# A Study on Mining Student Data Across Diverse Educational Systems

ジヘド, マクルフ

出版情報：Kyushu University, 2020, 博士（学術）, 課程博士
バージョン：
権利関係：

Kyushu University

Graduate School of Information Science and Electrical Engineering
Department of Advanced Information Technology

# A Study on Mining Student Data Across Diverse Educational Systems

*By*

JIHED MAKHLOUF

*A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy*

*Supervised by*

Assoc. Prof. Tsunenori Mine

February, 2021

**JIHED MAKHLOUF**

*A Study on Mining Student Data Across Diverse Educational Systems*

**Supervisor:** Assoc. Prof. Tsunenori Mine
**Advisory Committee:** Prof. Yutaka Arakawa, Prof. Sachio Hirokawa, Prof. Tsunenori Ishioka, and Prof. Atsushi Shimada

**Kyushu University**
Graduate School of Information Science and Electrical Engineering
Department of Advanced Information Technology

بسم الله الرحمن الرحيم

*(In the name of allah the most gracious, the most merciful)*

# Abstract

The growing usage of digital technologies in different fields is producing a far-reaching impact on our lives. Similarly, educational institutions are increasingly investing in Information and Communication Technology (ICT). This leads to a transformation in educational practices, research, and applications. In fact, the usage of ICT in education is empowered by a multitude of educational systems that keep producing a large quantity of data. Such data are analyzed for different purposes like understanding and improving students' learning.

Within this thesis, I present a set of studies where I used different types of data gathered from diverse educational systems. In each study, the aim and the outcome of the research are different. Thanks to this diversity, I can analyze both objective and subjective students' data. The students' objective data are implicitly gathered from an Intelligent Tutoring System and from an e-book reading system. The subjective students' data are explicitly gathered using questionnaires after each lesson. Therefore, the objectives and outcomes of our analysis are organized accordingly.

In the first study, I use a dataset gathered from an Intelligent Tutoring System for high school mathematics. The system gathers clickstream data of the students' usage. The objective is to predict which student will pursue a career in a STEM (Science, Technology, Engineering, and Math) related field. The models that I build are robust and generalize well to different distributions of students. Moreover, in my analysis, I prove that aggregating the data by school can improve the performance of the models. On top of that, I propose different ways of analyzing the data that opens the gate to more discussions about educational digital system designs.

In the second study, I use objective students' data as well. However, the data is gathered from an e-book reading platform where the students can access the lecture materials. The system provides many useful functionalities and stores students' usage data. I investigate the students' reading behaviors while using their grades as a validation of my analysis approach. I firstly propose to detect the students' reading sessions and investigate the optimal score threshold for considering highly performing students. Following that analysis, I examine the difference in the students' reading behaviors when they are attending the class compared to when they are accessing the materials outside of the lecture time.

In the third study, I explicitly gather students' subjective data using a questionnaire composed of predefined questions about their learning activities. I propose two research objectives in this study. Firstly, I aim at producing models that can automatically assess the students' learning experience using their questionnaires' input. Secondly, I design a system that evaluates the students learning activities to give them immediately the appropriate feedback.

While fulfilling all these research objectives, this thesis provides another contribution related to machine learning models. In fact, I introduce a unique procedure of training the machine learning models and comparing the different features engineering approaches by using genetic programming. In each study, I investigate several patterns and their effect on the prediction models by means of features engineering. Then, I use genetic programming to find the best machine learning settings for each features engineering approach before comparing them.

# Acknowledgment

An entire book won't be enough to thank all the people to whom I am grateful. Instead, I want to dedicate this part to an incomplete list of special people who had an impact throughout my journey in writing this dissertation.

I would like to give my profound gratitude to my parents and my brothers. Their presence, unconditional support, and happiness are my driving force and guiding light in times of darkness. I would not be who I am, now, without them.

It goes without saying, that I have the utmost respect, gratitude, and admiration toward my supervisor, Assoc. Prof. Tsunenori Mine. This work would not be possible without his wise guidance. He instinctively knew how to put me in an environment where I could conduct research effectively and ethically. I owe him an immeasurable debt.

I am also honored to have Prof. Yutaka Arakawa, Prof. Sachio Hirokawa, Prof. Tsunenori Ishioka, and Prof. Atsushi Shimada as advisory committee members. Their comments, advices, and suggestions added different dimensions to the research works that I conducted. Not only, they did point out the research aspects that I could improve, but also, they broadened my viewpoint with their valuable feedback. I hope that this dissertation will meet and exceed their expectations.

Finally, I was blessed to be surrounded by brilliant researchers and faculty members that inspired me with a special mention to Assist. Prof. Shigemi Ishida, Assoc. Prof. Yasutaka Kamei, Assist. Prof. Sato Ryosuke, and Prof. Naoyasu Ubayashi. Moreover, I would like to give my sincere thanks to all laboratory members with whom I shared memorable times especially, Mr. Mansur As, Mr. Billy Dawton, Mr. Ristu Saptono, and Mr. Kohei Yamaguchi.

# Contents

# List of Figures

# List of Tables

# List of Listings

# Introduction 1

## 1.1 Background

The implementation of educational software systems was not straightforward. Compared to other fields and industries, education is still lagging behind when it comes to adopting and taking advantage of the recent advances in Information and Communication Technologies (ICT) [42]. In fact, the usage of ICT in education went through different phases. The initial reactions from educators was skepticism about the advantages, effects and usability of educational software systems [71]. However, it quickly became clear that using digital technologies in education would have a positive impact on the learning and teaching outcomes [11].

Nowadays, educational software systems are an integral and ubiquitous part of students' learning in different educational institutions. Moreover, they cover all levels of learning, from pre-school to higher education. However, the usage, functionalities and outcomes are different at each stage. Therefore, we have seen the birth of many different systems and platforms that support education. Names like ITS (Intelligent Tutoring Systems), LMS (Learning Management Systems) MOOC (Massive Open Online Course) are more frequently used and people start to realize and take advantage of their power.

Along with providing useful functionalities, these educational software systems gather data about the students' usage and behavior. This collection of data allow different types of research and analysis. Not only the type of data used is different, but also the outcome of the analysis and research is different too. This diversity resulted in various topics of modeling students learning, performance predictions and many other applications. Therefore, it is in this context that I carry out my research. In fact, I use three different educational systems and two types of educational data. In each analysis, the objective and motivation of the research cover a particular aspect of the students learning and modeling.

## 1.2 Research Objectives and Contributions

There are different applications of and research ongoing using educational data. In this dissertation, I explain my work on three different topics that cover many applications of educational data mining. Moreover, I used different types and sources of data. In the first part, I use students' objective data. In the second part I use students' subjective data.

The students' objective data is composed of numerical data implicitly gathered from their usage of two different educational systems. It is implicitly gathered in the sense that the students' were not requested to input this data. But it was collected by the educational software system while the students were using it. However, the students' subjective data is explicitly collected by using a questionnaire. The students were requested to write their comments and answer 5 predefined questions.

In this dissertation, I work on three topics. In each one, I will explain the difference of the data used, the research objectives, the methodology and the research outcomes. The first two topics are related to the students' implicitly gathered objective data, while the third topic is carried out by analyzing students' explicitly gathered subjective data.

The first study is related to predictions and student modeling. In fact, the predictions are not associated to students' performance. The main goal is to predict the future career of the student. The starting point is a longitudinal study conducted by a team of researchers and developers of an ITS for high school mathematics called ASSISTments [1]. They collected data about the students' usage of the ITS. Then, they tracked the students' first career position after college. The dataset was anonymized and released in a public competition with the objective to predict, solely based on the data of the students' usage of the ITS, which student will pursue a career related to STEM (Science, Technology, Engineering and Math) field. While achieving this objective I made the following contributions:

- Predicting with a decent level of correctness which student will pursue a STEM-related career
- My models are robust and generalize well to different distributions of students
- I investigate the influence of the school on the prediction performances
- I prove that aggregating students usage by skills improves the performance of the model compared to aggregating based on the problem type

[1] https://new.assistments.org/

In the second study, I use a dataset composed of click-stream log files of students' usage of an e-book reading system. This e-book platform is called BookRoll[2]. The professors upload the lessons materials into the platform and students can access them anytime anywhere. There are several useful functionalities available for students when they access a document. For example, they can bookmark a page, write a memo, or set up a marker. All of this usage data is stored, anonymized then released accompanied by the students test scores. The main objective is to investigate the students' reading behaviors using the dataset. Since the research objective is open, I applied different approaches when investigating the students reading behaviors. This study covers the research topic of student modeling in EDM. The main contributions are as follow:

- I build a detector of students' reading sessions based on the document usage and the inactivity of the students
- Despite the skewed nature of the test scores, I carefully inspect the best test score threshold that maximizes the difference of the behavior between highly performing students and the less performing students.
- I investigate the difference in the behavior of the students when they access the materials during the lesson compared to their reading attitude outside of the lecture time
- I prove that the difference documents is an aspect that should be considered since it improves the prediction performances

In the third study, I use textual data coming from students' answers to a questionnaire where they have to provide their own assessment of their learning activities. The questionnaire is composed of five predefined questions taking into account temporal informations about the students' activities. The research objectives are two-fold. The first part is to automate the process of assessing the students' learning activities by predicting their learning experience. In the second part, I build models to automatically give feedback to students containing the assessment of their learning activities. The research objectives are related to topics of providing feedback to both the professors and to the students using text mining. The main contributions in this study are listed below:

- Building an automated assessment model of the students learning experience based on a five-scores likert scale

[2] https://www.let.media.kyoto-u.ac.jp/en/project/digital-teaching-material-delivery-system-bookroll/

- Proving that contextualizing the students' comments using the question type can lead to significant improvements of the prediction models
- Automating the process of giving feedback to students

Along with the research objectives and contributions stated above, I introduce a rare method of investigating the effectiveness of different features engineering approaches. In fact, during my analysis in all of the studies I try to find detect some patterns with sophisticated features aggregation. Then I test the effectiveness of this features transformation by comparing it to a baseline approach. However, to make sure that the comparison is fairly conducted, I search for the best performing machine learning method and the corresponding hyper-parameters for each approach. To achieve this objective I use genetic programming in the search process.

## 1.3 Importance of the Research

Each topic of research has its own importance. And the investigations that I carried out in this dissertation provide some insights that can be used to improve the learning outcomes of the students, and give the different educational stakeholders better understanding of students learning and behavior.

The first study is related to STEM(Science, Technology, Engineering, and Math) education and careers. STEM fields play an important role in the growth of nations economies. Therefore, it is very helpful to encourage and increase the students to pursue a STEM related education and career. In my analysis, I am able to predict the career outcome of students solely based on their learning behavior in an Intelligent Tutoring System. This analysis will give educators and stakeholder a powerful tool to encourage, and enhance STEM education by detecting students that lose motivation and interest toward STEM fields for different reasons.

In the second study, I investigate students reading behavior. This topic is important and very influential. In fact, students readings are an important aspect of their learning process. With the digitization of the learning materials, I could use the data to detect their reading sessions and investigate their reading behaviors. This analysis can help educators predict students performances, detect low performing students and students who are losing interest in the topic. As such, educators can give the appropriate guidance and interventions to help motivate the students and encourage them to perform better.

In the third study, we use questionnaires to get students own-assessment of their learning activities. For some time, the professors had to read and give feedback manually to the students. This was a very time-consuming task, especially when the professor is in charge of many classes. Automating the process of giving feedback and detecting the students' learning experience and activities will help the educator focus on which is more important. Therefore, they can provide better intervention and adapt the teaching content and methodology.

In the bigger picture, this dissertation exposes the usage of different types of educational data, with different types of applications and research objectives. The multitude of data sources enable different types of analysis, that we explain in details within each chapter.

## 1.4 Chapters Outline

The rest of this dissertation is organized as follows: Chapter 2 introduces the fundamental concepts about the background of this dissertation. Moreover, it explains the methodology used in different analysis and the usage of genetic programming. Later, Part 1 is composed of chapter 3 and 4 in which I use the implicitly gathered objective students' data from two different systems. Accordingly, chapter 3 discuss the first study using data coming from an ITS. The objective is to predict which student will pursue a career in STEM-related fields. In chapter 4, I use data from students usage of an e-book reading system. I investigate the students reading behaviors. The second part of this dissertation provides details about my study using explicitly gathered students' subjective data. Part 2 is composed of two chapters. Chapter 5 explains the steps I have followed to automate the process of evaluating the students learning experience based on their freely-written comments. In chapter 6, I use the same comments data to build an automated feedback system to the students. Finally, in chapter 7, I provide conclusions and detail a list of potential improvements fear each study.

# Fundamental Concepts 2

Among the advantages of the educational software systems is the collection of usage data. That data will be used by different stakeholders for the purpose of improving the students' learning. In fact, researchers are using the data generated by the educational software system to conduct different sorts of analysis using data mining techniques [56].

## 2.1 Data Mining and Education

Not only educational software systems could gather large quantities of data, but also it could collect different types and forms of data. Therefore, different trends of research topics exploited the available datasets. Furthermore, more researcher communities are formed which resulted in the creation of subsequent conferences. Conference themes are related to Intelligent Tutoring Systems, Artificial Intelligence in Education, Technology Enhanced Learning, Computer Supported Education and Advanced Learning Technologies. More recently, two more research themes started to gain traction. The first one is called Educational Data Mining (EDM), and the second is called Learning Analytics and Knowledge (LAK). They both use data mining and statistical methods and apply them on educational data. Therefore, they are similar in that aspect, nonetheless the objectives and methodologies are different [57, 4, 62].

In fact, EDM is "an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in" [62]. Meanwhile, as defined in its first conference, LAK is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs.

The differences do not manifest themselves only in the definitions. In fact, multiple studies and publications shaped the interest and emphasis in each one of them. Siemens and Baker [62] dressed a list of differences between EDM and LAK in 5 aspects. Table 2.1 summarizes the differences between EDM and LAK according to these 5 aspects.

| **Aspects** | **Differences** |
|---|---|
| Discovery | For LAK, human judgment is the goal and they use automated discovery to attain it. For EDM it is the reverse, the automated discovery is the key and the human judgment is a tool to achieve it. |
| Reduction & Holism | In LAK they focus more on understanding systems as wholes while in EDM they analyze the individual components of the systems. |
| Origins | LAK is closer to intelligent curriculum and interventions while EDM has its roots from educational software and student modeling. |
| Adaptation & Personalization | In LAK the focus is on reporting and informing the educator. For EDM automated adaptation is more important. |
| Techniques & Methods | SNS analysis, sentiment analysis and similar methods are used frequently in LAK. For EDM, methods like classification, clustering and discovery with models are more common. |

Tab. 2.1.: Main differences between LAK and EDM.

Despite the differences in several aspects, both LAK and EDM cover a wide range of research topics, tasks and applications. Also, several topics were commonly investigated by researchers in LAK and EDM [13]. There have been several classifications of research topics. For example Baker suggests four key areas of application for EDM [5]: improving student models, improving domain models, studying the pedagogical support provided by learning software, scientific research into learning and learners. However, another classification of EDM subjects was proposed by Castro [12]: applications dealing with the assessment of the student's learning performance, applications that provide course adaptation and learning recommendations based on the student's learning behavior, approaches dealing with the evaluation of learning material and educational web-based courses, applications that involve feedback to both teacher and students in e-learning courses, and developments for detection of atypical students' learning behaviors. A more recent study classified the research topics based on the publications made by the researchers [56]. The authors found a distinct trend and a clear separation between the applications of data mining in eduction. There are 11 types of applications that I will list and shorty define in the next section.

### 2.1.1 Common Application Topics of Data Mining in Education

#### Analysis and visualization of data

The main objective of the data analysis and visualization is to emphasize important information which gives a strong support for decision making. Visualization uses graphics techniques to simplify the analysis and understanding of data [35]. Statistics and visualization are the two main components.

#### Providing feedback

There are two ways of feedback. The first type of feedback is given to students to give them assessment about their performance, what they are doing wrong and why. The feedback to students can happen in various ways. It can be related to the students' answers to an exercise or after an analysis of their behavior or performance. The second type of feedback is provided to the educators. Most of the time, this time of feedback is used in decision making. It is different from the data analysis and visualization since data visualization is mainly showing basic informations in an easy way, feedback in the other side aims at giving more complex relationships and information [57].

#### Recommendations for students

The main idea is to adapt the learning contents, interfaces, personalized activities and their orders to each individual student. Therefore, providing recommendations accordingly [57].

#### Predicting student performance

This is one of the oldest and most active research topics and activities of data mining in education. The main purpose is to predict a certain unknown value that represents the performance of the student. This value can be a test score, a final term score or another value that assess the student knowledge [5].

### Student modeling

With student modeling, researchers can build cognitive models of the student. This models encapsulates the student skills and knowledge and consider its characteristics such as motivation, affective states, satisfaction learning styles and so on [25].

### Detecting undesirable student behaviors

One of the purposes of detecting negative student behavior is to intervene early and give the student the appropriate help. There are different types of undesirable student behaviors such as low motivation, cheating, dropping out, mistakes, distraction etc [57].

### Grouping students

The goal is to create groups of students according to their customized features, personal characteristics, personal learning data, and so forth. Then, this grouping can be used by stakeholders to build personalized learning system adapted to each group of students [2].

### Social network analysis

The purpose is to investigate relationships between individuals rather than their individual attributes and properties. A social network is composed by a set of people that are interconnected. Those connections represent a social relationship such as friendship, family bonds, or any sort collaboration [24]

### Developing concept maps

A concept map is a graph that shows the relationships between concepts and expresses the hierarchal structure of knowledge. The goal here is to automate the process of creating / developing the concept maps for the sake of helping teachers and educators [57].

### Constructing courseware

The main goal is to automate the process of creating the courseware and learning contents. Meanwhile, it also promotes the reuse and exchange of the existing learning resources across differents users and systems [57].

### Planning and Scheduling

The main purpose is to improve several traditional educational processes by helping the planning resources allocation and future courses. Also, a particular interesting application is helping students course scheduling and enhance the development of the curriculum [57].

## 2.1.2 Data Mining Methods Commonly Used in Education

There have been a wide range of data mining methods that have been used in education. While most of the methods are commonly used in other domains, there are several methods that are unique or differently used in educational settings [4]. Through the last several years there have been different categorizations of the data mining methods used in education. Depending on the application and the topic as seen in the previous sub section, some methods can be more used than others. In this section I do not provide an exhaustive list of data mining methods used in education, rather I detail the most used ones. There are different reviews that classify the EDM methods more thoroughly [5, 55, 57]

### Classification

Classification problems can be divided in two categories. Binary and multi-class classification. In the first category, the predicted variable has only two possible values. In the second category, the predicted variable has more than two possibilities. An important aspect in educational settings is the interpretability of the models. Therefore, "black-box" models such as neural networks are not particularly appreciated until recently. Moreover, in EDM, it is encouraged to apply cross validation on multiple levels. For example, apply cross validation on the student level to make sure that the model generalize well to new students, also apply cross validation on the learning content to verify its performances with new materials and topics. Classification is used in many EDM applications, mostly in predictions.

### Regression

In regression problems, the predicted variable has a continuous value. The most used regression model in EDM is linear regression. Even if support vector machines and neural networks have good results in different fields, they are still not used so frequently in EDM. Similarly to classification, regression methods are used a lot in EDM tasks that involve predictions among other tasks.

### Relationship mining

In main objective in relationship mining is to discover relationships between variables. For example it can detect which variables in a dataset are strongly associated with another variable of interest. In overall, there are four particular relationship mining techniques that are used in EDM. The first one is called association rule mining, which aims at finding simple if-then rules. The second method is called sequential pattern mining. The objective is to find temporal relationships between a set of events. The third method is the correlation mining. It is well known in statistics. The goal is to find positive or negative linear correlation between variables. Finally, the fourth relationship mining method is the causal data mining. The target is to investigate if an even is the cause of another event.

### Clustering

The objective of clustering is to group automatically data points that are similar in some dimensions. It is an unsupervised learning algorithm, therefore it is very useful when the data points categories are not know beforehand. The clustering can happen in different levels. For example, schools can be clustered to find similarities and differences between them. And using more fine-grained data we can cluster students to detect the common and different attributes that characterize them.

### Discovery with models

The first step in discovery with models is to develop a model of some phenomenon, usually with clustering or classification or regression. When the model is robust enough, it can be used as a component for a another analysis. For example, the values resulting from the first model are used as predictors for the second model. One of the most frequent cases of discovery with models is using the students' knowledge prediction model as a base for different other analysis. The Bayesian Knowledge Tracing (BKT) is a model that estimate the latent knowledge of the students [14]. It has been applied in different settings then used as a component for discovery with models [4].

## 2.2 Text Mining and Education

Usually the values used in the previously seen methods are numerical. However, there is another type of data that can contain very useful information. Textual data can be extremely helpful. Text mining techniques have been applied in different fields such as finance, business and medical, to cite a few. Text mining techniques can be very effective in educational settings. Basically, text mining is the process of extracting information and knowledge from textual data. The sources of textual data are diverse. Therefore, text can be heavily unstructured and adequate mining techniques are needed to process it.

In general, the applications of text mining are similar to the applications of typical data mining. However, textual data can enhance the performance of the analysis in many research topics. Moreover, using textual data opens the door to another set of applications that are not possible with purely numerical data and data mining techniques [20].

### 2.2.1 Common Applications of Text Mining in Education

There have been some surveys about the common topics of text mining in education [30, 20]. The most recent literature review about text mining in education regrouped the research topics into six main applications as follow [20]

#### Evaluation

One of the most used application of text mining in education is the evaluation of students' performance. It is similar to the students prediction in data mining, but the main difference is that the evaluation in text mining is applied especially to essays and online assignments.

### Student support

The collaboration among students is an essential part for pedagogical success. Therefore, it is necessary to engage them in online platforms, especially in the case of distance learning. One of the most frequent use case is providing help to students during their writing of traditional essays or academic manuscripts. Another type of support is encouraging students to collaborate and keeping the students motivated to avoid dropping out.

### Analytics

The purpose here is to provide the educators with different sets of informations to help them give the appropriate feedback to the students. Usually the sources of textual data are writings such as assignments, also forums, chats or emails.

### Question or content generation

The goal is to build helper systems for the educators. Basically, Automatic Content Generation (ACG) is able to generate content related to any given topic. Meanwhile, Automatic Question Generation (AQG) is capable of producing questions related to a particular content. Usually, the reference documents such as textbooks and teaching resources are used for this purpose.

### Student feedback

Generally, giving feedback is not a unique application for text mining, compared to data mining in education. However, using textual data adds another dimension to the feedback. In fact, among the objectives of student feedback is to automate the process of giving insight to students to improve themselves. There are two ways. The first one is to send the feedback directly to the students. The second approach is to provide a helper system to the educators when they elaborate the appropriate feedback to students.

#### Recommender system

Similarly to previous applications, recommender system are enhanced by the usage of textual data. It takes into account different aspects and dimensions that traditional data mining approaches do not.

