Tallinn University of Technology
School of Information Technologies
Department of Software Science

Eerik Sven Puudist 193467IAIB

Implementing Data Warehouse and Machine Learning Models for Student Segmentation and Academic Performance Prediction

Bachelor Thesis

Technical Supervisor
Ago Luberg
PhD
Academic Supervisor
Innar Liiv
PhD

Tallinn 2022

Tallinna Tehnikaülikool
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Eerik Sven Puudist 193467IAIB

Andmeaida ja masinõppe mudelite loomine üliõpilaste rühmitamiseks ja akadeemilise suutlikkuse ennustamiseks

bakalaureusetöö

Juhendaja
Ago Luberg
PhD
Kaasjuhendaja
Innar Liiv
PhD

Tallinn 2022

Acknowledgements

Working on such an interdisciplinary project has been a fascinating and thought-provoking experience. This endeavor would not have been possible without by supervisor PhD Ago Luberg who greatly supported me on every step of the project and introduced me to the research already conducted in that field. I am also very grateful to my second supervisor PhD Innar Liiv, especially for providing his knowledge on network analysis and for helping me to see data science in a wider context. This research project would not have been fruitful without the kind assistance of Kati Aus, a specialist in educational psychology and didactics who helped me to ask the right questions and interpret the findings from the psychological perspective.

I want to express my sincere gratitude to my Dharma teacher Ven. Bodhi Lama Erik Drew Jung and to all of my Sangha members for opening my mind to numerous new possibilities. I also want to thank my friend Mart Sõmermaa who introduced me to the IT world. Last but not least, my sincere gratitude goes to my beloved parents for their invaluable care and support.

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, the literature and the work of others have been referenced. This thesis has not been presented for examination anywhere else.

Author: Eerik Sven Puudist
2022-05-30

Abstract

The aim of this project is to make a step towards implementing a system which would provide the teacher with the necessary tools for better understanding and supporting their students in university IT courses.

The project consists of four major parts. Firstly, the requirements of the system and the available data sources are mapped out.

Secondly, a data warehouse is implemented to bring together data from various information systems and to provide a unified view on for subsequent analyses.

Thirdly, models are implemented using classification and regression method to predict students' academic performance.

Fourthly, distinct student segments are identified through clustering.

The thesis is written in English and contains 51 pages of text, 11 chapters, 19 figures, 7 tables.

Annotatsioon

Andmeaida ja masinõppe mudelite loomine
üliõpilaste rühmitamiseks
ja akadeemilise suutlikkuse ennustamiseks

Käesoleva projekti eesmärk on astuda samm süsteemi loomise suunas, mis pakuks õppejõule vajalikke töövahendeid ülikooli IT kursuste üliõpilaste mõistmiseks ja toetamiseks.

Töö koosneb neljast suuremast osast. Esiteks kaardistatakse ära nõuded loodavale süsteemile. Uuritakse olemasolevaid andmeallikaid ning töötatakse välja võimalused puuduvate andmete kogumiseks.

Teiseks luuakse andmeait, kuhu koondatakse kokku andmed erinevatest kursusega seotud infosüsteemidest eesmärgiga pakkuda terviklikku vaadet järgnevaks andmeanalüüsiks.

Kolmandaks luuakse mudelid üliõpilaste akadeemilise edukuse ennustamiseks kasutades klassifitseerimise ja regressioonanalüüsi meetodeid.

Neljandaks tuuakse klasterdamise abil välja eriilmelised tudengite grupid.

Lõputöö põhjal on valminud ingliskeelne teadusartikkel, mis käsitled tudengite rühmitamist väljalangemise aja ning õpitulemuste alusel ning uurib erinevate segmentide vahelisi psühholoogilisi ja käitumuslikke erisusi. Artikkel on tööle lisatud eraldi failina kui Lisa 2.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 51 leheküljel, 11 peatükki, 19 joonist, 7 tabelit.

List of abbreviations and terms

Moodle	a learning platform used in TalTech, moodle.org
Python	a general-purpose programming language used widely for data analysis
SSE	*sum of squared errors*, a metric for measuring clustering algorithm performance
SQL	*Structured Query Language*, a language used to interact with databases

Table of Contents

List of Figures viii

List of Tables ix

1 Introduction 1

2 Requirements for the system 4
- 2.1 Predicting the student's academic performance . . . 4
- 2.2 Understanding the reasons behind the student's performance . . . 4
- 2.3 Providing feedback for further developing the curriculum . . . 5
- 2.4 Providing information on how to promote the curriculum . . . 5
- 2.5 Providing feedback to the students . . . 5

3 Overview of learning analytics 7
- 3.1 The purpose and process of study analytics . . . 7
- 3.2 Psychological factors influencing academic performance . . . 7

4 Overview of data warehouse technologies 11
- 4.1 Necessity and goals of data warehouses . . . 11
- 4.2 Differences between operational systems and data warehouses . . . 11
- 4.3 General overview of data warehouse systems . . . 13
- 4.4 Database systems used for data warehouses . . . 16
- 4.5 Normalization and star schemas . . . 16

5 Overview of data analysis methods 18
- 5.1 Classification . . . 18
- 5.2 Regression . . . 20
- 5.3 Clustering . . . 21

6 Data sources 23
- 6.1 Existing data sources . . . 23
- 6.2 Novel data sources . . . 25

7 Study analysis data warehouse implementation 26
- 7.1 General notes on the implementation . . . 26
- 7.2 Why PostgreSQL? . . . 26
- 7.3 The layers in the database . . . 27

7.4 The data model 28

8 Exploratory data analysis on 2021 course data **32**
8.1 Overview of the 2021 students 32
8.2 Overview of study results 32
8.3 Overview of the dropout 33
8.4 Group chat messages 33
8.5 Grand survey 34
8.6 Homework submissions 36

9 Academic performance prediction **38**
9.1 Predicting on the first week if the student passes the course 38
9.2 Predicting on the first week the student's final score 40
9.3 Predicting if the student passes the course throughout the weeks 40
9.4 Predicting the student's final score throuought the weeks 42

10 Student segmentation **44**
10.1 Student segementation based on data available on the first week 44
10.2 Student segementation based on data available on the 14th week 46

11 Summary **50**

Bibliography **51**

Appendices

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis

Appendix 2 – Article: Psychological and Behavioural Differences Between IT Student Segments

List of Figures

1 Typical data access patterns in data warehouses vs operational systems . . 12
2 Data warehouse architecture 13

3 Students subsystem 28
4 Studies subsystem 28
5 Submissions subsystem 29
6 Chats subsystem 30
7 Plagiarism subsystem 30
8 Grand survey subsystem 31

9 Study results 33
10 Dropout times 34
11 Message count correlation with course outcomes 35
12 Submission count per study week 36
13 Submission count correlation with course outcomes 37

14 Accuracy of course passing prediction throughout the weeks 42
15 Accuracy of score prediction throughout the weeks 43

16 Student clustering on first week – choosing the best cluster count 45
17 Student clustering on first week – cluster centroids 46
18 Student clustering on 14th week – choosing the best cluster count 47
19 Student clustering on 14th week – cluster centroids 48

List of Tables

1 The confusion matrix 19

2 Predicting course passing on first week – 10 most important feature 39
3 Predicting course passing on first week – confusion matrix with threshold 0.5 39
4 Predicting course passing on first week – confusion matrix with threshold 0.7 40
5 Predicting score on first week – 10 most important feature 41
6 Predicting course passing – 10 most important features on different weeks 42
7 Predicting final score – 10 most important features on different weeks . . 43

1. Introduction

The high university dropout rate, particularly in the field of information technology, is a pressing problem in Estonia. Up to 32% of students in this field quit their studies already at their first year [1, p 3]. Designing a system for profiling students and tracking their academic performance with the aim to reduce dropout rates has therefore been on the table here in TalTech for many years and some steps towards this have already been taken.

In 2017 Brenda Uga defended her bachelor thesis where she implemented a system which predicted dropout among students of the Informatics curriculum based on their course outcomes and general personal data. [2]

It was found that the probability of the student graduating successfully depends greatly on their results in a few key courses including "Introduction to Programming" and "Object oriented programming in Java" [2, p 47]. Since many further courses assume that the student has acquired the skills presented in the two aforementioned courses, it is important to help students to finish those courses with good results.

It is therefore essential that the students in the risk group could be identified in advance so that they could receive special mentoring and support throughout the course. That mentoring could help the student to obtain the necessary understandings and study strategies to be successful on the course. If the complexity level of the course exceeds the student's capacity or if the student recognizes that this subject is too far from their real field of interest, the mentoring program could help them to find a curriculum which aligns better with their interests and abilities so that failure in the given curriculum would not cause the student to give up their academic pursuits.

That would require a system which does not only consider the final course marks, but is able to dig into the inner logic of the course to provide information and predictions during the course before the final mark is known.

There have been attempts to create general-purpose tools which would be applicable to a large variety of courses, e.g. all courses using Moodle. Such systems try to find

generic patterns and their correlation with the course outcomes. However, even if the courses use the same digital platform, e.g. Moodle, their inner logic is so different that predictions made based on such generic patterns are not very accurate. [1, p 28].

Moreover, besides common data sources such as Moodle logs, data from many course-specific sources should also be incorporated into the analysis.

This thesis focuses on designing a system (hereinafter referred to as "the system") specifically for programming related courses at TalTech using "ITI0102 Introduction to Programming" (hereinafter referred to as "the course") as an example.

The process of designing, implementing, testing, and deploying a system of that size is definitely far beyond the scope of a bachelor thesis. Nevertheless, the author aims to take a big step towards that goal. In particular the author sets out to:

- Map out the requirements of the system.
- Map out the existing data sources.
- Find out which data is not yet collected but would be necessary for fulfilling the requirements of the system.
- Implement methods for collecting that missing data.
- Design and implement a data warehouse for bringing all the data together.
- Start the research on using that data for academic performance prediction.
- Start the research on using that data for student clustering.

The author sincerely hopes that the system will continue to be enhanced in the future and that the methods introduced here will also be adopted by other courses and other universities.

Chapter 2 gives an overview of the requirements for the system. The available data sources and the novel data sources designed in this project are explained in chapter 6. Chapters 3, 4, and 5 provide an overview of learning analytics, data warehouses, and relevant data analysis methods, respectively. Building on that foundation, chapter 7 explains the design choices and the data model for the data warehouse implemented for the project. Chapter 8 introduces the 2021 dataset. Chapters 9 and 10 will then focus on the patterns in the dataset which would be useful for performance prediction and student segmentation. A conference paper with additional findings regarding student profiling is included as appendix 2 as a separate file.

The scientific Python stack, most notably pandas [3], was used for the analysis. Scikit-learn [4] was used for implementing predictive models and clustering. The plots were created with Matplotlib [5]. Gephi [6] was used to analyse the plagiarism network.

2. Requirements for the system

The primary user of the system is the teacher who mainly seeks answers to the following two questions:

- How is the student progressing on the course?
- Why so?

The system could also be used for making decisions for developing and promoting the Informatics curriculum at TalTech as well as for providing real-time feedback to the students together with personalized suggestions.

2.1 Predicting the student's academic performance

To provide special mentoring to the people in the risk group, students with potentially lower results need to be identified in advance. Depending on the quality and quantity of the available data, the prediction could range from a more generic "John is likely to not complete the course" to more fine-grained "John is likely to drop out in two weeks with the final score of 374 points".

2.2 Understanding the reasons behind the student's performance

The causes which can result in low academic performance can be seen as a spectrum with the two endpoints:

- internal or mental factors such as psychological beliefs, unproductive study strategies, gaps in previous education, language barriers;
- external factors such as unexpected health issues, problems within family or at work.

Since different causes require different types of support from the teacher, it is of supreme importance to understand the situation of the specific student. For example, if a student has not been active for a while because of a health issue, extending the deadline of the exercise could help them to cache up. The same remedy could

however backfire for a student who has constant habitual problems with procrastination.

2.3 Providing feedback for further developing the curriculum

The collected data can be used to find out how well on average the individual exercises were solved and if the time spend coincides with the EAP count of the course. For example, if solving the exercise on regular expressions required unproportional efforts from the students, further explanations on that topic or a more clear wording for the exercise might be needed.

2.4 Providing information on how to promote the curriculum

It is in the interest of the university to attract people to study informatics. However, it would be best to attract people who are actually able to cope with the complexity of the curriculum and direct other students to other curriculums. If it turns out, for instance, that students with math exam results below a certain threshold are very unlikely to graduate successfully, that could be made known to the potential students up front to give them a better understanding what the curriculum expects from them.

2.5 Providing feedback to the students

The collected data can be used to give the student a better understaning about themselves and their studies.

Firstly, the results of the psychological factors survey could help the student to understand their mental states, beliefs, and habits and their impact on the student's academic abilities. The student should be presented with the numeric metrics calculated from their survey answers accompanied by descriptions for interpreting the results and, probably most importantly, personalized suggestions and feedback.

Secondly, it would be helpful for the student to understand their performance on the course in a wider context. The student might know their current score, but it is hard for them to interpret in accurately. If the student could see that on previous years students with such a score on that week were unlikely to pass, they could take preventive measures before it is too late.

It is not only important to help the weaker segment of the students to pass, but also encourage the stronger segment to not be content with mediocre results. The system could help to identify that stronger segment and encourage them to work on the optional harder exercises to get more value out of the course.

3. Overview of learning analytics

3.1 The purpose and process of study analytics

The purpose of study analytics is to understand the process of teaching and learning in order to help the students to study more effectively. [1, p 21]

The cycle of study analytics can be divided into four phases:

- the process of teaching and studying
- gathering data about the efficiency of the process
- analysing the data
- intervention and improving the process

[1, p 21]

The cycle starts with the regular process of teaching and studying which nowadays – and especially during a lockdown situation – has a big digital component.

