**Wessel Turk**

in partial fulfillment of the requirements for the degree of

**Bachelor of Science**
in Computer Science

at the Delft University of Technology,

| | | |
|---|---|---|
| Supervisors: | Dr. N. Yorke-Smith | TU Delft |
| | Ir. J. Verdoorn | FeedbackFruits |
| | B. Hintemann | FeedbackFruits |
| Bachelor Project Coordinators: | Ir. O.W. Visser | TU Delft |
| | Dr. ir. H. Wang | TU Delft |

# Preface

FeedbackFruits is a company inspired to help teachers shape learning activities and spark students' active thinking, by creating an online platform on which teachers can create interactive learning activities. This report concludes the project 'Predictive Learning Analytics' for FeedbackFruits. It includes all the work that has been done in a ten week span at FeedbackFruits. The goal for this project was to extend the analytics tools for interactive presentations that are offered to teachers who use the platform. This hopefully results in giving them more insight in the participation of the class, understanding of the course material and the effort that students put in their school work.

# Summary

This report includes the design and implementation of analytics tools that can help teachers monitor the participation of students in their course and if students are at risk of failing a course. The goal of this project is to extend the interactive presentations tool offered by FeedbackFruits with a new set of analytics. The analytics that are added are both standard analytics as well as predictive analytics.

For the design of the code, the existing code for standard analytics for assignments was taken as an example. This was done such that we did not have to figure out the whole system, but we could take inspiration from existing code and adjust it to work for the presentations. The overall design of the standard and predictive analytics is almost identical. This process consists of obtaining the data needed for the analytic, processing this data, creating the insight, storing it in the database and displaying it in the user interface. The main difference in this process between the predictive and standard analytics is that the predictive analytics use a predictive algorithm written in Python to generate the information from which an insight is created.

The implementation consists of two major parts that operate separately, the front-end and the back-end. In both of the parts, a lot of work had to be done to be able to add all the features that were planned. In the back-end, the whole database structure for presentations and broadcasts had to be added in order to access the data that was needed for the analytics. When this was done, the analytics were implemented in the same way as the analytics that already existed for other features like assignments. This was done by first getting the data from the database, then processing it to extract the necessary information and then storing this information in an insight. In the front-end, the main focus was the user interface in which the analytics are shown to the teacher. The analytics that were planned had to be shown in bar and line graphs, both of which were not implemented yet. The rest of the user interface was less work because there was already a lot of features that could be used for our analytics by making minor changes, i.e. the view in which the analytics are shown.

In the final product, all the planned standard analytics were implemented. For the developers at FeedbackFruits, it is easy to add new analytics and for the user they are easy to understand. Only one of the predictive analytics was implemented. This was mostly due to the fact that there was not enough data to train the model, making it hard to let the model make accurate predictions. However, all the features needed to train such a model and create analytics based on these models are implemented, so FeedbackFruits could easily add other predictive analytics in a later stage when enough data is available.

Overall, the project was a success. Most features that were planned are implemented, so the usefulness of the presentations for teachers is improved. Teachers can now easily see what questions are hard, what slides are difficult to understand and how students participated in the presentation, among other things. This gives the teacher a more detailed overview than before.

# List of Figures

# Contents

# 1 Introduction

When it comes to teaching, it is not just important what knowledge you are trying to share with others. How you share this knowledge is just as important if you want to be successful as a teacher. Teaching is not just explaining, telling a story or giving a step by step plan on how to address problems [1]. It is also about getting the students involved in the process of sharing information, making him or her actively look for answers or think of solutions. It is a process in which both sides are actively involved.

FeedbackFruits (FbF) is a company that creates tools that can help teachers shape learning activities that spark students' active thinking. Their tools allow teachers to create interactive learning activities where students participate more in the learning process, not only with the teacher but also with each other. The tools also allow the teacher to monitor how students are doing, by providing a wide range of data about the students and their participation in class.

Although many analytics about activities in and around the class are already available to teachers, there is still a lot of potential to improve these analytics. Big parts of the stored data are not used yet, or are not shown to teachers. This information could help teachers understand how students are doing on their course, and if the material that they explain is understood.

During the ten weeks of the Bachelor End Project 'Predictive Learning Analytics', we worked together with FbF with the goal to extend their standard analytics tools and create predictive learning analytics for teachers. The scope of these analytics is limited to the interactive presentations tool of FbF, because there were hardly any analytics for this feature yet. We used the data that FbF already collects from its users as well as some data that is not actively collected yet, and created new analytics from this to create insights that are useful for teachers. There were no predetermined requirements for this project. We were free to determine what we thought would be a proper project goal with requirements. In this report we documented the progress and decisions that we made during development of the product. It includes the design and implementation of the product, as well as the research report and project plan that were created before starting development.

Chapter 2 explains what is needed to deliver during the ten weeks of the project, both the product and all the documentation that comes with this. In Chapter 3, the design of the product is discussed and in Chapter 4 the implementation of the product is explained. These chapters explain what design choices were made and how we came to these choices. The implementation of the product is also explained in detail, explaining what algorithms are used and how analytics are created from the raw data that is generated as users use the product in class. Chapter 5 includes all functionality of the final product that was developed in this project. Chapter 6 describes the process of doing research and developing the product. This includes problems that we ran into during development and alterations that were made to the requirements as the product developed. In Chapter 7, a reflection of the project is done. Here we discuss everything we did in this project and reflect on our work. Finally, Chapter 8 gives the conclusion of the project.

# 2

# Project description

This chapter describes the goal of the Bachelor End Project (BEP) 'Predictive learning analytics'. Section 2.1 describes the product that is created during this project. Section 2.2 explains how work was divided among team members in order to realize the product. In the last section, 2.3, we list all documentation that is written to ensure that all requirements for the product are met and the techniques that are used to create the product.

## 2.1. Product

The product that has been developed is an extension of the interactive presentations tool that is provided by FbF and is supposed to help teachers improve the quality of their courses. The extension will give the teachers insights about the progress that the students are making within the interactive presentations that the teachers host during their lectures.

The interactive presentations tool that FbF created is different from a standard presentation in that it allows the student to move through the presentation in his or her own pace and interact with the teacher and other students by starting discussions on parts of the presentation. A traditional presentation involves a teacher showing slides and a class of students following this presentation. This is a very passive process in which the student is not encouraged to participate in discussions or do critical thinking on what is explained, reducing the effectiveness of the presentation. FbF solved this by creating an interactive presentations tool. This is an online presentation that all students can join, allowing them to follow the presentation on their own laptop and view slides other than those that the teacher is currently explaining. Students can start a discussion on a slide when the information on the slide is incorrect or difficult to understand. The teacher can ask open and multiple choice questions in between slides, allowing him or her to get an idea of how well students understand the material. This increases the involvement of the student in the lecture, improving the quality of the classes and makes the presentations more effective to students.

**The goal of this project was to create a set of analytics for the interactive presentations tool that teachers can use to get a better insight in the participation of students and how well they are performing in class. These analytics already exist for other tools that FbF provides, but only very basic information is shown about the interactive presentations so far. Two types of analytics are created for interactive presentations, standard analytics and predictive analytics. Standard analytics give an insight in how students are doing in lectures so far. Predictive analytics use the information on how students are doing so far to make a prediction on how they will be doing in the future. Both analytics can help the teacher get a better understanding on the quality of their course and the effectiveness of lectures that are given.**

## 2.2. Work

The idea behind the BEP is to use the knowledge and skills that students have acquired in the three years of studying Computer Science & Engineering in a real world project, where each step of the process is documented. The project was carried out at the office of FbF in Amsterdam. All the functionality that was added to their existing product was created in their code databases. We adapted to the work style that the employees at FbF handled. This meant using the agile development technique Scrum, more on this is explained in Chapter 6.2.

## 2.3. Documentation

When this project was started, the first step that was taken was to create a project plan. In the project plan, all information regarding the product is documented. This includes a problem statement, a project goal and a list of requirements that the final product must have. The list of requirements also includes optional requirements that would further enhance the usefulness of the product but are not necessarily needed to call the product done. The project plan also states how the quality of the final product is assured, i.e., the functionality, maintainability and integrability of the product. The full project plan can be read in Appendix A.

Together with the project plan, a research report was created. Before making decisions on the implementation of the product, extensive research was done to find what techniques and methods are available in the field of creating analytics to ensure that the product will function optimally. The research gives insight in the important aspects of creating analytics, challenges that can occur when creating analytics and possible techniques that are used to create analytics. The full research report can be read in Appendix B.

# 3
# Design

In this chapter, the design considerations and choices made during the project will be discussed. We implemented multiple learning analytics for interactive presentations, similar to analytics that FbF already has available for other tools that they offer to the teachers. Since the analytics already exist for other tools, we could use the existing design of these tools when creating the new analytics, thus keeping the overall design coherent. We extended the code for the presentations using the same structure as used for the other tools, making the code easy to understand and maintain for the developers at FbF. In the following sections, the overall view of the different design aspects of our implementation will be discussed. In Section 3.1 it is explained what streams are and how they are used to create insights. Then, in Section 3.2 it is explained how the predictive insights work. Finally, in Section 3.3 the different aspects of our user interface (UI) will be discussed. The visualization of the architecture design can be found in Figure F.2.

## 3.1. Standard insights

The first step of creating analytics happens in the back-end, which is the part of the application that the users never see. In the back-end, information for an analytic is collected by requesting data from the database and performing operations on it. In order to obtain the data from the database, a query is created in which is specified what exact data from the database is requested. The data is returned by the database, after which it can be processed to create analytics. Processing is needed since the data in raw format can not be displayed in the front-end, also this data would not give the teacher a useful insight in the presentation.

When the data is processed to the right format, an insight is created containing the needed information and it is stored in the database. In the front-end the insights are requested and displayed to the user. Both the stream and insight classes were already implemented for other tools, which meant that we did not have to implement this system ourselves. We again adhered to the structure that the developers at FbF used to implement the analytics for other tools, making it easier for them to maintain the analytics for the presentations.

## 3.2. Predictive insights

Compared to the standard insights, creating predictive insights is a more complex task. The first step is the same as with the standard analytics; query data from the database and pre-process this for use. Unlike the standard analytics however, information is not drawn from the processed data itself. The data is instead fed to an algorithm that makes predictions based on this data. From the results of this predictive algorithm, the predictive insights are created.

The predictive algorithm that is used is a random forest. The reason we chose to use random forests is because of the explainability of this algorithm, a more detailed explanation of this choice can be found in Appendix B. This algorithm is a collection of decision trees. Each decision tree makes a prediction based on the data that is has been given and by looking at the prediction of all decision trees, the random forest algorithm makes a decision on how to classify the data.

