TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Hendrik Ütt 203925IAPM

INTEGRATING SENTIMENT ANALYSIS AND MACHINE LEARNING TO PREDICT STUDENTS' ACADEMIC PERFORMANCES IN AN INTRODUCTORY PROGRAMMING COURSE

Master's Thesis

Supervisor: Ago Luberg
PhD

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Hendrik Ütt 203925IAPM

TUDENGITE AKADEEMILISTE TULEMUSTE ENNUSTAMINE PROGRAMMEERIMISE ALGKURSUSE AINES MEELESTATUSE ANALÜÜSI JA MASINÕPPE KOMBINEERIMISEL

Magistritöö

Juhendaja: Ago Luberg
PhD

Tallinn 2023

Author's Declaration of Originality

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

Author: Hendrik Ütt

09.05.2023

Abstract

In this master's thesis, the critical issue of high student dropout rates is being addressed by analyzing two years of an introductory programming course. The primary objective is to develop a method for early identification of students at risk of dropping out by conducting an in-depth analysis of the data, which includes sentiment analysis. Based on this analysis, machine learning techniques are employed to predict academic performance. This approach, intended as a stepping stone for further development, enables lecturers and support staff to provide targeted assistance, ultimately contributing to improved retention rates.

The thesis comprises a comprehensive overview of relevant theories, a description of the data used, an explanation of the conducted sentiment analysis and discovered correlations. Additionally, it includes an exploration of various academic performance prediction methods and their respective metrics, as well as an examination of the development and utilization of an application that incorporates these findings for practical purposes.

Three research questions and corresponding hypotheses are formulated and tested. The results indicate that it is feasible to assign reasonably accurate numeric values to various text fields, representing sentiment. In most of the cases, these values align with human-labeled scores, exhibit correlations with academic performance, and play a crucial role in predicting academic outcomes. Furthermore, a method is developed that allows training a machine learning model on one academic year's data and making predictions for subsequent years. This methodology is integrated into an application capable of accepting any subset of features to predict different aspects of academic performance, including semester scores, exam pass rates, and course pass rates.

The thesis is written in English and is 96 pages long, including 7 chapters, 15 figures and 26 tables.

Annotatsioon

Tudengite akadeemiliste tulemuste ennustamine programmeerimise algkursuse aines meelestatuse analüüsi ja masinõppe kombineerimisel

Käesolevas magistritöös käsitletakse kriitilist küsimust, milleks on kõrge üliõpilaste väljalangevus mahukatel programmeerimise ülikoolikursustel, analüüsides programmeerimise algkursuse ainet kahel erineval akadeemilisel aastal. Peamiseks eesmärgiks on kõrge väljalangemisriskiga tudengite varajase tuvastamise meetodi väljatöötamine, viies läbi andmete analüüsi, mis sisaldab ka meelestatuse analüüsi. Selle analüüsi põhjal kasutatakse õppeedukuse ennustamiseks erinevaid masinõppe meetodeid, mis lõpuks võimaldaks õppejõududel ja tugipersonalil pakkuda sihipärast abi, vähendades tudengite väljakukkumise määra. Loodetavasti kasutatakse töös saavutatud tulemusi ja avastatud järeldusi ka edasisteks arendusteks selles valdkonnas.

Lõputöö sisaldab põhjalikku ülevaadet asjakohastest teooriatest, kasutatud andmete kirjeldusest, läbiviidud meelestatuse analüüsist ja avastatud korrelatsioonide selgitustest. Samuti käsitletakse erinevaid õppeedukuse ennustamise meetodeid ja nende vastavaid tulemusi ning lõpuks ülevaadet rakendusest, kus kasutatakse eelnevaid leide praktilistel eesmärkidel.

Töö käigus katsetati erinevaid lähenemisviise, kõigepealt prooviti panna tudengite tagasiside küsitluste tekstiväljadele käsitsi meelestatuse väärtused. Valminud andmestiku põhjal prooviti treenida masinõppe mudelit, mis suudaks määrata ükskõik millisele tagasiside tekstile sobivat numbrilist väärtust, mis iseloomustaks seda, kui positiivselt tudeng ennast tekstis on väljendanud. Selline lähenemisviis ei toonud oodatud tulemusi, misjärel otsustati olemasolevaid eeltreenitud suuri keelemudeleid selle ülesande tegemiseks kasutada. Lisaks sellele prooviti töö käigus erinevaid masinõppe mudeleid koos erinevate andmete atribuutidega, et leida kõige optimaalsem kombinatsioon.

Lõputöös sõnastatakse ja kontrollitakse kolme uurimisküsimust ning kolme hüpoteesi. Tulemused näitasid, et erinevatele tekstiväljadele on võimalik omistada suhteliselt täpseid arvulisi väärtusi, mis väljendavad meelestatust. Mõnel juhul ühtivad need väärtused ka inimeste poolt märgistatud tulemustega, saavutades kohati korrelatsiooni skooriks 0.91. Lisaks sellele on need väärtused tugevalt seotud akadeemiliste tulemustega ja mängivad üliolulist rolli akadeemiliste tulemuste ennustamisel. Lõpuks töötatati välja meetod, mis võimaldab treenida masinõppe mudelit ühe õppeaasta andmete põhjal ja teha täpseid ennustusi mingiks teiseks aastaks.