This generates a lot of data ranging from digitally submitted homeworks and digitally exchanged messages to activity logs in the various digital platforms used throughout the university [1, p 22]. This data has to gathered, organized and cleaned before it can be used for analysis.

The third stage in the progress is selecting the metrics, observing the patterns, designing the models, and visualizing the results. [1, p 22]

The insights born in that process can then be used to improve the study process. This can include coming back to the specific students whose data was used in the analysis and/or using the generalized results for improving the curriculum for the future students. [1, p 22]

3.2 Psychological factors influencing academic performance

3.2.1 Learning competence

Learning competence consists of a cognitive and a metacognitive aspect. The cognitive aspect includes skills for finding the relevant information, memorizing what has been learned, and using it in a variety of contexts to solve various problems. The metacognitive side includes skills for planning and organizing the study process and analysing one's weak and strong sides. [1, p 31]

3.2.2 Emotions felt in the study process

The study process triggers in the student a large variety of both emotions, both negative (anxiety, anger, shame, hopelessness, boredom) and positive (joy from learning, hope, glory, relief). These emotions have a strong impact on the student's abilities and academic performance. [1, p 31]

3.2.3 Beliefs regaring one's abilityes

PhD Carol Dweck has coined the theory about "fixed mindset" and "growth mindset". Fixed mindset means a set of beliefs suggesting that one's abilities are determined by congenital or otherwise unchangeable factors over which one has very little control. [7]

Growth mindset regards the one's abilities as something much more dynamic and emphasises one's ability to develop and transform through one's own willful efforts. [7]

Mindset has a strong influence on academic performance. Students with growth mindset are found to perform significantly better and also cope much better with failures. [7]

The teacher and the study environment have a significant impact on the student's mindset. Praising a student for their present abilities reinforces fixed mindset whereas praising a student for their ability to improve through making a willful effort reinforces growth mindset. [7]

One integral part of the growth-oriented mindset in self-efficacy: trust in one's abilities to accomplish a task [1, p 33]. That trust is built through recollecting and analysing past success, especially how one was able to acquire new skills that they previously did not possess but which they managed to adopt through a continuous

effort. Remembering that helps to build the certainty that one is able to repeat such process of improvement also for solving the present task as well as tasks undertaken in the future. [8]

It is also important for the student to acknowledge that learning is not easy and that the fact that they need to make an effort does not indicate lack of abilities. Making an effort is an integral part of the study process [1, p 33]. However, embracing that understanding requires self-efficacy and confidence, otherwise the student will fall back to fixed mindset maintaining that the task is just too difficult for them [8].

3.2.4 Study strategies

Students with higher expectations for academic success are likely to perform better and be more satisfied with their results. [1, p 33]

Task avoidance, on the other hand, correlates with worse performance and leads to lesser satisfaction with the results. This can form a self-perpetuating cycle since lower results and lesser satisfaction in turn foster task avoidance and anxiety. [1, p 34]

Efficient study strategies include associating new knowledge and understanding with previously known concepts, organizing new information into apprehendable categories and distributing the learning process over a longer timestamp. Efficient study strategies also require monitoring one's progress and adjusting the study strategies accordingly. [1, p 34]

However, it is not easy for the student to properly evaluate their study strategies because this topic is not emphasised enough in the public education system and it is hard to find the right metrics for measuring one's efficiency. For example, studying very intensely for a short while will help to temporarily remember a big corpus of facts and thus seems to be an effective strategy. However, it does not foster establishing long-lasting connections between different subjects and thus does not lead to permanent results. [1, p 34]

Students seem to prefer strategies that yield a quick result during the study process rather than considering the long-term outcomes. This includes, for example, looking answers up instead of trying to logically deduct them as well as avoiding novel types of exercises in the fear of making mistakes. [1, p 35]

The author would compare that tendency with “overfitting”, a common problem in machine-learning where the model performs well on known data but does not develop the ability to generalize and thus performs badly on novel data.

3.2.5 Motivation

Motivation influences one’s attention, goals, and study strategies and can in combination with other factors be used to predict performance. [1, p 35-36]

The teacher can foster the student’s motivation by creating opportunities for the students for asking curiosity driven questions. In the early phase of interest formation, positive emotions regarding the subject and a sound understanding of the field are very important. It is thus important that the teacher would be encouraging and also competent in the field. The teacher should provide optional tasks and support the student throughout the task solving process so that the student could experience a feeling of competence and success. Over time such supportive conditions support the formation of intrinsic interest in the student. [1, p 36]

The second phase of interest formation consists of focusing on the details and developing skills. The third phase includes helping the student to find their specific speciality and helping them to integrate that into their daily life. [1, p 36]

4. Overview of data warehouse technologies

4.1 Necessity and goals of data warehouses

Information can be one of the most valuable assets of the organization. However, for it to be of any value, it has to be presented in an easily usable format.

The idea, that computer systems could assist with decision making was born in the academic circles in the 1940s and gained popularity in the 1960s. Back then the analysis were performed using the databases that already existed in the organization. [9, ch 14.2]

However, the number and complexity of the organization's IT systems have grown significantly since since then. One of the main problems with usability is that the information is scattered around a variety of different systems, some of them developed in-house, some brought in from elsewhere. These systems have been designed to assist employees with their every-day duties, not for seeing the big picture and making strategics long-term decisions.

From the 1990s onwards it is generally believed that data analyses and decision support deserve systems built from the ground up with those requirements in mind. [9, ch 14.2]

4.2 Differences between operational systems and data warehouses

Systems designed to assist with every-day work (hereinafter referred to as "operational systems") usually share the following qualities:

- They are designed to process one record (e.g. create one new invoice) at a time
- They care about reflecting the *status quo*, not about preserving historical data
- They handle data only about a specific field that they are designed for

[10, ch 1]

Data warehouses – systems developed to support making long-term strategic decisions – approach data from a different angle:

- They are designed to work with a large number of records at a time
- They keep track of historical data
- They assemble data form a variety of sources together to form one unified big picture

[10, ch 1]

These differences in typical data access patterns are illustrated on figure 1: in the data warehouse on the left a small numer of columns from a large number of rows in queried, oftentimes to form an aggregate (e.g. to get the total sum of all sales in 2021) whereas in the operational system on the right lots of columns from a few rows are accesses (e.g. to retrieve all available data about a single order).

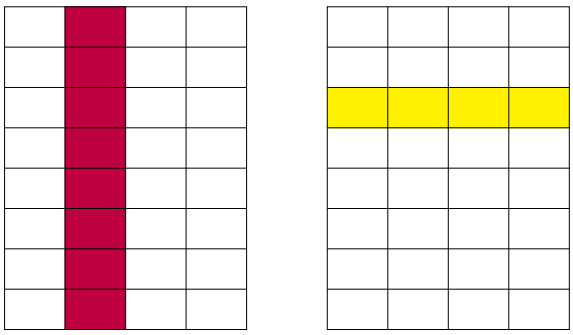

Figure 1. Typical data access patterns in data warehouses vs operational systems

This means that different data storage techniques should be used to assure optimal performance for both scenarios since the speed of the query is dependent on the amount of data blocks that has to be read from the storage device. For the operational systems, row based storage is used: all columns of a single row are physically located next to each other. For data warehouses, column based storage would be more efficient: values of the same column across different rows are stored physically together. Besides lesser disk usage that also allows more efficient data packing since values which are similar to each other of the same datatype can be packed more efficiently. [9, ch 14.4]

There are also requirements present in the operational systems which do not apply to data warehouses. Operational systems must deal with many concurrent write operations (e.g. many users sending messages at the same time). In data warehouses data is usually loaded in by a smaller set of systems. Also, unlike in operational

systems, data warehouses rarely need to modify or delete the data. Since they need to preserve the history of changes, alterations in data are usually just mean inserting a new version of the record not modifying the original one. Deleting data is usually needed only when required by the law (e.g. the right to be forgotten in the GDPR) or if the amount of data grows so big that the organization cannot afford to store it all. [9, ch 14]

Therefore, given the different requirements, it is oftentimes useful to develop the operational systems and data warehouses separately and not put both responsibilities on the same system.

4.3 General overview of data warehouse systems

There are multiple patters for building a data warehouse systems and these approaches can easily be mixed together to form a hybrid solution.

4.3.1 Physical data warehouse

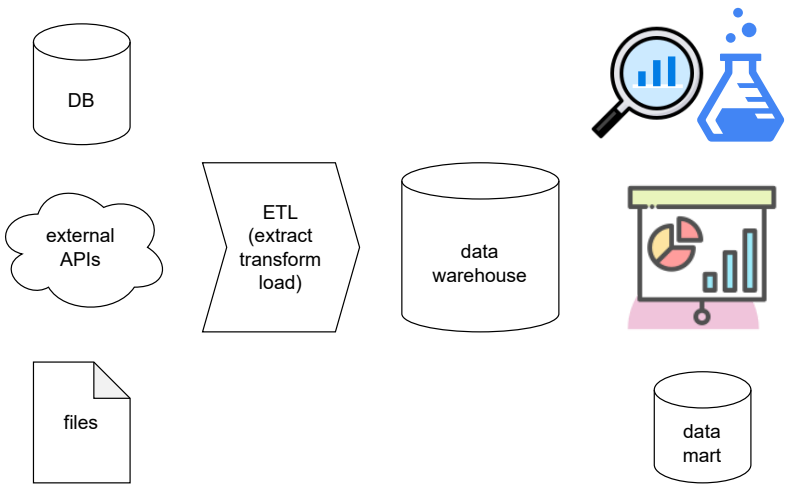

Figure 2. Data warehouse architecture

Figure 2 gives an overview of the general architecture of a data warehouse infosystem.

The data comes from multiple sources, including operational databases, data from external APIs (e.g. weather information, stock prices), data from files (e.g. contracts, meeting notes), sensors and logs (e.g. webpage access logs).

The data is then processed in the ETL (extract, transform, load) phase. Data in the various source systems might use different classifiers (e.g. 'M' and 'F' or '1' and '0' to mark the genders) or different granularity (e.g. location given as country name

or city name or specific address). Sometimes it is not clear if two sources talk about the same object or not (e.g. if "Mart Uustalu" and "Mark Erik Uustalu" and "M. E. Uustalu" are the same person or not). The same term might have a different semantics in different source systems (e.g. a "student" might refer to only university students in one system, but include previous school levels in another system.) This in turn can lead to data from different systems contradicting one another. All such issues must be tackled in the ETL phase. If some part of the data seems to contain too many problems, it might also be rejected and not loaded into the warehouse: that means finding a balance between the quality and quantity of data. [9, ch 4]

Usually data is loaded into the warehouse in big batches during nighttime when the systems' workload is smaller. However, there are also cases when the data must be loaded in real time as soon it is produced. [9, ch 4]

Besides the data itself, the warehouse also keeps track of the metadata (data about the data), e.g. from which source does this data come from, who is the author, when was it loaded, how reliable it is. [9, ch 4]

The data is then made accessible to business users via business analytics tools to assist with making tactical and strategic decisions and to data scientists for finding further use and value for the data. [9, ch 4]

The data from the warehouse could also be loaded into other smaller systems called data marts where it is structured based on the needs of a more specific group of users. [9, ch 4]

4.3.2 Logical data warehouse

In case of a logical data warehouse the data in not physically moved into the warehouse. The end users is still able to use the data as if it was stored in the warehouse, but under the hood the warehouse only keeps track of the metadata and connects to the other data sources when the data is needed. The warehouse thus functions like a gateway that integrates the different systems into one logical whole hiding the technical complexity from the end user. [9, ch 4]

4.3.3 Independent data marts

In this approach each department builds their own little system which is only concerned with the analytical needs of that department. Such small data warehouse is known as a data mart. Since multiple departments might need data from the same source systems, different data marts duplicate data, but metrics calculated using different data marts might contradict each other since different logic is used for loading the data into the marts and performing the calculations. [10, ch 1]

This is what tends to naturally happen if the process is not coordinated across the organization. Because of its smaller scope it provides some benefits in a short run, such a lower costs and faster development. However, it leads to a situation which perpetuate incomplete views to the organization's data and results in greater maintenance costs in the long run.[10, ch 1]

4.3.4 Data lake

Both the physical and logical data warehouse models require defining a unified schema which could accommodate all the data from the different sources. If the amount of data sources does not grow beyond a few dozens, that can be doable, but becomes very difficult once the number of data sources grows beyond that limit. [9, ch 4]

It is also common, that the organization has lots of data in semi-structured (e.g. JSON files) or unstructured (e.g. text documents, images, audio, video) formats which are assumed to be useful for data analysis in the future, but in which format the data will be needed, is not yet clear. [9, ch 4]

In this case the data can be assembled into a loosely structured collection called the data lake with the hope that the data analysts of the future will figure out how to structure and use that data. [9, ch 4]

If the data is thrown into the lake in a too unstructured format, the lake can easily turn into a data swamp from where the data becomes too hard to find and use to be of any real value. One tip to prevent that is to emphasise the collection of metadata: it must be clear, where the documents and JSON object are coming from and which point in time do they describe. [9, ch 4]

4.4 Database systems used for data warehouses

Data warehouses are usually built using SQL database systems. NoSQL systems such as document databases and Apache Hadoop platform are also used because of their good performance, which, however, comes at the price of less options for enforcing data quality and integrity. [9, ch 4]

4.5 Normalization and star schemas

The highly normalized table structure prevents many problems in database design, most notably helping to enforce data integrity: since normalization eliminates data redundancy and makes sure that each fact is stated in only one place, it prevents situations where tha data is updated in one, but not the other place which would lead to contradictions. Since each fact is saved only once, it also helps to reduce the need for storage space.