### 2.2.2 Text Mining Methods Commonly Used in Education

Since the nature of textual data differs from numerical data, the methods and techniques used are different as well. The most frequently used text mining methods are listed as follow [20]

#### Classification and clustering

Similarly to numerical data, textual data can be classified or clustered. However, it is necessary to proceed to textual features engineering before applying the usual machine learning methods to implement the clustering or classification.

#### Natural language processing

Natural Language Processing (NLP) is the process of manipulating natural language data including textual data or speech data. There are various algorithms for applying a semantic or syntactic analysis. Similarly to other fields, education is taking advantage of the recent breakthrough and advances in NLP to improve the performances in each application of text mining.

#### Information retrieval

Information retrieval (IR) is the process of finding documents within a large set of text, or organizing documents according to their themes or other types of characteristics. In educational settings, IR is mainly used to improve collaboration and online discussions in e-learning.

### Text summarization

The objective is to produce a shorter version of a document keeping mainly the essential informations. Text summarization is particularly helpful in shrinking the large amount of textual data present in digital libraries, scientific papers or many other data sources. There are two types of summary. The first one is called extractive summary, where the main idea is to pick the most significant sentences as they are. The second type is the abstractive summary. They try to improve the coherence between summarizing sentences by removing redundancies.

## 2.3 Research Methodology

Across the different topics of this dissertation, I introduce a novel methodology of building the machine learning models. In fact, one of the contributions of my work is a methodology for features engineering to find patterns and investigate some aspects of the dataset. In each study, I begin by making some hypothesis and look for the ways to validate it using the features available in the dataset. Then, I proceed to different transformations of the features of the dataset. This process is called features engineering. By applying the features engineering, I prepare for the analysis of the hypothesis and its effect on the machine learning models' performance. Finally, to validate the effects of the hypothesis, I compare it with a baseline approach in which the features engineering according to the hypothesis is not applied. In the comparison phase, I search for the best machine learning settings for each approach. It means that I compare the approaches by using the best performances that they can achieve. The purpose is to mitigate the effect of fixing a machine learning method and apply it to the compared approaches. Therefore, I used a particular AutoML technique to search and find the best machine learning settings for each approach. This AutoML technique is based on Genetic Programming.

## 2.4 Genetic Programming

Genetic Programming (GP) is an evolutionary computation technique derived from genetic algorithms in which program instructions are encoded into a population of chromosomes. The goal is to evolve this population using genetic operators to constantly update the population until a predefined condition is met. The update of the population is done using two famous genetic operators called crossover and mutation. Crossover is used to diversify the research in the research space by taking some parts of the parent individuals and mixing them into the offspring. On the other hand, mutation is the process of updating only some part of an individual and it is used to maintain the actual diversity, in other words, intensify the research in a certain area of the research space. The population is evolving from one generation to another while keeping the fittest individuals in regard to one or many objectives [53].

### 2.4.1 Basic Concepts

GP relies on a set of concepts that mimic biology in order to run and optimization process.

#### Individuals

An individual is the representation of a given solution to the problem that we want to optimize. An individual is composed by two forms of solutions: The chromosome and the phenotype. The chromosome is the "raw" information or value. The phenotype is the description of the chromosome in terms of the modeled solution. A chromosome is composed by a set of genes. Genes are the elementary value that composes a solution. Genes can hold a solution without being themselves the solution itself.

#### Fitness

The fitness of an individual is the result of the application of the objective function using the values of its chromosome. Therefore, the chromosome have to be decoded first. Then, the objective function is evaluated using the decoded values. The fitness does not only determine which is the best individual but also measures how far is the individual compared to the optimal result.

#### Population

A population is simply the collection of individuals that are being tested. The whole process starts with the initial population and them keep updating the population until the end of the process. The population size is a parameter that have to be defined. In general, the larger the population the faster we can explore the search space, however it requires more computational power and memory.

### Breeding

Breeding is the core concept of genetic algorithms. The breeding starts by selecting the parents to generate a new individuals, then operating the crossover and the mutation operations then replacing the individuals to form the population. There is a strategy adopted for each step of the breeding. The parent selection can be done randomly or it can be done after ranking the individuals and selecting the best ons. The crossover operation consists of taking some parts of each parent and recombining them in the new individuals with the hopes of generating better offspring. Figure 2.1 provides a simplified example of a crossover operation on a tree-based GP data structure. In this example, the crossover happens in the right branch of the first parent and in the left branch of the second parent. Then they are recombined into the offspring individuals.

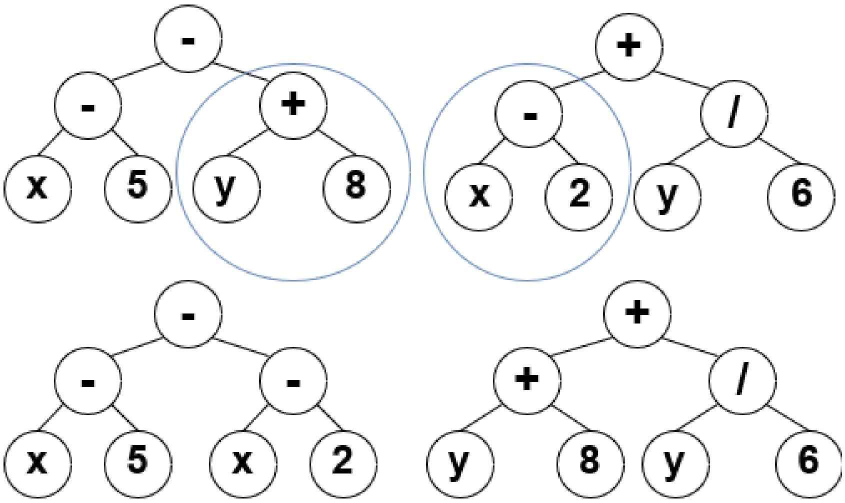


Fig. 2.1.: Crossover operation

On the other side, mutation applies a slight changes the genes of an individual. It is useful to avoid local optima by disturbing the genetic information and adding some noise to the solution. Figure 2.2 shows an example of a mutation operation. A small change happens in the gene represented by the bottom-right leaf.

Finally, after the generation of the new individuals, there is a choice to be made. Not all the individuals can integrate in the new population since the population size is fixed. Therefore, some individuals have to be discarded. There are several strategies such as automatically discarding the parents, keeping the best performing individuals only, or randomly choosing which individual to keep.

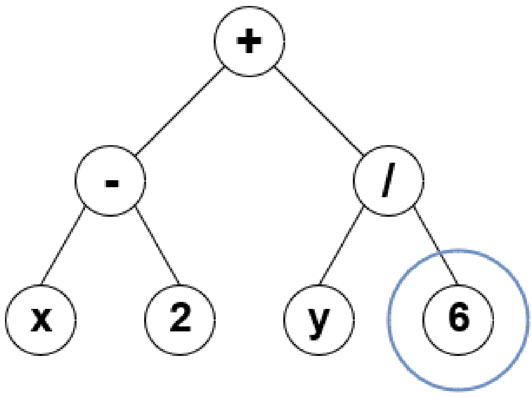


Fig. 2.2.: Mutation operation

### Stopping criteria

The whole repetitive process of updating the population is executed until a predefined set of conditions are met. The stopping can be fixed after a fixed number of generations, or a fixed execution time. Also, it can be stopped if there is no improvement of the fitness for several generations. Therefore, the stopping criteria is a parameter that have to be defined before starting the GP experiment.

## 2.4.2 Overview of the execution

The most important step of the GP is well defining the problem and providing a valid encoding of what should be a solution to be optimized. Then the search process will proceed by it self as follow:

```
create() the initial population
repeat:
  execute() the objective function to evaluate the fitness on
    each individual
  select() the subset of individuals for mating
  create() the new individuals by means of crossover and mutation
  discard() some individuals and keep the others in the
    new generation
until the stopping criteria
return best individual(s)
```

Listing 2.1: Execution of genetic programming.

### 2.4.3 Genetic Programming in Machine Learning

With the growing usage of data science techniques across many fields it became necessary to provide the right set of tools required to achieve the expected results. Beside the libraries and softwares used to apply data science, there is a trend of automating the process of building the proper machine learning model. Automated Machine Learning (AutoML) systems are designed to assist industries and researchers in the task of selecting the rights features and machine learning algorithm with its respective hyper-parameters. Using GP to optimize this task by investigating the huge search space represented by the countless possibilities and combinations of features selection and machine learning methods [32].

A machine learning workflow is generally composed by the following steps

- Initial data exploration
- Cleaning and preprocessing the dataset
- Apply different types of transformations to the features such as scaling, normalization or decomposition
- Proceed to the features selection using several strategies
- Selecting the machine learning algorithm and training it
- Validating the performance of the model using held-out data.

In AutoML, these steps are being automated by generating a so-called pipeline. A machine learning pipeline is the list of the successive operations applied to the dataset's features and to the machine learning method through a search into the combinations that give the best results.

Throughout the research topics of this dissertation, I used a tool called TPOT (Tree-based Pipeline Optimization Tool) which uses genetic programming to find the best machine learning pipeline for a particular dataset. However, I did not use all of its functionalities as I was interested in automating, only, the models training phase without automating the features transformation or selection steps. There are two reasons. The first reason is that my objective was to discover and validate some patterns in the datasets by means of features engineering and selection. Therefore, I did not need the tool to change any feature transformation that I apply. Moreover, the AutoML systems can generate strong performing machine learning pipelines but with a very weak interpretability. The second reason is that the process of applying GP to investigate the whole search space is time-consuming and by removing the features engineering steps I reduce the time and complexity of the pipeline. Figure 2.3 shows the automation process by TPOT. The parts that are included in the red box are the functionalities that I used from TPOT[1] [32, 47].

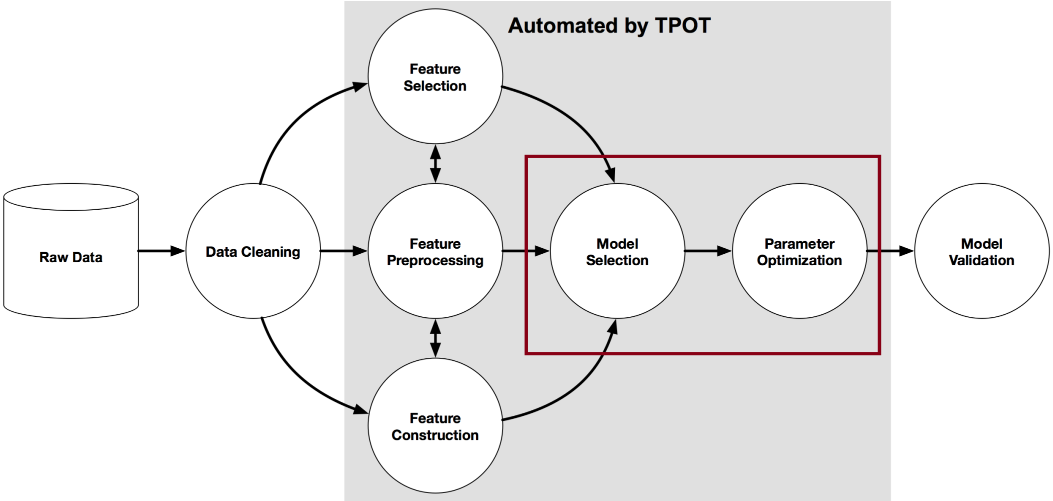


Fig. 2.3.: Automation Process (picture updated from TPOT documentation)

To run a GP experiment with TPOT there is a set of hyper-parameters to define beforehand. Table 2.2 explores the principal hyper-parameters that we have to initialize. The Generations count is the number of iterations of the whole optimization process. A bigger number gives better results but also takes more time to finish. It is the stopping criteria that was exposed earlier. The Population size is the number of individuals which will evolve in each iteration. The offspring size is the number of individuals that are supposed to be generated from the previous population using the genetic algorithm operators. Mutation and Crossover rates are the probabilities of having respectively a Mutation or a Crossover operation to evolve one or more individuals. The method used to measure the score is defined in the scoring hyper-parameters. In fact, in GP for machine learning we encode the pipeline in individuals. The fitness function is the performance of that individual after training and testing the pipeline. Therefore, as an example we can take the accuracy of the pipeline as an objective function to be maximized. Finally, the TPOT tool gives the possibility to cross-validate the pipelines internally.

[1] http://epistasislab.github.io/tpot/

Tab. 2.2.: Genetic Programming Hyper-parameters

| Hyper-parameter | Common Value |
|---|---|
| **Generations count** | 100 |
| **Population Size** | 100 |
| **Offspring size** | 100 |
| **Crossover rate** | 0.1 |
| **Mutation rate** | 0.9 |
| **Scoring** | Accuracy |
| **Cross-validation** | 5 Folds |

### 2.4.4 Summary

The usage of ICT in education provides many benefits. One of the benefits is the collection of students usage data. This opened the gate to sophisticated analysis of the students behavior and performances. Different research interest and fields related to educational data mining and learning analytics emerged. Different categories of applications using educational data exist and they have tangible impact on the students learning outcome and also in the improvement of the educational settings overall. As seen above, my research topics cover different applications of data mining and text mining in education. Moreover, my contributions do not stop only at the research topics using educational data. In fact, I introduce a unique approach to analyzing and finding patterns in the datasets by means of features engineering. Validating the patterns is done by comparing its associated features engineering approach with a baseline approach. To nullify the effects of the machine learning method I search for the best machine learning settings for each approach separately by using genetic programming. This allowed the comparison to be fair since it examine the best possible performances of each approach.

# Part I

## Implicitly Gathered Students’ Objective Data

# 3 Career Prediction Using High School Data

## 3.1 Background and Related Work

Science, Technology, Engineering, and Mathematics (STEM) fields are regarded worldwide as the building blocs for a nation's economy. Yet for several reasons, the number of open positions does not match the number of workers ready to take these positions. In fact, just in the United States, employment related to STEM occupations has grown a lot faster than for other non-STEM occupations. Over the last decade, STEM occupations have increased by 24.4% compared to "only" a 4% increase in non-STEM occupations [41]. However, STEM positions require the candidates to have appropriate STEM skills that are acquired in the course of completing a STEM degree or from advanced technical training. Thus, educating pupils in STEM majors and encouraging them to continue their studies are important steps toward filling the need for a STEM workforce which is constantly and rapidly increasing.

Previous research showed concern about student enrollment and retention in STEM fields when they get to college [74]. In fact, this can be explained by the individual choices made during one's academic career, more specifically during high school [52]. Many factors can influence student decisions. For instance, the financial situation of students plays a big role in their future enrollment [46]. Furthermore, quite often, students are influenced by their parents, whether directly or indirectly. That's why the education of parents has been investigated as a factor influencing students' higher education choices and outcomes [49].

External factors can impact personal choices, but stronger effects are more associated with academic success, proficiency in Maths and Science subjects and student's self-assessment of their level [72, 73]. These kinds of factors can be detected early, not only in high school but also in middle school. It is during this period that students acquire the necessary skills to help them prepare for college. Depending on their learning experience, students start to build their self-beliefs, objectives and career aspirations. Throughout their learning journey in middle school, they find themselves more engaged in or disengaged from the learning process at school, either starting to think about academic success and improving grades or becoming more disengaged and deviating from the track of academic success [60, 8].

Since integrating into a STEM career is closely related to graduating with a STEM major [74], the difficulty of responding to the growth of STEM positions is highly sensitive to the numbers of students enrolling in STEM majors. Continuous efforts have been made to increase STEM enrollments. But the promotion of the pursuit of a STEM major has to begin as early as middle school for two reasons. Firstly, the foundation of knowledge required in STEM fields is acquired during the years in middle school and high school. Secondly, very often, student decisions are still easily manageable during middle school, when it is possible to build their confidence in being able to pursue a STEM major [73]. That's why it is necessary to distinguish students who have difficulties and who are most likely to loose interest in STEM fields. These students need more support in order to help them overcome their problems and reignite their interest in STEM fields. Several detectors can indicate which students are most likely to pursue STEM college majors. Factors like family background and financial situation have an influence but they are not easily remediable [49, 46]. While student academic performance is a very strong indicator, it is too late to adjust the student's treatment, and teachers can no longer intervene, by the time a student finishes high school [18]. These detectors rely heavily on student grades and on-field observations. Thus, teachers find it difficult to identify problems and consequently to apply the appropriate type of support.

In a hopeful sign, the adoption of educational software has been expanding within different academic institutions in recent years. The utilization of this kind of software allows educators to gather data about student usage. The recorded data is fine-grained and relative to every student action within the system, opening up possibilities for extensive analysis, and ultimately growing into substantial sub-fields such as Educational Data Mining and Learning Analytics. With a large amount of data at hand, it became feasible to build predictive models capable of detecting student affects across a wide range of constructs such as gaming the system, boredom, carelessness, frustration, and off-task behaviours [6, 7, 59, 48, 58]. These affect detectors were the building blocks for subsequent research work that aimed at predicting learning outcomes [48], college enrollment [60] and more importantly predicting whether or not students will enroll in a STEM major in college [52].

## 3.2 Data Source

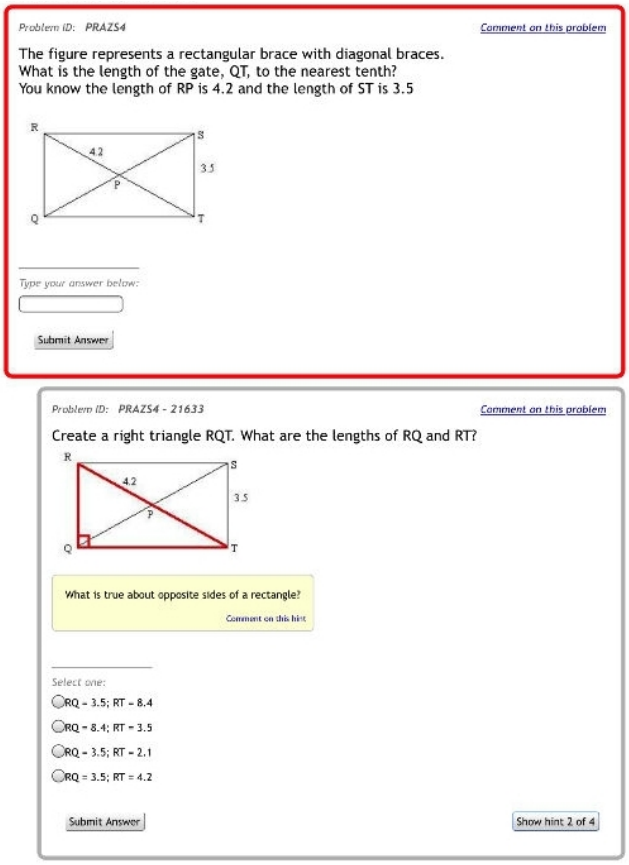
Following the research topics related to STEM enrollment, this study takes one step further and seek to predict the students' first job after college. This study uses click-stream data from students utilization of an Intelligent Tutoring System called ASSISTments [27].

### 3.2.1 ASSISTments ITS

ASSISTments[1] is a web-based Intelligent Tutoring System provided for free by Worcester Polytechnic Institute. It is intended for application to middle school mathematics where teachers can use a predefined set of contents or can create their own. The system provides students with the right assistance while assessing their knowledge. When students use the platform to work on problems assigned to them by their teachers, they receive immediate feedback as to whether their answers are correct or not. If they are right, they can proceed to the next problem, if not, the system provides them with scaffolding exercises which are sub-components of the original problem to help students master the required skills. Once those skills have been acquired, the student is directed back to the original problem to have another try. Then, after correctly answering this problem, they move on to the next one. Questions in the ASSISTments platform are related to specific skills, which makes tracking student performance more precise. On the other hand, teachers get full reports on student activities and their performance. That allows them to identify common mistakes and problems and find out who struggled to solve the problems; all of this can be done even before meeting their students in the classroom [46].

Figure 3.1 shows a small example of the user interface in ASSISTments. In this case, the student answers incorrectly to the problem. Therefore, he/she is redirected to a scaffolding problem. A scaffolding problem mimics the human tutor when breaking-down the problem into sub-components that are easier to understand by the students. The scaffolding method was proven to be effective in helping the student understand the required skills to solve the problem [27].

[1] https://new.assistments.org/



The figure represents a rectangular brace with diagonal braces.
What is the length of the gate, QT, to the nearest tenth?
You know the length of RP is 4.2 and the length of ST is 3.5

Create a right triangle RQT. What are the lengths of RQ and RT?

What is true about opposite sides of a rectangle?

○RQ = 3.5; RT = 8.4
○RQ = 8.4; RT = 3.5
○RQ = 3.5; RT = 2.1
○RQ = 3.5; RT = 4.2

Fig. 3.1.: Example of an ASSISTments problem where the student answered incorrectly, thus is led to solve a scaffolding problem

### 3.2.2 Dataset Composition

The gathered dataset is composed by action log files representing click-stream interactions of students with the ASSISTments ITS during the period between 2004-2006. We count 942,816 actions stored in the log files coming from different types of student interactions, such as, requesting help, answering a question or revealing a hint. Each action is specified by a set of recorded information, and those actions were carried out by a group of 591 students from 4 different schools which used ASSISTments. In ASSISTments, teachers can define exercise problems refined by the set of skills involved (summation, multiplication etc..). This dataset contains no less than 3765 problems related to a complete set of 93 skills. Moreover, other student information were recorded especially the first job after college graduation, which is the predicted variable for our models [50].

In overall, the dataset contains 82 features. These features described different aspects of the usage of the ASSISTments system. Some features were related to the general context of the usage, such as the school ID and the academic year. Features such as the Student ID, the Inferred Gender and the MCAS test score were related to the student who used the system. Another subset of features was associated with the action performed. In this subset of features, we obtained time-related information, such as the time taken to answer the question, or the detected long pauses after a correct answer. We also had access to features relevant to the correctness of the answers given by students and features that described the type of the answer, whether it was a fill-in or chosen answer (e.g., Multiple choice). The dataset also described some functionalities of the ASSISTments system. In fact, information about the hint and help request usage was registered. Moreover, ASSISTments provided problems at different levels: original problems and scaffolding problems. Finally, there is a subset of features related to models assessing students' knowledge, behaviors, and affective states such as boredom, engaged concentration, confusion, frustration, off-task and gaming-the-system behaviors.

As seen in the first chapter, discovery with models takes the results of previously created models as input for subsequent analysis. In fact, for many years, predicting student knowledge was an active field of research [14, 51, 54, 34] that has been characterized by the emergence of Bayesian Knowledge Tracing (BKT) [14] as one of the most used models. Indeed, BKT is able to estimate a student's latent knowledge of a specific skill given previous observable performances. Along with predicting student knowledge, different models were developed in order to estimate student affects and disengaged behaviours. Research such as [48] has produced 4 affective state detectors: Boredom, Engaged Concentration, Confusion, and Frustration. The disengaged behaviours appear in the form of an off-task attitude, gaming the system and carelessness. To build these models, field observations were recorded when students used the ASSISTments software. Then the recorded data was synchronized with the internal log data of the system, resulting in an automated model that can be used to replace the in-field experiments.