Näiteks on töös valminud masinõppe mudelite abil võimalik treenida mudelit ühe aasta andmete põhjal ning ennustada enne teise aasta kursuse algust kursuse tulemusi. Selleks kasutatakse semestri alguse küsitluse vastuseid koos tudengite atribuutidega, mille põhjal on võimalik 70% täpsusega ennustada, kas tudeng läbib kursuse või mitte. Neljanda nädala jooksul tõuseb see näitaja juba 79% peale.

Lõputöös valminud masinõppemudelid on ka lõpuks integreeritud rakendusse, mis suudab aktsepteerida mis tahes andmete alamhulka, et ennustada akadeemilise edukuse erinevaid aspekte, sealhulgas semestri skoorid, eksami läbimise määrad ja kursuse läbimise määrad.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 96 leheküljel, 7 peatükki, 15 joonist, 26 tabelit.

List of Abbreviations and Terms

CI/CD	methodology for software development that emphasizes frequent integration and testing of code changes, and automating the delivery of software to production.
CSV	comma-separated values
Docker	containerization platform that allows developers to package applications and their dependencies into portable containers that can be deployed easily across different environments.
Docker Compose	tool for defining and running multi-container Docker applications.
Git	distributed version control system designed to manage and track changes to source code during software development
GitLab	web-based Git repository manager that provides version control, issue tracking, continuous integration, and other development tools.
JSON	JavaScript Object Notation
NumPy	Python library for numerical computing, which provides powerful tools for working with arrays, matrices, and other multi-dimensional data structures.
Pandas	Python library for data manipulation and analysis, which offers flexible and efficient data structures such as DataFrames and Series, making it a popular choice for handling complex datasets in various fields.
Python	high-level programming language that is widely used in data science, web development, scientific computing, and other fields.
SSH	A network protocol that provides secure remote access to a computer or server.
TalTech	Tallinn University of Technology

Virtual machine software-based emulation of a computer system, which can run its own operating system and applications as if it were a physical computer.

WatchTower tool for automating the process of updating Docker containers.

Table of Contents

List of Figures

List of Tables

1. Introduction

In this master's thesis, the challenge of high student dropout rates in university courses with large numbers of students is addressed. The primary focus is on developing a method to identify students at risk of dropping out early in the course. This would enable lecturers and support staff to provide targeted assistance, ultimately helping to improve retention rates.

The author chose this topic because he believes that it is possible to create something useful and practical as a result of the work. In addition, the following areas that interest the author are planned to be used during the work: data mining, natural language processing and machine learning.

The topic is important because currently the lecturer has to deal with quite a lot of extraneous things, such as analyzing the learning results of several hundred students, identifying students who need help and giving personal advice to each student. If it were possible to automate these activities, the lecturer would be able to direct more of this time to the content of the subject, thanks to which the quality of the education provided by the university would also improve.

One of the most pressing problems of the chosen topic is the following: in Estonia, fully 32% of students studying information and communication technology drop out of university in the first year. [1] In addition, the number of students who dropped out of all TalTech students was 17-23%, and already in the first year of the university, a full 15% of all students who started their studies drop out, a third of whom had achieved a competitive result in the mathematics state exam, which was higher than 75 points. [2]

Most drop out of university is in the first year, so the course covered in the thesis has high importance, because it takes place in the first semester. In addition, the solution to the problem may discover students who have great potential but simply need a more personal approach.

There are especially many failures in the first semester, during the basic subjects. In science subjects, a situation often arises where a student gets stuck on a topic and is unable to progress in any way, which has several consequences. Motivation decreases and the impression may arise that the subject is uninteresting, too theoretical and the study load is heavy. Such impressions are also the reasons why university is left unfinished. [3]

The work would contribute to the quality of education offered by the university, simplify the work of the lecturer and motivate students. Currently, the situation is as follows: a student gets into difficulties in a subject, falls behind the schedule and loses motivation, and there are quite a few such students. Some time later, the lecturer discovers that the students are struggling and have been struggling for quite some time, by which time it is already too late to take action. All of this could be avoided if the automated system would continuously monitor student data and progress and, accordingly, signal to the lecturer if someone is in difficulty and what kind of help they need.

The task is relatively complex, the complexity lies in identifying the correct input data. A lot will be revealed during the work, the topic is relatively large and there are several ways to approach it. One solution to the problem would be to develop a system with which the lecturer can continuously monitor the progress of the students during the course and then guide or assist the students as needed. The solution would look like this: an application where it is possible to send students' data, which in turn forwards it to a machine learning model, which returns an overview that shows how students are progressing in the subject and predicts how they will do in the future.