However, such a table structure can grow relatively complex and become hard to understand for business users. In cases of large volumes of data, it might also negatively influence query performance since table join operation is relatively slow in most databases. [10, ch 1]

The star schema is a data model which is designed to address these two aforementioned problems. With this approach the data is divided into fact tables and dimension tables. Fact tables log data about events such as business transactions which can be measured numerically. Dimension tables contain mostly textual data representing different aspects of the facts in the fact tables. [10, ch 1]

For example, a row in the fact table might state that customer A brought product B in the store C on the data D with some price. The price is the numeric content that will be used to calculate aggregates (most commonly the sum) over a large group of rows. Tables for customer, product, store, and date are dimension tables, which contain mostly textual attributes such as the product's name, brand name, and category which are understandable to the business users. This allows the business user to easily construct a query like "give me the total sum of all sales of product B from customer A during the last month". [10, ch 1]

A row in the fact table is usually uniquely identified by a combination of references to the dimension tables and thus usually has a composite primary key. Dimension

tables do not have composite primary keys. [10, ch 1]

The dimension tables are denormalized: for example, the product table could contain several attributes regarding the brand, not a reference to the brand's row in a separate table. This makes the data model more easy to query for the business users and also speeds up the queries. [10, ch 1]

The approach of designing a database using the star schemas is known as dimensional modelling. [10, ch 1]

Proponents of the Kimball architecture (named after Ralph Kimball) advocate for structuring the entire data warehouse using dimensional modelling. Proponents of the hub-and-spoke Corporate Information Factory (CIF) approach advocated by Bill Inmon maintain that the data in the main warehouse must be highly normalize. That normalized data could then be used to provided a separate layer with the star schemas which is made available for the business users. That could mean a layer providing a ubiquitous overview of the entire warehouse or individual data marts addressing the needs of a particular group of users. [10, ch 1]

This approach is seen as providing the best of both worlds: the benefits of normalization guarantee data integrity in the main single-source-of-truth data warehouse, which provides data for the star schema layer optimized for performance and the needs of the business users. [10, ch 1]

5. Overview of data analysis methods

This project uses three types of data analysis methods: classification, regression, and clustering.

Classification and regression are both part of the supervised learning category. This means that the predicted value is known for a set of training data and the computer must learn how to predict that value for new records. [11, ch 1]

Clustering belongs to the unsupervised learning category: the wanted result is not known and it is the task of the computer to find patterns which would help to make sense out of the data. [11, ch 1]

5.1 Classification

Classification is a subcategory of supervised learning where the goal is to predict the categorical class labels of new data records based on the patterns observed in the training data. The class labels are discrete, unordered values indicating that the object belongs into some category. A famous example of a classification task is predicting the iris species based on the measurements of the blossom. [11, ch 1–2]

From the large variety of classification algorithms, random forest classifier proved to yield the best results in this project.

Random forest belongs to the ensemble learning category: several individual classifiers are combined to form a meta- classifier which has better generalization performance than any of the individual components. [11, ch 7]

This idea behind ensemble learning reflects the pattern that a committee of human experts tends to make better decisions than any of its individual members. [11, ch 1–2]

The committee can either consist of people from the same field of expertise or bring together members with different backgrounds. In a democratic setting, the decisions

are often made based on the majority voting principle: the solution which received the greatest number of votes is chosen.

An ensemble of classifiers functions very similarly. Its members can, but do not have to, use different algorithms and can be trained on different subsets of the training data. When a prediction is made, the choice that received the most votes from the members is selected as the prediction of the ensemble. [11, ch 1–2]

A random forest is an ensemble of decision trees. A decision tree is a classification method that works similarly to a plant identification book: it starts to split the dataset into smaller parts based on the value of an attribute. The attribute is chosen so that the two separate groups formed by the split would be as distinct as possible based on the class membership of their elements. In a simple case, the algorithm continues to split the data as long as all the subgroups only contain elements of the same class. When predicting the class membership of a new element, the classifier walks through the same splits and observers in which subgroup the element would land and predicts the class of the elements in that group. [11, ch 2]

The performance of a classifier can be measured based on different metrics. The simplest of them is the accuracy score: the percentage of correctly classified items. That metric is trustworthy when the different classes have a similar number of elements. [11, ch 2]

For example, if there are three classes with 100 elements in each, the probability of predicting correctly using a random guess would be 33%. If the accuracy of the classifier is 60%, that shows that the classifier is significantly better than a random guess.

However, the accuracy score might not be very useful if the classes have very different sizes. For example, in a fraud detection task only 1% of transactions might be frauds and thus a dummy model which would always predict that a transaction is not a fraud would already have an accuracy of 99%.

One useful method for getting a more accurate overview of the performance of the classifier is the confusion matrix depicted in table 1. [11, ch 6]

Table 1. The confusion matrix

true positives	false negatives
false positives	true negatives

The confusion matrix counts the numbers of true positive, false positive, true negative, and false negative predictions. [11, ch 6]

The accuracy metric groups both true positives and true negatives together as correct predictions as opposed to false positives and false negatives as incorrect predictions, but in practice the different mistakes can have very different consequences. [11, ch 6]

For example, falsely predicting that a person has some illness and should undergo further examination is usually a much more harmless mistake than incorrectly stating that the patient is health and does not need to worry.

In case of a binary classification – choosing between only two classes – the algorithm computes the probability of the element belonging to one of the classes and if the probability is higher than a given threshold, usually 0.5, the element is classified into that class. Otherwise, the element is seen as a member of the other class. That threshold can be adjusted to favour one type of mistakes over the other. [11, ch 6]

For example, the classifier can be adjusted to classify a patient as sick even if the probability of sickness is only 0.3. This results in fewer people incorrectly labeled as health with the price of incorrectly labeling more people as sick.

5.2 Regression

Regression also belongs into the supervised learning category. What distinguishes it from classification is that the predicted value is continuous (e.g. price, score), not a class label. [11, ch 1]

The main regression algorithm used in this project is LASSO which builds on linear regression. Linear regression constructs a linear equation where the attributes of the input data are the variable x_1 to x_n and the predicted value is y:

$$y = w_0 + w_1 x_1 + \cdots + w_n x_n$$

The training of the model involves finding the best weights (w_0 to w_n) so that the prediction would be as accurate as possible. [11, ch 10]

Linear regression could also be interpreted as fitting a straight line (a hyperplane in

a more general case) through the cloud of data points so that the sum of distances between the points and the line would be as small as possible. [11, ch 10]

One of the problems with that simple approach is overfitting, the situation where the model performs well on known data, but does not learn to generalize and thus performs badly on new data. The problem can be lessened by designing the model so that it would favour equations with relatively small weights or situations were some weights become zero. LASSO (least absolute shrinkage and selection operator) is one of such models. [11, ch 10]

The performance of a regressor is measured by the R^2 value. $R^2 = 1$ indicates flawless performance. $R^2 = 0$ corresponds to constantly predicting the mean value. Thus a regressor with R^2 values between 0 and 1 can be considered useful whereas a regressor with a negative R^2 value would be inferior to a dummy constant prediction. [11, ch 10]

5.3 Clustering

Clustering is an exploratory data analysis technique that organizes a pile of information into meaningful subgroups called clusters without having any prior knowledge of their group memberships. The clusters are formed so that objects in the same cluster would be similar to each other and objects in different clusters would be different from each other. [11, ch 11]

There is also no single technique for determining the number of clusters. It is possible to calculate which number would most accurately represent the groupings in the data, but ultimately it is up to the analyst to decide which similarity metrics and cluster count produces the most useful result. For example, in case of customer segmentation – a widely used application for clustering – the data might have 24 natural groupings but since that is way more than a human analyst can cognitively handle, a smaller cluster count cna be chosen. [11, ch 11]

The main clustering algorithm used in this project is k-means. The k-means algorithm belongs to the category of prototype-based clustering. [11, ch 11]

Prototype-based clustering means that each cluster is represented by a prototype. In case of continuous features it is a centroid which represents the average of the points in that cluster. [11, ch 11]

K-means requires the number of clusters (k) to be specified beforehand. The algorithm starts then to iteratively find the best positions to the k centroids to minimize the difference between elements within the same cluster (known as sum of squared errors or cluster inertia). K-means is both easy to understand and also computationally very efficient and is therefore one of the most widely used clustering algorithms. [11, ch 11]

6. Data sources

6.1 Existing data sources

At the time the thesis were written the following data sources containing relevant data about the course were found:

- Homework submissions
- Psychological factors survey
- General information regarding the student and their studies
- Chat messages
- Plagiarism detection service
- Moodle activity
- Time-tracking service

Due to the limited scope of the thesis, at this time only the first five data sources were examined in detail. Further research would be needed to cover the remaining data sources and possibly also discover some new.

6.1.1 Homework submissions

Throughout the course the students had to solve programming tasks which were then automatically tested and the results were instantly displayed to the student. A student could send as many submissions for the given task as they wanted and once they had received the score they were pleased with, they needed to defend their solution to redeem the points.

The tasks had different complexity levels, the main categories being:

- `MX` – a smaller exercise giving fewer points
- `EX` – the main exercise type giving plenty of points
- `XP` – extra difficult exercise requiring independent study of material beyond the scope of the course

There were also three special one of a kind tasks: `DJ` as an introduction to the Django web framework and special puzzles `WAT` and `AOC`.

Since the same technical solution was used for the exam and also the smaller tests, their submissions can also be found in the dataset. Throughout the course three main tests were made:

- `TK` – a smaller test
- `KT` – a bigger test
- `EXAM` – the final exam

6.1.2 Psychological factors survey

On the first week of the course the students filled in an extensive survey regarding beliefs, motivations, study strategies, burnout, problem-solving abilities and prior programming experience.

For shortness and simplicity it will be referred to as "the grand survey".

6.1.3 General information regarding the student and their studies

General information such as age and gender was known about most, although not all, students. Studies related information suh as their curriculum was also known.

6.1.4 Chat messages

Throughout the course a Discord server was used to give the students a place to ask questions regarding the homework assignments and other related subjects. There were also dedicated channels for sharing memes and other extracurricular content.

6.1.5 Plagiarism detection service

The plagiarism detection service provided information about similarities in the students' homework submissions, which could indicate that the exercises were solved together or that the student had copied the code written by someone else on the course.

6.2 Novel data sources

It was found that collecting the following information would contribute to formatting a more comprehensive overview of the students' study process.

6.2.1 Weekly check-in

Beside the grand survey which the students take at the beginning and at the end of the course, it would be useful to also track the weekly progress of the students. While the submission logs and chat messages can help to track certain aspects of the student's progress, some extra questions should be asked once a week. The questions are formulated as statements and the student is asked to express their agreement with the statement on a scale from 1 to 6. The statements are:

- The topics and exercises of that week were very difficult.
- The course is adding a lot of stress and anxiety to my life.
- What has been taught on the course is relevant and useful for me.
- I am confident that I will finish the course with good results.

6.2.2 Network survey

It can be assumed that the risk of a student's dropout increases when their friends drop out. If the hypothesis is valid, that data could be useful for dropout prediction. However, the currently available data is not sufficient to validate this hypothesis.

The necessary data could be collected by conducting a network survey where each student gives the names of their coursemates they are interacting with the most. Further research is needed to identify on which week the survey should be conducted and whether it would be necessary to conduct the survey multiple times within the semester.

6.2.3 Dropout reasons survey

Analysis of 2021 course data suggest that the reasons behind quiting the course are more complex than just not being able to keep up with the pace. There are also some students leaving despite their close to average results (appendix 2). However, the current data is not enough to understand these reasons and thus it would be necessary to ask the leaving students to explain the reasons behind their decision.

7. Study analysis data warehouse implementation

7.1 General notes on the implementation

The system is implemented as a physical data warehouse using the PostgreSQL database system. The data is stored in normalized tables, but made available for analysis through a layer of denormalized views.

7.2 Why PostgreSQL?

This data warehouse is implemented using PostgreSQL (a.k.a. Postgre) version 14.

Before explaining the rationale behind that choice, the requirements for the database system in this project are explained.

Firstly, the data volume is not expected to grow very big. The data from all the used sources regarding the 2021 course took about 300 MB of disk space which is insignificant given the capabilities of al modern database systems. Although the data volume is bound to increase every year as the historical data will not be deleted and novel data sources with potentially bigger amounts of data will be integrated, it is very unlikely that the database size would even come near to the concept of "big data". This means that performance considerations such as column based storage are not that critical.

Secondly, it would be preferable if the database system would be open-source to eliminate the risk of unfavourable changes in the product, its pricing, and service level agreements. Also, the author personally maintains that especially in the academic context it is important to propagate the use of open-source software and support the communities behind those projects instead of doing marketing for proprietary closed-source products.

Thirdly, the database system should be widely used and actively maintained to

ensure that is continues to be improved in the foreseeable future. This also assures that there will be enough documentation, tutorials, literature, and experts available.

Fourthly, the database system should provide a large variety of feature, most notable a good selection of datatypes, materialized views and support for custom functions.

When choosing the database system, the following options were considered:

- Oracle data warehouse solutions
- MS SQL Server
- Teradata
- Vertica
- MS SQL Server
- PostgreSQL
- MySQL

The first five are all proprietary and could thus only come into consideration when all the open-source alternatives fail to meet the set criteria.

PostgreSQL and MySQL are both open-source projects. Considering the provided features, Postgres was seen to be much more suitable for implementing a data warehouse:

- PostgreSQL has a larger variety of built-in datatypes.
- MySQL does not implement materialized views.
- Postgres is over all much more customizable. It is an object-relational database which enables the user to define their own types. It also has good support for extensions, foreign data wrappers, and procedural languages.