## 3.3 Aggregating Data by School

In the first analysis of the dataset, I investigate the effects of aggregating the data according to the school on the prediction performances.

### 3.3.1 Features Transformation and Selection

To make the predictions related to student enrollment in a STEM career, we needed to change the granularity of our data from the interaction level to the student level. To this end, we took the average of the selected features across all actions for each student. Picking the right features was done using univariate feature selection, only keeping features that have a strong relationship with the predicted variable. The results of the selection process are shown in Table 3.1

After running the test we observed that only some features have a strong relationship with the predicted variable. In fact, correctness is a strong predictor not only in this study but also in previous research focusing on college enrolment [60, 52]. This is more emphasised when we look at the correctness in the original problems, where the difference in the mean value is higher than the mean correctness in scaffolding problems. This is due to the fact that scaffolding questions aim to help the student acquire the skill and help him/her solve the original problem. In a way, having higher correctness in original problems gives us more insight about the skills of the student. Another strong predictor is the average of original problems, since it is the proportion of original problems over the total number of problems done by the student. A higher proportion of original problems translates to less of a "learning phase" involving scaffolding questions.

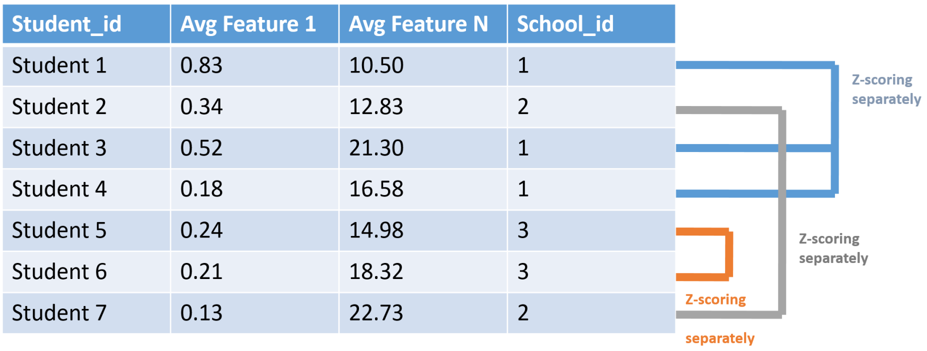
One interesting feature is the hint functionality usage. Hints give the student some advice on how to solve a problem while explaining the skill. That's why students with high hint requests are more likely to pursue a non-STEM career. Furthermore, bottom hints explain the problem from its basic notions. They are the lowest level of help, and that's why they are used less often, but the difference between the two groups of students is still significant. Extensive hints usage has been reported as a detector for gaming the system behaviour [7], which is another strong predictor for student enrollment in a STEM career. Students who loose interest in STEM have higher mean values in gamin the system.

Additional features that can be good predictors are carelessness and knowledge estimation. Similarly to STEM major predictions [52], the carelessness of students seems to increase when they are going to continue in a STEM career, which is a non-intuitive finding shared by the two pieces of research. Finally the average knowledge of a student is an estimation of his/her skills and to what extent he mastered the involved skill. It's the most straightforward predictor, since more knowledge means that the student has more aptitude to pursue a STEM career without serious problems.

**Tab. 3.1.:** Univariate Features Selection

| | STEM Career | Mean | Std | F-Value |
|---|---|---|---|---|
| Avg Bored | 0 | 0.252 | 0.033 | 2.90e-05 |
| | 1 | 0.252 | 0.031 | p=0.99 |
| ***Avg Bottom hint*** | ***0*** | ***0.046*** | ***0.035*** | ***10.811*** |
| | ***1*** | ***0.034*** | ***0.029*** | ***p<0.01*** |
| ***Avg Carelessness*** | ***0*** | ***0.12*** | ***0.065*** | ***18.207*** |
| | ***1*** | ***0.15*** | ***0.078*** | ***p<0.001*** |
| Avg Confused | 0 | 0.106 | 0.038 | 0.013 |
| | 1 | 0.105 | 0.035 | p=0.910 |
| ***Avg Correct Original*** | ***0*** | ***0.43*** | ***0.156*** | ***11.458*** |
| | ***1*** | ***0.485*** | ***0.176*** | ***p<0.001*** |
| ***Avg Correct Scaffold*** | ***0*** | ***0.584*** | ***0.106*** | ***4.494*** |
| | ***1*** | ***0.606*** | ***0.101*** | ***p<0.05*** |
| ***Avg Correct*** | ***0*** | ***0.417*** | ***0.152*** | ***16.516*** |
| | ***1*** | ***0.471*** | ***0.144*** | ***p<0.001*** |
| Avg Engaged Concentration | 0 | 0.647 | 0.03 | 1.209 |
| | 1 | 0.650 | 0.026 | p=0.271 |
| Avg Frustration | 0 | 0.127 | 0.047 | 1.834 |
| | 1 | 0.121 | 0.052 | p=0.176 |
| Avg FirstHelpRequest | 0 | 0.285 | 0.066 | 1.126 |
| | 1 | 0.292 | 0.071 | p=0.288 |
| ***Avg Gaming*** | ***0*** | ***0.113*** | ***0.124*** | ***4.115*** |
| | ***1*** | ***0.088*** | ***0.105*** | ***p<0.05*** |
| ***Avg Hint*** | ***0*** | ***0.266*** | ***0.141*** | ***14.108*** |
| | ***1*** | ***0.214*** | ***0.124*** | ***p<0.001*** |
| ***Avg Knowledge*** | ***0*** | ***0.224*** | ***0.135*** | ***16.881*** |
| | ***1*** | ***0.283*** | ***0.162*** | ***p<0.001*** |
| Avg Off-Task | 0 | 0.216 | 0.082 | 0.069 |
| | 1 | 0.219 | 0.074 | p=0.792 |
| ***Avg Original*** | ***0*** | ***0.298*** | ***0.125*** | ***8.904*** |
| | ***1*** | ***0.337*** | ***0.139*** | ***p<0.01*** |
| Avg Scaffold | 0 | 0.418 | 0.114 | 0.573 |
| | 1 | 0.426 | 0.118 | p=0.449 |
| Avg Time Original | 0 | 64.38 | 34.18 | 0.946 |
| | 1 | 67.82 | 38.16 | p=0.331 |
| Avg Time Scaffold | 0 | 32.51 | 17.16 | 0.416 |
| | 1 | 33.64 | 17.99 | p=0.518 |
| Avg Time Taken | 0 | 40.84 | 21.09 | 2.445 |
| | 1 | 44.25 | 23.51 | p=0.118 |
| Nb Problems | 0 | 236.3 | 139.5 | 1.754 |
| | 1 | 255.1 | 143.9 | p=0.185 |

Once the features selection is done, I proceed to the features aggregation by school. To do so, I separate students data by the school then apply the z-score for each feature separately school by school. Z-score is a statistics method that measure how many units of standard deviation a data point is far from the mean. Z-score has many applications such us normalization and ranking. Figure 3.2 explains how the school-based z-score method was applied. This process means that we are ranking the students in terms of standard deviation compared to their school-mates.



| Student_id | Avg Feature 1 | Avg Feature N | School_id |
|---|---|---|---|
| Student 1 | 0.83 | 10.50 | 1 |
| Student 2 | 0.34 | 12.83 | 2 |
| Student 3 | 0.52 | 21.30 | 1 |
| Student 4 | 0.18 | 16.58 | 1 |
| Student 5 | 0.24 | 14.98 | 3 |
| Student 6 | 0.21 | 18.32 | 3 |
| Student 7 | 0.13 | 22.73 | 2 |

Fig. 3.2.: Aggregating the data by school.

## 3.3.2 Validating the school-based approach

To validate the school-based approach, I compare it to a normal-approach where no particular feature transformation is applied. Figure 3.3 exposes the overall workflow. Therefore, I use GP as explained in Chapter 1. I run two separate optimization process, one for the normal-approach and another for the school-based approach. In the normal approach, the Random Forest Classifier give the best results during the optimization phase. In the school-based approach, the Gaussian Naive Bayes classifier has the best performance. Following the GP phase, I train each resulting model following a 10 fold cross validation.

Table 3.2 shows the mean of the cross-validated values for both models. This time the school-aggregated model showed an increase of RMSE to over 0.54 compared to its counter part. On the other hand, the normal approach attained 0.521 in ROC AUC score, but was still lower than the score of the school model (0.601).

Figure 3.4 shows more about the cross-validated ROC AUC scores. The values of the normal approach are spread from the minimum of 0.36 to the maximum of 0.65 with 25% of the values exceeding 0.63 and another 25% are being less than 0.44. On the other hand, the school-based approach is less diverse, since its minimum is

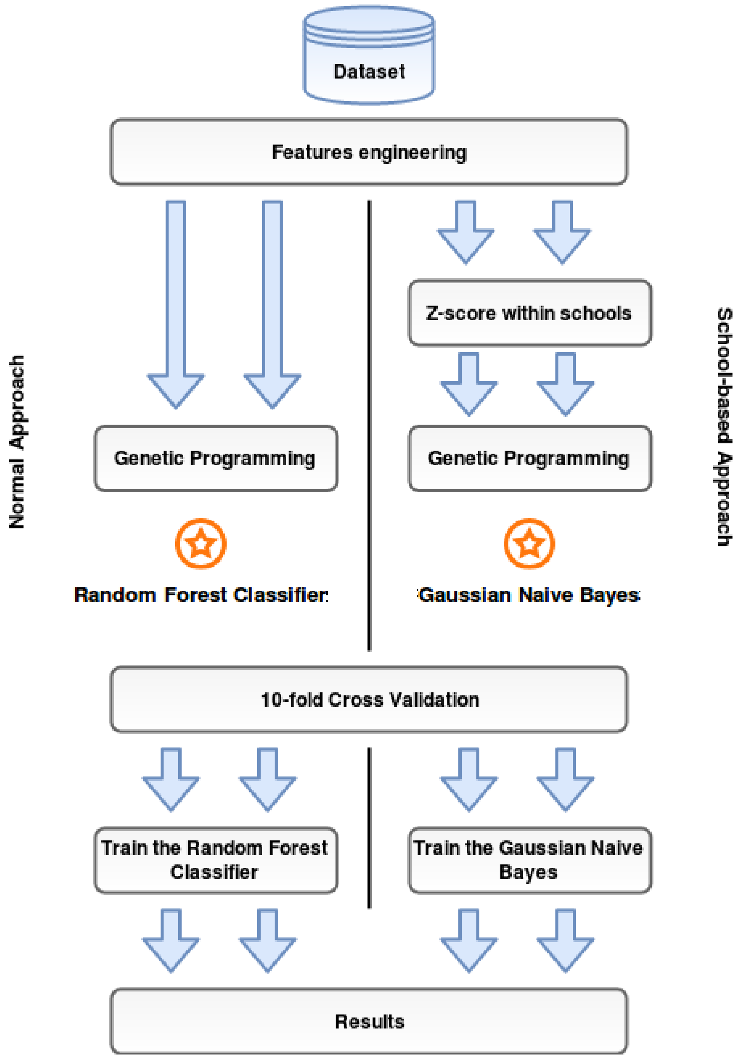


Fig. 3.3.: Overall workflow for validating the school-based approach.

Tab. 3.2.: Cross-validated scores for both approaches

| | School-based | Normal approach |
|---|---|---|
| **ROC AUC** | 0.601 | 0.521 |
| **RMSE** | 0.546 | 0.45 |

0.47 and maximum is 0.70. Half of the values exceed 0.59 and 25% of them are above 0.67.

Now when comparing the RMSE scores of the two approaches, we clearly see in Figure 3.5 that the normal approach is almost perfectly distributed around the minimum of 0.45 and the maximum of 0.47. While the school-based approach is

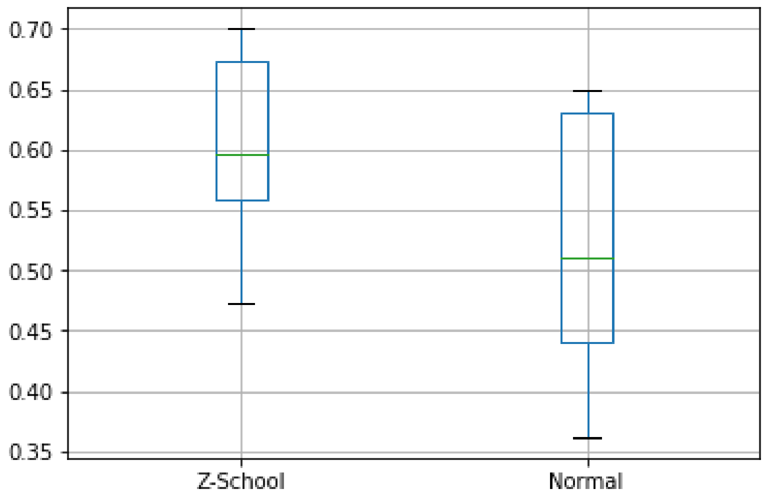


Fig. 3.4.: Cross-validated scores of ROC AUC for both approaches.

spread from the minimum of 0.47 to the maximum of 0.6. 25% of its values are under 0.51. Half of the data is above 0.555 and 25% of it is superior to 0.585.

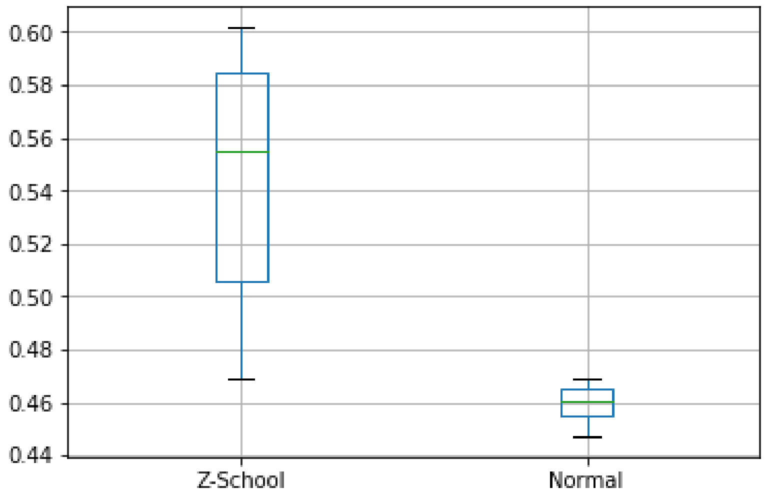


Fig. 3.5.: Cross-validated scores of RMSE for both approaches.

Even if the difference between the two approaches is statistically significant ($p<0.01$), the school-based approach has better AUC, while the normal approach has a lower RMSE, thus we cannot clearly confirm that the school-based approach has radically better results. The gain in terms of AUC is significant, but it suffers from a relatively high RMSE.

### 3.3.3 Effect of aggregating data by school

It is clear that aggregating the data by school improved the models performances in ROC AUC. However, the RMSE was higher. But, for a classification problem, ROC AUC is more often used as performance metric, therefore it is more significant. Therefore, it is fair to say that the aggregation by school improved the performances of the model but it is not very high itself. So, it is interesting to investigate other aspects of the dataset while keeping into consideration the improvement achieved by aggregating the data by schools.

## 3.4 Comparing the aggregation by skill and by problem

While the first analysis showed promising results by aggregating the data by school, there are different aspects of the dataset that are interesting and might add value to the analysis and improve the prediction models. The following analysis use several features implemented in ASSISTments ITS.

Some interesting features of ASSISTments and ITS in general are the decomposition of the problems by skills involved to solve them. This structure allows a more fine grained control and analysis of the students' performance and understanding. Moreover, it make it easier to address students' problems once we detect which skills they did not understand. Therefore, I aim at utilizing this information indirectly to improve the models performances. In fact, I do not intend on using the problem's id or the skill name as a direct predictor of the student career. Similarly to the school, I will use the information about the skill and the problem to conduct more fine-grained analysis and aggregation of the students data.

Firstly, I proceed to another round of features selection. I use the univariate feature selection combined with the ANOVA F-score to select the candidate features. In the Table 3.3, I list only the selected features that have a significant correlation with the predicted variable with an order of significance $p < 0.05$.

Later I check the correlation values between several of theses features and remove the highly correlated ones. Moreover, I discard other features such as the "original" feature since it will be used for features engineering and won't be used as a direct predictor in our machine learning models. Table 3.4 exposes the list of features considered for this study along with their meanings, after the manual selection and removal of the highly correlated features.

Tab. 3.3.: Features chosen by Univariate Feature Selection.

| Feature Name | F-Score | P-Value |
|---|---|---|
| AveKnow | 16.88 | 0.000045 ($p < 0.001$) |
| AveCarelessness | 18.20 | 0.000023 ($p < 0.001$) |
| hintCount | 11.11 | 0.000908 ($p < 0.001$) |
| hintTotal | 10.05 | 0.001601 ($p < 0.05$) |
| attemptCount | 7.19 | 0.007514 ($p < 0.05$) |
| frPast5HelpRequest | 8.58 | 0.003520 ($p < 0.05$) |
| frPast8HelpRequest | 5.86 | 0.015705 ($p < 0.05$) |
| past8BottomOut | 7.18 | 0.007538 ($p < 0.05$) |
| timeSinceSkill | 10.54 | 0.001234 ($p < 0.05$) |
| totalTimeByPercentCorrectForskill | 5.37 | 0.020812 ($p < 0.05$) |
| res_gaming | 4.11 | 0.042891 ($p < 0.05$) |
| Ln-1 | 16.10 | 0.000068 ($p < 0.001$) |
| Ln | 16.89 | 0.000045 ($p < 0.001$) |
| correct | 16.56 | 0.000053 ($p < 0.001$) |
| original | 8.95 | 0.002884 ($p < 0.05$) |
| hint | 14.12 | 0.000188 ($p < 0.001$) |
| bottomHint | 10.82 | 0.001062 ($p < 0.05$) |
| frIsHelpRequestScaffolding | 5.97 | 0.014831 ($p < 0.05$) |
| timeGreater10SecAndNextActionRight | 16.46 | 0.000056 ($p < 0.001$) |
| manywrong | 15.97 | 0.000072 ($p < 0.001$) |

At this point, I use another interesting feature of ASSISTments that is called scaffolding. In fact, when a student fails a problem, he/she is redirected to a subset of problems that decompose the skills involved in the original problem. Therefore, there are different types of problems: Original and Scaffolding. I take into account this difference and apply it simultaneously with the aggregation following the skills-based and problem-based approach and I add the suffix _o and _no accordingly. Thus, I generate {selected_features}_o which are the measured features when the problem is an original problem (original = 1). And {selected_features}_no are measured when the problem is not original (original = 0).

In the problem based approach, for each student, I apply the mean to all features for a given problem. Similarly, in the skill based approach I apply the mean for each student by skill involved. Figure 3.6 gives a whole overview of the transformation. In fact, starting from the action-level dataset, I apply an intermediate transformation for each approach separately. For example, in the problem based approach, each row of the intermediate dataset represents the data of a student in a particular problem. Each feature is measured differently depending if the problem was original or scaffolding. Therefore, the first row of the intermediate transformation of the

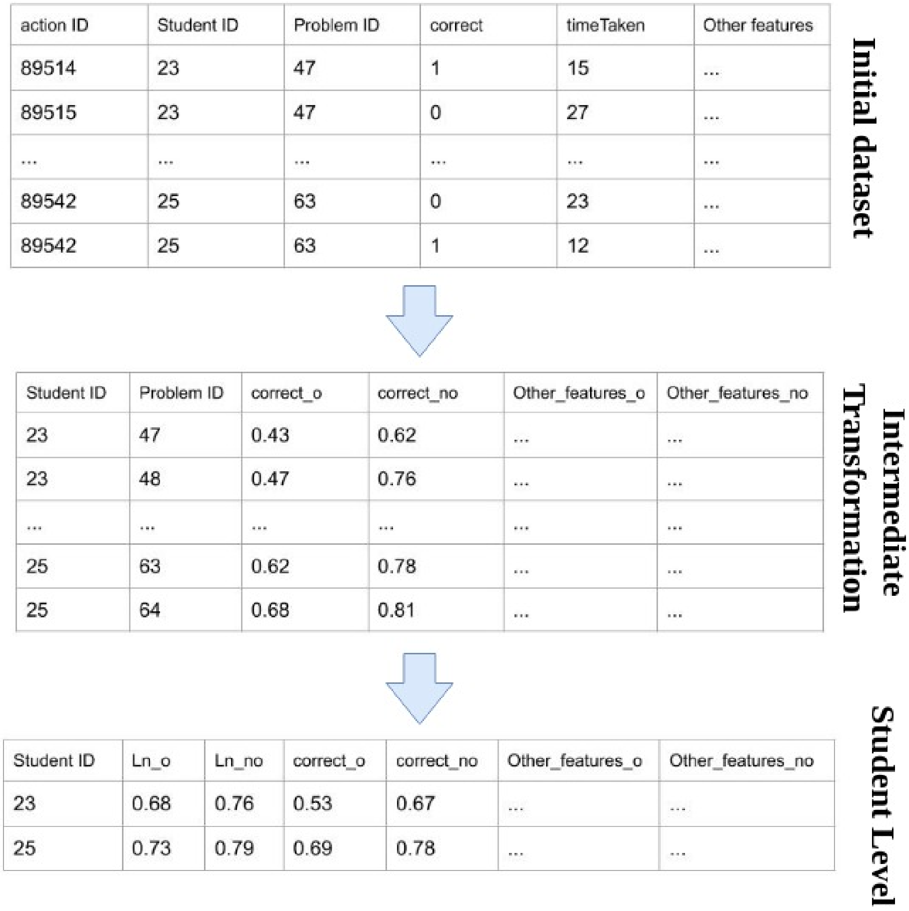
Tab. 3.4.: Feature set to be used when comparing skills to problems.

| Feature Name | Meaning |
|---|---|
| correct | Answer is correct |
| timeTaken | Time spent on the current step |
| bottomHint | Bottom-out hint is used |
| frIsHelpRequestScaffolding | First response is a help request Scaffolding |
| timeSinceSkill | Time since the current Knowledge Component (KC) was last seen. |
| hint | Action is a hint request |
| attemptCount | Total problems attempted in the tutor so far. |
| manywrong | Many wrong answers given |
| Ln | Bayesian Knowledge Tracing's knowledge estimate at the time step [14] |
| res_gaming | Rescaled of the confidence of the student affective state's prediction: gaming-the-system |
| frPast5HelpRequest | Number of last 5 First responses that included a help request |
| totalTimeByPercentCorrect Forskill | Total time spent on this KC across all problems divided by percent correct for the same KC |
| timeGreater10SecAnd NextActionRight | Long pause before a correct answer |

Figure 3.6 is interpreted as follow. The student 23, when working on the problem 47, achieved a correctness of 0.43 when the problem was original and a correctness of 0.62 when the problem was scaffolding. In the skill based approach, a similar transformation is applied by grouping according to the skill instead of the problem. Finally, the last step is to apply the mean across all problems/skill for each student.