The concrete problem that will be solved during the work is the following: Dropout of students in the subject ITI0102 Introduction to Programming course. Solving this problem, there are better prerequisites for fewer students to drop out of information and communication technology majors. After the work, the problem has been solved at least partially, and it would also be possible to further develop the solution to the problem.

The very important part of this course is that it is possible to get large quantities of data from each student. Even before the course begins the students will complete a grand survey that includes questions about education, anxiety and self-beliefs. Combining the data collected from the grand survey with students' attributes it is already possible to make some kind of predictions on how well students will fare in the course.

The course itself consists of weekly assignments and an exam, which are then combined to get the final grade for the students. Thanks to the structure of the course, it is possible to monitor students each week extensively by collecting data during the course. Each week students submit a homework assignment and complete a weekly feedback questionnaire (hereinafter referred to as "weekly questionnaire"), where they can report personal well-being, how they are progressing and give feedback about the course with numeric values and text fields.

Research has already been done in this area, and also this course has been analyzed, but a functioning solution has not yet been created. For example, in 2020, the master's thesis "Prediction of learning performance based on Moodle log data and psychological factors related to self-assessed learning" was done. [4] In addition, in 2022, the bachelor's thesis "Creation of Data Warehouse and Machine Learning Models for Student Grouping and Academic Capacity for prediction". [5]

These works did analyze and predict student learning outcomes, but nothing that lecturers could actually use to benefit from it has yet been done. However, in 2017, the bachelor's thesis "Prediction of TUT student dropout: calculating the probability using machine learning methods and displaying the results in a web application" was done, where it is also possible to display the prediction results. [6] The shortcoming of this work is the too general scope, i.e. subject completions are used as input data, i.e. this solution could not be used within a single subject.

This thesis represents a departure from prior research in several key ways. Firstly, the data employed is notably more concise and detailed way than that used in previous works, eschewing any vagueness. In contrast to prior research which has often relied on Moodle logs, chat messages and general information about students and their studies as a source of data, this thesis approach recognizes that such information may vary in structure between courses and years. While the input data is more limited in scope than in some prior works, it is also more targeted and specific.

In this thesis the choice for input data is made in favor of students' attributes, grand survey, weekly questionnaires and occasionally grades (although the latter is subject to fluctuations between courses and academic years). The part where it is planned to do deeper analysis is weekly questionnaires. More precisely analyzing text fields by converting them to a numerical value with sentiment analysis[7], which gives extra insight about the student and gives additional input data to analyze. It is planned to do it with the help of GPT[8] models, which would automate the process of a human going over weekly questionnaires manually.

1.1 Master's Thesis Goals

The primary objective are to analyze students' data and create machine learning models that would consistently yield highly accurate predictions about students' academic performances. This encompasses following metrics: such as estimating their semester scores, determining whether a student passes the exam or not and if the student passes the course or not. These predictions can provide valuable insights into a student's strengths and weaknesses, potential areas for improvement, and overall progress throughout their academic journey. An additional aim would be to develop an application that would utilize the created machine learning models by using student attributes, grand survey responses, weekly assignment results and weekly questionnaire responses as input data. The output will provide an overview of the student's current progress in the course and identify those at risk of discontinuing their studies. The creation of such an application that is supported by the data analysis, would be highly beneficial and practical, enabling the achievement of the following objectives:

1. **Early detection** - a lecturer can use it to identify students that may need extra attention early on in the course
2. **Focus on the important** - when there are hundreds of students, this application would significantly speed up analyzing and making sense of grades and weekly questionnaires, and would leave lecturer more time to focus on the content of the course
3. **Better chances to help students in need** - when finding out students who need help, lecturer can then just contact them
4. **Versatility** - because it analyzes weekly questionnaires, which are very generic, it is possible to use this application in any kind of course which uses these kinds of questionnaires

Research questions that need to be answered to reach those goals:

1. How accurately is it possible to predict academic performance with only few weeks of data?
2. How precise can bidirectional predictions be when forecasting one year academic performance using another year?
3. How are GPT sentiment analysis results connected with student's academic performances and how are they correlated with corresponding human results?

Even though this thesis focuses more on the weekly questionnaires, supplementary analysis is conducted to investigate the correlation and uncover new insights. It also tries to find correlations between Moodle log data, students' attributes, grand survey and weekly questionnaires results with respect to final grades, in order to enhance the depth and accuracy of the findings. This additional analysis can prove to be instrumental in improving the quality and efficacy of the course. By exploring the data further, it is possible to gain a deeper understanding of the underlying patterns and relationships within the information. This, in turn, can lead to more informed decision-making and a more comprehensive approach to course development.

Hypotheses about the thesis:

1. There is a strong correlation between grand survey and weekly questionnaires responses and academic performance
2. GPT models can assign sentiment analysis scores to texts that exhibit a stronger correlation with students' academic performance compared to human assessments
3. It is possible to predict if student passes the course or not with over 80% accuracy with few weeks of data

It is important to acknowledge that while universities strive to provide support and resources for all students, there may be instances where a student's personal life circumstances take precedence over their academic pursuits. These may include events such as family emergencies, health issues, financial struggles, or personal crises. In such cases, it is not always possible to "save" the student in question, or to ensure that they are able to meet their academic goals. Instead, it is crucial to approach these situations with empathy and understanding, recognizing that each student's situation is unique and that a one-size-fits-all approach may not be appropriate. By recognizing the complex interplay between personal and academic factors, universities can work towards a more compassionate and inclusive approach to education that prioritizes the well-being of all students.