7.3 The layers in the database

The database contains the following layers, each implemented as a separate schema:

- knowledge base layer (`kb`) – contains the main data in highly normalized tables
- metadata layer (`meta`) – contains metadata about the data in the knowledge base layer
- analytics layer (`aly`) – contains (materialized) views and other datastructures designed to make analytical queries simpler and faster

7.4 The data model

Figures 3, 4, 5, 6, 7, and 8 illustrate the developed datamodel. The datamodel has been divided into logical subsystems to make the figures more compact and easy to interpret.

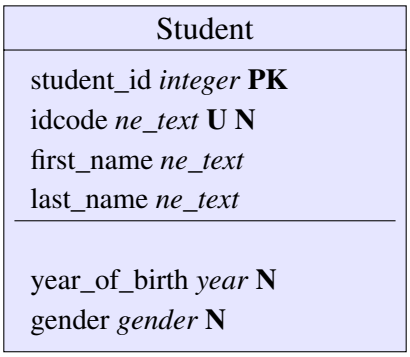

Figure 3. Students subsystem

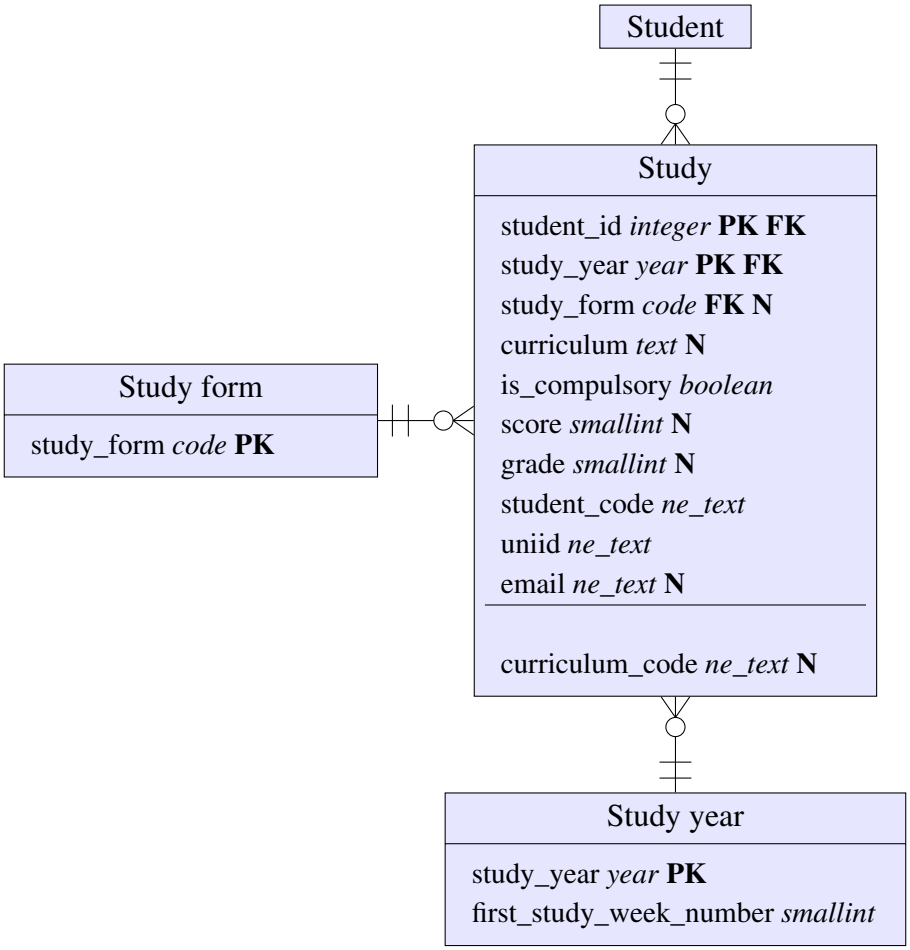

Figure 4. Studies subsystem

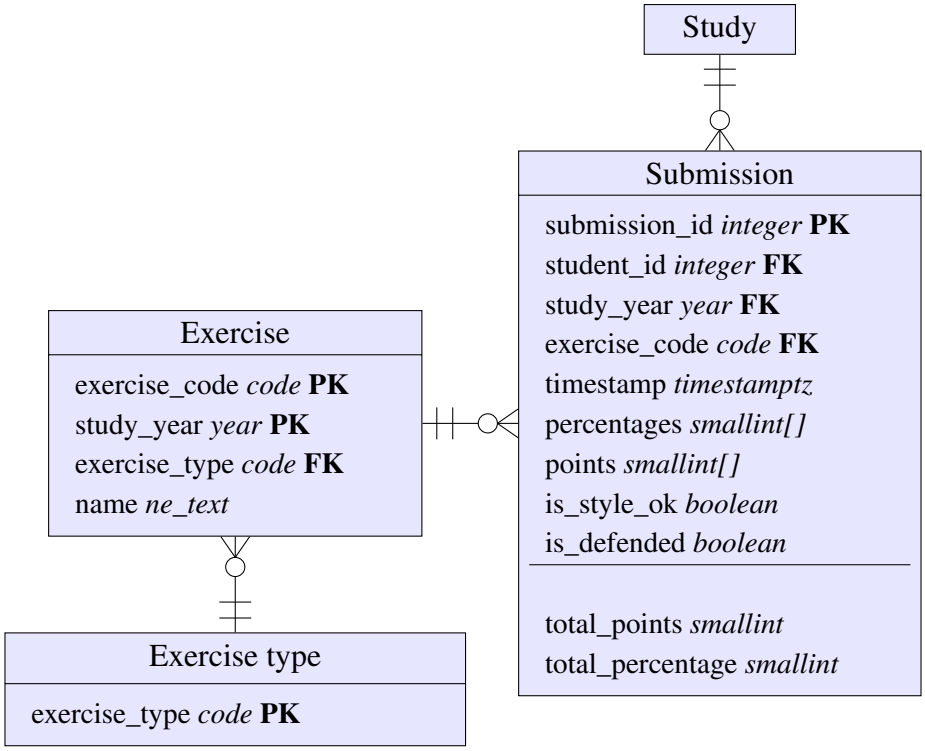

Figure 5. Submissions subsystem

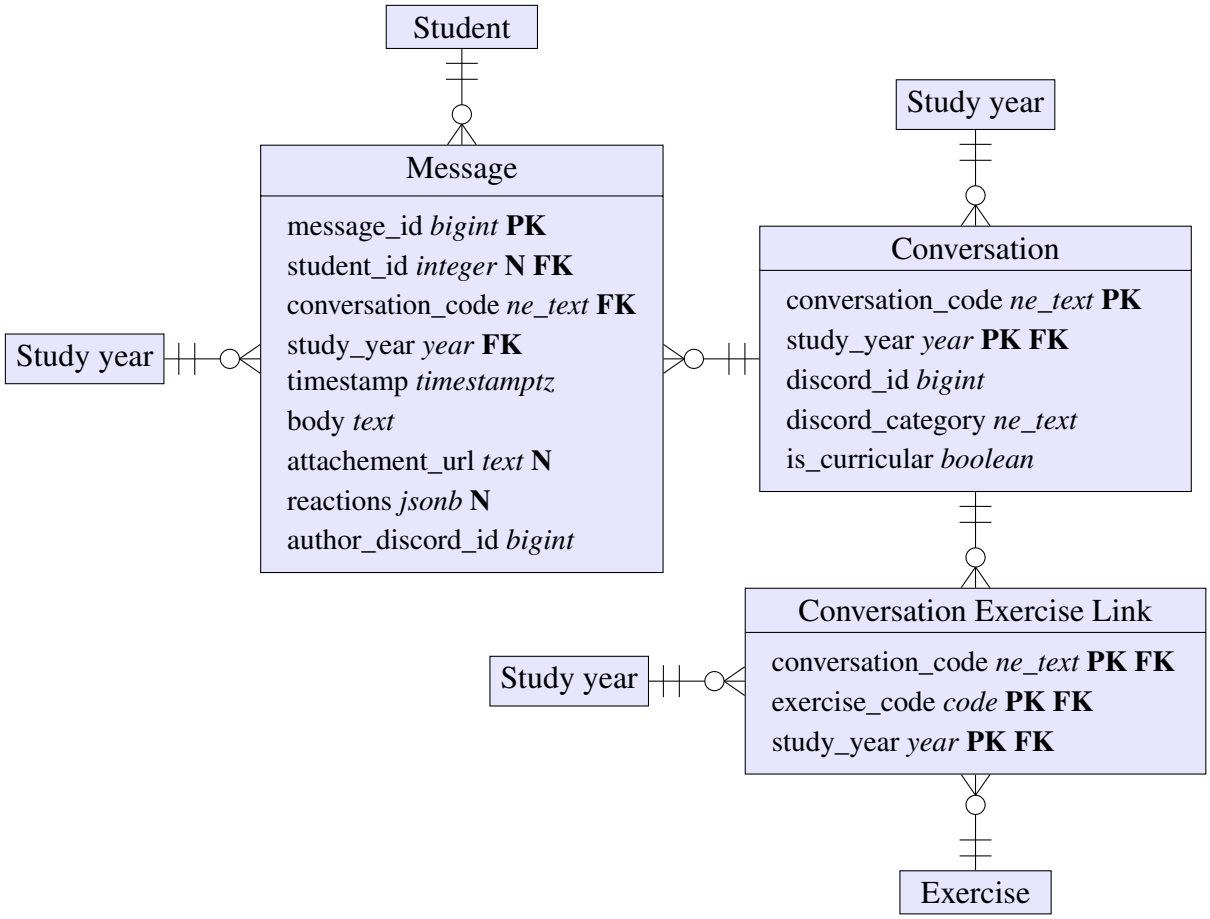

Figure 6. Chats subsystem

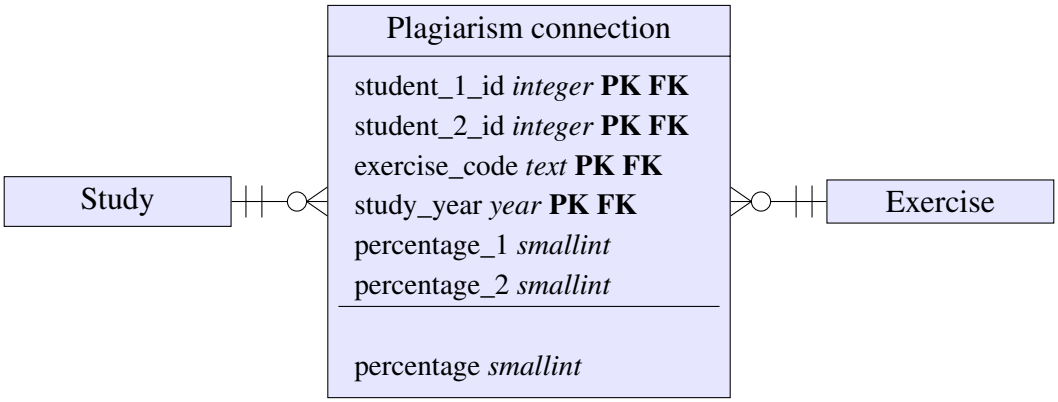

Figure 7. Plagiarism subsystem

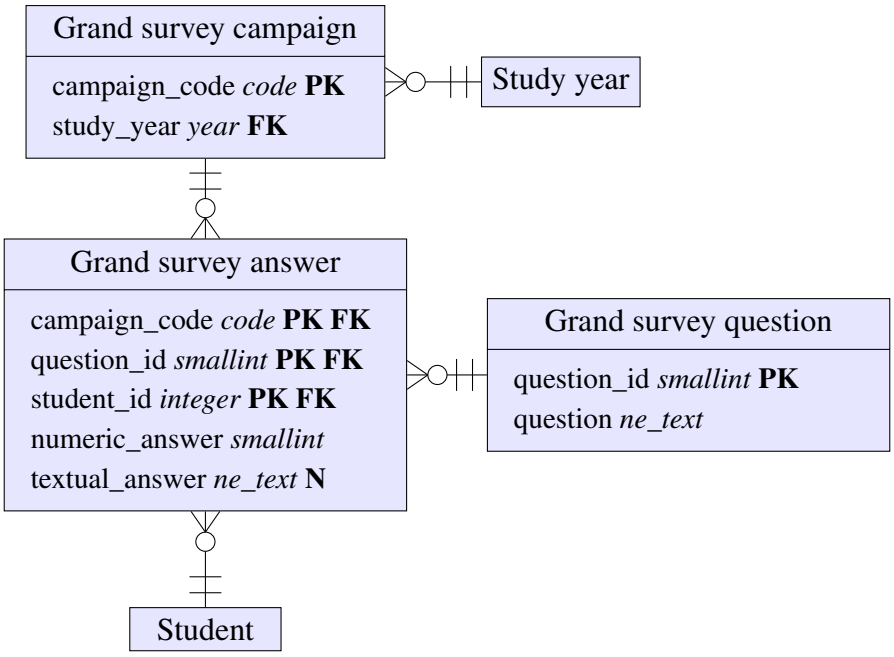

Figure 8. Grand survey subsystem

8. Exploratory data analysis on 2021 course data

The "ITI0102 Introduction to Programming" (est *Programmeerimise algkursus*) course aims to give an overview of programming assuming no prior programming experience. The course introduces a variety of general programming paradigms using the Python language.

8.1 Overview of the 2021 students

In 2021 the course was declared by 376 students of which were 97 female and 254 male (the gender is unknown for 25 students).

38% of the students where either 19 or 20 year old, 32% were between 21 and 29, 22% in their thirties and 5% older than that. The youngest student was 18 and the oldest 63.

For 281 (75%) students that course was a compulsory part of their curriculum.