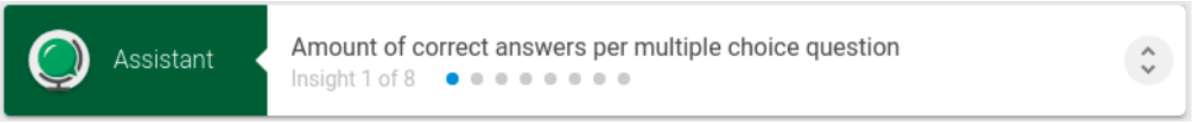
The predictive model needs to be trained in order to make predictions on data. The idea of the algorithm is as follows: the algorithm is given a large set of data. It processes this data and makes some prediction on what the output for this data should be. The actual output that corresponds to the data is already known. After making a prediction, the model compares the prediction it made with the actual outcome for the data and based on the difference between the predicted outcome and the actual outcome, the model adjusts itself to make more accurate predictions on the next run. In order for this to work accurately, a sufficiently large set of data is needed to train on. If the training data is not sufficiently large and diverge, the algorithm will be overfitted on the training data. This means that the model learns to make predictions based on a pattern it found in the training data, but this pattern is a simplified version of the pattern that appears in the real world data. Because of the lack of data from real world examples the algorithm assumes that the simplified pattern works. This results in the algorithm predicting for the training set with very high accuracy, but scoring poorly on real world data.

## 3.3. User Interface

The UI is where the analytics are displayed to the user. It is the front-end of the application, which is the part that the user actually interacts with when he or she is using the website. In order to give a clear overview of the analytics, a new section was added in the UI to display the information in. For the standard analytics, the information is displayed using bar and line graphs. These two were chosen because they are the most intuitive to understand for the average user and allow a lot of information to be displayed in a clear way. Furthermore, they were implemented with ChartJS, which is a framework that allows implementations of both line and bar graphs, among other things. Also, line and bar graphs are commonly used to display information and easy to understand, thus making them the best choice to easily display the information.

For predictive analytics, a new interface was created. The information that these insights display is different from the standard insights because it does not display large amounts of data in a compact format, but rather shows stand-alone numbers that show how students will do in the future based on their current performance. For this reason, predictive insights are explained in text rather than in graphs, explaining on what data the predictions are based and how they were derived.

Figure 3.1 shows how the analytics component looks when it is collapsed. This only shows the title of the analytic and what number the analytic is. When the user clicks on the arrows, the view expands and shows the full information, images of how this looks for each analytic are shown in chapter 5. An image of how the full page of the website looks with the analytics component is shown in Figure F.1.



Amount of correct answers per multiple choice question

Insight 1 of 8

Figure 3.1: The view of the analytics component when it is collapsed

# 4 Implementation

This chapter will go into detail about the ways the learning analytics were implemented in the project. Section 4.1 will explain the implementation of the back-end of the learning analytics and Section 4.2 will discuss the front-end. How the predictive analytics are implemented is explained in Section 4.3.

## 4.1. Back-end

In this section we explain the implementation of the back-end. The main point of the back-end is collecting and processing generated data to do something useful with it. In the case of analytics for FbF, this process consists of three main steps. The first step is making it possible to query the data so it can be examined and understood, which is described in Section 4.1.1. The second step is processing the data with the use of streams, which will be explained in Section 4.1.2. The final step is creating the insights that are send to the front-end to be displayed to the user, this will be discussed in Section 4.1.3.

### 4.1.1. GraphQL

In order to examine the data, a query language for APIs named GraphQL[1] was used. FbF was already using GraphQL, so that is why we had to use this query language as well to develop our product. In order to test the implementation of GraphQL, the ChromeiQL plugin on Google Chrome was used. This plugin shows all the documentation of the data in the database and makes it possible to perform queries on the data. Almost all of the information that was needed was already present when we started the project, except the information for presentations and broadcasts of presentations. In order to be able to query this data, presentation and broadcast type files had to be made, which are files that specify what fields certain types have. Fields can be specified by either explicitly specifying them in the type file of that item or by implementing an interface that already contains the fields. Once all the files for presentations and broadcasts were added to the graph structure, queries could be performed in order to fetch and process the data.

### 4.1.2. Streams

The first step in creating analytics is gathering data. Most data that is needed to implement the analytics was already stored in the database by FbF when we started the project. An exception to this was the information regarding the slides for students, such as how long they visited a slide or how many times they revisited a slide. This had to be turned on during the project such that the data could be stored and accessed. In order to fetch data from this database, a query is created in which the data that is needed from the database is specified. When this query is executed, the data comes back as a stream. This stream is then processed and manipulated to get the information for the insights that are created.

[1] https://graphql.org/

### 4.1.3. Insights

The analytics that FbF offers to teachers for their products are called insights. An insight is an object that contains all the information about different aspects of an activity that the front-end needs. It is created in the back-end by taking a data stream and extracting the correct information by mapping, filtering and finally storing the information. During this project we created two types of insights: standard insights that are based on standard analytics and predictive insights that are created by making predictions on data. When all the necessary information is extracted from the data stream, the actual insight is created and added to the list of insights which is then stored in the database.

## 4.2. Front-end

This section will explain the different aspects of the FbF product we had to use in order to show the generated analytics in the front-end. The page where the analytics had to be shown was already present, along with some very simple analytics, but we had to create a new section on the page to display the more advanced analytics in. This was done with two main parts. The first one was HandlebarsJS templates, which will be explained in Section 4.2.1. The second part was JavaScript files which served various purposes, this will be discussed in Section 4.2.2.

### 4.2.1. Templates

The templates that FbF used are HandlebarsJS[2] files, which are used to generate HTML code for a webpage. The view in which the analytics are displayed is called an assistant view, which was already made by the people of FbF for their assignment analytics, so we could use most of it for the assistant view for the presentations. The template files for the content however, which is mostly bar and line graphs, had to be developed by us, since these did not exist yet. For the charts, ChartJS[3] was used, which is a framework that allows implementation of graphs. To implement the graphs, a template file was made with the necessary fields, which are then set in the JavaScript files. In the template files, plain HTML code can be inserted, but other templates can also be used. This allows the code to remain structured and have separate template files for different purposes that can also be reused if needed.

### 4.2.2. JavaScript

For the front-end, the JavaScript files contained most of the code. The main goal of the JavaScript code is to set the fields that are declared in the template files discussed in Section 4.2.1. For the ChartJS charts, the data and the options for the graph had to be defined, which would all be done in the JavaScript. This includes the data points, the color of the graph, the labels of the axis and all other aspects of the graph as defined by the ChartJS documentation. In the template, it only needs to be specified that the component is a graph and that the data and options come from the JavaScript. When compiling the project, the template is automatically filled with all the correct values as specified and converted into HTML such that it can be displayed in the website.

## 4.3. Predictive analytics

The implementation of the predictive analytics consists of two major parts. Before being able to make predictions, the data must be pre-processed and the model that is used to make the predictions must be trained on this data. Section 4.3.1 explains how the data is pre-processed. The process of training the model will be explained in Section 4.3.2. After having trained the model predictions can be made, this will be explained in Section 4.3.3.

### 4.3.1. Pre-processing the data

The algorithm can not make predictions based on raw data, so the data is processed in Scala first. The data that is used originates from one course for which six presentations were held. The raw data is stored in a JSON file. The reason for not using streams as is done for the standard learning analytics is that FbF has an agreement with their customers that their data will not be used outside their course unless it is anonymized. For each student in each presentation, several features are calculated.

[2] https://handlebarsjs.com

[3] https://www.chartjs.org/

The output of the algorithm is whether the student attended the next presentation. The input of the algorithm is a set of the following three features:

- percentage of unanswered questions; the percentage of unanswered questions, including both open and multiple choice questions. This indicates whether the student is actively participating in the presentation.
- Score of the student as a percentage of the total score; for each multiple choice question a student can score between zero and one. The total score is the amount of multiple choice questions, as the open questions are not graded. This gives an indication of how well the students understand the material that is being presented.
- Average deviation from the average answering time; for each student the deviation from the average answering time for the corresponding question in the corresponding broadcast. For each student the average of the deviations is used as input to the model. This factor indicates how actively the student is participating and also how well they understand the material that is being presented.

After processing the data it is saved in a comma-separated values (CSV) file such that it can be used to train the model. In this file each row (except for the first row) represents a student and each student is represent by their features.

### 4.3.2. Training the model

The sci-kit learn[4] machine learning toolkit for Python is used to train the random forest classifier. The data is read in from the CSV file and before training the model, the output column is removed from the data. The model is then trained using 10 trees and a test size of 25%. After training the model, the percentage of incorrect predictions on the training data is equal to 39%. While training the model, the order of the presentations is not taken into account. Why this is a problem is explained in Section 7.2.4. After training the model it is saved to a file such that it can later be used to make predictions without having to retrain first.

### 4.3.3. Making predictions with the model

Similar to the standard analytics, data is being gathered and processed first. Using Python code an HTTP server is hosted. The data that is needed to make the predictions is sent to this HTTP server for each presentation. A method that reads in the trained model and makes predictions based on the received data is called. As a response the prediction is sent back and the insight is created.

## 4.4. Testing

Since the application of FbF is a website and the area we worked in for the project was mainly the front-end, it was difficult to test our code. FbF does not do any unit tests for the UI, but instead only has integration tests. In the first feedback from Software Improvement Group, they advised to add some unit tests for the most important methods, because there were none at the time. In the JavaScript, there were some methods that calculated certain values for the charts. Since these methods took in one parameter and returned an integer, they were easy to test. In the back-end however, we did not do any automated testing of our code. The reason for this was that it would require replication of a huge database in order to have relevant input for our tests. This would be too time consuming and simply not worth the effort with the limited time we had for the project. Instead we held our own presentations and verified that the insights were correct. Whenever something went wrong in the back-end, this error could be seen in the data in the front-end, so automated testing was not an absolute necessity.

[4] https://scikit-learn.org

# 5
# Final product

In this chapter, the implemented features of the final product will be explained in detail. These are all the features that were implemented in the ten weeks of the project. Section 5.1 will show the different standard analytics that were implemented. Section 5.2 will highlight the predictive analytic and how the results of it were shown to the user.

## 5.1. Standard analytics

In this section, the implemented standard analytics will be described in detail. There is a total of five standard analytics that were implemented in the final product. Three of these are related to the answers given to questions in the presentations and two of them are related to the slides of the presentation. The following sections will discuss the different standard analytics and show how the analytics are displayed to the user.

### 5.1.1. Number of correct answers per question

The first analytic that was implemented shows how each question from the presentation was answered by the students. The information is visualized using a bar graph with three different categories for the answers: correct, incorrect and unanswered. Figure 5.1 shows an example of how the graph of this analytic looks. The number 3 here is not shown on the x-axis below because only multiple choice questions are taken into account. When the cursor hovers over a certain bar in the graph, a preview of the question is given in a text box, along with the values for each of the categories. This is shown so the teacher does not have to go back to the slide-show in order to see what the question was, which makes the analytics a lot easier to use.

This metric is important for teachers because it gives a clear overview of the results of all the questions, which makes it easy to see what questions were answered well or bad. This can be used by the teacher to determine whether some questions were too hard or easy, or if students do not understand the material well enough. This is useful for the teacher when making test questions for example, since he or she will know what type of questions are easy or difficult for the students.

### 5.1.2. Number of students per correct answer amount

The second analytic shows how many students had a certain amount of correct answers. The information is visualized using a bar graph where the amount of correct answers is on the x-axis and the amount of students that had that amount of answers correct is on the y-axis. Figure 5.2 shows an example for how the graph of this analytic looks. The values on the x-axis of the graph go from zero to the total amount of multiple choice questions, even when the the value of the bar is zero. This is done to keep consistency between the presentations, so every graph has the same format.