Later, I proceed to another round of feature selection. For each approach, I separately use a combination of forward feature selection and backward feature elimination. Then, I take the union of the feature sets that emerge from each feature selection method.

For the problem-based approach, the selection gave us the following features listed in Table 3.5. The selected features set was quite small and contained predictors related to hint usage in original and non-original problems. Likewise, the behavior of gaming-the-system in non-original problems was detected as a strong predictor. The correctness, the longtime pauses after a correct answer and the number of the five last first responses that included a help request, all in non-original problems, were also selected as strong features. Finally, the average time since the skill has



| action ID | Student ID | Problem ID | correct | timeTaken | Other features |
|---|---|---|---|---|---|
| 89514 | 23 | 47 | 1 | 15 | ... |
| 89515 | 23 | 47 | 0 | 27 | ... |
| ... | ... | ... | ... | ... | ... |
| 89542 | 25 | 63 | 0 | 23 | ... |
| 89542 | 25 | 63 | 1 | 12 | ... |

| Student ID | Problem ID | correct_o | correct_no | Other_features_o | Other_features_no |
|---|---|---|---|---|---|
| 23 | 47 | 0.43 | 0.62 | ... | ... |
| 23 | 48 | 0.47 | 0.76 | ... | ... |
| ... | ... | ... | ... | ... | ... |
| 25 | 63 | 0.62 | 0.78 | ... | ... |
| 25 | 64 | 0.68 | 0.81 | ... | ... |

| Student ID | Ln_o | Ln_no | correct_o | correct_no | Other_features_o | Other_features_no |
|---|---|---|---|---|---|---|
| 23 | 0.68 | 0.76 | 0.53 | 0.67 | ... | ... |
| 25 | 0.73 | 0.79 | 0.69 | 0.78 | ... | ... |

Fig. 3.6.: An example of the feature transformation in the problem-based approach.

been seen across original problems was the last strong predictor in the features set.

We ran the same selection process in the skill-based approach and we found different features. Table 3.6 shows the list of selected features for the skill-based approach. The selected feature set for the skill-based approach was larger than that for the problem-based approach. Again, we found the behavior of gaming-the-system in non-original problems to be a strong predictor. The average BKT estimate and the average carelessness both in the original problems were selected this time. Surprisingly enough, they were not selected in the problem-based approach. We also found that the average time since the skill was seen in non-original problems was a good predictor, as well as the average correctness in both original and non-original problems. Likewise, the hint and the bottom hint usage in original problems were detected as strong predictors. Similarly to the problem-based approach, the longtime pauses after a correct answer and the number of the five last first responses that included a help request, both in non-original problems, were also selected as strong features.

Tab. 3.5.: Final feature set to be used in the problem-based approach.

| **Features measured in original problems** | **Features measured in non-original problems** |
|---|---|
| avg_hint_per_problem_o | frPast5HelpRequest_per_problem_no |
| timeSinceSkill_per_problem_o | res_gaming_per_problem_no |
| | avg_correct_per_problem_no |
| | avg_hint_per_problem_no |
| | avg_timeGreater10SecAndNext ActionRight_per_problem_no |

Tab. 3.6.: Final feature set to be used in the skill-based approach.

| **Features measured in original problems** | **Features measured in non-original problems** |
|---|---|
| Ln_per_skill_o | res_gaming_per_skill_no |
| AveCarelessness_per_skill_o | timeSinceSkill_per_skill_no |
| avg_correct_per_skill_o | frPast5HelpRequest_per_skill_no |
| avg_hint_per_skill_o | avg_correct_per_skill_no |
| avg_bottomHint_per_skill_o | avg_manywrong_per_skill_no |
| | avg_timeGreater10SecAndNext ActionRight_per_skill_no |

Along with the skill-based and problem-based approaches, I apply the school aggregation one more time. The school aggregation is done similarly with the previous analysis. We measure the z-score of all features for each school's students separately.

I compare all these alternatives to a baseline approach where no particular feature aggregation is applied. To fairly compare all approaches, I apply GP to find the best machine learning method with its optimized hyper-parameters. Then I validate the resulting pipeline using 10-fold cross-validation.

The results of the optimization phase are exposed in Table 3.7. For the baseline model, in which we just took the average values across all actions for each student, the optimization process generated a pipeline having Randomized Decision Trees as the prediction method. In the normal problem-based approach, the resulting pipeline contained a stacking technique using a Naive Bayes classifier combined with Logistic Regression. For the problem-based approach with school aggregation, we found that the Extreme Gradient Boosting algorithm had the best results. Similarly, for the normal skill-based approach, a Gradient Boosting Classifier was chosen.

Finally, for the skill-based approach with school aggregation the best pipeline used a Decision Trees Classifier.

Tab. 3.7.: Results of the optimization process.

| Approach | Best pipeline |
|---|---|
| Baseline | Randomized Decision Trees |
| Problem-based | Logistic Regression |
| Problem-based, school-aggregated | XGBClassifier |
| Skill-based | Gradient Boosting Classifier |
| Skill-based, school-aggregated | Decision Trees |

Table 3.8 exposes the results of the cross-validation step. The best scores for each measure are shown in boldface. The baseline model had the worst results in AUC and in the combined score, suggesting that simply taking the average values across all students' actions was not an effective concept. The problem-based approach had better results in AUC, attaining 0.629, but a worse RMSE of 0.482. Its combined score reached 1.146 which is better than the baseline score. Against our expectations, the aggregation of the features' values within schools did not improve the predictions in the problem-based approach. In fact, the school-aggregated model had a lower AUC, but better RMSE. However, the combined score was worse than the normal problem-based approach. Compared to the problem-based approach, the skill-based approach had a significant improvement in terms of RMSE, dropping to 0.461, which is the best RMSE score among all the models. With a combined score of 1.160, the normal skill-based approach had a better result than the normal problem-based approach and the school aggregated problem-based approach. The best AUC score was achieved by the skill-based approach with school aggregation, which showed a significant improvement, attaining 0.682 in AUC. However, its RMSE was the highest among all the models, reaching 0.513. Despite the high RMSE, this model had the best combined score of all the models considered.

Tab. 3.8.: Cross-validated scores of all approaches.

| Model | AUC | RMSE | Combined Score |
|---|---|---|---|
| Baseline | 0.521 | 0.466 | 1.055 |
| Problem-based | 0.629 | 0.482 | 1.146 |
| Problem-based, school-aggregated | 0.610 | 0.474 | 1.135 |
| Skill-based | 0.621 | **0.461** | 1.160 |
| Skill-based, school-aggregated | **0.682** | 0.513 | **1.169** |

These results can be explained by the fact that building features around skills gives stronger predictors than the problem-based model. And that's because skills are more fine-grained than problems and they better encapsulate the ability of students to master the subject. Moreover, problems can be related to one or many skills at the same time and that's probably why they are not as effective as the skills in terms of describing the failing students and the successful ones. Since problems can be related to different skills, when students master a skill, they are more likely to be successful applying it in different problems that involve that skill. However, the reverse is not always true, as you can't generalize from the problem viewpoint to the skill viewpoint. In other words, mastering one single problem does not mean mastering all the skills involved in that problem. Moreover, when we investigated the effect of comparing students' performances with their peer schoolmates, we found that such aggregation improved our models. Our aim was not to compare which school was the best or had the best students. Our objective was to verify whether students that had the best performance relative to their schoolmates were more likely to enroll in STEM-related fields.

## 3.5 Summary and contributions

In this study, I predict whether the student will pursue a career related to a STEM field. The model generalizes well to different distributions of students [50]. The used dataset consists of click-stream log files and a record about the student's job type after college. The dataset is rich in many features. And different meta-data are collected. In my analysis, I firstly identify that aggregating the data by school improve the models' performance. Following that analysis, I prove that aggregating the data based on the skills have a significant improvement compared to a baseline approach. It has even better results when the aggregation is applied by skill and by school.

# Students Reading Behavior

# 4

## 4.1 Background and Related Work

Understanding students' behaviors and provide them with a better learning experience has always been a driving motivation in learning science and educational technologies. Thanks to the continuous increase of the adoption of educational software, these goals are easier to achieve. With the introduction of Information and Communications Technology (ICT) to education, different types of educational software and teaching techniques have been implemented. Learning Management Systems, Intelligent Tutoring Systems, Blended learning and many more have been applied to educate people in K12 and higher education. Nevertheless, higher education is also taking advantage of the advances in educational technology. Learning Management Systems such as Moodle are being used in different educational institutions. Some of these educational software systems are part of an even bigger infrastructure which include different systems that are in cooperation.

A particular system is delivering the course content to the students in a seamless manner. The system is called "BookRoll"[1]. It is part of a bigger platform for sharing and reusing ubiquitous learning log (SCROLL) [22, 21, 43]. This platform is also composed by an integrated system for learning analytics [45].

These Digital-Learning-Materials Readers are useful in different ways. First, they are a good means of distributing the course material, second, they are a valuable data collection source for learning analytics as it serves to gather students' usage data. Finally, it also provides feedback to teachers about the students' learning experience. They also provide several usability advantages thanks to its practical functionalities [44, 40]. For example, the BookRoll digital teaching-material-delivery system allows teachers to upload lecture materials in a digital form which students can read anytime, anywhere [21, 22]. The system provides students with different functionalities, such as markers, bookmarks and memos [43].

In this study, I will examine the students reading behaviors through different aspects. I use a dataset gathered from the students' usage of the BookRoll system.

[1] https://www.let.media.kyoto-u.ac.jp/en/project/digital-teaching-material-delivery-system-bookroll/

## 4.2 Data Collection

The dataset is composed by different files. These files are organized by course. For each course we have 4 files as follow:

- Event Stream: This file lists all the events done by the students using the BookRoll system in that particular course
- Lecture Material: This file records the course material used for each lecture
- Lecture Time: This file contains the timings of each lecture
- Quiz Score: In this file, there is the list of the anonymized student id with their exam score in the respective course

In overall, the dataset contains almost 2 million rows of events, each one of them describes an action done by the student within the system. Different types of actions are recorded such as a request to open a file, a jump to a specific page, saving a bookmark and many more. For each event, the system stores several information, such as the anonymized student ID, the page number where the action happened, the device (PC or Mobile), the time-stamp, the action type and some other information that depends on the action type. The lecture is defined by an id, start and end time, the content used and its number of pages. When it comes to students, the data set contains only their anonymized ID and their score in the respective course. The most important features is the action type (named 'operationname' in the dataset). It is a categorical feature having 17 possible values describing the types of actions that the student can perform within the system.

In the first analysis, I will detect the students reading sessions based on their activity time. Such analysis is produced using the action types of the students.

## 4.3 Detecting Students' Reading Sessions

Basically, a reading session is related to opening a document and being engaged with it until the student closes it or the time when we detect an inactivity period exceeding a predefined 'Inactivity threshold'. If the student closes the document, then the session is closed normally; if the student is inactive then we terminate the session, but we keep track of the opened document, and we start another session when the student is back using the respective document. Since the student can open multiple documents, he can be engaged in many different reading sessions, but we only close the session when the student does not use a specific document during a period of time.

The choice of the adequate inactivity threshold, after which we consider a reading session closed, is subject to some experimentations. We wanted to find the most reasonable value which is not too long that it won't detect the inactivity behavior, but also in the same time don't be too short that we mark students as inactive when they come back to the document shortly after. Therefore, we investigate 4 different values of the inactivity threshold and compare the number of detected reading sessions. We chose 30 minutes, 60 minutes, 90 minutes and 120 minutes as the inactivity thresholds.

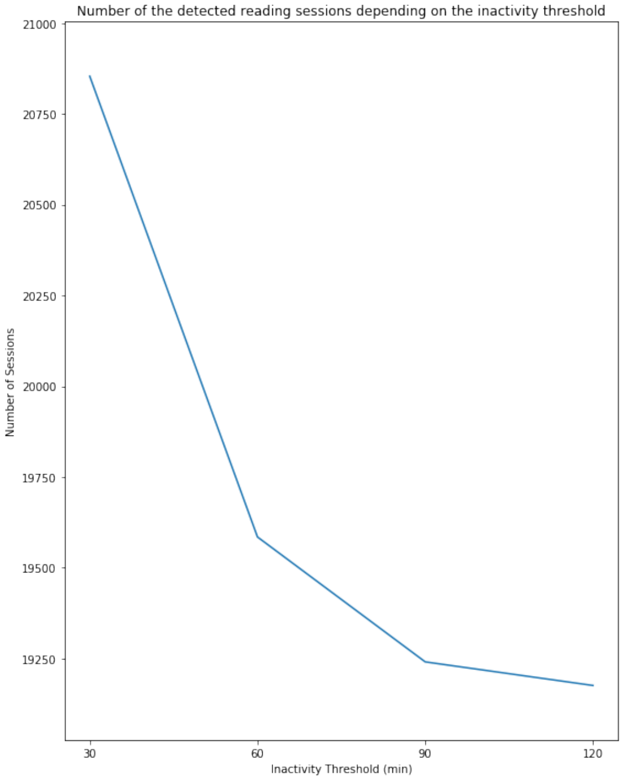


**Fig. 4.1.:** Effects of the inactivity threshold on the number of reading sessions.

In Figure 4.1, we see how the number of detected reading sessions is influenced by the choice of the inactivity threshold. The first choice, which is 30 minutes, detected the greatest number of reading sessions. But, when we increased that threshold to 60 minutes, we experienced a big reduction of 6% of the number of the detected reading sessions. From this change, we can see that setting the threshold to 30 minutes was very short and many students were labeled 'inactive' and closed their sessions while they came back again to the document and continued their activities shortly after that. Therefore, in the 30 minute threshold we detected more sessions simply because many of them were the same session, but they were split into two sessions due to the small limit of time. So, when we fixed the threshold to 60 minutes, a big number of these wrongly labeled inactive students kept their session open. We continue to investigate another threshold of 90 minutes and we remarked that the reduction in the number of detected sessions was not very significant. In fact, the difference in the number of sessions detected by 60 minutes and 90 minutes is about 1.75%. Moreover, when we selected 120 minutes as the threshold, the reduction was only 0.33% compared to the 90 minute threshold. So, we can say that choosing 120 minutes is somehow high and do not grasp the inactivity of students until they close normally the document. 60 minutes and 90 minutes are credible choices, but we chose 60 minutes. The reason is that 90 minutes is the duration of a lecture, and it is less likely for students to be inactive concerning the lecture material for the whole period of the class.

## 4.4 Detecting Highly Performing Students

The scores distribution is skewed toward the high scores. In fact, most of the students scores are between 75 and 90. Therefore, the objective is to find which score delimiter allow us to maximize the difference of the students' reading behavior. A first approach is to select multiple threshold scores and label students as highly performing when their score is higher than this threshold.

In Figure 4.2, we see that the score delimiters with the best class balance are, 85 and 80. The scores of 75 and below are low enough to consider almost everyone as highly performing. Meanwhile, a score of 90 is high so it only detects excellent students. Moreover, the class balance using the scores of 70, 75 and 90 is poor.

Based on the selected scores delimiter, we analyze which scores maximizes the difference in students reading behaviors to the point of having the best accuracy in predicting the students performance. Therefore, the problem is formulated as binary classification problem. There are two classes to predict: Highly performing student or not.

Using the students reading sessions detected, I generate different features to build the classifiers. Table 4.1 exposes the features transformations applied in the session-level data.

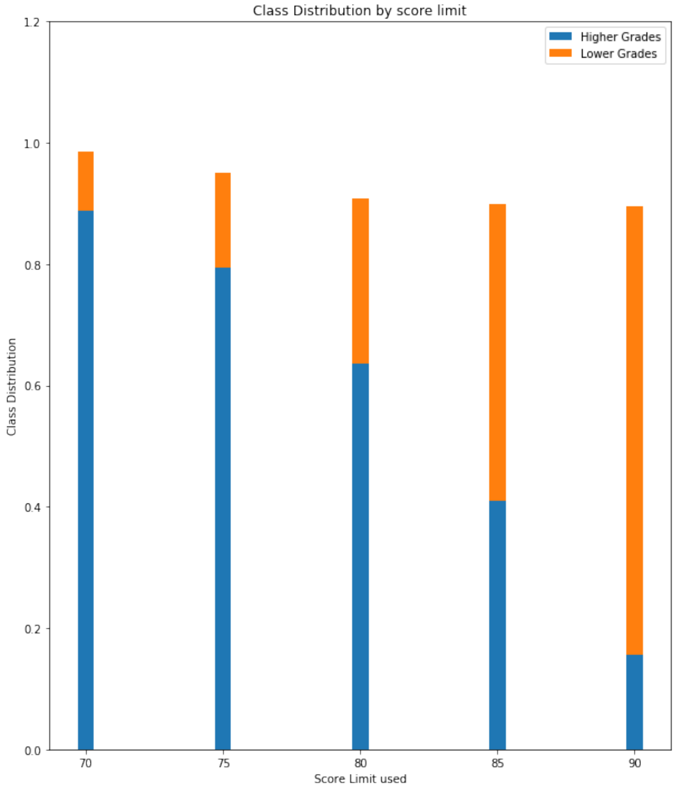


**Fig. 4.2.:** Class distribution depending on the score delimiter.

At this point, a features selection step is needed. I use a combination of three famous features selection techniques. They are the Univariate Features Selection, the Forward Features Selection, and the Recursive Features Elimination. Basically, I give a score to each feature based on its rank and whether or not it was selected in the respective features selection method. The aggregation of the score and ranks of all features selection is used then to select the subset of features to be chosen. This method is applied separately for each score delimiter. As a result, almost the same features are selected. Table 4.2 exposes the selected features. For the 85 score limit 12 features are selected, and for the 80 score limit, 11 features are chosen. Moreover, the 11 features are the same. The only difference is the 12th feature selected for the 85 score limit which is written in blod in Table 4.2.

After the features selection, I use GP to find the best machine learning pipeline for each score delimiter. Thereafter, I validate the models using 5-fold cross-validation. As shown in Table 4.2, Gradient Boosting Classifiers were chosen after the optimization process. With the score delimiter of 85, it does not have good performance on all metrics. In fact, the accuracy is low attaining 0.53, but the ROC AUC is fair since it attains 0.59. The precision is low too, approaching 0.52 and the Recall is 0.7.

| Column | Meaning and composition |
|---|---|
| Session length | Session length in seconds |
| Actions per page | Number of actions divided by the number of pages |
| Bookmark actions ratio | Number of actions related to bookmarks (add, delete) divided by the number of actions |
| Bookmark actions per page | Number of actions related to bookmarks (add, delete) divided by the number of pages |
| Memo actions ratio | Number of actions related to memos (add, change, delete)divided by the number of actions |
| Memo actions per page | Number of action related to memos (add, change, delete) divided by the number of pages |
| Link actions ratio | Number of link click actions divided by the number of actions |
| Link actions per page | Number of link click actions divided by the number of pages |
| Search actions ratio | Number of actions related to search (action, jump) divided by the number of actions |
| Search actions per page | Number of action related to search (action, jump) divided by the number of pages |
| Important actions ratio | Number of important marker actions (add, delete) divided by the number of actions |
| Important actions per page | Number of important marker actions (add, delete) divided by the number of pages |
| Difficult actions ratio | Number of difficult marker actions (add, delete) divided by the number of actions |
| Difficult actions per page | Number of difficult marker actions (add, delete) divided by the number of pages |
| Browsing actions ratio | Number of 'NEXT' or 'PREV' actions divided by the number of actions |
| Browsing actions per page | Number of 'NEXT' or 'PREV' actions divided by the number of pages |
| Jumping actions ratio | Number of jumping actions (from bookmark, memo or page) divided by the number of actions |
| Jumping actions per page | Number of jumping actions (from bookmark, memo or page) divided by the number of pages |

Tab. 4.1.: Features generated and their meaning.

While with the score delimiter of 80 the model has better results. Attaining 0.84 in Recall, 0.75 in Precision, 0.63 in ROC AUC and an accuracy of 0.68.

To further analyze the performances of the models, I check the confusion matrices and normalize the values. In fact, Figure 4.3 represents the confusion matrix for the model of the score delimiter of 85. The rate of True Positive is 0.7 and for the True

| | |
|---|---|
| Avg browsing actions per page | Avg browsing actions ratio |
| Avg difficult actions per page | Avg in lecture |
| Avg difficult actions ratio | Avg important actions ratio |
| Avg jumping actions per page | Avg jumping actions ratio |
| Avg memo actions per page | Total number of actions |
| Sessions count | **Avg session length** |

Tab. 4.2.: Features selected.

| | Score delimiter of 85 | Score delimiter of 80 |
|---|---|---|
| Machine learning method | Gradient Boosting Classifier | Gradient Boosting Classifier |
| Accuracy | 0.53 | **0.68** |
| ROC AUC | 0.59 | **0.63** |
| Precision | 0.52 | **0.75** |
| Recall | 0.7 | **0.84** |

Tab. 4.3.: Validation scores.

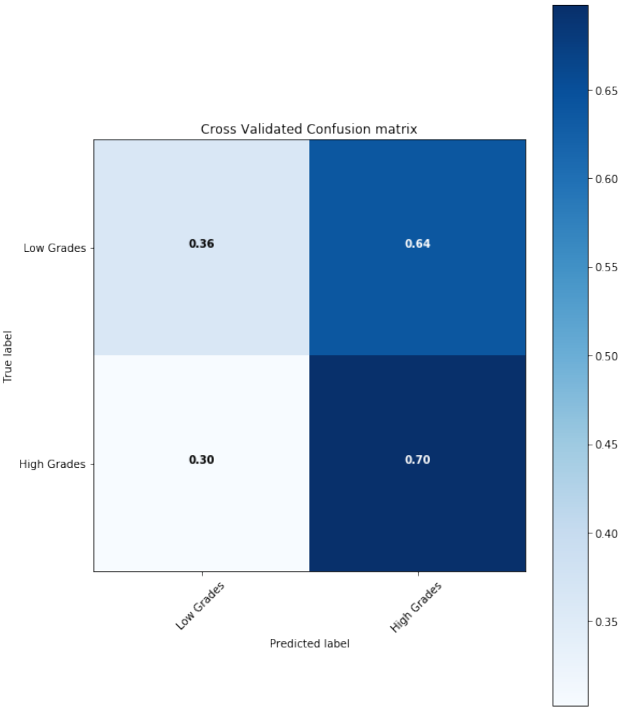


Fig. 4.3.: Confusion Matrix for the score of 85.

Negative is 0.36, while the False Positive and False Negative rates are 0.64 and 0.3 respectively.

In the other side, Figure 4.4 shows the confusion matrix for the models of the score delimiter of 80. The True Positive rate is better, reaching 0.84, but the True Negative rate is 0.25 while the False Positive rate attains 0.75. Finally, the False Negative rate is 0.16.