8.2 Overview of study results

Study outcomes of 2021 are shown on figure 9. The non-negative results correspond to the final grades. In addition, two extra outcomes are described:

- -2 *(not seen)* – the student did not submit any homeworks, although they had registered to the course
- -1 *(no exam)* – the student had submitted some homeworks, but they did not take the final exam

Thus 224 students (60%) passed the course and 152 (40%) did not.

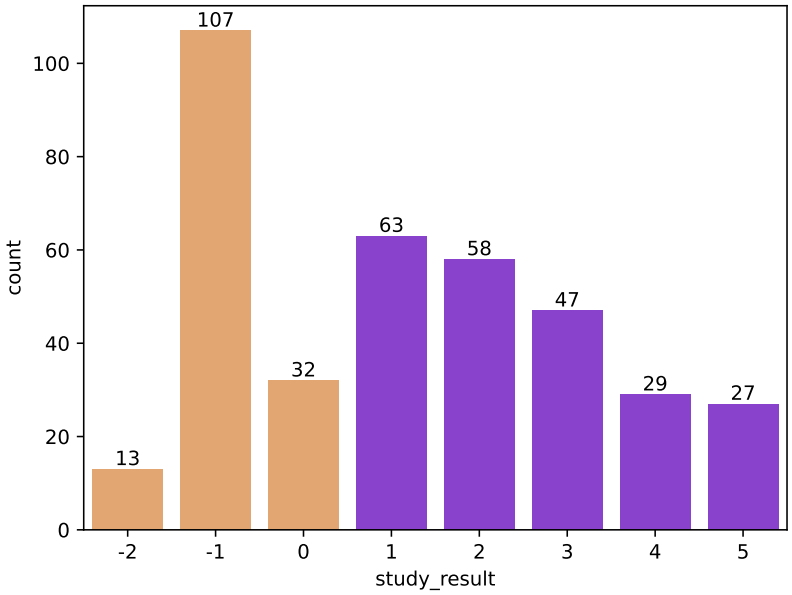

Figure 9. Study results

8.3 Overview of the dropout

Figure 10 shows the time of the last homework submission on the x-axis and the student's final score on the y-axis with their grade indicated by the hue.

Firstly, it is seen that all students with positive grades have taken the final exam. The long vertical lines in the end of the x-axes between `2022-01` and `2022-02` indicate exam dates.

The figure also shows that some students with grade 0 have participated in the exam, but the majority of students with grade 0 have stopped submitting homeworks way before the exam session began. The vertical line after `2021-11` indicates a bigger test and it seems that for some students that has been their final attempt on the course.

8.4 Group chat messages

Figure 11 shows how the student's curricular and extracurricular message counts correlate with the student's final score and their last activity time. Logarithmic scale is used for most comprehensive result.

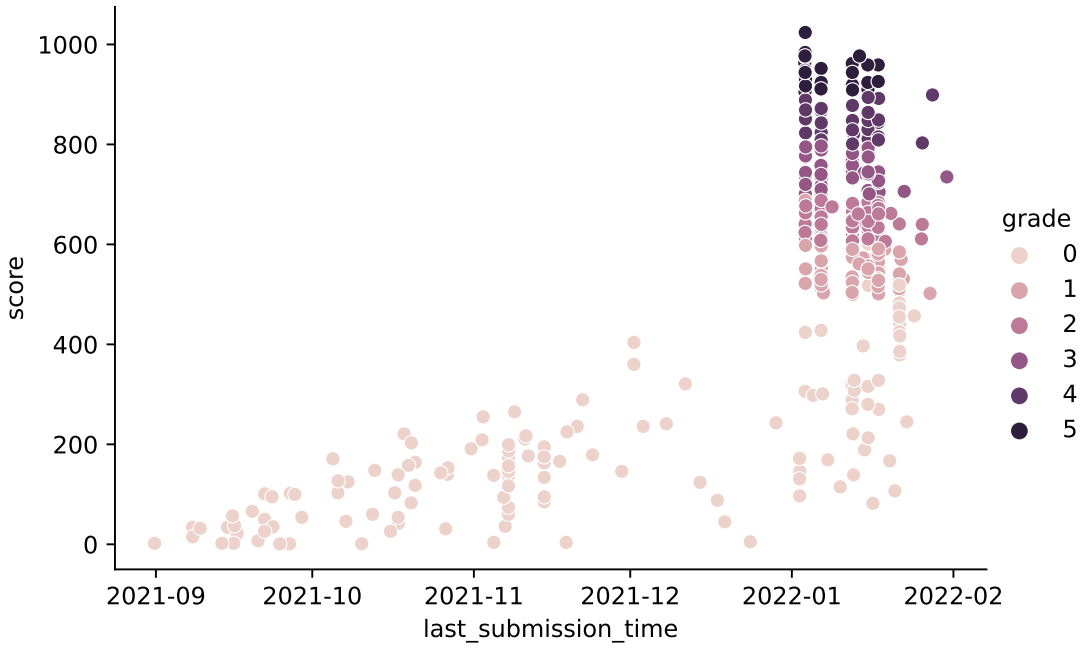

Figure 10. Dropout times

8.5 Grand survey

Via factor analysis 20 dimensions were identified:

1. fixed mindset regarding mind
2. enjoying and appreciating education
3. anxiety level
4. importance of reputation
5. skillfull study methodology
6. self-confidence
7. procrastination
8. satisfaction with teacher
9. fixed mindset regarding math
10. dutifulness
11. effortless gratification
12. fear of looking stupid
13. computer games and math results
14. prior experience with programming
15. studying together with course mates
16. willingness to see big picture
17. study stress impact on health and wellbeing
18. mathematical problem solving abilities

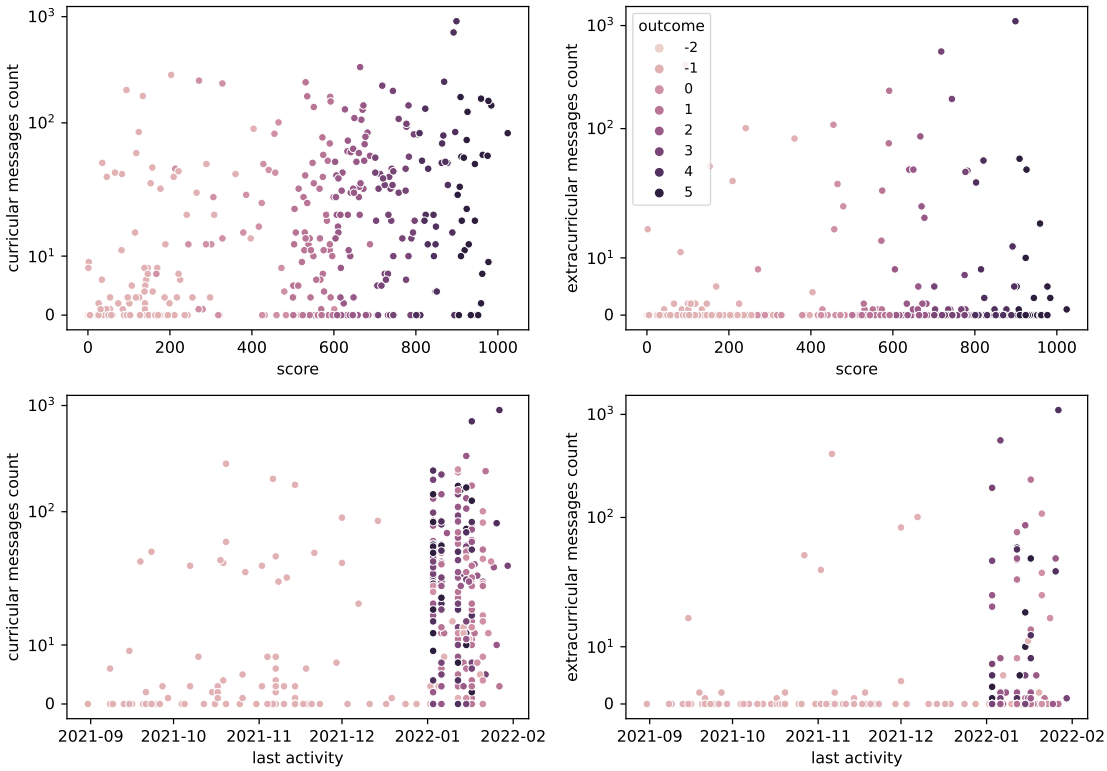

Figure 11. Message count correlation with course outcomes

19. avoiding effort
20. appreciating education

Five individual questions had a correlation with the score with absolute value higher that 0.3:

- question nr 45: (correlation -0.37) "I procrastinate with the homework so long that I do not get it ready by the deadline."
- question nr 43: (correlation 0.36) "I believe that I am doing well on the course compared to by coursemates."
- question nr 86: (correlation 0.35) "What grade will you get?"
- question nr 63: (correlation -0.35) "I procrastinate with the homework until the last minute."
- question nr 75: (correlation 0.35) "Math exam score."

Two factors out of 20 had a correlation with the score with absolute value higher than 0.3:

- self-confidence (correlation 0.34)
- procrastination (correlation -0.32)

Questions 45 and 63 and the self-confidence factor had the strongest correlation with last activity time (-0.23 for both questions and 0.2 for self-confidence).

8.6 Homework submissions

Figure 12 shows the number of submissions per study week.

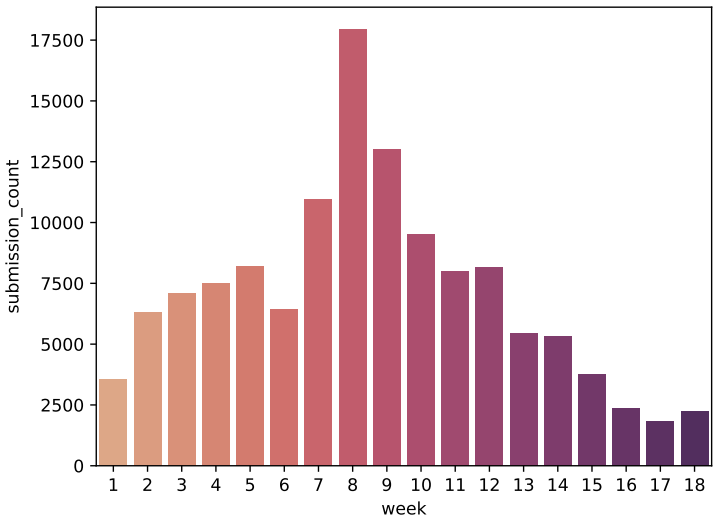

Figure 12. Submission count per study week

Figure 13 shows the correlations between submission count and the student's final score. The figure in the upper left corner plots the total submission count while the succeeding figures filter out some specific exercise type. It is seen that for `EX` exercises (which make up the vast majority of all submissions) students with mediocre results have made the most submissions. Students with high results have completed the exercises with fewer submissions and students with lower results have not started so many exercises. However, for more challenging exercises such as `XP`, `DJ`, `WAT` students with higher results have generally also made more submissions. Due to the vast difference in submission counts, logarithmic scale had to be used on these three plots.

Both started and finished exercise count have a strong correlating with the final score as well as with each other. It is also seen that even some people who quited the course had started and even solved one `XP` task. However, those who started at least two `XP` tasks where guaranteed to come to the exam and those who started at least three or solved at least two were guaranteed to pass the course.

The variety in average points for submission is much greater among the early quitters,

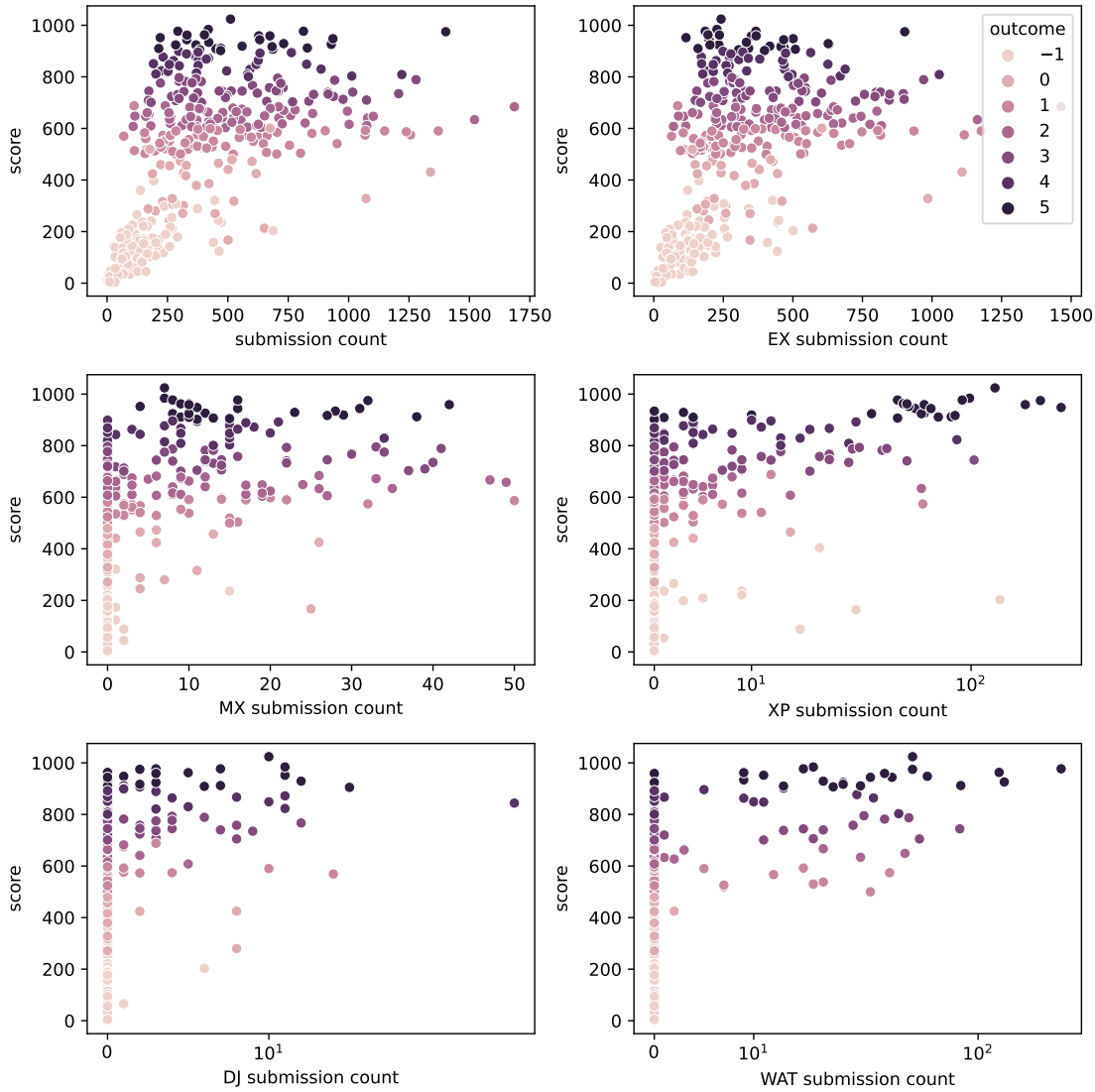

Figure 13. Submission count correlation with course outcomes

most likely due to the smaller number of submissions. Interestingly the students with the highest average finished with the grade three or four, not five.