This metric is important for the same reasons as the first analytic, namely to see whether the students have a good understanding of the material and the questions are of the appropriate difficulty. The

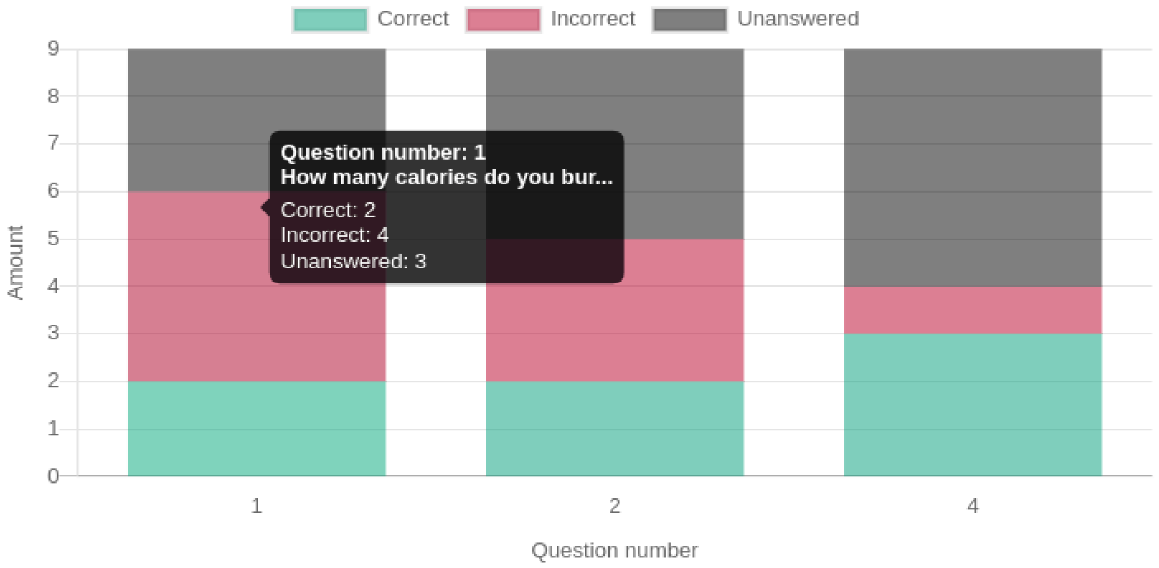


Figure 5.1: Bar graph of the amount of correct answers per multiple choice question

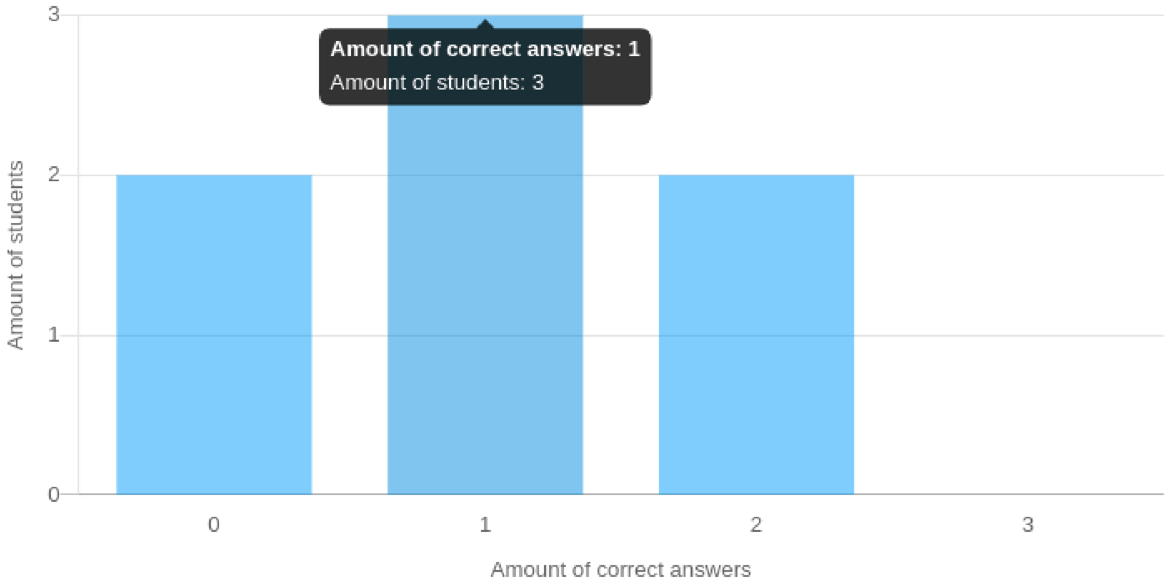


Figure 5.2: Bar graph of the amount of students per amount of correct answers for the multiple choice questions

main difference here is that this analytic gives this information on the whole presentation as opposed to individual questions.

### 5.1.3. Correlation between time taken to answer and correctness

The third analytic shows the correlation between the time taken to answer the question by the students and the amount of correct answers for that time. This is done by making separate bar graph for every multiple choice question with at most four time intervals on the x-axis and the amount of answers on the y-axis. The bars show the correct and incorrect answers in green and red respectively, similar to the first analytic. An example of how this analytic looks is shown in Figure 5.3. When constructing the intervals, a similar number of students is placed into each time interval. This is done so the answers of each interval are easy to compare, since having one student in an interval does not give a good insight in the performance of that interval.

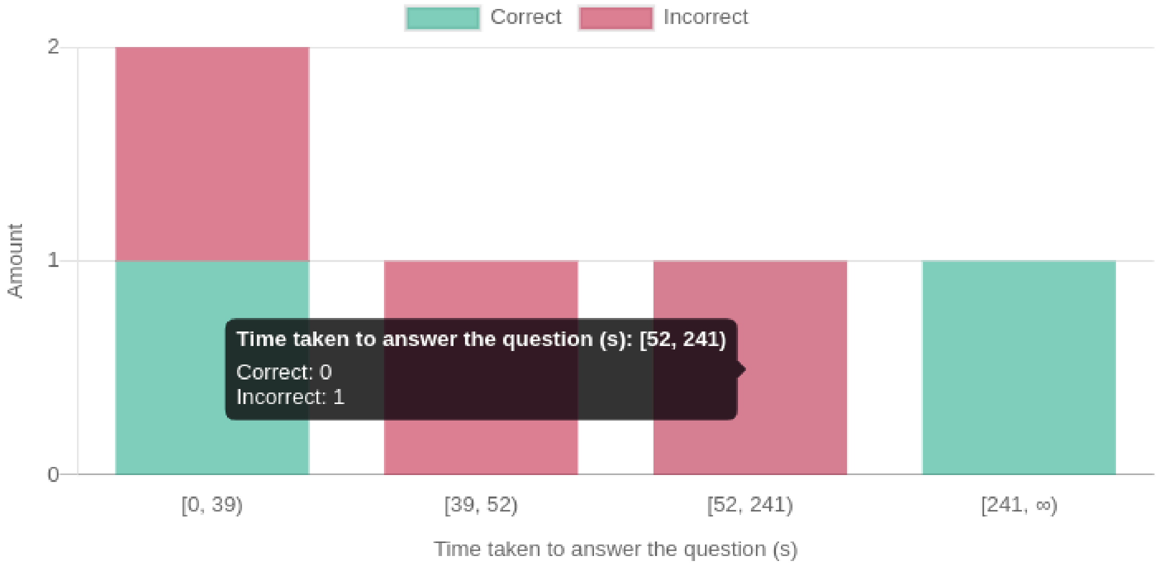


Figure 5.3: Bar graph of the time taken for a certain question in correlation with correctness

The goal of this analytic is to allow the teacher to possibly discover a correlation between the time taken to answer a question and the correctness. A correlation can have a lot of different causes. For example, if a lot of students answer quickly but with wrong answers, this could be caused by a lack of interest causing them to guess and answer, but also because the question seemed easy but was actually more difficult. The teacher can then act on this information by telling the students to read more carefully or take more time to think about the question for example.

### 5.1.4. Number of total views per slide

The next analytic is the total number of views for every slide in the presentation, excluding question slides. This is visualized with a line graph where the slides are on the x-axis and the amount of views for that slides are on the y-axis. Figure 5.4 shows how this graph could look like for a total of nine participants. The slide numbers on the x-axis are hidden so it gives the impression of a timeline rather than separate values for each slide. When the mouse hovers over the graph, it gives the exact values for that x and y in a text box. This is useful for when the presentation has a lot of participants or slides and the individual values are not easy to make out visually.

The main point of this metric is to give the teacher an insight into how many students were in the presentation at a given time. With this graph, he or she can see whether a lot of students came in late or left the presentation early, or whether people were not there for certain slides.

### 5.1.5. Number of times a slide was revisited

The final standard analytic that was implemented was how many times each slide was revisited by the students, again excluding question slides. This is visualized with a bar graph that has the slide numbers on the x-axis and the amount of times that slide was revisited on the y-axis. Figure 5.5 shows an example for how the graph of this analytic would look. Unlike the views per slide metric in Section 5.1.4, the slide numbers are shown on the x-axis because for this metric it is relevant to see the data for individual slides. When the user hovers over a certain bar with the cursor, a preview of the slide corresponding to that bar is shown. This is important because the teacher does not have to keep going back to the actual presentation in order to identify which slides are revisited.

This analytic is useful for a teacher so he or she can identify which slides are revisited a lot by the students. This can potentially indicate that some slides are difficult or unclear and students have to look at the material multiple times before understanding it.