1.2 Plan of action

To find answers to the research answers and hyptheses, the plan is the following:

1. **Identifying the problem** - The first stage of the master's thesis involves identifying the problem and planning the research objectives, questions, and subsequent activities by reviewing relevant scientific literature.
2. **Getting to know the course and the data** - in the second stage, the author studies the structure of the course and tries to determine which data sources would be the most advantageous to use.
3. **Data pre-processing** - in the third stage the data is processed and cleaned to ensure it is in a suitable format for analysis.
4. **Data analysis** - in the fourth stage, the author tries to create models that would predict students final grades based on weekly assignment results and questionnaires. In addition, the relationships and patterns within the input data are being investigated, using various methods.
5. **Building the application** - in the fifth stage, the author builds an application that takes in input data and makes predictions about students final grade.
6. **Conclusion** - finally, in the last stage, the author draws conclusions and offers recommendations for future work based on the research findings.

1.3 Overview of Master's Thesis

In chapter "Theory Overview", the author reviews relevant literature related to machine learning, educational data mining, and programming education. Additionally, theories that support the implementation of analysis and predictions are discussed.

In the "Data" chapter, the author presents the process of data collection, detailing the pre-processing and cleaning steps employed to get the data ready for analysis. Additionally, a closer look is taken at data distribution and prominent trends.

In chapter "Analysis", statistical methods used to analyze the data and sentiment analysis results are described. The results of the analysis and any insights gained from the data are also discussed.

In chapter "Predictions", the machine learning models developed for predicting final grades based on the weekly questionnaires and assignment results are described. The metrics of the models are also being discussed.

In chapter "Application", the development of an application to implement the machine learning models is discussed. After that, the design of the application is outlined. Finally, the description of how the application can be used is provided.

2. Theory overview

In the ensuing chapter, we shall provide a comprehensive overview pertaining to the realm of learning analytics, a historical perspective on prior research conducted within the field, an introduction to the Moodle learning management system, an elucidation of the specific course under examination, a theoretical exposition of the data analysis methodologies employed in this work, and their subsequent practical applications.

2.1 Learning analytics

Learning analytics is a growing field in education that focuses on using data to understand how students learn and perform. When creating learning analytics tools, it's crucial to consider several important factors, such as understanding the nature of knowledge, the purpose of the tools, and the ways assessments are conducted. [9, ch 1]

2.1.1 Theoretical Foundations

First, it's essential to think about what knowledge is and how it's measured in learning analytics. This means considering how we define knowledge and the methods used to evaluate it in different learning situations. Next, we need to consider the educational goals of these tools and who they are for, such as teachers, students, or administrators. We also need to be aware of the ethical concerns when designing learning analytics tools, for example making sure they don't favor certain groups of students or require specific technologies that some students might not have access to. [9, ch 1]

Lastly, the assessment process itself is important, including where and how it takes place, and when feedback is given. This involves understanding if the learning analytics tools are designed for ongoing learning (formative) or for evaluating the final outcome (summative). By carefully considering these factors, developers can create learning analytics tools that work better for everyone involved in education, including students, teachers, and school administrators. [9, ch 1]

2.1.2 Learning Analytics Cycle

The Learning Analytics Cycle consists of four main components: (1) learners generating data, (2) applying that data to develop useful charts or graphics, (3) providing feedback to learners through various actions, and (4) completing the cycle to ensure effective learning. The objective is to employ proven learning theories and gather ideas for enhancing learning analytics projects, such as accelerating feedback and involving more people. [10, p. 134-135]

The Learning Analytics Cycle emphasizes the importance of closing the loop by delivering the appropriate feedback. It relies on extensive educational research to establish a strong foundation for learning analytics and offer practical insights. The cycle begins with diverse learners and proceeds with collecting and examining data about their learning experiences. [10, p. 134-135]

The third component of the cycle converts raw data into valuable information that aids in understanding the learning process. This aspect, often the central focus of learning analytics projects, has experienced significant advancements in tools, methods, and techniques. The cycle is complete when these insights guide actions that benefit learners, such as personalized feedback or performance comparisons with others. [10, p. 134-135]

Learning analytics projects may not always include all four components, but those lacking a feedback mechanism for improving learning might be less effective. By implementing the Learning Analytics Cycle, this study aims to enhance the learning process and increase the overall impact of learning analytics initiatives. [10, p. 134-135]

2.1.3 Methodologies and Techniques

Learning analytics methodologies and techniques encompass a wide range of computational and statistical methods tailored to the unique challenges and opportunities within the educational domain. These methods focus on the development of predictive models designed to enhance students' educational experiences by identifying at-risk individuals, predicting academic performance, and improving learning outcomes [9, ch 5]. One common application, as demonstrated in various case studies, is the development and evaluation of predictive models aimed at identifying students at risk of underperforming academically or dropping out. K-fold cross-validation and various performance metrics are employed to assess these models' effectiveness in predicting student outcomes, with the insights gained informing targeted interventions to support at-risk students and improve their chances of academic success [9, ch 5].