Students with the highest results solved the EX exercises much faster than the others while the students with mediocre results invested the most time. Similar patterns can be observer with the started and finished exercise ration: due to the smaller nuber of started/finished exercises the variety is much higher among the early quitters. Students with the highest results finished a larger percentage of started exercises compared to the students with mediocre results.

9. Academic performance prediction

The following experiments implement classification and regression models for academic performance prediction.

Regression models provide more nuanced output as they predict the final grade, not just a boolean value (dropout vs no dropout) as the classification models do.

However, for risk group identification, classification models might yield better results as their prediction threshold can be adjusted to change the ratio of false negative and false positive results.

In the future both models should be deployed to work in parallel. Observing cases where the models yield contradictory results (e.g. the student is predicted to drop out, but have a very high grade) can be used to discover corner cases and further enhance the models.

9.1 Predicting on the first week if the student passes the course

In this experiment the boolean value indicating whether the student will pass the course was predicted based on the data know on the first study week, i.e. the general information about the student (their year of birth, gender, and whether this course is a compulsory part of their curriculum) and their studies and the psychological factors derived from the grand survey.

Knowing that 60% of students passed the course, it would be easy to implement a dummy classifier which always predicts that the student passes the course. The accuracy of such a classifier would be 0.60 and thus the experiment can be considered successful if the accuracy of the implemented classifier exceeds that threshold.

The experiment used classifiers provided by `scikit-learn`. The accuracy of the classifiers was evaluated using cross validation. Besides the accuracy score, confusion matrix was used to understand the behaviour of the classifiers.

From the observer classifiers `RandomForestClassifier` performed best achieving the accuracy of 0.718 ± 0 .021 which is almost 12% better that the dummy classifier described above.[1]

Next classifiers in the ranking were `GaussianProcessClassifier` and `LogisticRegression` with accuracies of 0.694 ± 0.051 and 0.689 ± 0.076, respectively.

Since only 70%[2] of the students has taken the survey, psychological factors were not known for the other 30% and were replaced with zeros. Since people who had takes the survey were more likely to pass the course, the sole fact that the survey was taken was already useful information for classification.

However, 30% of missing data for the factors did impact the classifier's ability to use them in the predictions. The left-hand column in table 2 shows that `feature_importance` is relatively low for all psychological factors. The importance of these factors rises significantly when the classifier was trained only on the data regarding students who did take the survey as seen in the right hand column. General information is shown in italic to distinguish it from the psychological factors.

Table 2. Predicting course passing on first week – 10 most important feature

data with missing factors included		data with missing factors excluded	
feature name	importance	feature name	importance
year of birth	0.27	*year of birth*	0.081
gender	0.053	procrastination	0.057
is the course compulsory	0.05	interest	0.051
procrastination	0.046	inefficient study strategies	0.051
emotional support	0.041	social learning	0.049
task avoidance	0.04	performance approach motivation	0.048
inefficient strategies	0.035	self efficacy	0.048
social learning	0.034	introjected regulation	0.047
introjected regulation	0.033	intrinsic regulation	0.044
self efficacy	0.033	burnout	0.044

Table 3. Predicting course passing on first week – confusion matrix with threshold 0.5

true positives: **54**	false negatives: **7**
false positives: **36**	true negatives: **16**

The confusion matrix (table 3) shows that the algorithm is more inclined towards

[1] The classifier used the following parameters:
`RandomForestClassifier(max_depth=7, n_estimators=1000, max_features=2)`.

[2] The survey was taken by 264 students out of 376.

giving false positives, i.e. too optimistically predicting that the student is going to pass.

Since the aim behind the classification would be to detect people with potential difficulties, it is important that everyone with a problem would be correctly classified into the risk group. If people who do not have problems also accidentally land in the risk group, that is a less severe problem.

Table 4. Predicting course passing on first week – confusion matrix with threshold 0.7

true positives: **34**	false negatives: **27**
false positives: **12**	true negatives: **40**

The problem can be fixed by using a higher threshold instead of 0.5 which is the default. Table 4 shows that when the threshold is raised to 0.7, the number of false positives has dropped significantly without negatively impacting the total number of accurately classified students.

9.2 Predicting on the first week the student's final score

In this experiment the student's final score was predicted based on the same data used in the previous experiment in section 9.1.

In this case a simple dummy regressor could always predict the mean score and the goal of this experiment is to implement a regressor which would perform better that than.

The best results were achieved using the `Lasso` regressor giving the R^2 value 0.263 $\pm$0.125. It was followed by `Ridge` ($R^2 = 0.259 \pm 0.133$) and `LinearRegression` ($R^2 = 0.258 \pm 0.135$).

Most important attributes used by the `Lasso` regressor are listed in table 5. General information is shown in italic to distinguish it from the psychological factors.

9.3 Predicting if the student passes the course throughout the weeks

This experiment builds on the experiment described in section 9.1, but adds to the dataset various new attributes reflecting the student's performance and behaviour throughout the course.

Table 5. Predicting score on first week – 10 most important feature

feature name	coefficient
is the course compulsory	156.422
gender	56.862
procrastination	-21.216
self efficacy	14.837
interest	12.816
efficient strategies	-11.984
social learning	8.501
math beliefs	-7.603
social anxiety beliefs	6.589
inefficient strategies	-6.244

The analysis is repeated for all of the 16 study weeks to observ how the new data coming in each week influences the accuracy of the model.

The new attributes are:

- number of homework submissions
- number of homework submissions regarding XP exercises
- number started exercises
- number of messages sent to the group chat
- total score on that week

Figure 14 illustrates how the accuracy of the model changes throughout the weeks. The blue line indicates the accuracy, the broader light blue area corresponds to the standard deviation. It is clear that as the weeks progress, the model becomes more accurate and its standard deviation decreases. However, the accuracy does not rise above 0.888 as even on the 16th week the exam results are not known yet.

Table 6 lists the ten most important features on different weeks. General information is underlined and psychological factors shown in italic to distinguish them from the performance indicators. On the first week general information about the student and their studies as well as the psychological factors from the survey are still important for the classifier. However, as the weeks progress, attributes related with the student's performance and behaviour start to become more important.

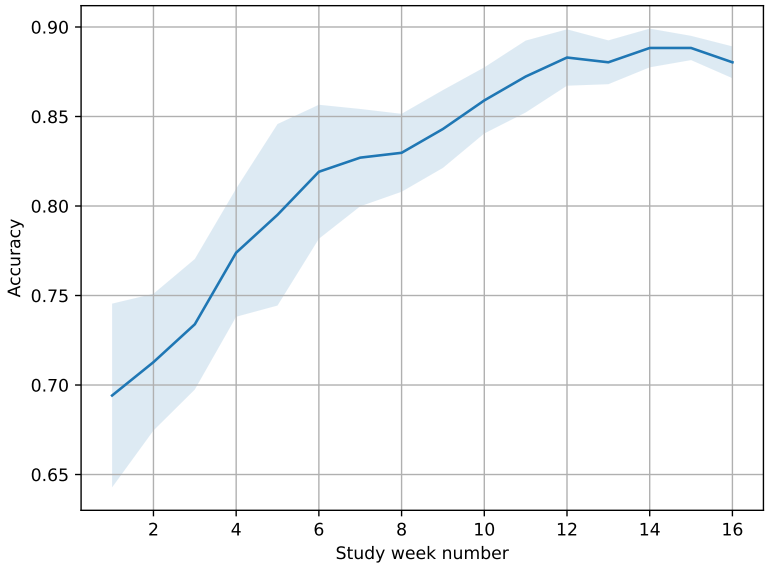

Figure 14. Accuracy of course passing prediction throughout the weeks

Table 6. Predicting course passing – 10 most important features on different weeks

	week 1	week 7	week 14
1	year of birth	score on week	score on week
2	score on week	started exercise count	started exercise count
3	submission count	year of birth	submission count
4	started exercise count	submission count	year of birth
5	is the course compulsory	message count	message count
6	*emotional support*	xp submission count	xp submission count
7	*anxiety*	*self efficacy*	*anxiety*
8	message count	*efficient strategies*	*efficient strategies*
9	gender	*introjected regulation*	*procrastination*
10	*interest*	*anxiety*	*emotional support*

9.4 Predicting the student's final score throuought the weeks

This experiment builds on the experiment described in section 9.2, but adds to the dataset the new performance and behaviour metrics described in section 9.3.

Figure 15 illustrates how the accuracy of the regressor changes throughout the weeks. The blue line indicates the accuracy, the broader light blue area corresponds to the standard deviation. The regressor seems to abide by a similar patters as the classifier depicted in figure 14.

Table 7 lists the ten most important features on different weeks. General information is underlined and psychological factors shown in italic to distinguish them from the

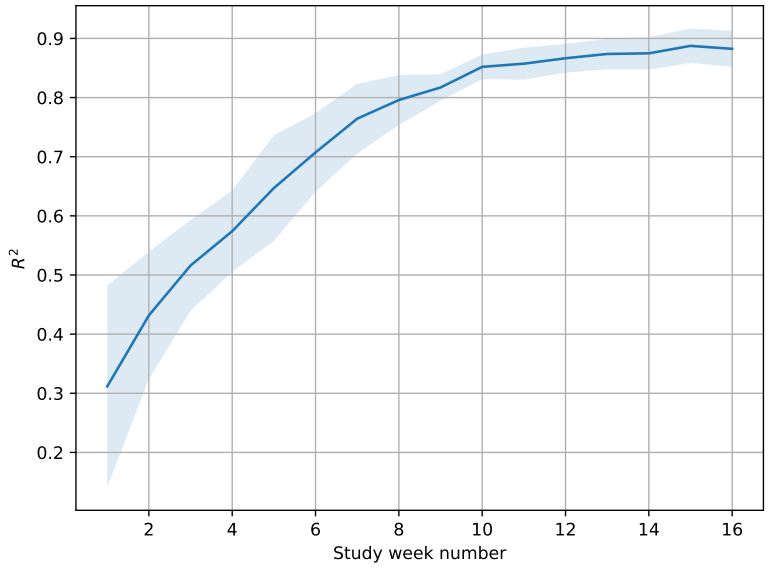

Figure 15. Accuracy of score prediction throughout the weeks

performance indicators. The patterns are very similar to the findings presented in table 6.

Table 7. Predicting final score – 10 most important features on different weeks

	week 1	week 7	week 14
1	score on week	score on week	score on week
2	year of birth	started exercise count	started exercise count
3	submission count	submission count	submission count
4	is the course compulsory	year of birth	year of birth
5	*self efficacy*	is the course compulsory	message count
6	*interest*	message count	xp submission count
7	*introjected regulation*	xp submission count	*loss of interest*
8	*emotional support*	*anxiety*	*inefficient strategies*
9	*easy tasks*	*performance approach motivation*	*procrastination*
10	*procrastination*	*adopted regulation*	*introjected regulation*

10. Student segmentation

The purposes of that experiment is to find the logical groupings of the students to better understand their needs and abilities.

Only the students who have taken the grand survey and who's general information is known were included into the dataset to ensure a more accurate outcome.

The experiment is firstly performed using the data available on the first week and then repeated incorporating the new data available by the 14th week.

10.1 Student segementation based on data available on the first week

10.1.1 Selecting the features

To make the results more easy to interpret, only a subset of the available features were used. The featured were selected based on their importance in the classification and regression experiments described in chapter 9 as well as the author's domain knowledge.

After experimenting with different sets of attributes, the following combination was chosen as it yielded the most meaningful results:

- age
- is the course compulsory
- math beliefs (higher value indicates that the student has a more fixed mindset believing that mathematical abilities cannot be developed)
- social anxiety beliefs (higher value indicates that the student has a more fixed mindset believing that social skills cannot be developed)
- interest (higher value indicates that the student is more interested in the course)
- procrastination (higher value indicates that the student has a higher tendency to procrastinate)
- self efficacy (higher value indicates that the student has a more self-confidence)

- emotional support (higher value indicates that the student perceives more emotional support from the teacher)

These are the features given as input for the clustering algorith, however some more features such as gender and final score are used later on to describe the formed clusters.