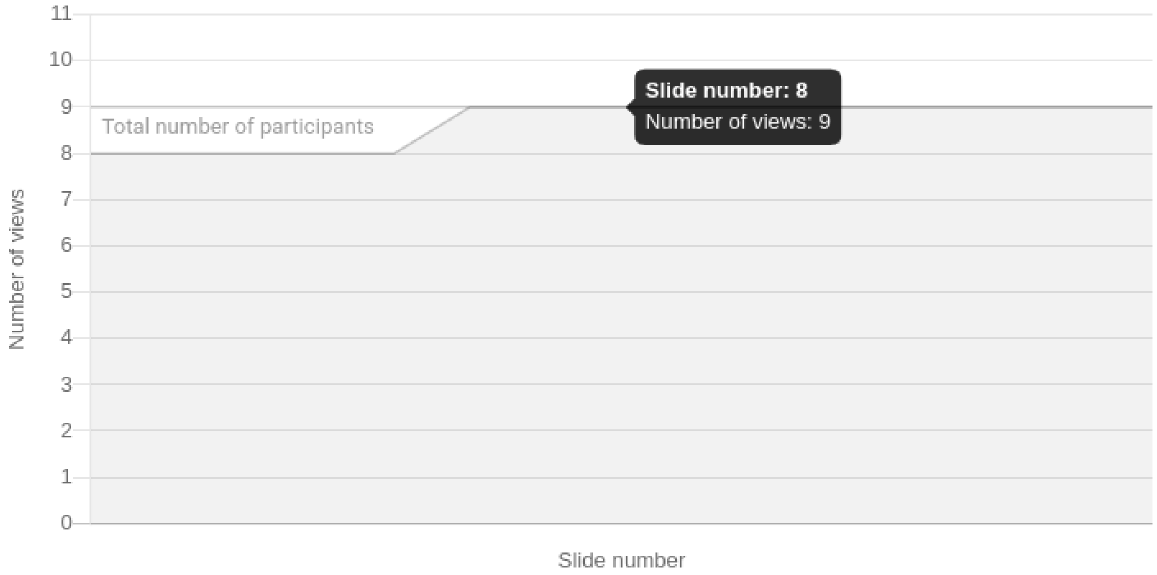


Figure 5.4: Line graph of the amount of views per slide (excluding questions)

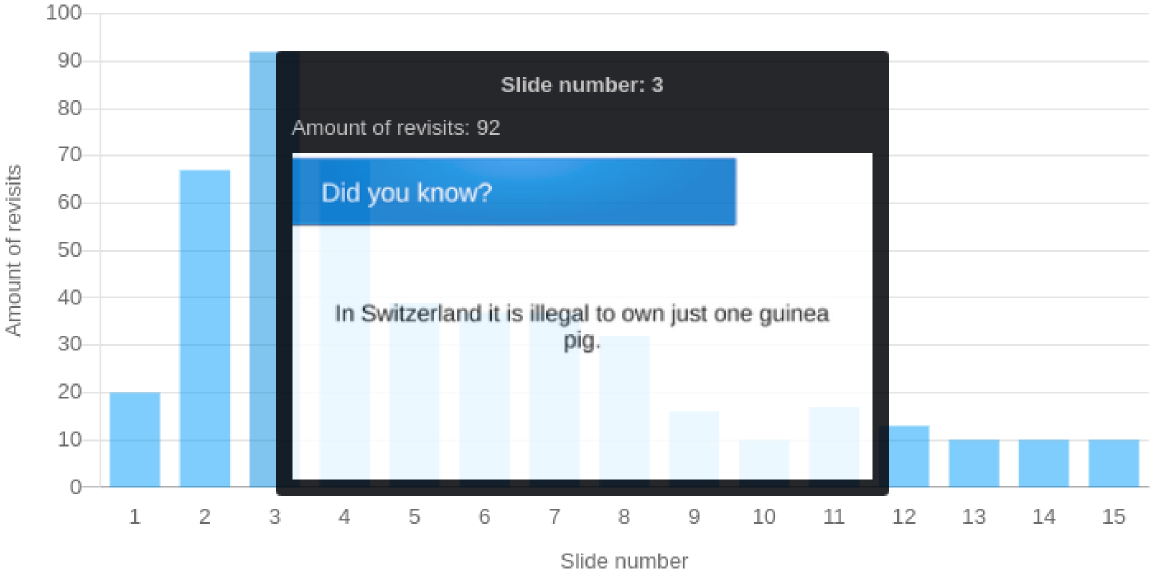


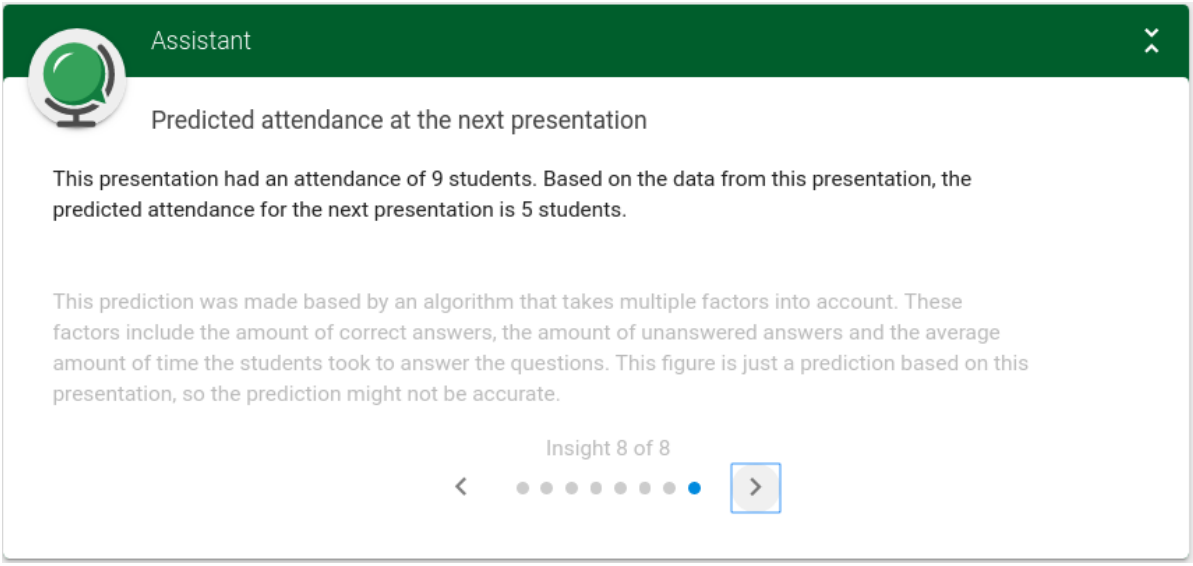
Figure 5.5: Bar graph of the amount of times a slide is revisited by students

## 5.2. Predictive analytics

This section will discuss the predictive analytic that was implemented during the project in detail. Since there was more work to do with the standard analytics than expected at the start, most of the project was spent on that part and predictive got less of our attention than initially expected. As a result, not all of the predictive features that were planned at the start were implemented. However, the back-end and front-end of one predictive analytic are fully implemented, which required all the basics to be implemented. Since the basics are now there, adding new analytics should not be a lot of work, so the final product is still useful. The following section will discuss the implemented predictive analytic in detail.

### 5.2.1. Future presentation attendance

The predictive analytic that was implemented predicts how many people will be present at the next presentation, based on the attendance of previous presentations. As opposed to the standard analytics, it was decided to present the predictive analytics with text instead of graphs. Since the analytic only consists of two numbers, there is no useful application of a graph that would present the information in a clearer way than simple text. In Figure 5.6, an example is shown for how this analytic would look. The analytic itself consists of two values, which is the amount of students that were present at the current presentation and the amount of students that is predicted to be present at the next one. Underneath this, some extra information is given about the factors that were taken into account when generating the prediction. This is explained to give the teacher all the information so he or she knows how the prediction was generated. The teachers can then decide for themselves how important this prediction is for them and whether they will act based on the information they were given.



Predicted attendance at the next presentation

This presentation had an attendance of 9 students. Based on the data from this presentation, the predicted attendance for the next presentation is 5 students.

This prediction was made based by an algorithm that takes multiple factors into account. These factors include the amount of correct answers, the amount of unanswered answers and the average amount of time the students took to answer the questions. This figure is just a prediction based on this presentation, so the prediction might not be accurate.

Figure 5.6: Textual information about the predicted attendance for the next presentation

This metric can be useful for a teacher to see whether the general interest of the lectures is going down. If there were a lot of people at the previous presentations, the prediction will most likely say there will be a lot at the next one too. This means that the teacher is doing a good job because the students think the lectures are useful. If the attendance goes down every week however, then the algorithm should also predict a lower amount for the next presentation. With this information, the teacher can try to improve the lectures for example in order to make more students attend. Another useful application could be that an important lecture is coming up and the teacher sees that not a lot of people are predicted to show up. The teacher can now act on this estimate before the next lecture actually takes place by putting an announcement online that the lecture is important. This could result in more attendance for important lectures and possibly a higher passing rate for the course.

# 6
# Process

This chapter describes the process of developing the product in detail, as specified in the project plan. In Section 6.1, the research phase of the project is explained. During the research phase, information is gathered on technologies, methods and obstacles that already exist in the field of predictive learning analytics. In Section 6.2, the development phase of the project is explained. This section explains the process of building the actual product, following the schedule and goals that were set in the project plan. Finally, what version control and programming languages were used can be found in Section 6.3.

## 6.1. Research phase

The research phase of a project is used to get an insight into the problem that has to be solved, find existing work that is used to resolve the problem and find information that is useful to create a new solution for the problem. During these two weeks, a research report and a project plan are made.

The first two weeks of the project were dedicated to the research phase. During these two weeks, we gathered literature about learning analytics and predictive learning analytics. The research report summarizes the result of the research phase. In the two weeks of research, different methods and techniques for creating learning analytics were found. In the report we compare these and determine what the optimal approach will be to achieve the project goal. In the project plan we specify the project goal and the requirements that have to be met in order to say that we achieved the project goal. It also explains what development process we used. Both the project plan and research report can be found in Appendix A and B, respectively.

## 6.2. Development phase

The development phase is a six week period in which the project goal is achieved. The project plan that is created in the research phase is followed during the development phase to ensure that all project requirements are met by the end of the six weeks.

From an earlier course of the bachelor Computer Science, we learned the agile development method Scrum. Scrum is a development method where work is done in cycles (or sprints), usually with a length of one to two weeks. At the beginning of a cycle, a backlog is created in which all tasks for that cycle are set. These tasks represent all the work that should be finished in this cycle. At the beginning of a new cycle, the work of the previous cycle is reflected upon. By reflecting on the process of the previous cycle we can determine what went well and where we could improve for the next cycle. This helps to optimize the workflow for the upcoming cycles.

For development we maintained two week long work cycles, because FbF also maintained work cycles of two weeks. Each cycle started on Tuesday, where we made a backlog for the upcoming two weeks in Trello[1]. We created all tasks for the new sprint, each having a description of the task, the

[1] https://trello.com/en

assigned team members, the estimated time needed and the importance of the task for this sprint. In a sprint retrospective document we wrote down what went well, where we could improve and how we would improve on the next sprint to ensure that we would do better. After completing a task it would be commented how much time it took to complete the task, this made it easier to create the sprint retrospectives.

### 6.2.1. First Sprint

The first sprint included getting familiar with the code base and setting up the project. This included getting access to all repositories and learning how to connect to the database and the staging environment. This extra work combined with our (at this point) limited knowledge with the code base made us decide to only create standard learning analytics tasks in the first cycle. We wanted to take into account that the first sprint would require a lot of research into the code in order to figure out how data was processed and how insights were created and displayed to the user. To keep the workload of the three cycles equal we planned the less complex tasks in this first cycle.

We divided the tasks over two groups. Two team members would work on back-end tasks such as getting the needed data for the insights, process this data to get the required information, and storing it in the database. The other two team members would work on the front-end. With the prepared data that was received from the back-end they made sure that the insights are displayed to the teacher.

During this sprint, we started with an insight showing how many students got each multiple choice question in a presentation correct. This insight was relatively easy to implement and gave a good view on the difficulty of creating insights. Next, we implemented an insight that shows how many students have how many questions correct. Work on a third insight was started. This insight would show the correlation between the time taken to answer a question and the correctness of that answer. This insight was not finished in this sprint however.

Since this was the first time that we worked in the repositories of FbF, some unforeseen problems occurred with creating data and displaying this in the staging environment. Small problems, such as only being granted read access to a repository or figuring out how data was processed, were easily resolved. Because of bigger problems with data storage we were not able to implement one insight that shows the number of students that have seen a slide during a presentation. The data necessary to create this insight was no longer stored in the database. This prevented us from implementing this insight, since we did not know how the actual data structure in the database would look.