The results from the confusion matrices suggest that the prediction models tend to predict that the student is among the highly performing students even if he/she is not. That's why the True Negative score is low (0.36) which is the models' performances when predicting the low performing students. Accordingly, the false negative, is high. Therefore, it shows that both models tend to label a low performing student as highly performing. This problem is less troublesome when I use the score delimiter of 85 compared to a score delimiter of 80.

Moreover, a particular perspective is that defining only two classes might not encapsulate the diversity of students performances and perhaps using a more fine-grained grade delimiter could improve the predictions and also the understanding of the students behavior. For example, if I use 5 grades delimiter, it can give more details about each subgroup of students behavior.

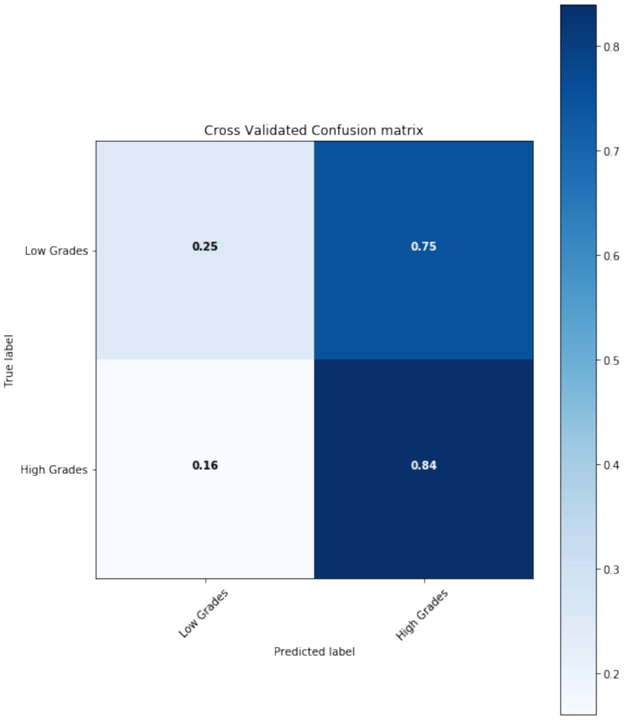


Fig. 4.4.: Confusion Matrix for the score of 80.

## 4.5 Investigating the influence of the document's content

Students' reading behaviors can be depending on the documents which they read. In fact, documents have different number of pages, and also different contents. For a defined course, the materials used in the introductory lessons are different from the materials used in more detailed and advanced topics of the same course. Therefore, I investigate the effect of aggregating the data by document in the performances of the prediction models.

### 4.5.1 Initial Data Analysis

To proceed with the document-based aggregation it is important to verify the usage of the documents by each students. The main objective is to make sure that the documents are used and that each document is actually related to one lesson.

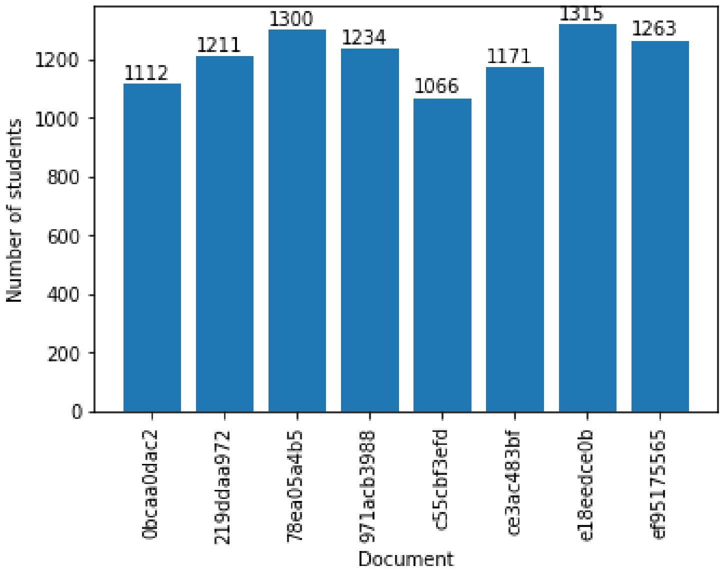


Fig. 4.5.: Number of students using each document.

In fact, as shown in Figure 4.5, each document has not been used by some students. But most of the documents were accessed multiple times and this makes the analysis doable and significant since the least used document was accessed at least by 1066 unique student. However, using only Figure 4.5, we cannot make sure that students who did not use the documents are the same. Therefore, by using the Figure 4.6, we notice that there are only 6 students who used only one document, 16 students used 2 documents, 17 students used only 3 documents and we notice that the majority of students did use 6 documents or more.

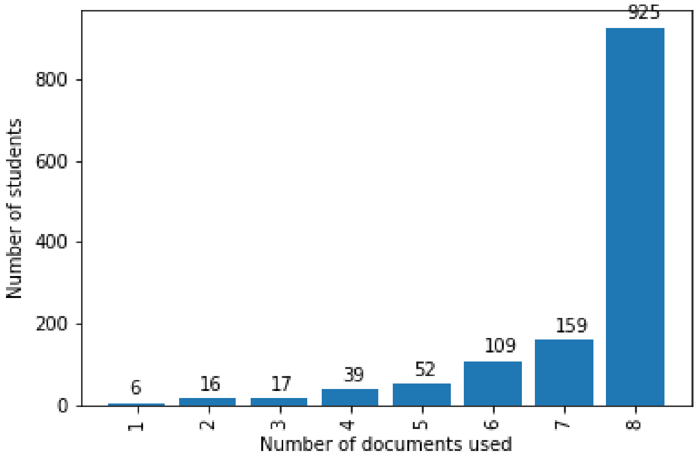


Fig. 4.6.: Number of students for each number of document usage.

### 4.5.2 Refining the Students' usage

There are several actions that a student can do using the system. In fact the action type is composed by 15 different values. Not all action types reflect an active type of interaction between the student and the "BookRoll" System. To express this difference, I separate the actions in two categories. Browsing and Interaction.

As explained in Table 4.4, the browsing actions are all actions related to displaying the contents of documents and the actions that allow the student to read through the document. The interaction actions are all actions that allow the student to act on the documents like the bookmark, the marker and the memo. Using the original categorical feature called “operationname”, we can count these “Browsing” and “Interaction” actions separately, for each student. After that, we divide them by the total number of actions that were done by the respective student. Using this division, we calculate the ratio of “Browsing” and “Interaction” actions. In addition to the “Browsing” and “Interaction” features I also calculate other features such as the action counts, the documents used, the total length of the memos and the ratio of difficult and important markers.

| Feature name | Feature composition |
|---|---|
| Browsing | OPEN, CLOSE, NEXT, PREV, SEARCH, SEARCH_JUMP, PAGE_JUMP, LINK_CLICK |
| Interaction | {ADD, DELETE} BOOKMARK, BOOKMARK_JUMP, {ADD, DELETE} MARKER, {ADD, DELETE, CHANGE} MEMO |

Tab. 4.4.: Browsing and Interaction actions.

### 4.5.3 Taking the class time into account

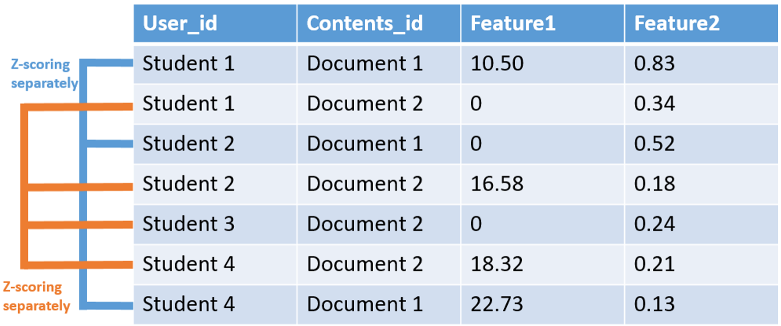
Even if the students can access the course material anytime, anywhere. Most of the time they use the BookRoll system during the class time. Hence, there is another dimension of the analysis. I investigate the difference between the students usage of the system when they are inside the lecture compared to when they are outside of the class time.

| Feature name | Mean | Standard Deviation | T-test |
|---|---|---|---|
| in_lecture_actions_count | 1219.33 | 685.46 | 48.211 |
| out_lecture_actions_count | 233.96 | 306.72 | (p-value <0.01) |
| in_lecture_docs_used | 7.22 | 1.40 | 44.784 |
| out_lecture_docs_used | 3.86 | 2.33 | (p-value <0.01) |
| in_lecture_total_memo_length | 822.91 | 2977.43 | 9.752 |
| out_lecture_total_memo_length | 23.34 | 167.81 | (p-value <0.01) |
| in_lecture_browsing | 94.07 | 7.88 | 4.322 |
| out_lecture_browsing | 91.09 | 23.81 | (p-value <0.01) |
| in_lecture_interaction | 5.84 | 7.45 | 9.587 |
| out_lecture_interaction | 3.08 | 7.37 | (p-value <0.01) |
| in_lecture_important | 55.1 | 42.69 | 24.676 |
| out_lecture_important | 17.08 | 36.30 | (p-value <0.01) |
| in_lecture_difficult | 23.42 | 33.79 | 12.939 |
| out_lecture_difficult | 8.31 | 25.74 | (p-value <0.01) |

Tab. 4.5.: Features comparison between inside and outside the lecture time.

Table 4.5 shows this comparison using the features defined in the previous section. In fact, the {in, out}_lecture_browsing and {in,out}_lecture_interaction features are measured according to the composition in Table 4.4. We can easily notice that there is a statistically significant difference in the students' usage when they are in the lesson and when they are not. Firstly, they don't use the system a lot when they are not taking a class. It is clear from the number of actions and the number of documents used. Both of them are reduced drastically. Also, when they use the memo function, they deal with a lot shorter memos. However, the type of usage does not change a lot. In fact, the ratio of browsing actions is dropped slightly similarly to the interactive actions. Similar results appear to be happening for the marker usage. Overall, students do not use the system a lot when they are not in the lesson time.

The final step of preparing the document-wise approach is to use the z-score function to measure the ranking of the student features compared to the other students that used the same document. This is done by taking a sub-part of all students that used a particular document and we apply the z-score function to their features. In this way, the features are transformed from normal values to “ranking” values. Figure 4.7 exposes the z-scoring based on documents. Students’ behavior is compared to other students’ behavior separately in each document.



| User_id | Contents_id | Feature1 | Feature2 |
|---|---|---|---|
| Student 1 | Document 1 | 10.50 | 0.83 |
| Student 1 | Document 2 | 0 | 0.34 |
| Student 2 | Document 1 | 0 | 0.52 |
| Student 2 | Document 2 | 16.58 | 0.18 |
| Student 3 | Document 2 | 0 | 0.24 |
| Student 4 | Document 2 | 18.32 | 0.21 |
| Student 4 | Document 1 | 22.73 | 0.13 |

Fig. 4.7.: Z-scoring based on documents.

To validate this approach, we compare it to a baseline model where no sophisticated feature engineering was applied. I start by applying the univariate features selection separately for each approach. Table 4.6 shows the features selected in each approach.

| Features selected for the baseline approach | Features selected for the document-wise approach |
|---|---|
| in_lecture_browsing | in_lecture_browsing |
| in_lecture_difficult | in_lecture_difficult |
| out_lecture_difficult | out_lecture_difficult |
| in_lecture_important | in_lecture_important |
| out_lecture_important | out_lecture_important |
| in_lecture_interaction | in_lecture_interaction |
| in_lecture_docs_used | in_lecture_total_memo_length |
| out_lecture_total_memo_length | out_lecture_interaction |

Tab. 4.6.: Features selected for baseline and document-wise approaches.

After selecting the appropriate features, I proceed to optimizing the machine learning pipeline using GP then I validate using the held-out data. Table 4.7 exposes the outcome of the validation phase. For both, boosting regressors have been chosen.Overall, the prediction performances have improved in the document-based approach. In fact, the RMSE attain 7.19 in the document-based approach while the baseline has an RMSE of 7.31. Similarly, the max error in the document-based is lower compared to the baseline. However, the error variance in the baseline is higher, thus better, when compared to the document-based approach. In this case, both models' error variance is not very good. In fact, the error variance in the baseline is about 0.017 while the error variance in the document-based approach is slightly worse, attaining 0.014.

| | Baseline | Document-based |
|---|---|---|
| ML Method | Gradient Boosting Regressor | eXtreme Gradient Boosting Regressor |
| RMSE | 7.31 | 7.19 |
| Max error | 34.05 | 31.29 |
| Error Variance | 0.017 | 0.014 |

Tab. 4.7.: Validation results comparison between the baseline and the document-based approach.

Experimental results suggest a minor improvement of the prediction performance of the document-based approach. In fact, the document-based model had better results in all metrics. This suggests that aggregating the students reading behaviors using the documents gives a fair comparison since the documents' difference has an influence in the way the students use it.

## 4.6 Summary

This chapter elaborates the second study using implicitly gathered students objective data. There are several contributions in this study. Firstly, I explored the students' scores to find which one can be used as a threshold to distinguish between highly performing students and the others. Secondly, I prove that the students reading behavior changes if they are not in the lecture time. Finally, I examined the effects of aggregating the students data by document and found that it improves the performances of the prediction models.

In the second part of this dissertation, I use explicitly gathered students subjective data.

# Part II

## Explicitly Gathered Students’ Subjective Data

# 5 Assessing the Students' Learning Experience from their Comments

## 5.1 Background and Related Work

Thanks to the continuous advances in educational technology, more educational institutions are adopting educational software systems. Indeed, the usage of such systems opens up countless opportunities of gathering and analyzing insightful data. Moreover, it allows building different sophisticated models used to help improve the students learning experience [17, 33, 63]. Additionally, predicting students' performances and behaviors is a growing subject of interest in education-related fields. Researchers are building different predictions models. Thereafter, instructors use the results of these prediction models to improve their decision making and provide better guidance and assistance to their students, especially the ones that need it the most. However, it is essential to establish methods of assessing students' performance before trying to build the predictive models. Diverse crafty and innovative solutions were designed to improve the students learning experience. But, the assessment of the students' performance and learning experience has to be considered as a continuous process which aims to increase the quality of students' learning [28]. In fact, many different means are used to assess the students' performance. Some of them rely on careful observations during class time, while others are more explicitly elaborated such as test scores and questionnaires [37, 38].

The usage of different assessment methods is influenced by the educational settings and environment. In fact, in the classroom, the teachers and professors do not only have to teach and convey the content of the course to their students but also have to guess the students learning experience through careful observation of their behavior and attitude. While giving immediate feedback in the classroom is very effective, it is hard to keep track of all students learning experience across different classes during the whole academic semester or year, especially if they are not responsive or do not express their problems [26]. Hence, carefull observation is an effective but challenging mean of assessing the students' learning experience. Other means like the traditional exercises assessment, test scores and attendance are very handy and helpful but sometimes they are not enough to fully grasp the range of students' behavior and learning experience [23, 75]. Therefore, finding different ways of gathering insightful data about students is a vital step toward improving the educational environment. The diversity of such educational data allows the development prediction models that cover a broader range of students' behavior, performance, and situation in general. Moreover, this educational data can be gathered from different sources and stored in different ways.

Indeed, one source of very valuable data is questionnaires and surveys. They have been used for a long time, however, research using solely data coming from questionnaires is still limited compared to other sources of data. For instance, severalresearchers designed designed a questionnaire that measures the students affect such as personality, motivation and attitude, then they built a predictive model of students' english langauge aptitude based on reading, speaking and writing independently [3]. In a different context, Jiang et al. [29] used a large collection of course evaluation survey for undergraduate and build a predictive model using linear regression to extract the aspects that influence the evaluation of the course and the responsible teacher.

Predictive models using data gathered from questionnaires are not abundant. And it is even more rare to find research topics that use solely textual data coming from questionnaires. For example, Sliusarenko et al. [64] used the textual data gathered from a course evaluation rating survey. The survey's textual data consists of open-ended comments. Then the authors extracted the most important aspects of the students' comments and how they do influence their rating of the course. In another work, Minami et al. [39] used the term-end questionnaire to extract students textual input. They combined the textual data with other sources of data like attendance, test scores and homework evaluation scores and identified the common writing characteristics of highly successful students.

In a different context, Goda et al. [26] designed a questionnaire where students are requested to self-reflect on their learning experience using freely written comments. The survey is conducted after each lesson. The authors also proposed the PCN method. PCN is the abbreviation of Previous, Current and Next. It provides the ability to acquire temporal information of each student's learning activity relatively to the corresponding lesson. The first subset P (Previous) covers all the student's activities prior to the lesson. It can be in the form of preparation of the actual lesson or a review of the previous lesson. The second subset C (Current) is related to all activities made during the class. It particularly covers the students' understanding of the content of the lesson, the problems that he / she have faced and the activities that involve teamwork or communication with peer classmates. Finally, the subset N (Next) encapsulates the students' comments about plans to review the actual lesson and prepare for the next lesson. After that, These comments are reviewed by their professor who give back his own feedback to the students. This allows the students to get guidance when needed, and also the professors to gather valuable data about the students' learning experience. The authors declared that the PCN method incited students to improve their self-reflection on their learning environment and to better strategize on their learning activities planning.

Several subsequent researches were made using the PCN method, mainly to predict students performance and grades. In [68], the authors used a clustering method combined with Latent Semantic Analysis to predict students' scores. In a later research, they used an Artificial Neural Network to predict user grades with a tweak in the labels using an overlapping method [67]. The same authors used differnt techniques such as topic modeling [69], treated the students' comments as a time-series problem [66], and built prediction models using the majority vote while taking into account the succession of the lessons [65].

These previous research topics proved that the students' comment data are reliable sources of information. However, to assess the students' learning experience the professors have to read a huge quantity of comments. This method is not scalable, especially when the same professor is teaching many different classes. In this chapter, I detail how I address this problem by building an automated evaluator of students' learning experience using their freely-written comments.

## 5.2 Comments Collection

To gather the students' comments I use a questionnaire following the PCN method. The questionnaire is composed by five predefined question. Table 5.1 exposes how the questions are divided into subsets of P C and N. Firstly, the subset P (Previous) contains only one question related to the students' activities before the lesson.In the subset C (Current), we have 3 different questions. Firstly, students describe their problems and which content they did not understand well. The second question is about their discoveries during the lesson and finally they report their interactions and cooperation with their peer class mates. Finally, in the subset N (Next), students detail their plans for the next lesson. This questionnaire is provided to students' after each lesson in a programming course.

Tab. 5.1.: Questions and comments following the PCN method.

| Subset | Question | Example of comment |
|---|---|---|
| P (Previous) | What did you do to prepare for this lecture? | I scrolled throught the syllabus |
| C (Current) | Do you have anything you did not understand? Any questions? | I did not understant the recursive functions. |
| | What are your findings in this lesson? | I understood how to declare a function. |
| | Did you discuss or cooperate with your friends? | I worked with my friends to solve the exercice. |
| N (Next) | What is your plan to do for the next lecture? | I will do the homework and submit the report. |

Since there are 5 different questions, the answers are also different. However, there is a pattern in which students are more expressive answering some questions compared to others. Accordingly, I investigate how long are the students' comments depending on the type of the question. Figure 5.1 shows the distribution of comments' length depending on the question type. From the figure, we can see that P and N comments have a somehow similar distribution, except that students tend to write more about their next plans than about their preparations. Comments that describe students' problems and findings have also similar distribution and their lengths are more spread than the other comments. Nevertheless, the median length of comments on problems is lower (10) than the median length of comments on findings (17). Meanwhile, teamwork comments are shorter in general than the others.

## 5.3 Rating comments with one score

### 5.3.1 Manual Annotation

To automate the process of assessing students comments, it is necessary to manually annotate the data in the first place. The task is relatively simple. Therefore, it is done by two students in their Master program. A deep understanding of the programming course materials is not necessary. In fact, both the questions and the students' answers are related to the students' own assessment of their learning activities. The manual rating of the comments follows the given grid. Table 5.2 lists the scores

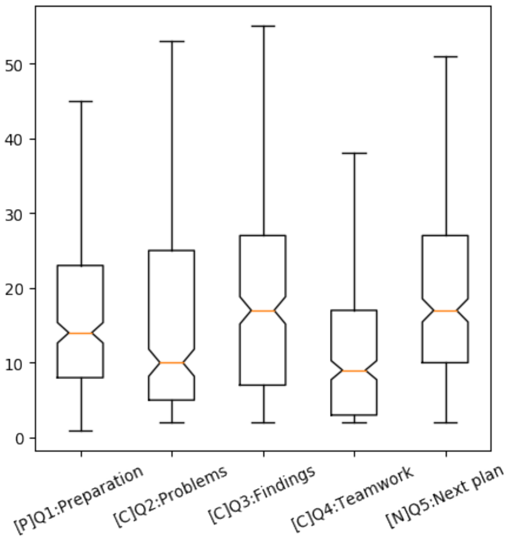


**Fig. 5.1.:** Comments length per question type

and their respective meanings and attributes. In fact, scores are between 1 to 5. The higher the score, the better the comment and the relative learning experience expressed by the student.

**Tab. 5.2.:** Scores grid and the appropriate meaning.

| Score | Meaning |
|---|---|
| 1 | No description of the learning actions, or expressions showing a lack of commitment, giving up or negative attitude. |
| 2 | Small description of the learning activity without details that make easy to understand the problem or the effort made by the student. |
| 3 | Comments that describe briefly with some level of attention to detail the learning activity or showing a moderate degree of commitment, results or troubles. |
| 4 | Students expressing their learning activity in details and have a good level of achievement compared to the expectations at that level of the course. |
| 5 | Students that achieved the expected level of commitment or practice and who successfully described their learning experience. |

## 5.3.2 Text Transformation

Since the comments are written in Japanese, the textual data have to be processed accordingly. Comments are cleaned, normalized and parsed using MeCab. MeCab[1]

[1] https://taku910.github.io/mecab/

is a dictionary-based Part-of-Speech and Morphological Analyzer of the Japanese language.

In order to use textual data with machine learning, it has to be transformed into numerical values. There are different methods for transforming textual data to numerical data. Some of the most famous encoding methods are the following:

### TF-IDF Matrices

The Term Frequency – Inverse Document Frequency, TF-IDF for short, is widely used in different tasks involving text mining such as information retrieval, text classification, and ranking documents' relevancy. It is composed by two parts. The first part is the Term Frequencies, which are the counts of each word in a document and the second part is the Inverse Document Frequency which is obtained by dividing the total number of documents by the number of documents that contain the word. The Inverse Document Frequency was firstly proposed by Karen Sparck Jones in 1972 in a paper called "A statistical interpretation of term specificity and its application in retrieval" [70].

### Doc2Vec

Doc2Vec [31] is an unsupervised machine learning algorithm that encodes documents and paragraphs into vector representations. It was inspired by its predecessor algorithm Word2Vec [36] that generates word vectors from texts. Generating the Doc2Vec weights can be done using two different methods : Distributed Bag of Words (DBOW) and Distributed Memory (DM).