To address the challenges and opportunities associated with implementing learning analytics methodologies and techniques, such as supporting non-computer scientists in the field, promoting community-led educational data science challenges, and engaging in second-order predictive modeling incorporating the effects of interventions, it is essential to bridge the gap between diverse scholars in the field and reconcile differing research goals, methodologies, and perspectives. This fosters a more collaborative and effective approach to learning analytics [9, ch 5].

Natural language processing (NLP) plays a crucial role in this field as a powerful tool for analyzing language due to its ubiquity and ability to provide indices related to various aspects of language [9, ch 8]. NLP has been used to identify a range of constructs, including predicting native languages of writers, evaluating essay quality, and identifying differences between spoken and written English. Despite potential drawbacks, such as relying on simplified representations and limitations in generalizing to different contexts, NLP remains powerful in providing information about individuals and their learning processes. In learning analytics, NLP can help automate understanding of learning processes and learners, inform feedback systems, and provide insights into student attitudes and motivation [9, ch 8].

Learning analytics aims to model student characteristics and skills for more effective instruction, and researchers are increasingly using large, complex data sources and various analytic techniques, including NLP, to predict and assess comprehension across contexts [9, ch 8]. However, a complete understanding of learning requires integrating multiple sources of data and various approaches to data analysis.

One aspect of learning analytics is the research on the interplay of emotions, learning, analytics, and educational data mining, which has primarily centered on individualized learning through intelligent tutoring systems, educational games, or interfaces supporting fundamental competencies [9, ch 10]. Recent work has expanded to investigate affect across broader interaction contexts, reflecting the sociocultural context of learning. Key research areas include affect-based predictors of attrition and dropout, sentiment analysis of discussion forums, classroom learning analytics, and teacher analytics. These approaches explore the influence of factors like behavioral engagement, sentiment analysis of written language, and automatic analysis of teacher instructional practices to better understand their impact on student affect and engagement [9, ch 10].

2.1.4 Ethics, Privacy, and Responsible Use

Student privacy laws have changed over time, and they're mainly about controlling who can see and use students' personal information in school records. These laws aren't keeping up with the fast-paced world of technology. Some new rules try to limit how student data is used, but they don't fully address the challenges brought up by modern data analysis in education. To make sure that data tools help everyone and are fair, people working with student data need to be open, responsible, and think ahead. For a long time, people thought that privacy rules in education were good enough, even though they didn't offer much control over personal information. However, with the rise of big data, which includes things like cloud storage and instant data transfers, it's become clear that these old rules don't do enough to protect students' privacy. [9, ch 28]

Some of the new rules try to completely ban certain types of data collection or limit how it can be used, but this can cause problems and limit the benefits of using data to help students learn. Also, many of the new laws don't cover information in colleges or online learning platforms, which means these areas are left with weaker privacy protections. Privacy and data use in education come with unique challenges. Traditional ways don't protect students' privacy well, and students can't really avoid using modern tech. These tools might accidentally discriminate against certain groups and cause students to be more careful with what they say. The shift from human decisions to algorithms raises questions about openness and responsibility. [9, ch 28]

To deal with these problems, we need to think about ethics, have clear review steps, talk with people involved, and make sure algorithms are accountable. Being transparent, checking algorithms, and allowing people to understand decisions will help build trust in data-driven education. By thinking about the bigger picture, we can make sure these tools reach their full potential. [9, ch 28]

2.2 Earlier research

In recent years, there has been a growing interest in leveraging machine learning techniques to predict student performance and dropout rates in higher education. Various researchers have explored different methodologies and data sources to enhance the accuracy and applicability of their prediction models. This section presents an overview of some notable studies in this domain, discussing their objectives, approaches, and key findings. These studies have investigated topics such as student segmentation, academic performance prediction, dropout rate estimation, and the impact of learning-related psychological factors on academic achievement. The insights from these works contribute to a better understanding of the potential benefits and challenges associated with the use of machine learning in the education sector.