### 6.2.2. Second Sprint

The goals for the second sprint were to finish all standard insights and get started with the predictive analytics. The predictive analytics are the most important aspect of this project, and should definitely be implemented.

There were three standard insights left to implement. The insight for the correlation between time taken and correctness of an answer was almost finished in the previous sprint and only took one more day to complete. The event of a student seeing a slide was now stored in the database as well, allowing us to create the last two insights. These insights show the number of students that have seen a slide and the number of times that a slide is revisited. Implementing how many students had seen a slide was an easy task, since it only required the count of distinct user ID's. Implementing the number of times a slide was revisited was more of a challenge, since we had to distinguish between different broadcasts (i.e., different sessions in which the same presentation was presented).

The predictive analytics tasks for this sprint were to find and implement a suitable algorithm, and find a way to send data to this algorithm from the Scala code. The reason for this is that the predictive algorithm is written in a Python script and is not connected to the the Scala project in which the data is queried and processed. The data from the Scala interpreter needs to be shared with the Python interpreter in order for the algorithm to train on this data and make predictions. The tasks of researching which algorithm to use and implementing this algorithm were both done during this sprint. We did not manage to connect the two interpreters and share data between the two.

In a meeting one week into the sprint some suggestions were made by the FbF team. The standard insight that shows how many times a question is answered correctly only shows the question number of each question. It was suggested to add the question itself to the overview, allowing a teacher to immediately know what question he or she is looking at. Furthermore, the insight that shows by how many students each slide was seen, and how many times a slide was revisited also only shows the slide number. The suggestion was made to add a preview of each slide, allowing the teacher to easily see which slide corresponds to the slide number of each slide in the insight. These two suggestions were added as tasks and in the last week of this sprint the first of these two tasks was implemented in the product.

### 6.2.3. Third Sprint

The goals for the last sprint were to finish the predictive insight and, if there was enough time, implement the suggestions that were made during the meeting in sprint two. Multiple tasks still had to be done in order to create the predictive insights. This includes querying data and pre-processing this to use as training data, finding a way to share data between Scala and Python, allowing the predictive algorithm to read in the data and sending the predictions back to Scala to create the insight to show to the user.

Gathering training data for the predictive algorithm proved more difficult than originally anticipated. We originally planned to share data between Scala and Python via HTTP service by sending CSV style strings. However, the dataframe structure that the data has when it is sent from the database to Scala is difficult to save in a string format. This made sending the data over HTTP difficult. The second problem we faced was the lack of data that was stored in the testing environment. We had been working in the testing environment since the start of the project. This environment does not allow access to the actual database containing data of users of FbF. This meant that there was not enough data to train the model on. Together with our supervisor from FbF, we decided to do a request from the actual database, anonymize this data and store it in a local file. The data was given to us in JSON format. This was then processed to CSV format and stored in a local file. We were now able to train and make predictions. The HTTP service that was already built was used to send the predictions from the algorithm back to Scala where a predictive insight could be created. Creating the predictive insight was an easy task, since the data was already received from the HTTP service in correct format, only a textual explanation had to be added.

The remainder of the time for this sprint was used to implement one of the suggestions that were made in the meeting during the previous sprint. When looking at the insights that shows the number of times each slide was revisited, a preview of each slide can be shown to allow the teacher to easily see what slide it is that was revisited.

## 6.3. Version control and programming languages

The product that we develop in this project is directly build into the existing product of FbF. This means that we had to use the same tools for development as them and implement all functionality in the same languages as they used.

For version control FbF uses Git via GitHub. GitHub is a hosting facility for the open software Git. Git allows users to easily branch from the main code base in order to create new functionality. After implementing and testing, this can than easily be merged with the main code base. All four of us are familiar with Git, as we have been using this for all our previous projects. This made working in their code base much easier as we knew how to create our own branches and work on development without disrupting the workflow of the software engineers from FbF.

FbF used a variety of languages, but the main languages we worked with were Scala and JavaScript, two languages we had already used previously and were familiar with. These languages were used to request and process data and build the interface that users see when using tools from FbF. Other languages that were used were Ruby, Python and HandleBarsJS.

# 7 Discussion

This chapter will reflect on the project as a whole. Both positives and negatives will be discussed about the product and the development process. In Section 7.1, the biggest challenges we faced during this project are explained. In Section 7.2, some improvements that could be made on the final product are discussed. Section 7.3 reflects on the development process during the project. Finally, Section 7.4 will discuss the ethical aspects of the project that have to be considered.

## 7.1. Challenges

This section will discuss the various challenges we faced during the ten weeks of this project. Despite the project going well overall, there are always certain things that do not go according to plan. The following sections will explain these challenges in detail.

### 7.1.1. Familiarity with the code

The biggest challenge of this project was getting familiar with the product. In order to extend the existing product that FbF delivers with new functionality, we needed to work in a large number of files. Finding out in which files changes needed to be made and where new functionalities needed to be implemented took a lot of time in the first weeks of development. This is partly because no documentation was provided to explain what each file did. Some of the programming languages that needed to be worked with were new to us. Of all the programming languages that were used, Scala, Python and JavaScript were the only ones with which we had experience.

### 7.1.2. Limited test data

Another challenge we faced was the limited amount of data that we could use to both test our work and train the predictive algorithm that was used to make predictions for the predictive insights. We were given access to the testing environment, which did not store any data from the users of the FbF tools. We were able to create data by hosting presentations ourselves, but the number of participants for these presentations were low and the data was not a good representation of a presentation that is given in a lecture. The lack of data meant that we were not able to train the predictive algorithm. We worked around this by anonymizing some of the actual data and using this for training, but when splitting this data in a training set and a validation set, the validation set showed that the error percentage of the predictions was still high. This meant that there was not enough training data for the algorithm to learn how to make accurate predictions with the data it is given.

## 7.2. Future improvements

Although most of the goals that were set at the start of the project have been accomplished, there is always improvements that can be made on the product. This section will highlight some ideas that could further improve the product, but did not get implemented in the final product of this project.

### 7.2.1. Textual advice based on analytics

When the teachers see the analytics for the first time, they might not understand the information immediately. Furthermore, even if they do understand the information, they might not know how to act on that information to fix a possible problem shown in the analytics. For this reason, a possible improvement on the product could be to analyze the data automatically and provide the teacher with textual advice in the analytic screen. This advice could be a possible action the teacher could take to fix an issue, or simply point out a correlation that might not be obvious in the graphs. Since teachers may not be able to get the most out of the product by themselves, this change could help improve the efficiency a lot.

### 7.2.2. Revisited slides with a timer

Currently, when a student goes back to a slide, the number of revisits for that slide and student is incremented by one, except for when the teacher takes them there. A problem with this occurs when students are using their arrow keys when navigating through the presentation. When the student goes back two or more slides with the arrow keys, it will count all the slides in between as revisited as well, even though the student was not interested in these slides. In the insights, this could make it look like some slides were revisited much more than was actually the case, giving teachers the false idea that some slides were important or difficult based on their revisited numbers. This problem could possibly be solved by introducing a minimum time requirement that the student has to be on a slide before it counts as revisited.

### 7.2.3. Real times of slide views

The current analytic for the number of total views per slide is meant to give the impression of a timeline rather than information about individual slides. In order to achieve this, the numbers on the x-axis are currently hidden instead of showing the slide numbers. In the feedback FbF gave, they said that the x-axis could also show the actual times that the slide first got opened by the teacher. This would preserve the impression of a timeline of the whole presentation while still providing some extra details about the individual slides. This is something that did not get implemented in the project, but could be a good improvement to the analytic.

### 7.2.4. Predictive learning for similar presentations

The current predictive algorithm is given information about the participation of a student during a lecture and predicts whether that student will attend the next lecture based on the participation in this lecture. The algorithm does not know what presentation the student attended, or from what course the presentation was. An improvement would be to extend the data with information about the presentation itself. This would allow the algorithm to not only predict whether the student will attend a next presentation based on his or her participation, but also on what type of presentation this was. The algorithm could then compare with similar presentations to decide whether the student will attend a next presentation, probably resulting in a higher accuracy.

## 7.3. Development process

During our first week of development, we decided to divide the work between team members. There was a lot of code to go through in order to get familiar with the code and it would be more efficient if each team member focused on one part of the process of creating insights. By doing this, the amount of work for each team member would be reduced and the complete process could be understood faster. Due to the absence of documentation and the lack of experience with the technology that was used, help from each other was needed in order to understand the process of creating insights. For this reason little progress was made in the first two weeks of the project.

Once we knew how an insight was made and the first insight was created, the process of creating more insights went a lot faster. The time needed to create an insight was mostly dependent on the complexity of processing the data. For some insights very little data processing needed to be done in order to get the desired information for the insight. An example of this is number of students per correct answer amount. For this we only needed to count the number of answers a student had correct, and count how many times each amount correct appeared. The insight that shows the correlation between time taken to answer a question and the correctness of that answer was a lot more work however. This insight required us to calculate how long a student took to answer the question. This required the opening times of each question and the update time for the answer of each student to calculate the time taken to answer the question. Lastly intervals had to be created such that all intervals contained about the same number of answers.

Communication between the team and FbF was great. Two times a week we worked in the office with the other employees and supervisor, usually on Tuesdays and Fridays. The days in between we would work from home. This schedule worked well since enough progress could be made from home, and when we ran into problems we could consult with the supervisor via Slack, the online platform that was used for communication.

## 7.4. Ethical implications

Measuring the participation and results of students in class can be of great value to determine the quality and effectiveness of a lecture, but we have to be careful not to make ethical irresponsible decisions based on this data. With the product in its current form, this is not possible. All analytics that are generated show how well the students did that attended the presentation. This group is shown as a whole, where no personal information is shown to identify a student from this data.

It is possible however that the predictive analytics are later extended to predict the change of a student passing the course, or show the list of students that will be attending later lectures instead of just the number of students that will be present. Although useful, such a predictive insight can be wrongly interpreted by a teacher who might see that a student will not attend upcoming presentations and is likely to fail the course. It should not be the intention that a student has a disadvantage in the course because he or she is refused help or extra time to understand the material explained in class, with as reason that the student does not seem to want to put in enough effort in the course.

# 8 Conclusion

To summarize, for our BEP we extended the interactive presentations tool of FbF. The goal was to add standard and predictive analytics to the interactive presentations to help teachers by giving them better insights in the progress of their students in the presentation and to improve their courses. Before extending the product, we looked at how the product that FbF already provides was structured in order to get familiar with how extensions for this product would be built. This was not an easy task since FbF uses various programming languages and we were not familiar with some of them (e.g., HandleBarsJS).

In the beginning the tasks were divided over two groups to make it easier to get familiar with the code. One group worked on the front-end and one group worked on the back-end. The back-end group was responsible for converting the data to a format which allowed easy visualization in the UI by the front-end team.

To process the standard insights we only needed to tweak and filter the data in the correct way to be able to use it. To generate predictive insights, more elaborate methods were needed. After doing research, we concluded that random forests would be the best option to use as the model to make predictions with. We trained the model with a CSV file in Python and used an HTTP server to communicate between Scala and Python in order to be able to create insights.