### Pre-trained Word Embedding

Word embedding is a vector representation of a document's vocabulary. It can grasp the relationship between words and their context. Word2Vec is a famous method of generating this word embedding. However, pre-trained word embedding means that the model was already trained on a large corpus of documents, such as Common Crawl[2] and Wikipedia[3] texts. Therefore, the vector representations are already generated.

[2] https://commoncrawl.org/

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

### 5.3.3 Single or Multi Model

The comments depend on the questions. Moreover, there are two questions that carry a contradictory meaning. In fact, when students answer "Yes" to the question: "Did you have problems?" the learning experience is negative. Meanwhile, when the students answer "yes" to the question: "Did you understand?" the learning experience is positive. So I compare the performances by building a single model for all types of comments and compare it to a multi-model approach in which I make 4 models. 1 model for the problems, 1 model for the findings, 1 model for teamwork and 1 model for preparation and plan since they are similar in the composition and in the length distribution.

Moreover, for the single and the multi model I use the three text encoding methods. Therefore, there are six alternatives to test. Table 5.3 provides a summary of the name of each alternative and their differences.

Tab. 5.3.: Summary and names of the investigated alternatives

| Name | Characteristics |
|---|---|
| single_tf-idf | Analyze all comments regardless of the type of the question and generate features using the TF-IDF method. |
| single_doc2vec | Analyze all comments regardless of the type of the question and generate features using Doc2Vec sentence vectors. |
| single_pre-trained | Analyze all comments regardless of the type of the question and generate features using pre-trained Japanese language word vectors. |
| multi_tf-idf | Generate 4 models (P and N; Misunderstanding; Findings and Teamwork) and analyze the comments relative to each model using TF-IDF. |
| multi_doc2vec | Generate 4 models (P and N; Misunderstanding; Findings and Teamwork) and analyze the comments relative to each model using Doc2Vec. |
| multi_pre-trained | Generate 4 models (P and N; Misunderstanding; Findings and Teamwork) and analyze the comments relative to each model using pre-trained Japanese word vectors. |

Figure 5.2 summarizes the whole workflow. The first step is to split the data into testing and training. In fact, as it is custom in prediction models, I hold out part of the dataset as unseen validation data. For the Multi-model approaches, I held out comments from each corresponding question (e.g. holding out P or N comments from the P/N model). For the single model, in which all comments are mixed regardless of the type of the question, I held out comments using a stratified split. The stratified split allows the respect the proportions of each type of comment in the

whole dataset. I used a ratio of $1/4$ of the dataset for unseen validation only data. Then, I use MeCab to extract words and their Part-of-Speech. After that, I proceed to build the models. For each approach, I use the three discussed methods for feature engineering. Therefore, I have 6 different alternatives for comparison. However, in the multi-models approach I create 4 different models and use the appropriate comments for training: Previous and Next comments, Problems comments, Findings comments, and Teamwork comments. When I evaluate each alternative in the multi-models approach, I take the average of the 4 models' performances using equation (5.1):

$$P_{multi} = \frac{\sum_1^4 PM_i}{4} \tag{5.1}$$

Here $multi$ is the multi-model alternative and $PM_i$ is the $i^{th}$ model of the respective alternative.

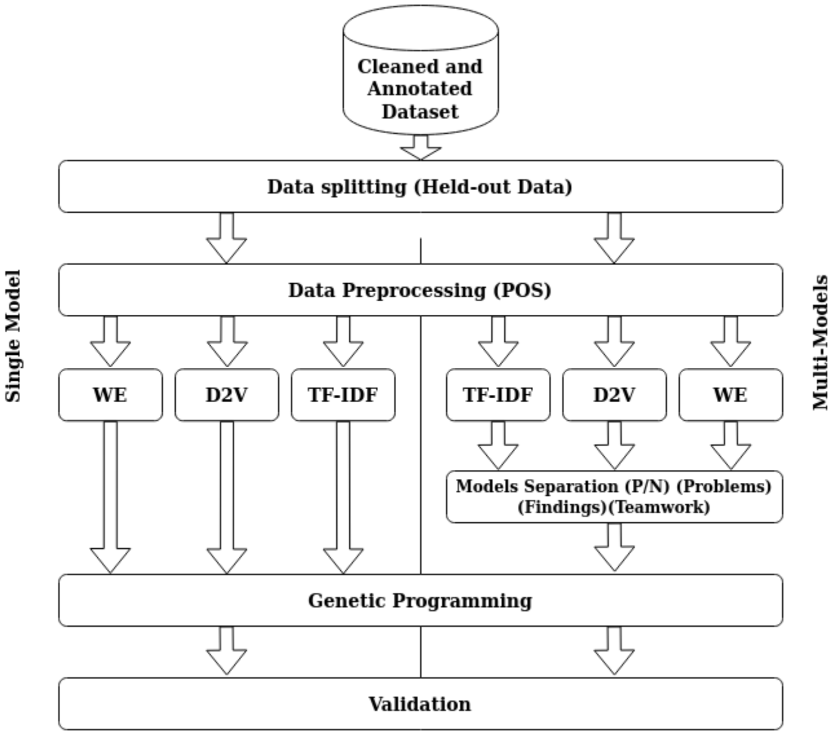

Cleaned and Annotated Dataset
Data splitting (Held-out Data)
Data Preprocessing (POS)
Single Model
Multi-Models
WE
D2V
TF-IDF
TF-IDF
D2V
WE
Models Separation (P/N) (Problems) (Findings)(Teamwork)
Genetic Programming
Validation

Fig. 5.2.: Models building workflow.

Afterward, each of the 6 alternatives will be optimized separately from the others. Once the optimization phase finished, I validate each alternative "pipeline" and compare their results using the held-out data. Table 5.4 shows the results of the optimization phase, with the chosen machine learning technique and its best score.

Tab. 5.4.: Results of the optimization process

| Alternatives | Best Method | Best Score |
|---|---|---|
| single_tf-idf | Random Forest Classifier | 0.633 |
| single_doc2vec | Random Forest Classifier | 0.619 |
| single_pre-trained | Random Forest Classifier | 0.676 |
| multi_tf-idf | K-Nearest Neighbors | 0.705 |
| multi_doc2vec | P+N: SVM<br>Misunderstand: Random Forest Classifier<br>Findings: K-Nearest Neighbors<br>Teamwork: Random Forest Classifier | 0.662 |
| multi_pre-trained | Random Forest Classifier | 0.740 |

In the first approach, in which I mix all comments regardless of the type of the question, I notice that for all its alternatives, the Random Forest Classifier was the best machine learning method, with the best scores of 0.633, 0.619, 0.676 respectively for the usage of TF-IDF, Doc2Vec and the pre-trained word embedding. For the separated models, I have various machine learning methods, giving the best results. In fact, when using the TF-IDF weighting, all four separated models have K Nearest Neighbor as the best classifier with an average score of 0.705. However, when using the Doc2Vec technique, different machine learning methods give the best results to each separated model. In fact, for the P (Previous) + N (Next) model, Support Vector Machine (SVM) is the best. For the misunderstanding model, the Random Forest Classifier achieves the best score similarly to the teamwork model. Finally, the model of findings uses K Nearest Neighbors. The last alternative, in which I use a pre-trained word vector and four different models, get the Random Forest Classifier as the best performing method for all the models. The average score in this alternative is 0.740.

Once I decided which machine learning methods I will use, I train the models accordingly and proceed to validate each performance using unseen data. Table 5.5 shows the validation scores of all models. The best scores are written in boldface. The first notice is that all models did perform better than chance in giving the right score to the comments. The best results in terms of accuracy and precision are achieved by the multi-models approach using the pre-trained word embedding having attained an accuracy of 0.740 and a precision of 0.668. This model also scored second best in the recall with 0.630 and also second-best in the F1-score having 0.635. With the same approach, but using the TF-IDF weighting matrix, it achieve the best scores in recall with 0.660 and in F1-score having 0.650. This model is the second-best in accuracy attaining 0.662 similarly in precision by obtaining 0.655.

Tab. 5.5.: Validation scores for all models

| | Accu | Prec | Recall | F1-score |
|---|---|---|---|---|
| single tf-idf | 0.603 | 0.560 | 0.600 | 0.560 |
| single doc2vec | 0.586 | 0.510 | 0.590 | 0.530 |
| single pre-trained | 0.590 | 0.550 | 0.590 | 0.550 |
| multi tf-idf | 0.662 | 0.655 | **0.660** | **0.650** |
| multi doc2vec | 0.548 | 0.545 | 0.548 | 0.505 |
| multi pre-trained | **0.740** | **0.668** | 0.630 | 0.635 |

Separating models for each question is the best approach in this dataset, however, there are a few shortcomings in following this approach in the long term. In fact, this approach is not easily maintainable or scalable. First, any improvement or update in this approach has to be replicated as many times as there are separated models. Furthermore, if we are planning to add more questions or implement an interactive interface to gather students' comments, these models cannot respond very well. Nevertheless, in the actual situation and scale, they might be very valuable and effective in assessing students' comments without much human intervention. On the other hand, building a model capable of generalizing well toward comments regardless of the initial question or aspect is considered to be the right way. Not only in terms of scalability, but also in the complexity of the whole system.

## 5.4 Extending the comments score

When rating the comments, it was ambiguous to give just one score. In fact, the are two aspects to rate in a comment: The explicit quality of the comment itself and the learning experience of the students. Somehow, there are comments that are well written and have a good description, but the learning experience of the students is negative. Consequently, the good quality of the comment mitigate the negative learning experience and the overall score is not accurately describing the learning experience of the student. The reverse is also true. For example, students provide a very short comment without any detail describing a positive learning experience. A frequent example is answering "No" to "Did you have problems?". In this case, the learning experience is positive but the comment quality is poor. Once again the quality of the comment mitigate the overall score which make the reviewer rating inaccurate. The solution to this problem is to add one more score value for the reviewer. Hence, a comment, now, have 2 scores: The descriptive score and the Learning experience score.

### 5.4.1 Enhanced Manual Annotation

To avoid the confusion, not only I added an additional score to each comment, but also I build a user interface where the reviewers can easily rate the comment and give feedback to them. The feedback will be used in the next chapter. Furthermore, I set the scoring using a 5-values Likert scale. Basically, in a Likert scale the distance between the candidate values is the same. Table 5.6 and Table 5.7 expose the possible values of each score and their meanings. The learning experience score ranges from very bad to very good, and the descriptive score ranges from very short to very detailled.

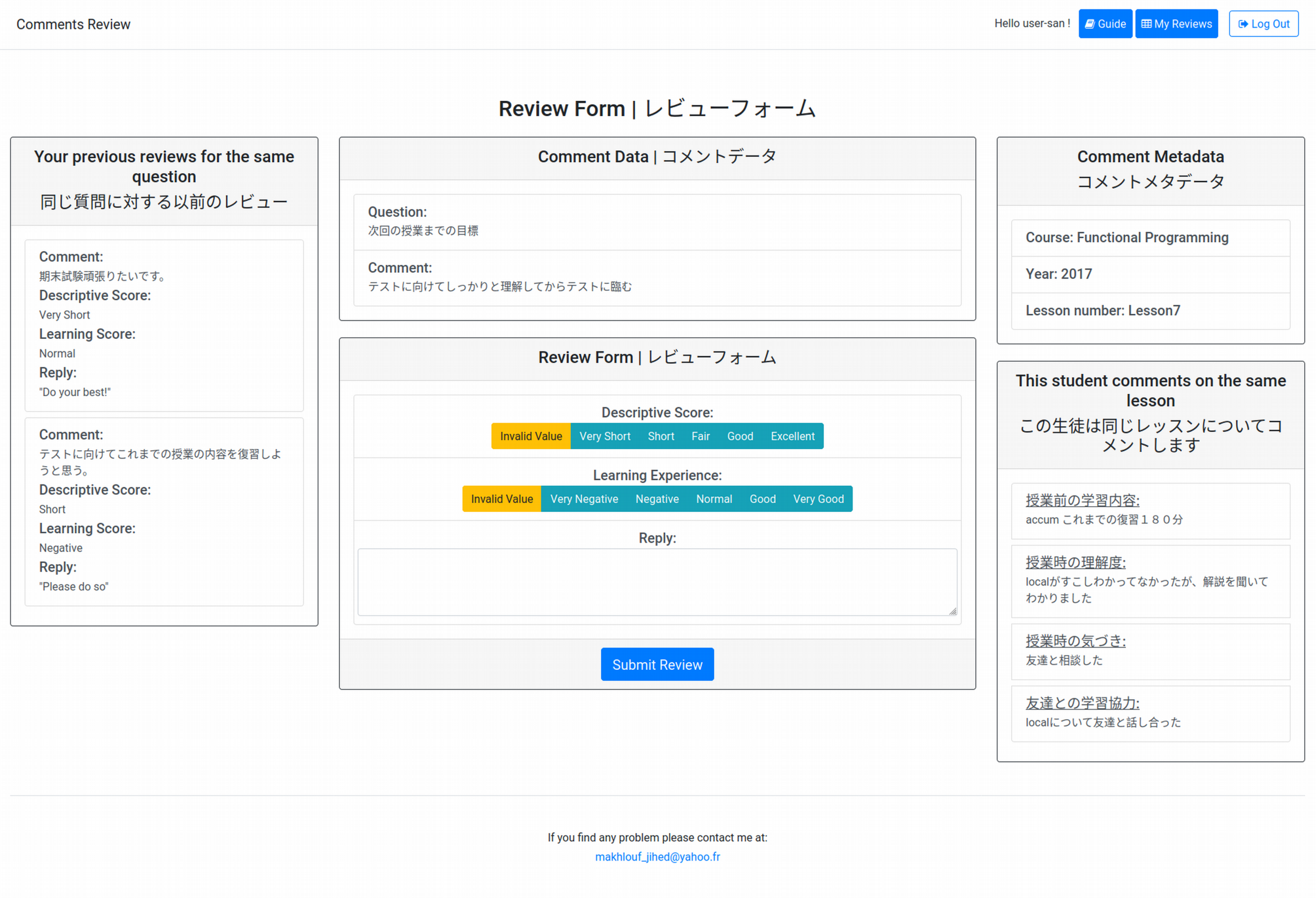
**Tab. 5.6.:** Scoring for the learning experience.

| Score | Meaning |
|---|---|
| 1 | Very bad learning experience |
| 2 | Bad learning Experience |
| 3 | Fair learning experience |
| 4 | Good learning experience |
| 5 | Very good learning experience |

**Tab. 5.7.:** Scoring for the quality of the comment.

| Score | Meaning |
|---|---|
| 1 | Very short comment |
| 2 | Short comment |
| 3 | Normal comment |
| 4 | Detailed comment |
| 5 | Very detailed comment |

One more functionality added to the rating interface is the possibility to view the meta-data about the comment, such as the lesson, the comments of the same student in the same lesson or similar comments to the same question coupled with the reviewer rating to it. Figure 5.3 shows a screenshot of the reviewing interface.



## Review Form | レビューフォーム

**Your previous reviews for the same question**
**同じ質問に対する以前のレビュー**

**Comment:**
期末試験頑張りたいです。
**Descriptive Score:**
Very Short
**Learning Score:**
Normal
**Reply:**
"Do your best!"

**Comment:**
テストに向けてこれまでの授業の内容を復習しようと思う。
**Descriptive Score:**
Short
**Learning Score:**
Negative
**Reply:**
"Please do so"

**Comment Data | コメントデータ**

**Question:**
次回の授業までの目標

**Comment:**
テストに向けてしっかりと理解してからテストに臨む

**Review Form | レビューフォーム**

**Descriptive Score:**

**Learning Experience:**

**Reply:**

**Comment Metadata**
**コメントメタデータ**

Course: Functional Programming

Year: 2017

Lesson number: Lesson7

**This student comments on the same lesson**
**この生徒は同じレッスンについてコメントします**

授業前の学習内容:
accum これまでの復習１８０分

授業時の理解度:
localがすこしわかってなかったが、解説を聞いてわかりました

授業時の気づき:
友達と相談した

友達との学習協力:
localについて友達と話し合った

If you find any problem please contact me at:
makhlouf_jihed@yahoo.fr

Fig. 5.3.: Reviewing interface

### 5.4.2 Adding context to the comment

To avoid building a model for each question type, I investigate the effectiveness of using a padding that contains the type of the question before each comment. For example, if the student commented: "I reviewed the content and practiced at home" when answering the question "What did you do to prepare for this lecture?", then the comment is transformed by adding a padding like: “<preparation> I reviewed the content and practiced at home.”. The same padding technique but with different content is applied to each of the 4 other questions.

To validate this approach of scoring the comments I apply the same text encoding techniques used when we operate with a single score. Hence, we have the same baseline approach compared to a padding approach instead of a multi-model mode. Table 5.8 lists the 6 alternative that I will investigate.

Tab. 5.8.: Names and description of the model building approaches.

| Name | Description |
|---|---|
| baseline_tf-idf | Baseline approach using TF-IDF matrices |
| baseline_doc2vec | Baseline approach using Doc2Vec |
| baseline_pre_trained | Baseline approach using the pre-trained word embedding |
| padded_tf_idf | Using the padding with TF-IDF matrices |
| padded_doc2vec | Using the padding with Doc2Vec |
| padded_pre_trained | Using the padding with loading the pre-trained word embedding |

### 5.4.3 Workflow summary

The figure 5.4 demonstrate the workflow for this study. As explained above, students’ comments are rated according to 2 scores. The descriptive score and the learning experience score. In this research I will build models to predict the learning experience of the students. Therefore, I discard, for now, the descriptive score. After that, I proceeded to clean the data and formulate properly the dataset to prepare for the prediction models. As it is common in prediction models, we split our data and held out a part of the dataset to serve for testing purposes. I proceeded to use a stratified split to respect the proportions of the predicted variable which is the learning experience score. $1/4$ of the dataset is used for held out validation and $3/4$ are used for training. After that, the feature engineering consists of encoding the text into numerical values using the 3 techniques similarly to the previous section. For the padding approach, I transform the comments just before applying the text encoding. Afterwards, each approach will be optimized separately from the others using GP. Finally, approaches are compared with each other after the GP phase.

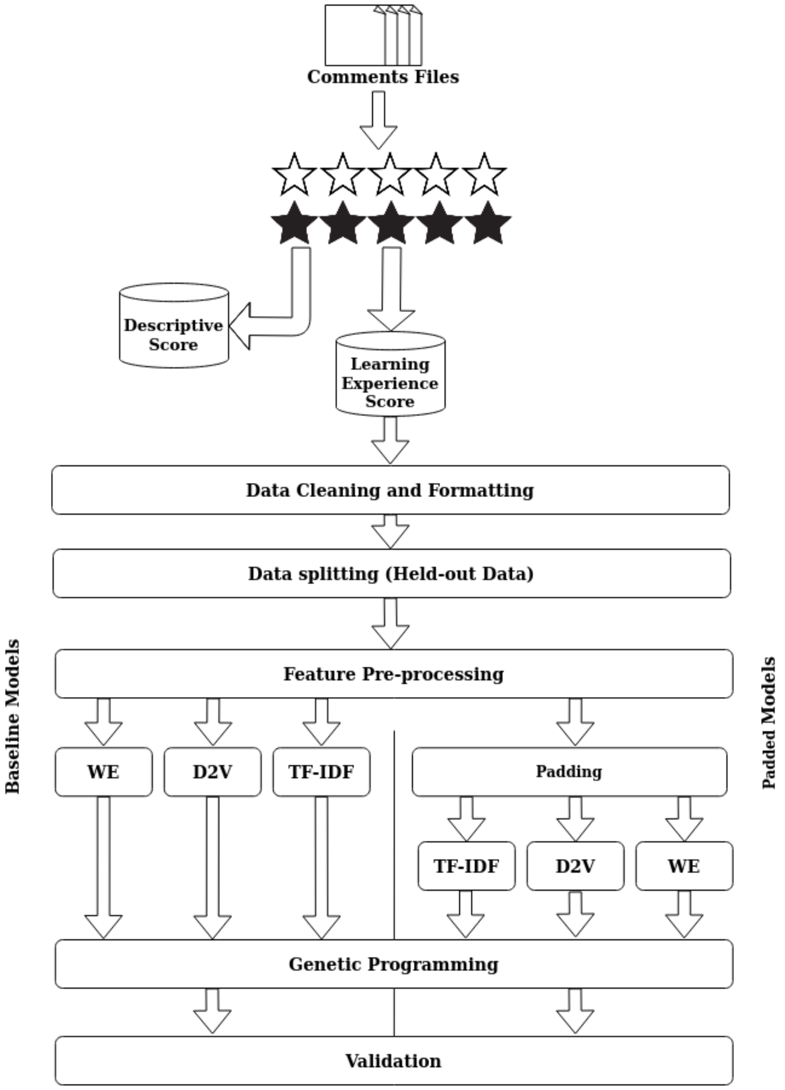


Fig. 5.4.: Models' Building Workflow using two scores

### 5.4.4 Effects of using two scores

The results of the test phase are shown in Table 5.9. The first things we notice is that no value in any metric is under 0.6. The second thing we notice is that the paddded approach using the pre-trained word vectors performed the best in all metrics. It achieved an accuracy, precision and recall values of 0.74 and an F1-score of 0.72. Moreover, we can see that all padding approaches performed better in every metric compared to the baseline approaches. In fact, in the baseline approach, no metric achieved a value higher than 0.67. On the other hand, the padded approaches had all their metrics scores higher than 0.67.

Tab. 5.9.: Validation scores for all models

| | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| baseline_tf-idf | 0.65 | 0.60 | 0.65 | 0.60 |
| baseline_doc2vec | 0.67 | 0.64 | 0.67 | 0.63 |
| baseline_pre-trained | 0.66 | 0.63 | 0.66 | 0.62 |
| padded_tf-idf | 0.71 | 0.70 | 0.71 | 0.70 |
| padded_doc2vec | 0.71 | 0.70 | 0.71 | 0.68 |
| padded_pre-trained | **0.74** | **0.74** | **0.74** | **0.74** |

To better understand the performance of our models we look at the F1-scores in each learning experience score respectively. Figure 5.5 shows the performances of the baseline models. Similarly, the Figure 5.6 shows the F1-scores of the padded models. We can clearly notice that they expose a similar behavior. In fact, most of the models have a low performance when predicting the comments having a learning score of 1 or 5. The baseline models also perform poorly when the learning experience score is 2, but the padded models have better performances, at least more than 0.5. Both approaches models have better results when dealing with comments have a 3 rating, and they reach their peak performances when dealing with comments having a learning experience score of 4.