Eerik Sven Puudist conducted an in-depth analysis in his bachelor's thesis titled "Implementing Data Warehouse and Machine Learning Models for Student Segmentation and Academic Performance Prediction". In one experiment, he utilized general information about the students to predict whether they would pass the course during the first week, achieving an approximate accuracy of 72%. Furthermore, he conducted another experiment by predicting the students' dropout rates and final scores throughout the weeks. The accuracy of the prediction for the dropout rate reached as high as 0.888 by the 14th study week. To evaluate the results of predicting the students' final scores, he used the metric R^2, which almost reached a value of 0.9 in the final study weeks. However, the metric scores never reached perfect value as the exam results were unknown until the last study week. [5]

Heleriin Ots employed innovative approaches in her thesis called "Predicting academic achievement based on Moodle log data and self-assessed learning-related psychological factors", where she used Moodle logs, grand survey responses, and running results as input data to classify students into three groups based on their final grades. First class were those who received a grade of 0, second class were those who received a grade of 1, 2, or 3, and third class were those who received a grade of 4 or 5. For instance, the multiclass classifier had an accuracy of 65.63% when using one month of Moodle logs and running results as input. Moreover, she endeavored to train the model on one course and then use it to predict the outcome of another course, demonstrating the model's versatility. [4]

In a bachelor's thesis entitled "Predicting Dropouts Among TUT Students: Calculating Probabilities Using Machine Learning and Displaying Results in a Web Application", Brenda Uga conducted an elaborate experiment by testing multiple machine learning algorithms to predict the likelihood of students dropping out. After careful analysis, it was found that the decision tree algorithm emerged as the most proficient among the various machine learning algorithms. Moreover, the accuracy of predicting dropouts even before the first semester based only on the general information of students was over 85%. Ultimately, Brenda Uga identified the number of credits obtained by the students as the most salient variable that exerted a significant impact on the prediction of dropout. [6]

In the research paper "Machine Learning-Based Student Grade Prediction: A Case Study," the authors attempted to predict student GPAs using a methodology akin to how movie streaming services estimate user ratings for unseen films. The study employed Collaborative Filtering, Matrix Factorization, and Restricted Boltzmann Machines for making predictions. The dataset comprised general student information, course credits, and achieved grades. Students and courses were arranged in a matrix. The paper specifically explains the approach to forecasting a student's GPA for a course they will enroll in the future. If the predicted GPA is insufficient, the course lecturer will be informed that the student needs to invest more effort into the course. [11]

At the University of Washington, a study titled "Predicting Student Dropout in Higher Education" was conducted, which analyzed students' demographic information, pre-college entry data, and extensive transcript records to estimate dropout rates. A total of 69,116 students were enrolled during the research period, with half of them used for training, validation, and testing through logistic regression, random forests, and k-nearest neighbors. The models generated binary outputs, where 0 represented students who did not obtain a degree within six years of university attendance (with four years being the standard duration), and 1 indicated the contrary. The highest accuracy rate, 66.59%, was achieved by logistic regression. [12]

2.3 Moodle

The course discussed in this paper is managed by Moodle, which is an open-source Learning Management System (LMS), it plays a crucial role in modern education by collecting large amounts of data related to various aspects of student interactions. These interactions include content consumption, completion of assessments, and communication with peers and instructors. The data obtained from these interactions can be effectively utilized as proxy indicators to measure student engagement levels, as well as predictors of their academic performance. [13, p. 180]

Moodle has garnered widespread recognition and usage across 242 countries worldwide. With an impressive user base exceeding 358 million individuals and spanning over 165,000 sites, this platform has established itself as one of the leading solutions for online learning. Boasting more than 44 million courses, Moodle's diverse offerings cater to the educational needs of a vast array of learners, solidifying its status as an indispensable tool in the realm of modern education. [14]

The examination of cross-platform and LMS-specific tools in relation to varying Moodle versions is critical, given the inconsistent availability of analytic tools. These tools primarily analyze user interactions through LMS log data, extracted and transformed from the mdl_log database table. Prior to Moodle version 2.7, log formats were inconsistent due to a lack of standardization. However, with the introduction of a new log system in version 2.7, this issue was addressed. The new system not only gathers more detailed user interaction data but also offers a standard API for enhanced log writing, reading, and overall system performance. While both log systems can coexist in Moodle 2.7 and later versions, adapting tools to the new log system's capabilities remains necessary. The focus lies in comparing learning analytics tools across different Moodle versions, rather than addressing the platform's log storage issues. These tools are analyzed based on specific categorizations. [15, p. 52]

Moodle Dashboard, the main dashboard application for Moodle, is provided as a block, allowing users to display the results of any Moodle query graphically or textually. In standard course formats, the block grants access to an additional page that exhibits the data corresponding to the specified query. Various options are available to visualize the information, including tables, plots, geospatial and map graphs, and timelines. Moodle Dashboard can display the data directly or combine with other blocks to create a complex, customizable dashboard. It boasts robust data filtering capabilities and can automatically generate data exports. Moodle Dashboard is supported up to Moodle version 2.5. [15, p. 52-53]

In addition to Moodle Dashboard, the default Moodle reporting tool can also serve as a dashboard. This tool facilitates the analysis of user interactions within the platform across different contexts, such as site, course, or activity. Reports display information on user comments, course activity, LMS event logs, live logs, and statistics about user activity and view/post actions. Further filtering of this information is possible, and at the course and activity levels, data on course and activity completion, time spent on activities, and grading information can be gathered. [15, p. 52-53]