The final product contains all the standard insights we initially wanted to implement. However, the product does not contain all the predictive insights that were specified as a requirement in the project plan. The only predictive insight that is implemented is predicting how many students will be attending the next presentation. Due to a lack of data the model can not accurately predict how many students will be present at the next presentation. The lack of data is also one of the reasons why the other predictive insights were not implemented.

Overall, the process of developing the product during the ten weeks of our BEP went relatively smooth. During the first week of development we had some delay with setting up our systems in order to get access to FbF's product and get started with development. Even with this and the fact that we started the development one week late, we managed to finish the product within the given time. Additionally, we implemented certain features based on feedback given by FbF. These features were not specified in the project plan.

# A
# Project plan

## A.1. Introduction

This report gives a detailed explanation of the plan of action for the Bachelor End Project 'Predictive learning analytics'. The project is for the company FeedbackFruits (FbF), which is an online learning environment aimed to improve education for both lecturers and students.
The goal of this project is explained in Chapter A.2, which is to create an analytics tool for interactive presentations that can be used to give insights in the data gathered regarding student activity. Next to giving insights it should also be possible to predict what students will do based on the data gathered (e.g., will a student pass the course). In Chapter A.3 the exact requirements for the project are given. This is done using the MoSCoW method. Next is explained how we can assure that our product meets all requirements to be useful for the client and is both functional, maintainable and integrable with their existing product. In Chapter A.4 the development process is explained. We use Scrum to plan the development of the product, with 2 week sprints. This gives us 3 sprints to create our product. In Chapter A.5 it is explained how different documents will be created and how they will be stored. Finally, in Chapter A.6 an overview of all important dates of the project is given.

## A.2. Project goal

The goal of this project is to give teachers insight into the data that is gathered related to student activity. The main goal of these insights is that the teacher sees a potential problem with students or the material and has a chance to act on it before it has negative consequences. The data will also be used to predict future events or numbers, giving the teacher more insights into the current state of the course.

To fulfill this goal we will extend the existing system with the necessary features. We will first research the possible ways of predicting certain events based on student activity within the interactive presentations. After the research phase we will implement the best solution that was found during the research phase.

## A.3. Project requirements

The final product is a piece of software that should contain all features as discussed in Section A.3.1. It should also adhere to all the quality requirements as specified in Section A.3.2.

### A.3.1. MoSCoW

The requirements for the project will be outlined using the MOSCOW method [2].

- Must have
    - Standard analytics for interactive presentations:
        - Show for each slide how many students have seen it

        - Show slides that have been revisited the most
        - Show for each question how it was answered
        - Show time taken to answer each question and correlation with correctness of answer
        - Show for each number of correct answers (range of 0 to total number of questions) how many students had it
    - Pass all tests on the integrated implementation
    - Proper documentation
- Should have
    - Predictive analytics for interactive presentations:
        - Predict how students that joined late would have answered the questions they have missed
        - Predict whether students are likely to pass the course
        - Predict how many students will be present at the next presentation
- Could have
    - Visualization of the decision making for predictive analytics
- Won't have
    -

### A.3.2. Quality assurance

To ensure that our product is developed in a way that it is useful to our client, we have to specify some criteria other than the features of the product. These criteria are functionality, maintainability and integrability and will be discussed in the following sections.

#### Functionality

Since our product might be integrated into the core code of FbF, it has to fulfill all of the functional requirements explained in Section A.3.1. This ensures the product works as it is supposed to and follows the requirements set out by the client. Along with the functional requirements being met, the code should also be tested and support proper error handling to ensure there are no bugs.

#### Maintainability

Since we are only working on this product for a total of 10 weeks, our product should be maintainable by the developers of FbF after our bachelor project is finished. To ensure this is the case, we will use the same code style as used by the current core code base of FbF. All code should be formatted correctly and easily readable for people outside of our group. If needed, we will also provide external documentation of our code such as class diagrams.

#### Integrability

Since our product might be integrated into the core code of FbF, we have to make sure that it functions well with the existing code base of FbF and is easy to integrate once the product is finished. To achieve this, we will build on top of existing software and test our code continuously to ensure it functions as it is supposed to, even when it is integrated.

## A.4. Development process

During this project we will make use of Scrum[1] cycles. In total we will have three sprints that each have a length of two weeks. During each sprint we will have a prioritized product backlog that will give a clear overview of what has to be done first. During the sprint tasks might be assigned to other team members due to unexpected changes in work load, e.g., the work load of a certain task is higher than expected. This can be done during the daily Scrum, where each team member gives a short update on what he has done the previous day and what he will do during that day. At the end of each sprint session we will have a meeting to review the sprint and to plan the next sprint, this will most likely be on Tuesdays.

[1] https://www.scrum.org/resources/what-is-scrum

## A.5. Documentation

The final report will be written in LaTeX. All team members have experience with LaTeX. Meeting minutes and sprint cycle documentation will be written using Google Docs and stored on Google Drive. We have chosen for this because it is easier to write and access small documents in this way compared to LaTeX.

## A.6. Project planning

Because of a late start, the research phase will end at or before the end of week 19 (4.3). If the research phase is ended before the end of this week this time can be added to the first sprint. If the research phase is done at the end of the week we will have three sprints of two weeks each. Unless we will finish the research phase early in week 19 (4.3), the list shown below is a summary of our planning. If the research phase is done early in this week the second and third sprint session can be started one week earlier.

- 10/5/2019 - End of the research phase.
- 13/5/2019 - First sprint session.
- 28/5/2019 - Second sprint session and code submission to SIG.
- 11/6/2019 - Third sprint session.
- 21/6/2019 - Second code submission to SIG.
- 27/6/2019 - Submit the final report.
- 4/7/2019 - Final presentation.

# B

# Research report

## B.1. Introduction

With the amount of data being stored about the activity and participation of students in class, it becomes increasingly difficult for teachers to keep track of how their students are doing and if they are putting enough effort in their study. Next to the increasing amount of data that teachers have to work with, the number of students for some studies is also increasing drastically. This makes that the teachers have to divide their attention over a larger group of students, which even further reduces the possibility to look at the participation and progress that each individual student is making on the teacher's class.

Not only teachers, but also the students could benefit from a good analytics tool that monitors their participation. The transition to a university can be difficult for a student, as the approach to learning can be quite different from what they are familiar with. Students can put much effort in studying, but it is important to know where to put this effort in. Analytics tools could help here, as they can show the student where they are failing. The tools can help clarify what the student should do to improve their efficiency when studying.

In this paper we look at which analytics methods exist and what different classification techniques are used for predictive learning analytics. We analyze the most popular classification techniques, looking at both their advantages and disadvantages. With this information we decide what classification technique we will use to build our predictive learning analytics tool.

## B.2. Problem statement

Teachers have access to a large set of data about their course, regarding the effort that students put into their course. From this data teachers can get a good insight in the progress that a student is making, if the student is likely to pass the course or whether the student needs extra help in order to pass the course. The problem however is that finding patterns in this data that indicate how well a student is doing is time consuming work. A teacher would have to do this process for each of his students, multiple times throughout the duration of the course to keep track of the progress that each student is making and whether extra attention should be given to students. Teachers often can not afford to spend this much time on their students. To help them, Learning Analytics (LA) exist that can process data and find patterns that suggest which students are at risk of not passing a course.

### B.2.1. Learning analytics process cycle

"The overall LA process is often an iterative cycle and is generally carried out in three major steps: (1) data collection and pre-processing, (2) analytics and action, and (3) post-processing." [3]. The following sections will discuss these three phases in detail and highlight their purpose.

#### Data collection and pre-processing

Data collection is the start of all LA processes. Without data, there would not be anything to analyze, so obviously this phase is crucial to get good results. The data should be collected from different institutions and courses, to ensure that the LA can be widely used. The next step is to convert the raw data to a format that is suitable for analytics, this is called pre-processing. This step also includes reducing the size of the dataset if it is too large or taking out irrelevant features from the data. There are a lot of pre-processing techniques with different uses, which one is the optimal one often depends on the project.

#### Analytics and action

When the data is pre-processed, different LA techniques can now be applied to discover hidden patterns in the data that give a relevant insight into the learning process of the students. This phase also includes the action step, which means acting on the patterns that were found in order to prevent possible problems in the future. This step is the primary goal of the whole LA process.

#### Post-processing

After analyzing the data, it is possible that improvements on the dataset can still be made, this is called post-processing. This can include adding more data, determining new features to look at, refining the data or possibly even switching to a different method of analysis.

### B.2.2. Important aspects of learning analytics

In [3], a reference model for LA is given based on four questions about the LA process. These four questions will be answered in the context of the project for FeedbackFruits (FbF) in the following sections.

#### What?

In the 'What?' dimension of the reference model for LA it is being questioned what kind of data the system gathers, manages and uses for the analysis. The system stores data about the students activity in the interactive presentations tool. The data is stored in a database which is used when the progress of the student with the interactive tool is requested.

#### Who?

Who is targeted by the analysis is another dimension of the reference model. In this case the teachers are the ones targeted by the analysis.

#### Why?

Why does the system analyze the collected data? FbF wants to help the teachers and the students to have a better experience with the courses that the teachers give and the students follow. In LA there are many possible objectives. In the context of FbF, prediction and intervention will be used [3]. This way the teacher can see what subjects are difficult for the students and how the presentation might be improved.

#### How?

How does the system perform the analysis of the collected data? There are various techniques in LA and we will use the first three that are described in Section 4.4 of [3]. We will use statistics, visualization of the data and data mining in the form of prediction.

### B.2.3. Challenges of learning analytics

Like almost all technologies, LA faces some challenges that can make the widespread implementation difficult. This section will discuss the most common challenges that people mention about LA.

The first challenge is the fact that LA often goes hand in hand with machine learning algorithms that automate the analyzation process. Machine learning in itself is not a problem, but more often than not, machine learning is used as a black box model [4]. A black box model means that the algorithm is given an input and produces an output, but how it constructed this output is unknown. This makes understanding the technology quite difficult for people that have to use it, because there is no way to check the validity of the output.

The second issue is the fact that a machine learning algorithm can only be as good as its training data. If there is a bias or inaccuracy in the training data, the algorithm will reflect this in the results. It also means that a model will most likely only work in a very specific context if the training data was gathered exclusively in that context. This results in LA models possibly being useless in contexts other than what it was training on, such as different courses or universities. Collecting data from different contexts could be a solution to this problem, but that is often more difficult and time-consuming, so there is always a trade-off between resources and accuracy.

The last problem with LA is that teachers do not know how to use the software or how to interpret the results [5]. Furthermore, when they do know how to navigate the software and get a result, they do not know the best way to take action for at-risk students [5]. This is a tough issue, because it means that all the software could work perfectly, but still it is not able to be effectively used by the people it is meant for.

### B.2.4. Output of our Learning Analytics

Based on the data that is being gathered a decision should be made. The goal is to provide teachers with a structured overview of all the data and provide relevant insights about the students or activities. This can include at-risk students, what questions are too difficult or correlations between different statistics. With these insights, the teacher can recognize possible issues and act on them before they become bigger. The analytics should be a benefit for education for both the teacher and the students.