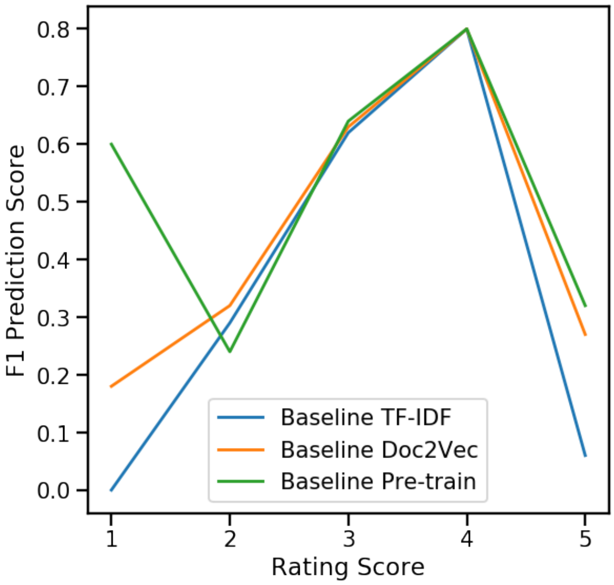


Fig. 5.5.: F1-Scores of the baseline models in each rating score.

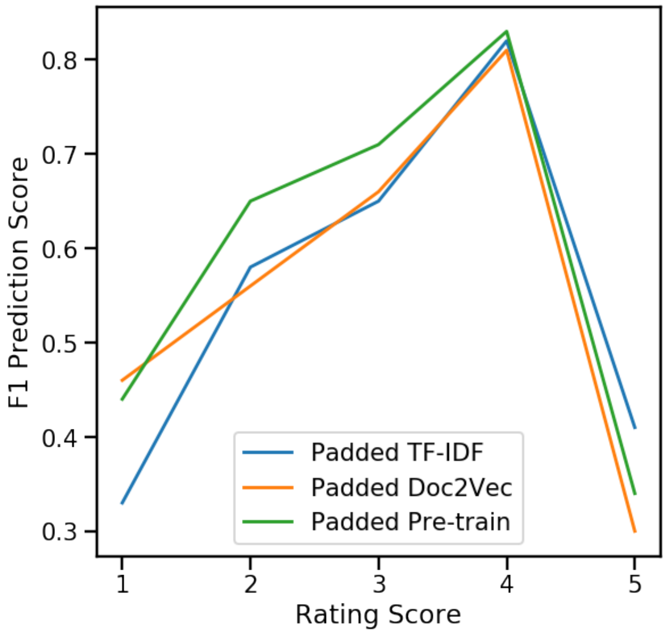


**Fig. 5.6.:** F1-Scores of the padded models in each rating score.

## 5.5 Summary

The aim of this study is to automate the process of assessing the students' learning activities. The research topic was achieved successfully. Several contributions also are made in this study. Firstly, I prove that the question's context is influential in the overall performance of the model. The experimental results show that a simple padding technique can provide a huge boost in the performance. Moreover, I investigate the effect of using two scores. Using two scores instead of only one is helpful and improves the performances across all models. Moving to the fifth chapter, I use the same dataset to build automated feedback to students.

# 6 Automated Feedback to Students

## 6.1 Background and Related Work

It has been demonstrated that proper feedback can lead to a better learning of the students [15, 10, 9]. Different factors, such as the timing and content of the feedback, and the characteristics of the learner, contribute to the effectiveness of the feedback [76]. The timing of the feedback can be delayed or immediate. Different studies found that, in classroom settings, immediate feedback is more effective in improving the learning of the students [1, 61]. The feedback can be as simple as the correctness of the student in a task, or can contain detailed explanation and the reasons of the mistake of the student. Compared to simple feedback, the more detailed and elaborated feedback has been found to be more effective and helpful to the students, especially in the more complex and advanced topics [61].

Beside the factors that influence their effectiveness, feedback have different sources, forms, and structures. In fact, the feedback can be originating from classroom settings or from online classes. Moreover, the feedback can be related to exercises, peer-reviews, group feedback, students self-assessment, and so on [10, 9, 15].

In the same context as the previous chapter's study about students' comments, this chapter's study uses students' freely written comments gathered explicitly using questionnaires. In fact, the purpose of gathering students' comments is two fold. Firstly, it helps professors and educators acquire temporal informations about students' learning activities. Secondly, it allows them to provide feedback to the students and give them the appropriate guidance according to their comments.

However, this task became quickly hard to maintain since the professors find themselves quickly overwhelmed by the number of students comments. Thus, the professors cannot read the students' comments to acquire fine grained individual information about the students. On the other side, students have to wait for a long time before receiving any feedback from the professor. In the previous chapter, I addressed the first half of the problem by building an automated assessment tool of the students' learning activities based on their freely written comments. It can be used to give reports to the professor without the need for them to read all students' comments.

In this chapter's study I tackle the second half of the issue by building an automated feedback model that gives the appropriate reply to students' comments in real-time.

## 6.2 Data Source

Similarly to the previous study, the data is gathered from a PCN-based questionnaire. The comments span from the same course for 2 consecutive years: 2017 and 2018. The course is programming for undergraduate. Each year the course have 7 lessons. Therefore, there are 14 lessons included in the dataset.

Figure 6.1 shows the number of comments gathered for each lesson. Lessons' numbers from 1 to 7 belong to the 2017 class and the lessons from 8 to 14 are relative to the 2018 class. At the first look, we can see that students in the 2018 class were more consistent in providing their comments. While the previous year's class had many missing comments; ultimately having very few comments in the last lesson (lesson 7). In average, we have 38 comments for each lesson. In the 2017 course that average drops to 30 comments per lesson, while in the 2018 course the average number of comments per lesson is 46.

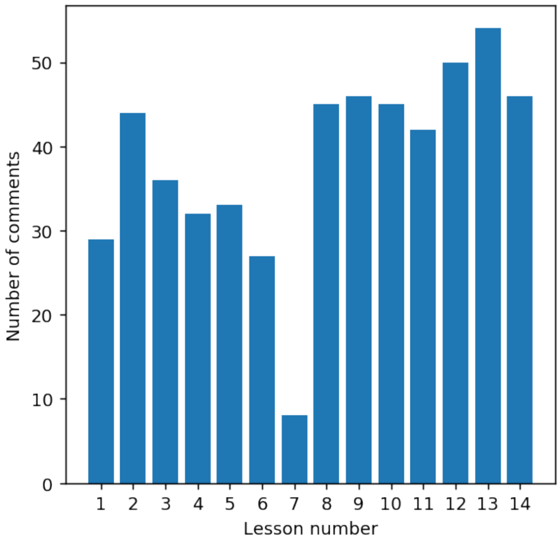


Fig. 6.1.: Number of comments per lesson.

During the manual data annotation, we asked two students in their Master program to annotate and give feedback to the comments. Figure 5.3 of the last chapter shows the interface used by the reviewer to provide the feedback to undergraduate students' comments.

## 6.3 Baseline Feedback Model

The simplest way of building the feedback model, is to gather the feedback messages without transformation. A retrieval-based feedback model formulates the research topic as a multi-class classification problem. After some initial cleaning and preprocessing, the final set of feedback classes is composed by 112 unique feedback message. The textual data was cleaned, normalized then parsed using MeCab.

Padding the question type within the comment have been effective when building the automatic comments assessment models. I investigate its effectiveness while building the automatic reply system. Therefore, it is compared with a normal process where no padding is applied. To encode the textual data, I use TF-IDF and Doc2Vec for each approach. Table 6.1 exposed the four approaches that I investigate

Tab. 6.1.: Feedback models for the baseline approach

| **Model name** | **Characteristics** |
|---|---|
| TF_Simple | Normal approach using the TF-IDF text encoding technique |
| D2V_Simple | Normal approach using the Doc2Vec weights |
| TF_Padded | Padding the comment with the question type and encode the text with TF-IDF |
| D2V_Padded | Padding the comment with the question type and encode the text using Doc2Vec |

Finding the best machine learning pipeline was done using GP. Later I validate the performances of the models using held-out data. The results of optimization phase are exposed in table 6.2. The optimization process found that the Support Vector Machine had the best accuracy result in the TF-IDF based model with a score of 0.433. When using the Doc2Vec, Random Forest Classifier was the best method by reaching 0.379 accuracy score. However, when we integrate the context of the question in the comment we have an improvement in the accuracy score for both models. In fact, in the TF-IDF model we found that the Extreme Gradient Boosting algorithm achieved the best results with an accuracy score of 0.650 and when using the Doc2Vec, we have again the Random Forest Classifier as the best method attaining 0.519 accuracy score.

Tab. 6.2.: Results of the GP phase

| Model's name | Best method | Best Accuracy |
|---|---|---|
| TF_Simple | Support Vector Machine | 0.433 |
| D2V_Simple | Random Forest Classifier | 0.379 |
| TF_Padded | XGBoost | 0.650 |
| D2V_Padded | Random Forest Classifier | 0.519 |

The validation results are shown in table 6.3 We can see that the model using TF-IDF vectors and padding the question type in the comments achieved the best scores across all validation measures. In fact, it have reached 0.247 in the Macro F-score, 0.664 in the Micro F-score, 0.107 in the average word mover's distance and a maximum WMD of 0.507. In the other hand, the baseline approach using Doc2Vec had the worst performances by having the lowest Macro F-score of 0.106, the lowest Micro F-score as well, going down to 0.356. Its average WMD is the highest attaining 0.212 which means that when it does not classify the feedback correctly, it still does not choose a close enough class. The worst value of Max WMD is achieved by the baseline model using TF-IDF

Tab. 6.3.: Results of the validation phase

| Model's name | Macro F-score | Micro F-score | Avg WMD | Max WMD |
|---|---|---|---|---|
| *TF_Simple* | 0.133 | 0.429 | 0.184 | 0.543 |
| *D2V_Simple* | 0.106 | 0.356 | 0.212 | 0.529 |
| *TF_Padded* | **0.247** | **0.664** | **0.107** | **0.507** |
| *D2V_Padded* | 0.158 | 0.467 | 0.136 | 0.514 |

These results can be explained by the high imbalance of the feedback classes. In fact there are 5 feedback classes that are repeated more than the others. Figure 6.2 shows the histogram of the feedback classes. It is noticeable that around $3/4$ of the feedback classes are unique. And by regrouping the feedback classes by questions there are still plenty of feedback classes that appear only once.

## 6.4 Refining the feedback messages by Clustering

### 6.4.1 Manual Clustering

In the baseline approach, I did not proceed to any particular data engineering. With the high number of feedback classes and their distribution it was hard to achieve remarkable results. One way to solve the issue is clustering. The most

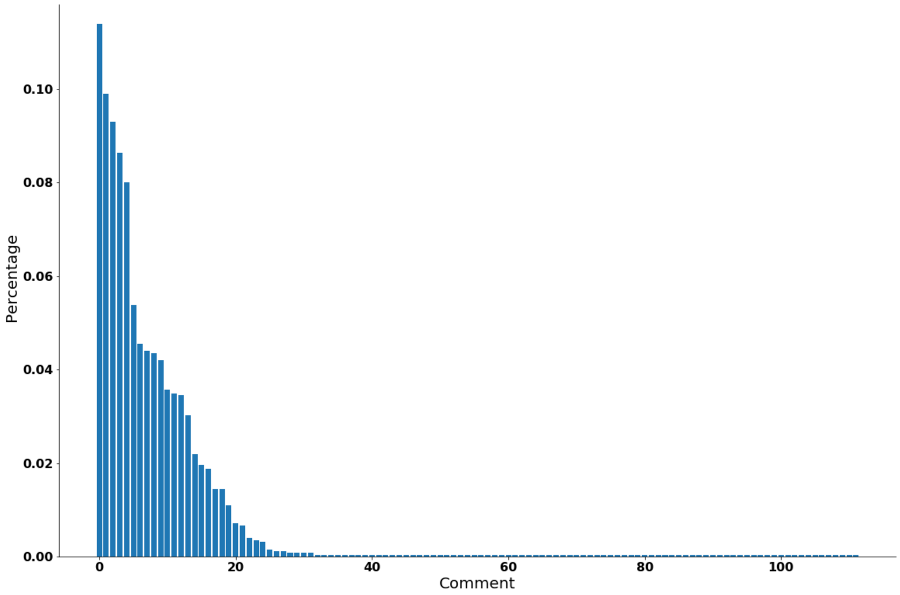


Fig. 6.2.: Histogram of the feedback classes.

straightforward way of clustering the comments is to use the question types first. Once the comments were separated by the question types, I check the feedback provided by the reviewers. I found that many feedback messages are similar but formulated differently, therefore they are counted as different feedback messages. So, I proceed to manually regrouping and clustering the comments based on the the meaning of the feedback messages provided. With this clustering, I managed to reduce drastically the number of unique feedback classes.

Also, in the process, we made sure that the feedback are not shared in between questions, which means each feedback message is unique in the dataset, even outside of its corresponding question. From the 120 feedback messages, we only kept 22 unique feedback messages. Figure 6.3 exposes the number of comments for each class. We easily notice that there are some predominant classes, and that reflects the type of comments that the students provided as well. For example, many students reply with "None", "Nothing" or "Nothing in particular" when they answer the question "Did you have any problems?". At such times, the feedback given by the reviewer was to encourage them to self-reflect more. Even if there are predominant classes within the questions types, there is not any class that has the absolute majority of data points. To investigate the distribution of the feedback classes between the questions types we count the number of feedback classes for each question type, before and after the clustering.

In fact, Table 6.4 shows the distribution of the number of feedback classes for each question type. In average, we have 24 feedback classes by question before the clustering. After the clustering, we have 4.4 feedback classes for each question.

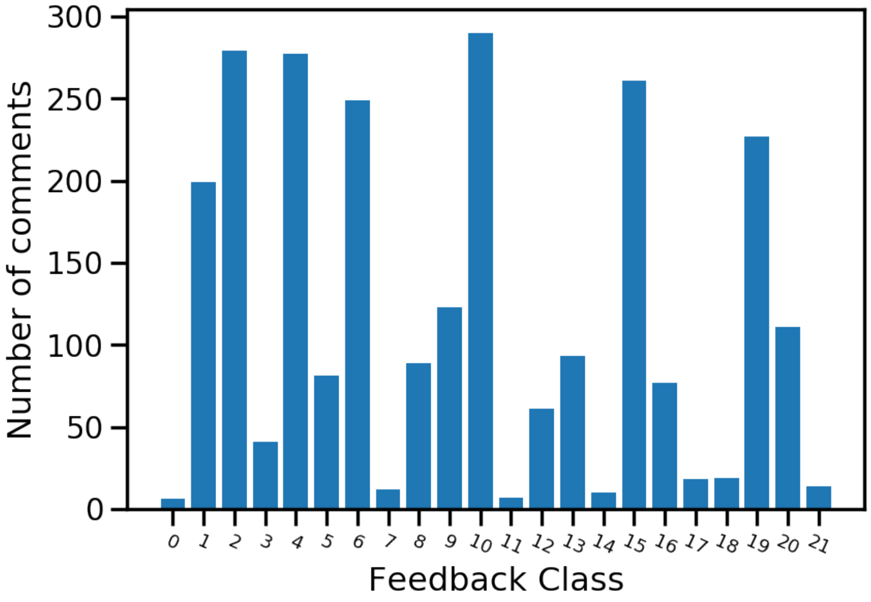


Fig. 6.3.: Number of comments per feedback message.

Tab. 6.4.: Distribution of the feedback classes by Question.

| Subset | Question | Before | After |
|---|---|---|---|
| P | Preparation | 26 | 5 |
| C | Problems | 30 | 6 |
|  | Findings | 34 | 5 |
|  | Teamwork | 10 | 3 |
| N | Plans | 20 | 3 |

### 6.4.2 Features Preprocessing

During my previous analysis, I did not take advantage of the variety of the content of the students' comments. In fact, some students include lines of code in their comments. While other include temporal data. In the previous cleaning phase, some special characters used for detecting the source code were removed within the cleaning process. This time, I utilize the full range of informations available in the comments. Therefore, The first step in the pre-processing phase is to remove line breaks, redundant or extra blank spaces. Special characters and punctuation are kept for later usage. After that, English texts were transformed to all lower case. For the Japanese text, it was firstly normalized to avoid problems between half-width and full-width writings and similar issues. This step was done using the neologdn[1] normalizer for the Japanese language. Just by normalizing the feedback messages we could spare some feedback classes due to small issues in text encoding during the review phase.

After cleaning up and normalizing the text, we proceed to some pattern detection. In fact, there are two main patterns that we noticed. The first one is related to the "Preparation" question. Many students write the duration of their preparation. Some students write in minutes, while others write in hours. So the main idea was to replace any occurrence of the time of preparation by a special token called "studyTime". The second pattern was the incorporation of source code inside the comment. Since it was a programming course, we had to detect the syntax of the programming language within the comments using the special characters kept in the previous pre-processing phases, and replace the source code by the special token "Code". After the pattern replacement, we clean again our comments from the unused special characters. Finally, we use MeCab for the parsing and POS (Part Of Speech) tagging.

At this point of the analysis, we have proven in several occasions that using a simple padding to include the question's type within the comment improves the performance. Therefore, we include this padding in the features engineering phase, right before the text encoding.

On top of clustering the comments, I investigate the application of state-of-the-art deep learning language model called BERT. BERT is the abbreviation of Bidirectional Encoder Representations from Transformers. It is a language model released in 2018, that achieved state of the art performances in different natural language processing tasks [16]. BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. BERT is very useful since fine-tuning a pre-trained BERT model does not require heavy changes in the neural network architecture.

### 6.4.3 Experiments

To accelerate the experiments I reduce the number of machine learning methods that we test. In fact, a study about machine learning found that Random Forests

[1] https://github.com/ikegami-yukino/neologdn

and Support Vector Machines have the best performances across a wide range of applications [19]. Then I add eXtreme Gradient Boosting since it performs very well in machine learning competitions. The general workflow is exposed in Figure 6.4. After clustering the feedback classes, the data is split to training and testing. After that, I apply the features engineering steps and proceed to add the adequate padding to each comment. Then, I proceed to fine tuning the three machine learning methods using a grid search. Once the best machine learning method is found, we train and compare it to BERT language model.

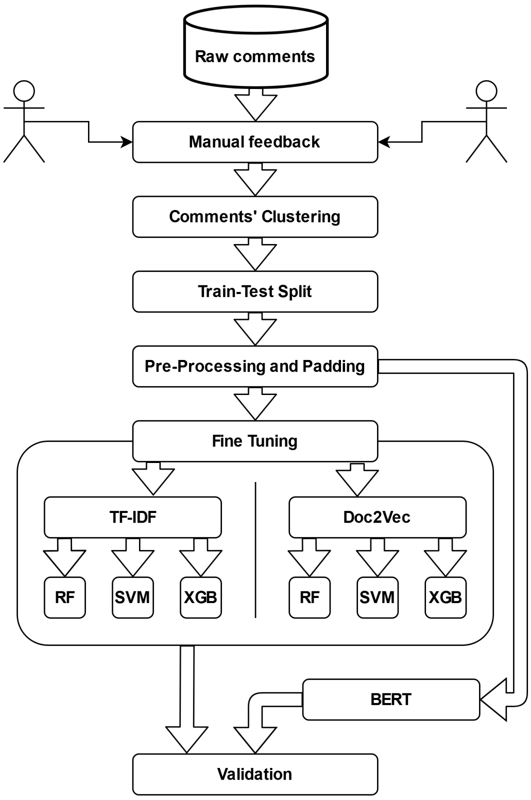


Fig. 6.4.: Summary of the process.

The results of the fine tuning phase are exposed in Table 6.5. When using TF-IDF for text encoding, the Random Forest classifier achieved the best average accuracy attaining 0.77, followed by the SVM with 0.76, and lastly came XGBoost attaining 0.75 average accuracy. On the other side, when Doc2Vec text representation was used, XGBoost had the best average accuracy score reaching 0.73. Random Forest came second with a score of 0.72 and worse performance was achieved by SVM with 0.70 average accuracy score.

Tab. 6.5.: Average accuracy score after the cross validated grid search.

| | Random Forest | Support Vector Machines | eXtreme Gradient Boosting |
|---|---|---|---|
| TF-IDF Encoding | **0.77** | 0.76 | 0.75 |
| Doc2Vec Encoding | 0.72 | 0.70 | **0.73** |

After finding the best performing machine learning algorithm for each textual encoding method, we compare them to the usage of BERT. We train the models on the training data and validate our results using the unseen testing data.

Since we are analyzing Japanese text, we loaded a pre-trained model trained on Japanese Wikipedia. The pre-trained model was provided by Tohoku University [2]. After that, we add an output layer to the deep neural network to adapt it to the classification problem that we have and the number of classes of our feedback.

The results of the validation phase are shown in Table 6.6. We can see that the usage of BERT language model improved significantly the performances of our classification model. It outperformed the other techniques. It achieved 0.81 accuracy. And when checking the performance class-wise, we can see that it also achieved robust performances in the weighted precision attaining 0.78, the weighted recall by reaching 0.81 and also the weighted F1 score obtaining 0.79. The two other methods achieved results similar to each other with a slight advantage for TF-IDF encoding with the usage of Random Forest algorithm. However, when we measure the Macro F1 score, all models do not perform very well. In fact, the TF-IDF model attained 0.50 while the two other models achieved a score of 0.53. Indeed, this is caused by the imbalance between the feedback classes and the number of data points in each class.

Tab. 6.6.: Validation scores on unseen data.

| Metric | Accuracy | Weighted Precision | Weighted Recall | Weighted F1 | Macro F1 |
|---|---|---|---|---|---|
| TF-IDF_RF | 0.75 | 0.74 | 0.75 | 0.74 | 0.50 |
| Doc2Vec_XGB | 0.74 | 0.74 | 0.73 | 0.73 | **0.53** |
| BERT-based | **0.81** | **0.78** | **0.81** | **0.79** | **0.53** |

[2] https://github.com/cl-tohoku/bert-japanese

### 6.4.4 The Effect of the Number of Comments on the performance

Figure 6.5 shows the Weighted F1 scores achieved by each model for each feedback class. The performance is not consistent across all classes. To investigate the effects of the number of comments of each class on the models performances, I ordered the feedback classes by the number of comments. The performance in general is decreasing with the reduction of the number of comments. However, in many cases the models still performed well even with small number of comments. Moreover, when the number of comments is below 20, the models have an F1 score of 0 except the Doc2Vec model in class 18. Also, the models had a sudden drop in the performance where the number of comments are relatively high, particularly in class 9. This inconsistency in the performances can be explained by the number of comments in general. Also, the imbalance between the classes, especially the imbalance between the classes within the same question, has an effect on the performances of the models.

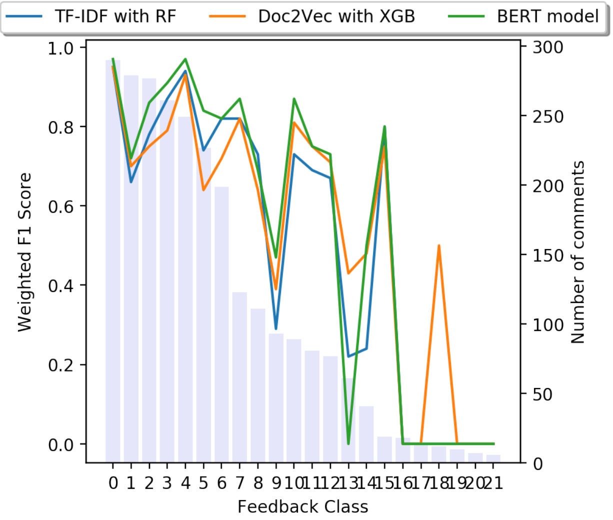


Fig. 6.5.: F1 Scores for each class by all models.