Dashboards offer a visually rich, aggregated presentation of student and teacher activity within a learning platform, typically featuring tables and graphs with varying interactivity levels. These tools can be applied to different platforms or tailored to a specific one, focusing primarily on describing LMS activity through specific metrics. While they display relevant indicators at a glance, they generally do not provide insights into the relationships between these metrics. [15, p. 52-53]

Ad hoc tools, designed to track or analyze specific information within a concrete context, often lack flexibility and scalability. One such tool, Interactions, is a Moodle plugin that groups interaction types for later analysis. Compatible with Moodle versions 1.9 and 2.0 to 2.3, it functions as a reporting block and expands the default reporting tool's capabilities by creating a Microsoft Excel spreadsheet with two worksheets. The first worksheet is an exact replica of the log reporting tool's output, while the second processes each record and assigns it to a category within three classifications. The results are in Excel format, allowing for easy graph creation and integration with statistical analysis tools. Another ad hoc tool is a web service for assessing student performance in teamwork contexts. Based on the Comprehensive Training Model of the Teamwork Competence (CTMTC) framework, this tool extracts students' interactions from forums, cloud-based file storage services, and wikis, enabling individual student assessment and conflict detection. It utilizes the Moodle Web service layer and extracts data from Moodle logs, focusing on forum posts and threads, and offers three different view modes: forum-based, team-based, and thread-based. [15, p. 53-54]

Two tools for social network analysis, SNAPP (cross-platform) and GraphFES (Moodle exclusive), help detect disconnected students and provide information on class social interactions. GraphFES connects to both types of Moodle logs, extracting information from all message boards in a course, and generates three different graphs that are best analyzed using specialized tools such as Gephi. SNAPP, a bookmarklet that works with various platforms, builds social networks in a Java applet and offers interactive tabs for users to manipulate graphs, display values, and export data in different formats. Engagement Analytics, a Moodle plugin, gathers real-time information on student progress, offering insight into student engagement levels and providing a set of indicators and a risk-alerting algorithm to help teachers detect at-risk students and decide when to intervene. [15, p. 54-55]

VeLA (Visual eLearning Analytics) is a framework that extracts information from LMS logs using web services, providing interactive representations of the data. VeLA offers functionalities such as a semantic spiral timeline, an interactive semantic tag cloud, a social graph, and a tool to compare and establish relationships between LMS data and user activity. GISMO, a graphical interactive monitoring tool, visualizes students' activities in online courses as a Moodle plugin. It allows teachers to examine information about students, such as course attendance, material reading, or assignment submission, and provides comprehensive visualizations for the entire class, including seven different visualization types. [15, p. 55]

However, the process of interrogating the vast amount of data generated within Moodle presents several challenges. Not only is the data difficult to access and analyze, but it is also cumbersome to translate the insights gleaned from this data into actionable strategies for educators. As a result, there is a need for more efficient methods to extract, process, and implement the valuable information obtained from Moodle, in order to enhance educational outcomes and improve overall student experiences. [13, p. 180]

There have been developed learning analytics tools for Moodle. The MEAP+ was designed to address these issues by improving the information representation and providing actionable insights. The plugin enables the analysis of gradebook data, assessment submissions, login metrics, and forum interactions, as well as facilitating personalized emails to students based on these analyses. In the context of higher education, numerous institutions are adopting learning analytics to optimize learning and support decision-making processes. To close the analytics loop, student data must be understood and acted upon, with staff-facing dashboards and intervention systems being developed in response. The enhanced MEAP+ aims to contribute to this landscape by leveraging the capabilities of existing LMSs like Moodle. [13, p. 180-181]

The Moodle Engagement Analytics Plugin (MEAP) provides unit convenors and student support staff with insights into student engagement within a Moodle unit site based on login activity, assessment submission activity, and forum participation. Although these data have limitations and may not fully capture student learning, they can still offer valuable insights into engagement and predict performance. MEAP's customization features allow users to weight indicators according to their perceived importance and adjust parameters for each indicator. However, MEAP lacks the advanced functionality of other learning analytics tools, such as complex visualizations or built-in intervention systems. A design-based research approach was adopted to enhance MEAP, guided by the IRAC (information, representation, affordances for action, change) framework. The study aimed to improve MEAP's utility and impact by addressing three key questions: identifying meaningful additional information, improving data representation, and implementing affordances for action to facilitate staff interventions. [13, p. 181-182]

2.4 Course

The course under study in this research is ITI0102 Introduction to Programming. Although the course is primarily scheduled during the first semester, it is also offered later. Additionally, students who did not succeed in their first attempt may retake the course after the first semester.

To commence the course, students are provided with onboarding materials that familiarize them with the course requirements. Following this, they are required to fill in a Grand Survey which includes questions relating to their anxiety levels, self-beliefs, mini-quizzes in programming, and their expectations and perceptions about the course.

ITI0102 has a wide range of learning materials, including slides, videos, and a Discord channel, which students can use to get consultation. The course mainly focuses on teaching the Python programming language, covering topics such as variables, functions, testing, and object-oriented programming, in addition it teaches how to use Git and APIs.