### B.2.5. Ethics regarding Learning Analytics

With the product that we are creating, it should not be the case that it can have a negative impact on the students. For instance, students that are classified as at-risk students, should not be treated different than students that are on track with their progress. It should not be the case that a teacher is less willing to help the at-risk student rather than a student that is on track.

Another issue is the privacy of the students using the tool. Since FbF is a third party that handles and stores all the data generated by the students, the privacy of the students using the tool is an important concern. We believe that the LA tool does not violate the privacy of the students because the teacher of the course is the only person that can view the data. It does not get uploaded anywhere or shown to anyone other than the course instructor. Since the information is relevant to the course and thus also to the teacher of that course, we believe that there is no violation of the students privacy when using the LA tools for educational improvement.

## B.3. Possible solutions

This section will highlight the different technologies that are being used for LA in practice. The following sections are the main implementations we will be looking at for our project. In Section B.3.1, the technologies of standard LA will be discussed. Section B.3.2 will explain the different classifiers that could be used for predictive LA.

### B.3.1. Standard learning analytics

”LA applies different techniques to detect interesting patterns hidden in educational data sets. In this section, we describe four techniques that have received particular attention in the LA literature in the last couple of years, namely statistics, information visualization (IV), data mining (DM), and social network analysis (SNA).” [3]

#### Statistics

Statistics are arguably the most basic form of learning analytics. As a result, most LA tools implement some form of statistics in their methods. Statistics are simple, but can sometimes still provide a lot of useful insights for the user. Like all LA, data has to be collected first and ideally get represented as numbers. This makes the operations in statistics, such as mean, average and standard deviation a trivial process, which is why it is so popular.

#### Information visualization (IV)

The results of a particular LA method are not always clear and easy to understand for the users. Putting the information in a form that everyone can understand is very important so the users can get the most out of the system. Since humans are very visually orientated, visual representation of data often works better than plain text or tables [6]. Obviously, the optimal visualization depends on the context, but overall they work better than textual representations.

### Data mining (DM)

”Data mining is defined as the process of discovering useful patterns or knowledge from data sources” [3]. This is a broad definition, which includes many techniques. Some examples of this are finding nearest neighbours, clustering, finding frequent itemsets and social network analysis. This last one is explained in the next subsection.

### Social network analysis (SNA)

In social network analysis data is modelled as a graph, where each data point is an entity, and entities are connected via edges. The importance between edges is measured, and can be used to cluster the network in disjoint communities. Communities are locally densely connected sub-graphs of the social network.

## B.3.2. Predictive learning analytics

Several popular classification techniques used in Educational Data Mining (EDM) are mentioned in [7]. In the following sections these classifiers will be explained and it will be reviewed if it will be possible to use them in our project.

### Decision tree

As the name implies, this classifier uses a tree to classify the data. At each non-leaf node, the decision is made whether to continue with the left or right child node. A teacher might want to know whether a student is done with a learning activity even though the student did not indicate he or she is done. An example decision tree is shown in Figure B.1. If the student answered less than 90% of the questions, continue with the left child node, otherwise continue with the right child node. The leaf nodes determine to what class the data belongs. In this case there would be two classes, one class would indicate the student is not done yet with the learning activity while the other indicates that the student is done with the learning activity.

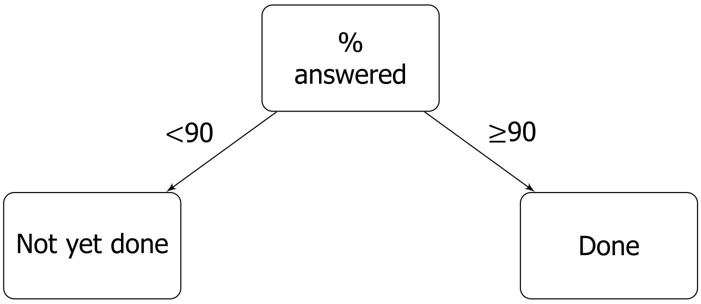


Figure B.1: Example of a decision tree

This method of classification has several advantages and disadvantages. One advantage is that it can be visualized how the decision is reached, while this is not possible for other techniques (e.g., neural networks). Visualizing the decision makes it easier to understand how the decision is reached and allows somebody to reason about what would happen if certain features of the data would be changed. This does however not hold for trees with a high depth, since these have more nodes. An important disadvantage of decision trees is that overfitting to the training dataset can easily occur.

### Random forests

Random forests make use of multiple decision trees to classify the data. The output is the mode of the classes that were output by each of the decision trees. This reduces the chance of the problem of overfitting mentioned in the previous section. Because multiple decision trees are used it becomes harder to understand how the decision is made.

There are different ways of selecting random trees. One way of creating such a forest is described in [8]. The algorithm works as follows:

1. Draw $n$ bootstrap samples from the original data.
2. Build a tree on this bootstrap sample. Choose the best split of a random subset of the bootstrap sample instead of the best split of the bootstrap sample.

The disadvantage of the methods of creating random forests is that bootstrap samples are drawn from the data. For this problem there is not a lot of data available and therefore many of the trees will be the same. As a result, random forests might not work for this problem.

#### Decision rules

Decision rules can be seen as IF-THEN statements. An example is ”IF percentage_answered > 90 THEN Done”. Decision rules are easily interpretable, unless there is a large amount of decision rules used, as is also the case for decision trees. If we would have to choose between decision rules and random forests, we would favor random forests, since we think this will give a better overview.

#### Logistic regression

Logistic regression is similar to linear regression. Logistic regression can be used for binary classification which is needed for this problem. In linear regression a linear function is fit to the data, while in logistic regression a logistic function is fit to the data. In logistic regression the line is fit to the data using the maximum likelihood. In linear regression this is done using the least squares method. The curve of the logistic function indicates the probability of the data point belonging to the positive class (e.g., the student is done with the learning activity). This probability can be used to determine to what class the data point belongs. For example, if the probability is larger than 90% predict that the student is done with the learning activity, else predict the student is not yet done with the learning activity. Compared to a classification technique like decision trees interpretation is more difficult. Logistic regression does however provide probabilities which could be useful for interpretation. If for example a lot of the predicted probabilities are close to 50%, this could indicate that the predictions could be inaccurate.

#### Conclusion

Because of the explainability random forests will be favoured as classification technique. It could however be the case that it is not possible to build random forests due to limited categories of data that are available. This will have to be tested and therefore this can not be determined up front. If it turns out that it is not possible to use random forests, logistic regression will be used. With logistic regression it is not possible to show how the decision is made but it is possible to show what the probability of data belonging to a certain class is.

## B.4. Conclusion

In this report, we determined what would be the best method for making predictions with LA. To achieve this we stated a problem with current LA, which is that gathering useful information from data is difficult and time consuming. It is also explained how LA works and what the important aspects of LA are. In Chapter B.3 we looked at different methods for both standard LA and predictive LA and analyzed their advantages and disadvantages. From this, we have drawn the conclusion that using the random forests classification technique would be most valuable. However, this might not be possible given the restricted amount of data available. If this does not work logistic regression will be used.

# C
# Software Improvement Group evaluation

The Software Improvement Group (SIG) is a part of the TU Delft that evaluates the quality of the code written by students during projects. The main focus of the evaluation during the bachelor end project was on maintainability, since the product is made for a company that needs to be able to work with the code after the project finishes. Two submissions had to be made during this project, one in week 6 and one in week 9. The following sections will discuss the feedback obtained from these two submissions.

## C.1. First submission

The first submission was made on the 31st of May and the feedback on this submission was obtained on June 11th. During the first three weeks of the project, we did our best to keep the code as clean and organized as possible. Comments were written for every part of the code that was not trivial to understand when looking at it and long methods and lines were split up in different smaller parts wherever possible. The score for the submission was 4.3 on a scale of 5, which is above average in terms of maintainability. The maximum possible score was not obtained due to lower scores for unit size and unit interfacing.

Unit size is related to the length of methods and some methods in our code were found to be too long. This was mainly done because the operations in that method needed all of the parameters and thus splitting them would require all the parameters to be sent to the smaller method, which seems unnecessary. After examining the method after the SIG feedback however, some calculations were identified that could be split from the rest of the method. This change will be made during the third sprint.

Unit interfacing is related to the amount of parameters methods have. In the code, there was a method that had 6 parameters, which is above average for a single method. The feedback here suggested to merge some of the parameters that are related to each other into one object and then pass that object as a parameter. This is something that will be taken into consideration and will be changed, if possible, in the third sprint as well.

The last point of the feedback was the lack of (unit)tests in the code. This was hard to do since FbF only did integration tests in the front-end part of the code, so we could not add onto existing tests if we wanted to do unit tests in our parts. The advice here was to at least test the most important parts, so it could be guaranteed that additional changes did not result in unwanted behavior. In the third sprint, we will add some unit tests for the calculations done in the front-end, along with tests for the back-end.

## C.2. Second submission

The second submission was on the 21st of June and feedback for this submission was received on the 1st of July. In the three weeks between receiving the first evaluation from SIG and doing the second submission we tried to process as many points of improvement that were recommended in the first evaluation, in order to improve our score.

The first point of improvement was reducing the Unit Size of some of our written methods. This was done by reducing the length of methods by combining calculations were possible, limiting the number of intermediate values that need to be saved. We also checked whether certain functionality could be extracted from methods and be placed in its own method. In order to keep the SIG score at least equally high we made sure to keep method length short for all new code that would be written during the third sprint.
The second point of improvement was to reduce Unit Interface. This is the number of parameters that a method takes. This was done by creating objects in which the data is held. The objects hold related data, and can be given to a method as one parameter. For a method that takes four parameters an object can be created that holds all data needed for that method. This method then only has to take one parameter, the object, in order to function. By making sure that objects are created for related data we ensure that the complexity of the method is reduced by reducing the number of parameters, making the code easier to understand.
The last recommendation by SIG was to implement tests for our code. No tests had been written yet, so a first set of unit tests for the most important front-end features was written.

The second evaluation did acknowledge that we tried to make improvements to the code based on the recommendations that were done in the first evaluation, but this was not sufficient to improve our score. The remark was mode that Unit Size was not improved enough to make a structural improvement to the code. The number of tests that were written was low for the amount of code that was produced, even when taking in consideration that there were no tests up until the first SIG evaluation.

# D

# Original project description by FbF

## D.1. The challenge

It is incredibly valuable for a teacher to have proper insight into how their students are progressing. They can find which topics students are struggling with, allowing them to revisit such topics; and maybe more importantly, they can also reach out to students who are at risk of dropping out, for whatever reason. However, increasing student numbers—Computer Science at the TU Delft has grown from 120 students to over 800 students in a decade—result in considerably less interaction between teachers and their students. This dramatically reduces the insight teacher have into their students' progress.

Every day, we gather tons of data on student activity within the learning activities teachers organize on FeedbackFruits. When properly analyzed, this data can be incredibly valuable for teachers, maybe even allow prediction of future success (or failure) of students. We want you to step up to the challenge.

We would like you to create a system which is fed student activity, and predicts a student's risk of dropping out or failing a course.