10.1.2 Selecting the number of clusters

Figure 16 shows the relation between the sum of squared errors (lower values indicating that the cluster centroids are closes to the data points) and the number of clusters. If the line made a notable "elbow curve" at some number of clusters, that would be a good number to choose as it would reflect well the logical groupings in the data.

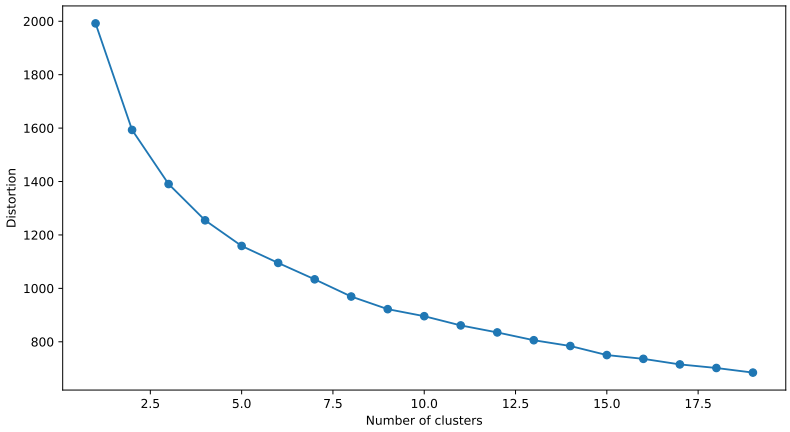

Figure 16. Student clustering on first week – choosing the best cluster count

The line does not have a notable elbow curve, but after experimenting with different numbers of clusters, four was found to produce the most interpretable result.

10.1.3 Interpreting the results

Figure 17 shows the centroids of the four clusters. They can be interpreted as the average representatives of their respective clusters.

Group 1 represents the youngest segment with the average age of 22 for whom the course is compulsory. They are highly interested in the course and have high self-confidence, but very fixed mindset. With 682 points they have the highest average score among the four groups. They tend to be mostly male.

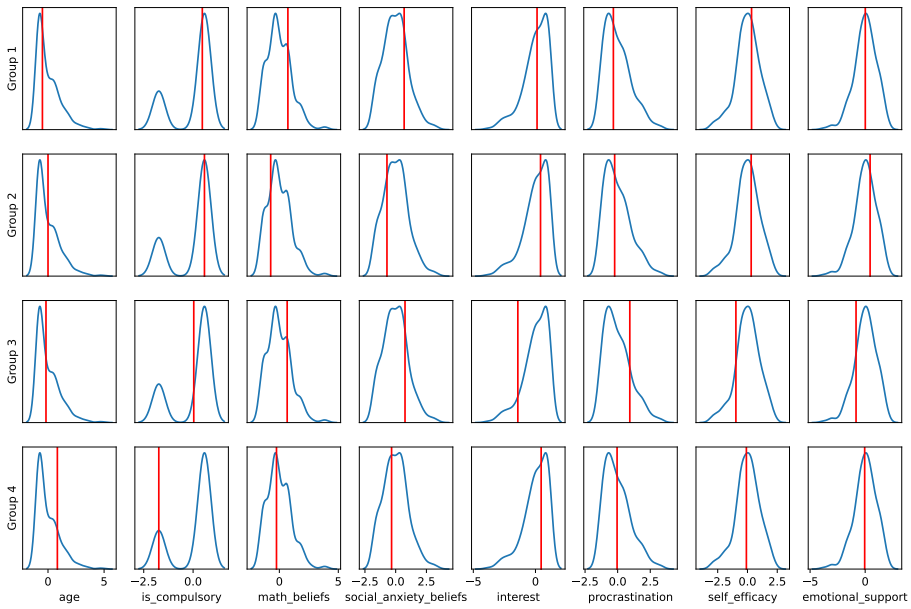

Figure 17. Student clustering on first week – cluster centroids

Group 2 also represents a bit older people with the average age of 29 for whom the course is compulsory. They have high interest in the course and high self-confidence. However, compared to group 1, they have a much more growth oriented mindset and also perceives more emotional support from the teacher. They, too, have a high average score of 610 points and are mostly male.

Group 3 consists of younger people with the average age of 25. They have a very fixed mindset, are not at all interested in the course, procrastinate a lot, have low self-confidence and do not perceive much emotional support from the teacher. Their average score is only 425. This group contains both male and female members.

Group 4 consists of more mature females with the average age of 33 years for whom the course is not mandatory. They have a high interest in the course, but the lowest average score, 427 points.

10.2 Student segementation based on data available on the 14th week

10.2.1 Selecting the features

The features were selected based on the same logic as in section 10.1. The selected feature are:

- age
- is the course compulsory
- emotional support (higher value indicates that the student perceives more emotional support from the teacher)
- interest (higher value indicates that the student is more interested in the course)
- procrastination (higher value indicates that the student has a higher tendency to procrastinate)
- submission count
- submission count for hard exercises
- score on the 14th week

10.2.2 Selecting the number of clusters

Figure 18 shows the relation between the sum of squared errors and the number of clusters.

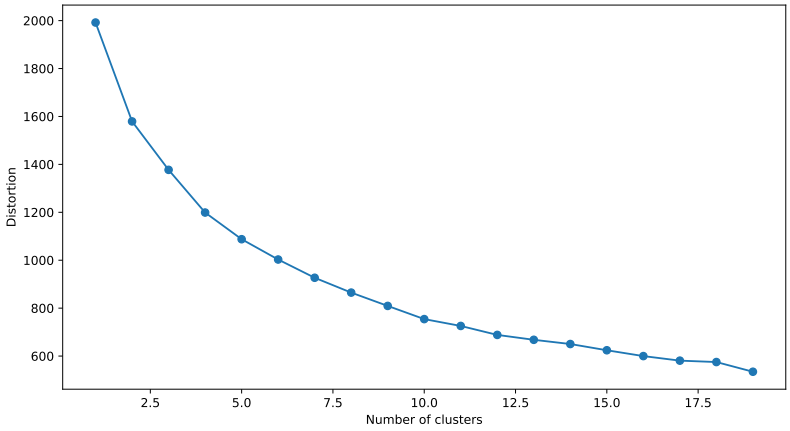

Figure 18. Student clustering on 14th week – choosing the best cluster count

The line does not have a notable elbow curve, but after experimenting with different numbers of clusters, four again turns out to produce the most interpretable result.

10.2.3 Interpreting the results

Figure 19 shows the centroids of the four clusters.

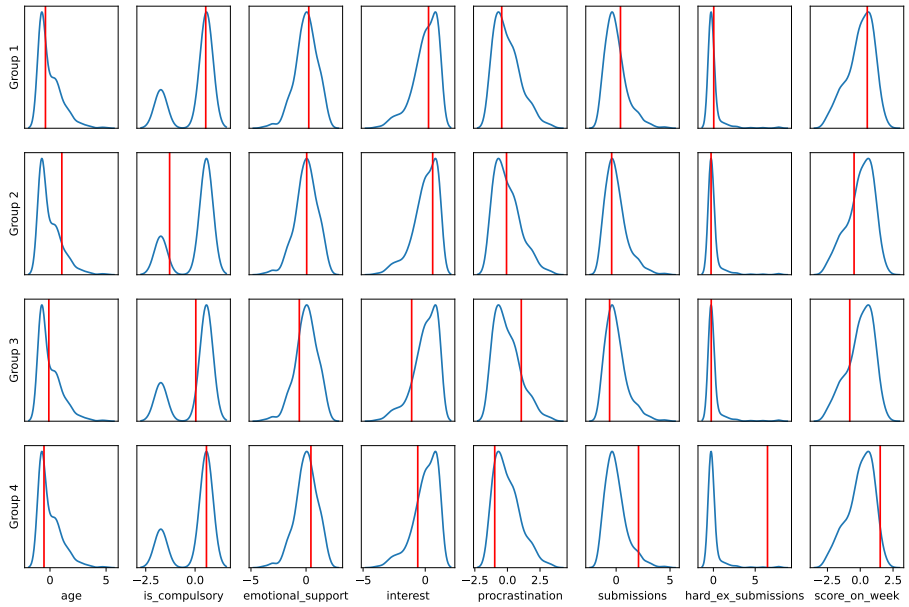

Figure 19. Student clustering on 14th week – cluster centroids

Group 1 represents young people with the average age of 23 for whom the course is compulsory. They feel a lot of emotional support from the teacher, are highly interest and do not procrastinate much. Their submission count is above average, but they do not put much effort into solving the XP exercises and thus they have a rather average score of 693 points. This is the largest cluster with 128 members.

Group 2 consists of older people (34 on average) for whom the course is not compulsory. They have a high interest in the course. They do not put much effort into solving the XP exercises and thus their score falls below average (424 points).

Group 3 represents people with the average age of 25, for most of them the course is compulsory. They do not perceive a lot of emotional support from the teacher and their interest in the course is very low. They have problems with procrastination, do not solve the XP exericises and have the lowest average score among the four groups (370 points).

Group 4 represents the youngest people (22 years on average) for whom the course is compulsory. They perceive a lot of emotional support from the teacher, but are not very interested in the course. They do not procrastinate, their submission count is very high and they are working heavily on solving the harder XP exercises. They have the highest average score among the groups: 964 points. This group consists of males exclusively. This is the smallest cluster with only 4 members.

11. Summary

As a result of the project, a data warehouse was implemented to facilitate data from various sources regarding the "Introduction to programming" course.

The aim of that project was to design a system which would help the teacher to better understand and support their students.

It was shown that the collected data can be used to fulfil that purpose. In particular, it was shown that the collected data can be used to predict the student's academic performance and that some predictions can already be made on the very first study week.

It was also shown that the students on the course can be divided into different segments based on psychological and behavioural factors. That in turn will help the teacher to better consider the needs and abilities of each segment.

Conducting three more surveys was suggested for the following study years to provide a more comprehensive dataset for further research.

Bibliography

[1] Heleriin Ots. “Predicting academic achievement based on Moodle log data and self-assessed learning-related psychological factors”. Tallinn University Of Technology, 2020. URL: https://digikogu.taltech.ee/et/Item/2ddcb69a-27d9-492c-8c51-886ad60e3478.

[2] Brenda Uga. “Predicting Dropouts Among TUT Students: Calculating Probabilities Using Machine Learning and Displaying Results in a Web Application”. Tallinn University Of Technology, 2017. URL: https://digikogu.taltech.ee/et/Item/1bf6b6b2-052a-451a-a528-88670b968539.

[3] Jeff Reback et al. *pandas-dev/pandas: Pandas 1.4.2*. Version v1.4.2. Apr. 2022. DOI: 10.5281/zenodo.6408044. URL: https://doi.org/10.5281/zenodo.6408044.

[4] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[5] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[6] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. *Gephi: An Open Source Software for Exploring and Manipulating Networks*. 2009. URL: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

[7] Laura Välik. *Mindset teooria – kas vanad koerad õpivad uusi trikke?* accessed 2022-03-08. 2019. URL: https://mihus.mitteformaalne.ee/mindset-teooria-kas-vanad-koerad-opivad-uusi-trikke/.

[8] Venerable Bodhi Lama Erik Drew Jung. *Lecture notes from courses on Buddhist psychology*. 2021.

[9] Erki Eessaar. *Andmeaidad ja andmevakad*. accessed 2022-04-01. 2021. URL: https://maurus.ttu.ee/download.php?aine=346&document=32870&tyyp=do.

[10] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition*. 2013.

[11] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning - Third Edition*. 2019. ISBN: 978-1-78995-575-0.

Appendices

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis

I, Eerik Sven Puudist,

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Implementing data warehouse and machine learning models for student segmentation and academic performance prediction" , supervised by PhD Ago Luberg and PhD Innar Liiv
   (a) to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   (b) to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

Appendix 2
Work-in-Progress: Exploring psychological and behavioural differences between university IT student segments formed based on dropout time and academic performance

Eerik Sven Puudist[1], Ago Luberg[1], and Kati Aus[2]

[1] Tallinn University of Technology, Estonia,
eerik.sven@gmail.com,
ago.luberg@taltech.ee
[2] Tallinn University, Estonia,
katiaus@tlu.ee

Abstract. This research project investigates the role of self-regulated learning competencies and other psychological factors in the dropout times and academic performance of students on their first semester at university in a programming course. The students were broken into seven segments based on their dropout times and study outcomes. The psychological and behavioural differences between the segments were investigated. It was found that students who leave at the beginning of the course, who leave after a negative test result, and who fail the final exam have different psychological profiles and require different kinds of support. Based on that, suggestions were given for decreasing dropout rates and increasing academic performance.

Keywords: learning analysis, university dropout, self-regulated learning, machine learning, segmentation

1 Introduction

The high university dropout rate, particularly in the field of information technology, is a pressing problem in higher education. In Estonia up to 32% of students quit their studies already at their first year [1, p 3].

A common reason for resigning in programming related courses is the accumulation of small gaps in apprehending the subject which makes it hard for the student to keep up with the pace of the studies. How well the student is able to handle such a backlog depends depends largely on their self-regulated learning competencies [2] e.g., their adaptive beliefs, motivation, and study strategies.

The problems leading to the student quitting their studies thus begins long before they become apparent to the teacher and once the problems have already

developed that far, the right timeframe for supporting the student might already be over. This issue is especially topical with bigger group sizes where the teacher does not have the opportunity to personally monitor each student. Especially during a pandemic situation when study in the classrooms is not possible, it might take several weeks before the teacher recognises that a student is not participating anymore.