## 6.5 Summary

The research objective of this study is to build an automated reply system to students' comments. I asked students in their Master degree to give feedback to these comments. In the baseline method, I did not apply a particular transformation and the results were far from perfect. After clustering the comments and using the content of the comments to its maximum by detecting special tags such as "code" or "studyTime" we could achieve much higher results. The application of state-of-the-art language model achieved the best result with a very strong accuracy and F1 scores.

With this results, I can fairly say that it is possible to automate the process of giving feedback to students. Therefore, the professors can focus more on what they can improve in the classroom to help students who have problems.

In the next chapter, I will address the conclusion and present some future improvements of the work.

# Conclusion and Future Work

# 7

## 7.1 Results and Findings

Across the different studies in this dissertation, I could witness the diversity of the educational dataset. Also, I could certify how valuable these dataset are. Each different type of data enables a set of analysis that cannot be done elsewhere. This diversity in the educational data is heavily displayed across the research topics that I worked on. The results and findings are diversified as much as the dataset are.

In the first study, I successfully build prediction models that have a strong level of correctness in guessing which student will pursue a STEM-related career. These predictions are based on click-stream data of the students' usage of an intelligent tutoring system. The models that I build are robust and generalize well to different distributions of students.

In the process of building the prediction models I investigate different patterns. Firstly, I examine the effect of the school on the prediction models by proposing a school-based aggregation of the students data. This method provided an good improvement in AUC score but also a worsening in the RMSE score. After that, I use some of the functionalities of the ASSITSTments ITS. In fact, in ASSISTments the task are organized in problems and skills. Each problem involves one or more skills. I examine which one holds more information by proposing another type of aggregation that I call skill-based approach vs problem-based approach. Meanwhile, I also apply school-aggregation to both approaches and compare all combinations. I found that the skill-based approach with school aggregation achieved a significant improvement compared to a baseline. Also, it achieved the best results over all. With or without the school aggregation, the skill-based approach outperformed the problem-based approach. These findings, provides some evidence to what is suspected. If students acquire a skill, then they are most likely to perform well in problems that involve that skill. However, when a student complete a problem successfully it does not automatically mean they mastered all the involved skills.

In the second study, I used data collected from the students' usage of an e-book system called BookRoll. I conducted different analysis. The first discovery is finding the optimal inactivity threshold that defines a reading session. In fact, I introduce the reading session which consists of a student opening a document and interacting with it until he/she closes the document or until there is a long inactivity time. The inactivity threshold was chosen after comparing the effect of different thresholds on the number of reading sessions detected. I found that 90 minutes is a correct threshold. This finding also aligns with the fact that the class time usually lasts 90 minutes. On top of that, the majority of the students actions happen during the lecture time, therefore the reading sessions inactivity threshold corroborates with the general behavior of the students. I also detect the score limit that separates the "highly" performing students from the rest. This finding was indicated by the balance between the number of classes according to the score limit and also by the predictions accuracy that maximize the difference in the behavior between the highly performing students and the rest. In a later analysis, I explore the change in students reading behaviors during the class time and outside of the class time. The difference in behavior is significant but the prediction of the student scores as a regression problem is not significantly different.

In the third study, I explicitly gather students subjective data by providing a questionnaire composed by 5 predefined question about their learning activities. The objective is two fold: The first part is to automate the process of assessing the students learning activities and report it to the professor. The second part is to build an automatic feedback system to students comments. In process of successfully building the models, I have proven that the context of the question is important and can lead to significant improvement of the performances. To incorporate the context of the question it is not necessary to build a model for each question. In fact, I proposed a simple padding method in which we hard-code the type of the question just before the rest of the comment. This method was tested multiple times and showed evidence of improvement of the models performances. Moreover, I found that pre-trained word embedding achieve better results than training my own word embeddings on this dataset, mostly because of the limited vocabulary used, therefore the models won't generalized well to new comments especially if they use new words. Finally, the usage of state-of-the-art language models outperforms the rest of the approaches. Therefore, it will be helpful to exploit the new advances of NLP as much as possible.

## 7.2 Future Work

The results and findings that I have seen while working with these dataset provided me with inspirations for future improvements knowing what theses systems are capable of.

In the first study using ASSISTments, move focus should be given toward the skills, and allow more fine grained decomposition of the skills would be an interesting research topic.

In the second study using BookRoll data, it would interesting to have access to students memos and apply NLP methods to investigate and detect students problems or good understanding throught textual data.

Finally, for the third study we plan to build an integrated platform where the professor can have full reports on the students activities and where the students can engage in an interactive discussion without the limit of the predefined question.

# Bibliography

[1]Terry Anderson, Liam Rourke, Randy Garrison, and Walter Archer. “Assessing Teaching Presence In A Computer Conferencing Context”. In: *Online Learning* 5.2 (2019) (cit. on p. 75).

[2]E. Ayers, R. Nugent, and N. Dean. *A comparison of student skill knowledge estimates*. 2009 (cit. on p. 10).

[3]F Bachtiar, Katsuari Kamei, and Eric Cooper. “An Estimation Model of English Abilities of Students Based on Their Affective Factors in Learning by Neural Network”. In: *proceedings of IFSA and AFSS International Conference 2011*. 2011 (cit. on p. 60).

[4]Ryan Baker and Paul Inventado. “Educational Data Mining and Learning Analytics”. In: May 2014, pp. 61–75 (cit. on pp. 7, 11, 13).

[5]Ryan Baker and Kalina Yacef. “The State of Educational Data Mining in 2009: A Review and Future Visions”. In: *Journal of Educational Data Mining* 1 (Jan. 2009), pp. 3–17 (cit. on pp. 8, 9, 11).

[6]Ryan Shaun Baker, Albert T. Corbett, and Kenneth R. Koedinger. “Detecting Student Misuse of Intelligent Tutoring Systems”. In: *Intelligent Tutoring Systems*. Ed. by James C. Lester, Rosa Maria Vicari, and Fábio Paraguaçu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 531–540 (cit. on p. 26).

[7]Ryan S.J.d. Baker. “Modeling and Understanding Students’ Off-task Behavior in Intelligent Tutoring Systems”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Chi ’07. San Jose, California, USA: Acm, 2007, pp. 1059–1068 (cit. on pp. 26, 30).

[8]Robert Balfanz. “Putting Middle Grades Students on the Graduation Path A Policy and Practice Brief”. In: (Jan. 2009) (cit. on p. 26).

[9]Trevor Barker. “An Automated Individual Feedback and Marking System: An Empirical Study”. In: *9th European Conference on eLearning 2010, ECEL 2010* 9 (Apr. 2011) (cit. on p. 75).

[10]John Biggam. “Using Automated Assessment Feedback to Enhance the Quality of Student Learning in Universities: A Case Study”. In: *Technology Enhanced Learning. Quality of Teaching and Educational Reform*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 188–194 (cit. on p. 75).

[11]Gautam Biswas, Krittaya Leelawong, Daniel Schwartz, and Nancy Vye. “Learning By Teaching: A New Agent Paradigm For Educational Software.” In: *Applied Artificial Intelligence* 19 (Mar. 2005), pp. 363–392 (cit. on p. 1).

[12]Félix Castro, Alfredo Vellido, Àngela Nebot, and Francisco Mugica. “Applying Data Mining Techniques to e-Learning Problems”. In: *Evolution of Teaching and Learning Paradigms in Intelligent Environment*. Ed. by Lakhmi C. Jain, Raymond A. Tedman, and Debra K. Tedman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 183–221 (cit. on p. 8).

[13]Guanliang Chen, Vitor Rolim, Rafael Ferreira Mello, and Dragan Gašević. “Let’s Shine Together! A Comparative Study between Learning Analytics and Educational Data Mining”. In: *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. Lak ’20. Association for Computing Machinery, 2020, 544–553 (cit. on p. 8).

[14]A. T. Corbett and J. R. Anderson. “Knowledge tracing: Modeling the acquisition of procedural knowledge”. In: *User Modeling and User-Adapted Interaction* 4.4 (Dec. 1995), pp. 253–278 (cit. on pp. 13, 29, 37).

[15]Loris D’antoni, Dileep Kini, Rajeev Alur, et al. “How Can Automatic Feedback Help Students Construct Automata?” In: *ACM Trans. Comput.-Hum. Interact.* 22.2 (Mar. 2015) (cit. on p. 75).

[16]Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805 (cit. on p. 81).

[17]Beth Dietz and J.E. Hurn. “Using learning analytics to predict (and improve) student success: A faculty perspective”. In: *Journal of Interactive Online Learning* 12 (Jan. 2013), pp. 17–26 (cit. on p. 59).

[18]Donald F Whalen and Mack Shelley. “Academic success for STEM and non-STEM”. In: 11 (Jan. 2010) (cit. on p. 26).

[19]Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?” In: *Journal of Machine Learning Research* 15.90 (2014), pp. 3133–3181 (cit. on p. 82).

[20]Rafael Ferreira-Mello, Máverick André, Anderson Pinheiro, Evandro Costa, and Cristobal Romero. “Text mining in education”. In: *WIREs Data Mining and Knowledge Discovery* 9.6 (2019), e1332 (cit. on pp. 13, 15).

[21]Brendan Flanagan and Hiroaki Ogata. “Integration of Learning Analytics Research and Production Systems While Protecting Privacy”. In: Dec. 2017 (cit. on p. 43).

[22]Brendan Flanagan and Hiroaki Ogata. “Learning Analytics Infrastructure for Seamless Learning”. In: Mar. 2018 (cit. on p. 43).

[23]María Teresa Flórez and Pamela Sammons. *Assessment for Learning: Effects and Impact.* Eric, 2013 (cit. on p. 60).

[24]Linton Freeman. “The Development of Social Network Analysis–with an Emphasis on Recent Events”. In: *The SAGE Handbook of Social Network Analysis* (Jan. 2011) (cit. on p. 10).

[25] Enrique Frias-Martinez, Sherry Chen, and Xiaohui Liu. “Survey of Data Mining Approaches to User Modeling for Adaptive Hypermedia”. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 36 (Dec. 2006), pp. 734–749 (cit. on p. 10).

[26] Kazumasa Goda and Tsunenori Mine. “PCN: Quantifying Learning Activity for Assessment based on Time-series Comments.” In: *Proceedings of the 3rd International Conference on Computer Supported Education - Volume 2: ATTeL, (CSEDU 2011)*. Insticc. SciTePress, 2011, pp. 419–424 (cit. on pp. 59, 60).

[27] Neil Heffernan and Cristina Heffernan. “The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching”. In: *International Journal of Artificial Intelligence in Education* 24 (Dec. 2014) (cit. on p. 27).

[28] Anne Hume and Richard Coll. “Assessment of learning, for learning, and as learning: New Zealand case studies”. In: *Assessment in Education: Principles, Policy and Practice* 16 (Nov. 2009) (cit. on p. 59).

[29] Yuheng Jiang, Sohail Javaad Syed, and Lukasz Golab. “Data Mining of Undergraduate Course Evaluations”. In: *Informatics In Education* 15 (May 2016), pp. 85–102 (cit. on p. 60).

[30] Kenneth R. Koedinger, Sidney D’Mello, Elizabeth A. McLaughlin, Zachary A. Pardos, and Carolyn P. Rosé. “Data mining and education”. In: *WIREs Cognitive Science* 6.4 (2015), pp. 333–353 (cit. on p. 13).

[31] Quoc V. Le and Tomas Mikolov. “Distributed Representations of Sentences and Documents”. In: *CoRR* abs/1405.4053 (2014). arXiv: `1405.4053` (cit. on p. 64).

[32] Trang T Le, Weixuan Fu, and Jason H Moore. “Scaling tree-based automated machine learning to biomedical big data with a feature set selector”. In: *Bioinformatics* 36.1 (June 2019), pp. 250–256. eprint: `https://academic.oup.com/bioinformatics/article-pdf/36/1/250/31813758/btz470.pdf` (cit. on pp. 20, 21).

[33] Leah Macfadyen and Shane Dawson. “Mining LMS data to develop an “early warning system” for educators: A proof of concept”. In: *Computers and Education* 54 (Feb. 2010), pp. 588–599 (cit. on p. 59).

[34] Joel Martin and Kurt VanLehn. “Student assessment using Bayesian nets”. English (US). In: *International Journal of Human Computer Studies* 42.6 (June 1995), pp. 575–591 (cit. on p. 29).

[35] Riccardo Mazza. *Introduction to Information Visualization*. 1st ed. Springer Publishing Company, Incorporated, 2009 (cit. on p. 9).

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: `1301.3781 [cs.CL]` (cit. on p. 64).

[37] Toshiro Minami and Yoko Ohura. “A correlation analysis of student’s attitude and outcome of lectures investigation of keywords in class-evaluation questionnaire”. In: *Advanced Science and Technology Letters* 73 (Dec. 2014), pp. 11–16 (cit. on p. 59).

[38]Toshiro Minami and Yoko Ohura. “How Student’s Attitude Influences on Learning Achievement? An Analysis of Attitude-Representing Words Appearing in Looking-Back Evaluation Texts”. In: *International Journal of Database Theory and Application* 8 (Apr. 2015), pp. 129–144 (cit. on p. 59).

[39]Toshiro Minami and Yoko Ohura. “Investigation of Students’ Attitudes to Lectures with Text-Analysis of Questionnaires”. In: *Proceedings - 2nd IIAI International Conference on Advanced Applied Informatics, IIAI-AAI 2013*. Aug. 2013, pp. 56–61 (cit. on p. 60).

[40]Toshiya Nakajima, Shun Shinohara, and Yasuhisa Tamura. “Typical Functions of e-Textbook, Implementation, and Compatibility Verification with Use of ePub3 Materials”. In: *Procedia Computer Science* 22 (Dec. 2013), pp. 1344–1353 (cit. on p. 43).

[41]Ryan Noonan. *STEM Jobs: 2017 Update*. Office of the Chief Economist, Economics and Statistics Administration, U.S. Department of Commerce(ESA Issue Brief 02-17). Mar. 2017 (cit. on p. 25).

[42]Oecd. *Innovating Education and Educating for Innovation*. 2016, p. 152 (cit. on p. 1).

[43]Hiroaki Ogata, Chengjiu Yin, Misato Oi, et al. “e-Book-based Learning Analytics in University Education”. In: Dec. 2015 (cit. on p. 43).

[44]Hiroaki Ogata, Misato Oi, Kousuke Mouri, et al. “Learning Analytics for E-Book-Based Educational Big Data in Higher Education”. In: May 2017, pp. 327–350 (cit. on p. 43).

[45]Hiroaki Ogata, Mengmeng Li, Bin Hou, et al. “SCROLL: supporting to share and reuse ubiquitous learning log in the context of language learning”. In: *Research and Practice in Technology Enhanced Learning* 6 (Jan. 2011) (cit. on p. 43).

[46]F.R. Olenchak and T.P. Hébert. “Endangered academic talent: Lessons learned from gifted first-generation college males”. In: 43 (Mar. 2002), pp. 195–212 (cit. on pp. 25–27).

[47]Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. “Evaluation of a Tree-Based Pipeline Optimization Tool for Automating Data Science”. In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. Gecco ’16. Denver, Colorado, USA: Association for Computing Machinery, 2016, 485–492 (cit. on p. 21).

[48]Zachary A. Pardos, Ryan S. J. D. Baker, Maria O. C. Z. San Pedro, Sujith M. Gowda, and Supreeth M. Gowda. “Affective States and State Tests: Investigating How Affect Throughout the School Year Predicts End of Year Learning Outcomes”. In: *Proceedings of the Third International Conference on Learning Analytics and Knowledge*. Lak ’13. Leuven, Belgium: Acm, 2013, pp. 117–124 (cit. on pp. 26, 29).

[49]Ernest T. Pascarella, Christopher T. Pierson, Gregory C. Wolniak, and Patrick T. Terenzini. “First-generation college students: Additional evidence on college experiences and outcomes”. English (US). In: *Journal of Higher Education* 75.3 (May 2004), pp. 249–284 (cit. on pp. 25, 26).

[50]T. Patikorn, R. S. Baker, and N. T. Heffernan. "“ASSISTments Longitudinal Data Mining Competition Special Issue: A Preface"”. In: *Journal of Educational Data Mining* 12 (2020) (cit. on pp. 28, 41).

[51] Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. “Performance Factors Analysis –A New Alternative to Knowledge Tracing”. In: *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2009, pp. 531–538 (cit. on p. 29).

[52] Maria Ofelia San Pedro, Jaclyn Ocumpaugh, Ryan Shaun Joazeiro de Baker, and Neil T. Heffernan. “Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software”. In: *Edm*. 2014 (cit. on pp. 25, 26, 30).

[53] Riccardo Poli, William Langdon, and Nicholas Mcphee. *A Field Guide to Genetic Programming*. Jan. 2008 (cit. on p. 17).

[54] Jim Reye. “Student Modelling Based on Belief Networks”. In: *Int. J. Artif. Intell. Ed.* 14.1 (Jan. 2004), pp. 63–96 (cit. on p. 29).

[55] C. Romero and S. Ventura. “Educational Data Mining: A Survey from 1995 to 2005”. In: *Expert Syst. Appl.* 33.1 (July 2007), 135–146 (cit. on p. 11).

[56] Cristóbal Romero and Sebastian Ventura. “Data Mining in Education”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3 (Jan. 2013) (cit. on pp. 7, 8).

[57] Cristóbal Romero and Sebastian Ventura. “Educational Data Mining: A Review of the State of the Art”. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 40 (Dec. 2010), pp. 601–618 (cit. on pp. 7, 9–11).

[58] Jennifer Sabourin, Bradford Mott, and James C. Lester. “Modeling Learner Affect with Theoretically Grounded Dynamic Bayesian Networks”. In: *Affective Computing and Intelligent Interaction*. Ed. by Sidney D’Mello, Arthur Graesser, Björn Schuller, and Jean-Claude Martin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 286–295 (cit. on p. 26).

[59] Maria Ofelia Clarissa Z. San Pedro, Ryan S. J. d. Baker, and Ma. Mercedes T. Rodrigo. “Detecting Carelessness through Contextual Estimation of Slip Probabilities among Students Using an Intelligent Tutor for Mathematics”. In: *Artificial Intelligence in Education*. Ed. by Gautam Biswas, Susan Bull, Judy Kay, and Antonija Mitrovic. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 304–311 (cit. on p. 26).

[60] Sweet San Pedro, Ryan Baker, A.J. Bowers, and Neil Heffernan. “Predicting college enrollment from student interaction with an intelligent tutoring system in middle school”. In: *Edm*. Jan. 2013, pp. 177–184 (cit. on pp. 26, 30).

[61] Valerie J. Shute. “Focus on Formative Feedback”. In: *Review of Educational Research* 78.1 (2008), pp. 153–189 (cit. on p. 75).

[62] George Siemens and Ryan Baker. “Learning analytics and educational data mining: Towards communication and collaboration”. In: *ACM International Conference Proceeding Series* (Apr. 2012) (cit. on p. 7).

[63] George Siemens and Phil Long. “Penetrating the Fog: Analytics in Learning and Education”. In: *EDUCAUSE Review* 5 (Jan. 2011), pp. 30–32 (cit. on p. 59).

[64] T. Sliusarenko, Line Clemmensen, and Bjarne Ersbøll. “Text mining in students’ course evaluations: Relationships between open-ended comments and quantitative scores”. In: *CSEDU 2013 - Proceedings of the 5th International Conference on Computer Supported Education* (Jan. 2013), pp. 564–573 (cit. on p. 60).

[65] Shaymaa Sorour, Kazumasa Goda, and Tsunenori Mine. “Comment Data Mining to Estimate Student Performance Considering Consecutive Lessons”. In: *Educational Technology Society* 20 (Jan. 2017), 73–86 (cit. on p. 61).

[66] Shaymaa Sorour, Kazumasa Goda, and Tsunenori Mine. “Evaluation of Effectiveness of Time-Series Comments by Using Machine Learning Techniques”. In: *Journal of Information Processing* 23 (Nov. 2015), pp. 784–794 (cit. on p. 61).

[67] Shaymaa Sorour, Tsunenori Mine, Kazumasa Goda, and Sachio Hirokawa. “Predicting students’ grades based on free style comments data by artificial neural network”. In: *Proceedings - Frontiers in Education Conference, FIE* 2015 (Feb. 2015) (cit. on p. 61).

[68] Shaymaa Sorour, Tsunenori Mine, Kazumasa Goda, and Sachio Hirokawa. “Prediction of Students’ Grades Based on Free-Style Comments Data”. In: *The 13th International Conference on Web-based Learning*. Vol. Lncs 8613. Aug. 2014, pp. 142–151 (cit. on p. 61).

[69] Shaymaa Sorour, Kazumasa Goda, and Tsunenori Mine. “Student Performance Estimation Based on Topic Models Considering a Range of Lessons”. In: *Aied2015*. June 2015 (cit. on p. 61).

[70] K. Sparck Jones. “A statistical interpretation of term specificity and its applicationin retrieval”. In: *Journal of Documentation* 28 (1972), 11–21 (cit. on p. 64).

[71] UCIO, E. R., EHBEIN, HRISTINA, and Reston. “Developing Educational Software: A Professional Tool Perspective”. In: 2010 (cit. on p. 1).

[72] Xueli Wang. *Modeling Student Choice of STEM Fields of Study: Testing a Conceptual Framework of Motivation, High School Learning, and Postsecondary Context of Support*. WISCAPE Working Paper. Wisconsin Center for the Advancement of Postsecondary Education. 2012 (cit. on p. 25).

[73] Xueli Wang. “Why Students Choose STEM Majors: Motivation, High School Learning, and Postsecondary Context of Support”. In: *American Educational Research Journal* 50.5 (2013), pp. 1081–1121 (cit. on pp. 25, 26).

[74] Donald F. Whalen and Mack C. Shelley. “"Academic Success for STEM and Non-STEM Majors"”. In: *Journal of STEM Education* 11 (2010) (cit. on pp. 25, 26).

[75] Varaporn Yamtim and Suwimon Wongwanich. “A Study of Classroom Assessment Literacy of Primary School Teachers”. In: *Procedia - Social and Behavioral Sciences* 116 (Feb. 2014), pp. 2998–3004 (cit. on p. 60).

[76] Mengxiao Zhu, Ou Lydia Liu, and Hee-Sun Lee. “The effect of automated feedback on revision behavior and learning gains in formative assessment of scientific argument writing”. In: *Computers & Education* 143 (2020) (cit. on p. 75).