## D.2. About us

We are FeedbackFruits, a vibrant scale-up founded some years ago by students of the TUDelft. We strive to improve education. Last year, we grew out of the Yes!Delft startup incubator and are now located on the FreedomLab Campus in the heart of Amsterdam, a place for companies working on disruptive technologies to call home. Here, you will find people working on AI in healthcare, blockchain in food and agriculture, and the circular economy.

We believe that the latest technologies can help us out on our mission. We use a diverse set of tools, languages, and frameworks, always picking the right tool for the job. Our services are running on a combination of Ruby on Rails, NodeJS, Python, and Scala, with REST APIs, GraphQL APIs, and a Kafka data bus tying these services together.

## D.3. Perks

We offer free lunches, host dinner for the entire team every Tuesday, and a regular Vrijmibo (Friday drinks). Our co-working space FreedomLab Campus is located next to the Artis zoo in Amsterdam. You will get your own Swapfiets in Amsterdam, giving you a great opportunity to explore the city.

## D.4. Other information

We offer free lunches, host dinner for the entire team every Tuesday, and a regular Vrijmibo (Friday drinks). Our co-working space FreedomLab Campus is located next to the Artis zoo in Amsterdam. You will get your own Swapfiets in Amsterdam, giving you a great opportunity to explore the city.

# E

# Information sheet

**Project title**: Predictive learning analytics
**Name of the client organization**: FeedbackFruits
**Date of the final presentation**: 04/07/2019

**Description:** The goal of this project was to create a set of analytics for the interactive presentations tool of FbF that teachers can use to get a better insight in the participation of students and how well they are performing in class. These analytics already exist for other tools that FbF provides, but only very basic information is shown about the interactive presentations so far. Two types of analytics are created for interactive presentations, standard analytics and predictive analytics. Standard analytics give an insight in how students are doing in lectures so far. Predictive analytics use the information on how students are doing so far to make a prediction on how they will be doing in the future. Both analytics can help the teacher get a better understanding on the quality of their course and the effectiveness of lectures that are given.

**Members of the project**:
Name: Murtadha Al Nahadi
Interests: Computational Intelligence, data mining.
Contribution: Testing and refactoring code

Name: Robin Faber
Interests: Artificial intelligence, software development.
Contribution: Front-end of standard analytics and front-end of predicting analytics.

Name: Frank Ooijevaar
Interests: Computational Intelligence, software development.
Contribution: Back-end of predictive analytics.

Name: Wessel Turk
Interests: Back-end development, artificial intelligence, data mining.
Contribution: Front-end of standard analytics, back-end of predictive analytics.

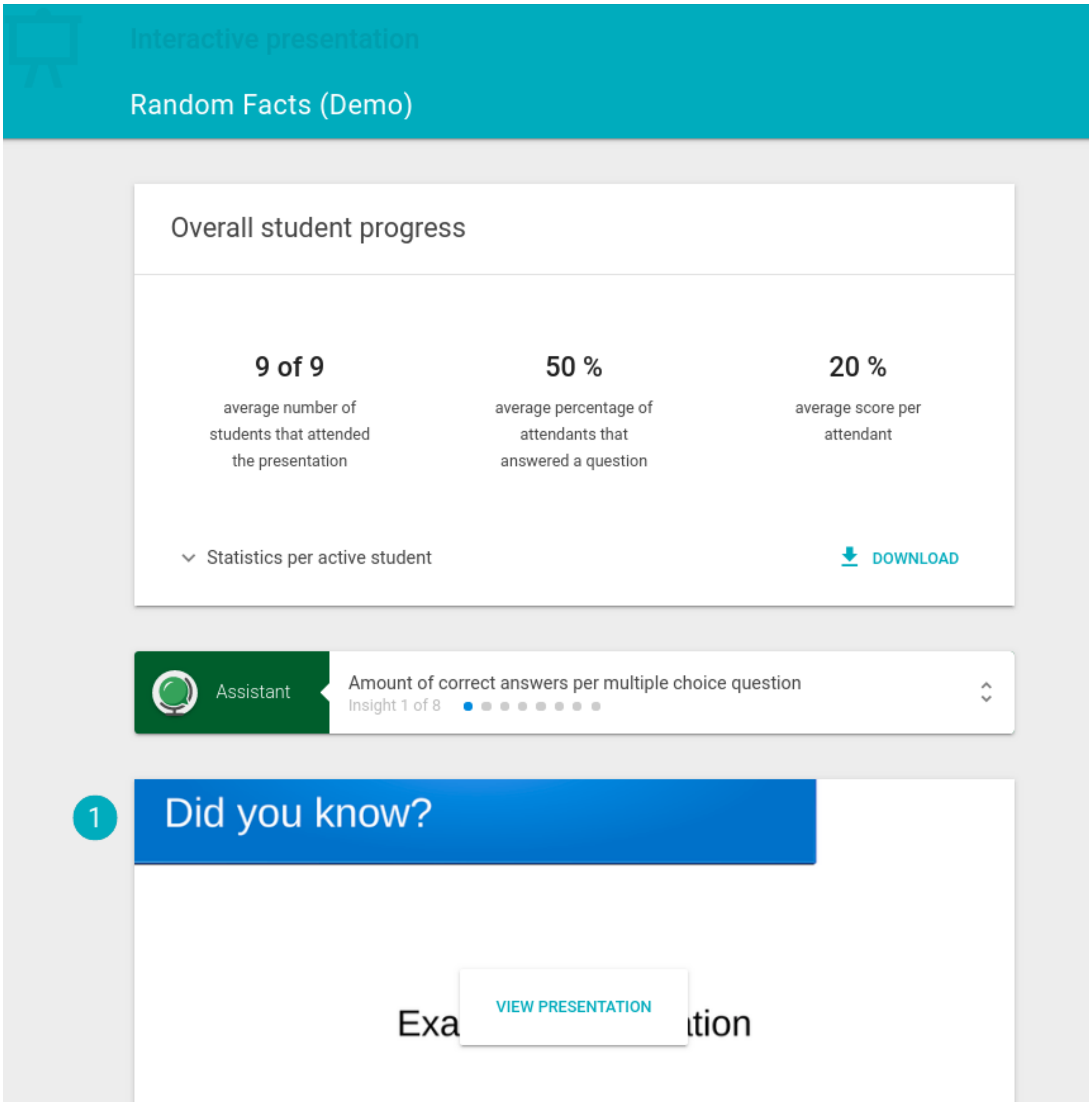
All team members contributed to the back-end of standard analytics, writing the final report and creating the final presentation.

**Name and affiliation of the Client**: J. Verdoorn, head of R & D, FeedbackFruits
**Name and affiliation of the TU Coach**: Neil Yorke-Smith, Algorithmics group, TU Delft
**Contact person**: Wessel Turk, E: wesselturk@gmail.com
*The final report for this project can be found at: http://repository.tudelft.nl*

# F

# Figures

This appendix contains all the figures that are referenced in the report but not important or relevant enough to put them in the place where they are referenced.



Interactive presentation

Random Facts (Demo)

Overall student progress

**9 of 9**
average number of students that attended the presentation

**50 %**
average percentage of attendants that answered a question

**20 %**
average score per attendant

Statistics per active student

Amount of correct answers per multiple choice question
Insight 1 of 8

Figure F.1: View of the website when the analytic component is collapsed

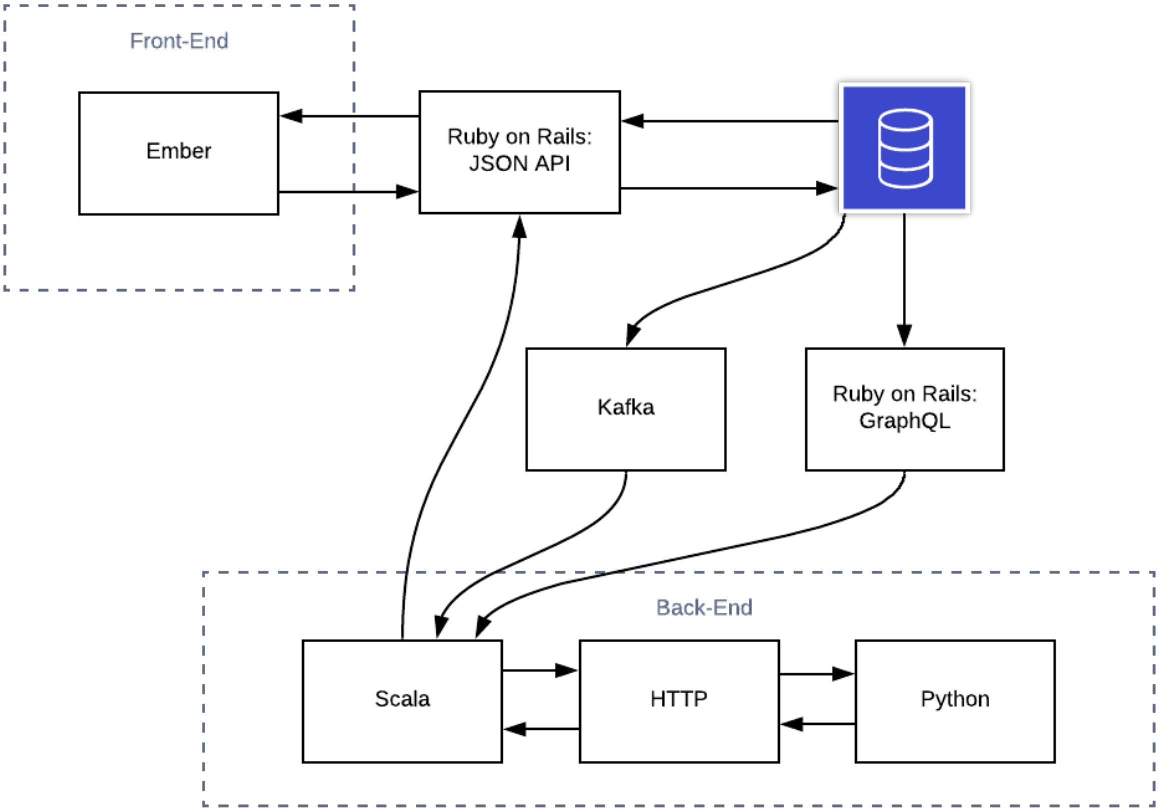


Figure F.2: Architecture diagram of the project code

# Bibliography

[1] J. Saphier, R. R. Gower, and M. A. Haley-Speca, *The skillful teacher: Building your teaching skills*. Research for Better Teaching Acton, MA, 1997.

[2] D. Clegg and R. Barker, *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[3] M. A. Chatti, A. L. Dyckhoff, U. Schroeder, and H. Thüs, "A Reference Model for Learning Analytics," *International Journal of Technology Enhanced Learning*, vol. 9, no. 1, pp. 1–22, 2012.

[4] A. Essa and H. Ayad, "Student Success System Risk Analytics and Data Visualization using Ensembles of Predictive Models," p. 158, 2012.

[5] 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," no. March, pp. 267–271, 2017.

[6] R. Mazza, *Introduction to information visualization*. Springer Science & Business Media, 2009.

[7] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning analytics*. Springer, 2014, pp. 61–75.

[8] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R news*, vol. 2, no. December, pp. 18–22, 2002.