It is therefore essential that the students in the risk group could be identified in advance so that they could receive special mentoring and support throughout the course. That mentoring could help the student to obtain the necessary understandings and study strategies to be successful on the course. If the complexity level of the course exceeds the student's capacity or if the student recognises that this subject is too far from their real field of interest, the mentoring program could help them to find a curriculum which aligns better with their interests and abilities so that failure in the given curriculum would not cause the student to give up their academic pursuits.

Fruitful research for implementing a binary classifier for identifying the students with higher dropout risk for similar courses has already been conduced, such as [3]. However, this project aims to identify more nuanced student segments, since both the risk group and the better performing group are very heterogeneous with different segments needing very different approaches from the teacher.

This research is part of a joint project of Tallinn University of Technology and Tallinn University for developing a more personalised approach to programming courses, where psychological differences and various backgrounds of the students could be taken into consideration to help each student to derive most value from the course.

2 Methods

The research is based on the data regarding the "ITI0102 Introduction to Programming" course (hereinafter referred to as "the course") taught in Tallinn University of Technology at the fall of 2021. The aim of the course is to give an overview of programming in Python language assuming no prior experience. For the majority of the students the course is taught on the first semester of their studies.

2.1 Measured features

The following sets of features were measured:

- Information about the student: age, gender;
- Information about the studies: curriculum, form of study (daily vs session), whether the given course is compulsory;
- Study motivation based on achievement goals theory [4] and self-determination theory [5];

- Beliefs and mindsets based on implicit beliefs theory [6, 7];
- social and emotional aspects related to studying [8, 9];
- study strategies [10, 11, 12, 13, 14].

2.2 Data sources

The data was collected from a variety of sources:

- General information regarding the student and their studies from the study information system ÕIS;
- Psychological factors survey (hereinafter referred to as "the survey");
- Homework submission logs;
- Course chat messages;
- Data from plagiarism detection service.

The survey consisted of 87 questions regarding self-regulated learning competencies, prior programming experience, and problem-solving abilities. The survey will be published as a separate paper.

The students took the survey on the first week of the semester, thus it reflects primarily their general tendencies and predispositions, not their experiences with the specific course. Taking the survey was not compulsory. From the 376 students who declared the course 264 took the survey and gave the permission for using the data for research.

The majority of the exercises on the course where programming tasks with automated tests. While the student was solving the exercise, they could submit it to the test system as many times as they wanted and use the automatic feedback to enhance their solution. The homework submission logs can thus be used to estimate when and for how long the student was solving each exercise.

The course had a dedicated Discord server where the students could ask for help for solving the exercises. There were also separate channels for extracurricular communication such as sharing programming-related jokes.

2.3 Software and algorithms

The scientific Python stack, most notably pandas [15], was used for the analysis. Scikit-learn [16] was used for calculating feature importance. The hierarchical clustering module[3] from SciPy [17] was used for drawing dendrograms. The plots were created with Matplotlib [18].

[3] `scipy.cluster.hierarchy`

3 Results

Based on the dropout time and study outcome students where divided into seven segments:

- `above avg dropout` – students who resigned although their study results were close to or above the course average;
- `early dropout` – students who dropped out before the bigger test on the 10^{th} study week;
- `post test dropout` – students who dropped out directly after the test, usually after receiving a negative grade;
- `late dropout` – students who dropped out later, but still before the exam;
- `failed exam` – those who did not pass the exam;
- `mediocre results` those who finished with mediocre results (grades 1–3);
- `excellent results` – those who finished with excellent results (grades 4 and 5).

From the 376 students who declared the course, 224 (60%) passed and 152 (40%) did not. The dropout times and final scores of the students who did not pass are depicted on figure 1.

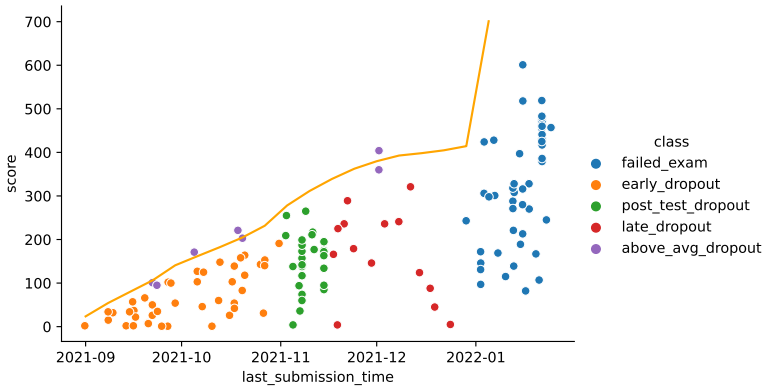

Fig. 1. Dropout times

The mean values of the measured attributes for each segment are depicted on figure 2. For attributes which can be correlated with academic performance, the red end of the spectrum indicates worse values (e.g. more fixed mindset, higher anxiety level) whereas the blue end indicates better values (e.g. more growth-oriented mindset, lower anxiety level). The meaning of the colorcoding for each attribute is explained next to the attribute name. The relative similarity of the segements is shown on figure 3.

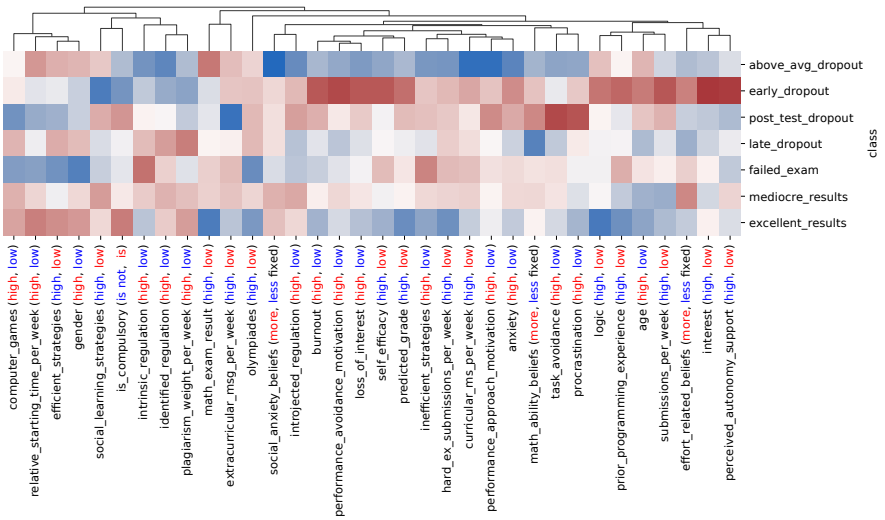

Fig. 2. Clustermap of student segments

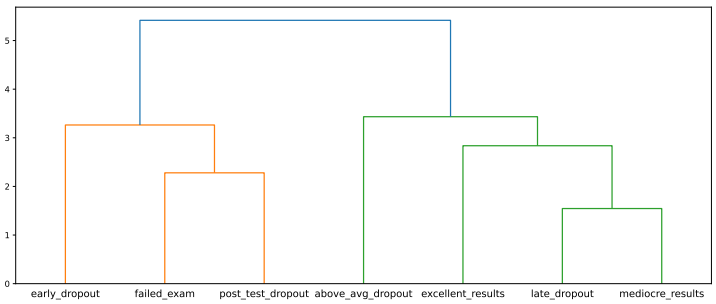

Fig. 3. Dendrogram of student segments

That the students who leave despite their good study results are very similar to the students who finish with excellent results: both are very active in the Discord chat and put time into solving the harder exercises. However, the leavers group contains more members for whom the course is not compulsory and who do not have prior programming experience and their average age is higher.

The early dropout group consists primarily of people without prior programming experience. Moreover, they have very little interest in the subject, high burnout, and very low perceived autonomy support.

Students who leave after receiving a negative test result have problems with task avoidance and procrastination. However, their math exam results and prior

programming experience are both above average and higher than for any other group who did not pass the course. Thus, it seems that they would have potential to complete the course if they would receive support for fixing the issues with task avoidance and procrastination.

Most people who fail the exam are starting without prior programming experience. They seem to be very diligent people as they start early with the exercises and have participated on olympiades during school time, but they have a lot of inefficient strategies and low intrinsic regulation.

The group with mediocre results already has some prior programming experience, but they do not develop as far as they could as they are bounded by a relatively fixed mindset. The group with excellent results could possibly perform even better if they had learned efficient study strategies and developed a more growth-oriented mindset.

4 Discussion

Although the course is designed for people without any prior programming experience, even a little prior experience turned out to greatly increase the probability of passing the course. It can be speculated that there is so much novelty for the student on their first semester at the university that it is hard to keep up with a course with such high pace. Since simplifying the course is not an option for the knowledge is required for subsequent programming courses, it might be useful to suggest the students to go through some online introductory materials regarding programming in Python in their own pace before the course starts.

The study showed that psychological factors have a strong influence on the students' dropout rates and study outcomes and that students with fixed mindset and tendencies for task avoidance and procrastination are likely to quit the course even though their mathematical abilities are sufficient.

The students are already receiving some psychological coaching in terms of self-regulated learning on their first semester, but that does not seem to be sufficient. Further research is required to explore that deficiency in detail, but it can be assumed that the students do not develop the ability to apply what has been taught in real life. Such training should therefore be integrated into the programming course itself. That would be especially necessary after students have received a negative test result and when they are most likely to resign.

The most capable students would also need more psychological training as they could achieve even more if they were introduced to efficient study strategies and if they could become more conscious of their beliefs and motivations.

References

[1] Heleriin Ots. "Predicting academic achievement based on Moodle log data and self-assessed learning-related psychological factors". Tallinn University Of Technology, 2020. URL: https://digikogu.taltech.ee/et/Item/2ddcb69a-27d9-492c-8c51-886ad60e3478.

[2] Ernesto Panadero. "A Review of Self-regulated Learning: Six Models and Four Directions for Research". In: *Frontiers in Psychology* 8 (2017). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2017.00422. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2017.00422.

[3] Natalja Maksimova, Avar Pentel, and Olga Dunajeva. "Predicting First-Year Computer Science Students Drop-Out with Machine Learning Methods: A Case Study". In: Mar. 2021, pp. 719–726. ISBN: 978-3-030-68200-2. DOI: 10.1007/978-3-030-68201-9_70.

[4] Elizabeth A. Linnenbrink and Paul R. Pintrich. "Achievement Goal Theory and Affect: An Asymmetrical Bidirectional Model". In: *Educational Psychologist* 37.2 (2002), pp. 69–78. DOI: 10.1207/S15326985EP3702_2. eprint: https://doi.org/10.1207/S15326985EP3702_2. URL: https://doi.org/10.1207/S15326985EP3702_2.

[5] Maarten Vansteenkiste et al. "Identifying configurations of perceived teacher autonomy support and structure: Associations with self-regulated learning, motivation and problem behavior". In: *Learning and Instruction* 22.6 (2012), pp. 431–439. ISSN: 0959-4752. DOI: https://doi.org/10.1016/j.learninstruc.2012.04.002. URL: https://www.sciencedirect.com/science/article/pii/S0959475212000321.

[6] Samira Moumne et al. "Implicit Theories of Emotion, Goals for Emotion Regulation, and Cognitive Responses to Negative Life Events". In: *Psychological Reports* 124.4 (2021). PMID: 32674669, pp. 1588–1620. DOI: 10.1177/0033294120942110. eprint: https://doi.org/10.1177/0033294120942110. URL: https://doi.org/10.1177/0033294120942110.

[7] Hans Schroder et al. "The Role of Implicit Theories in Mental Health Symptoms, Emotion Regulation, and Hypothetical Treatment Choices in College Students". In: *Cognitive Therapy and Research* 39 (Sept. 2015), pp. 120–139. DOI: 10.1007/s10608-014-9652-6.

[8] Marcos Carmona-Halty et al. "School Burnout Inventory: Factorial Validity, Reliability, and Measurement Invariance in a Chilean Sample of High School Students". In: *Frontiers in Psychology* 12 (2022). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2021.774703. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2021.774703.

[9] Kati Vinter, Kati Aus, and Grete Arro. "Eesti ja vene õppekeelega põhikooliõpilaste akadeemiline läbipõlemine". In: *Eesti Haridusteaduste Ajakiri. Estonian Journal of Education* 7.1 (May 2019), pp. 128–156. DOI: 10.12697/eha.2019.7.1.06. URL: https://ojs.utlib.ee/index.php/EHA/article/view/eha.2019.7.1.06.

[10] Kati Aus et al. “Kus tegijaid, seal nägijaid? Akadeemilise prokrastineerimise õpetajapoolse märkamise seosed õpilaste individuaalsete erinevustega”. In: *Eesti Haridusteaduste Ajakiri. Estonian Journal of Education* 2 (Apr. 2014). DOI: 10.12697/eha.2014.2.1.09.

[11] Robert A. Bjork, John Dunlosky, and Nate Kornell. “Self-regulated learning: beliefs, techniques, and illusions.” In: *Annual review of psychology* 64 (2013), pp. 417–44.

[12] John Dunlosky et al. “Improving Students’ Learning With Effective Learning Techniques”. In: *Psychological Science in the Public Interest* 14 (2013), pp. 4–58.

[13] M. J. Lawson et al. “Teachers’ and students’ belief systems about the self-regulation of learning”. In: *Educational Psychology Review* 31.1 (2013), pp. 223–251.

[14] Jari-Erik Nurmi et al. “The role of success expectation and task-avoidance in academic performance and satisfaction: Three studies on antecedents, consequences and correlates”. In: *Contemporary Educational Psychology* 28 (Jan. 2003), pp. 59–90. DOI: 10.1016/S0361-476X(02)00014-0.

[15] Jeff Reback et al. *pandas-dev/pandas: Pandas 1.4.2*. Version v1.4.2. Apr. 2022. DOI: 10.5281/zenodo.6408044. URL: https://doi.org/10.5281/zenodo.6408044.

[16] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[17] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[18] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.