Throughout the course, students are required to complete weekly assignments by solving programming problems and uploading their solutions to Moodle. The automated testing system allows students to submit their solutions multiple times and receive immediate feedback, which motivates them to continue practicing. Nevertheless, students must still defend their assignments with an assistant lecturer to obtain credit for the assignment. In addition to the weekly assignments, students are encouraged to complete weekly questionnaires in which they can give information about their current status regarding the pace, difficulty, and feelings about the course on a numeric scale, as well as express positive and negative thoughts about the course in text fields.

When students have gained sufficient points from the weekly assignments, they are eligible to take the final exam, which covers all the material in the course. The final grade is calculated based on the combination of the weekly assignments and the exam.

2.5 Data Analysis

This section serves as an introduction to the theoretical framework for the data analysis methods employed in the thesis, providing a comprehensive overview of the relevant concepts and techniques that are central to the research. Through a critical examination of the literature, this section will present a detailed exploration of the various statistical and computational models that have been developed to analyze and interpret complex data sets, thereby providing a solid foundation for the practical implementation of the chosen methods.

2.5.1 Machine learning

Machine learning is a subset of artificial intelligence that involves the use of algorithms and statistical models to enable computers to improve their performance on a specific task by learning from data. It allows systems to automatically learn and improve from experience without being explicitly programmed. Machine learning algorithms can be used for a wide range of applications, including image and speech recognition, natural language processing, predictive analytics, and autonomous systems. There are several types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning. [16, ch 1]

A training loop is an iterative process in machine learning where a model is trained on a dataset to improve its performance. During each iteration of the training loop, the model makes predictions based on the training data and the predictions are compared to the true values to calculate a loss function. The loss function measures how well the model is performing and the model's parameters are updated to minimize the loss. [16, ch 1]

Cross-validation is a technique used to assess the performance of a machine learning model. It involves dividing the dataset into k subsets and training the model k times, each time using a different subset as the validation set and the remaining subsets as the training set. Hyperparameters are parameters that are set before training a machine learning model. They control various aspects of the training process, such as the learning rate and the number of iterations. Feature selection is the process of selecting a subset of relevant features for use in building a machine learning model. Metrics are used to evaluate the performance of a machine learning model. Common metrics include accuracy, precision, recall, and F1 score. [16, ch 1]

2.5.2 Supervised machine learning

A common technique used in machine learning classification problems is supervised learning, which involves training a model to learn from labeled data, where the algorithm compares its predicted output with corrected outputs to find errors and modifies the model accordingly. The trained model can predict the label of an object based on the set of features. Supervised learning can be used in applications that predict likely events based on historical data. The tasks in supervised learning are divided into two categories: classification and regression, where the label is discrete and continuous, respectively. The supervised learning algorithm distinguishes between the observed data and the training data, where the model is trained to predict the most likely labels for a new set of samples in the testing set. [17, p. 2-5]

Classification

Classification is a subcategory of supervised learning that focuses on predicting the categorical class labels of new instances based on past observations. These class labels represent discrete, unordered group memberships, which can be understood as the distinct categories to which the instances belong. In a binary classification task, the machine learning algorithm learns a set of rules to differentiate between two possible classes, such as predicting whether a student will drop out of a course or not. By analyzing various factors, including the student's engagement, academic performance, and attendance, the classification algorithm can determine the likelihood of a student's continued participation or withdrawal from the course. [18, p. 3-4]

Two common classification methods also utilized in this thesis are Random Forests and K-nearest neighbors (KNN). Random Forests consist of multiple tree predictors, with each tree relying on a random vector sampled independently and following the same distribution across all trees. The generalization error of the forest converges to a specific limit as the number of trees increases and depends on the individual trees' strength and the correlation between them. Internal estimates monitor error, strength, and correlation, demonstrating the impact of increasing the number of features employed in the splitting process and helping gauge variable importance. [19] On the other hand, KNN operates on the principle that the labels of the nearest patterns to a target pattern provide valuable information for classification. It assigns the class label based on the majority of the K-nearest patterns in the data space, necessitating a defined similarity measure within the data space.[20, p. 14]

Regression

Regression analysis, another form of supervised learning, focuses on predicting continuous outcomes by establishing relationships between predictor variables and a continuous response variable. In this approach, the aim is to develop a model that enables accurate outcome predictions based on the explanatory variables. For instance, predicting students' final scores in a course could involve analyzing various factors such as attendance, engagement, and prior test scores. By understanding the relationships between these variables and the final score, a regression model can help anticipate a student's performance and facilitate timely interventions to improve outcomes. [18, p. 4-5]

Two common regression methods also utilized in this thesis are Linear Regression with its variants and Random Forest Regressor. Linear regression is a statistical technique used to model the relationship between dependent variables (y) and independent variables (x). It seeks to establish a linear connection between these variables, enabling estimation and testing of the model's parameters. The choice of which variables to include in the model depends on the specific problem. The independent variables can be either fixed or random. Fixed variables are under the experimenter's control, while random variables are observed and not controlled by the researcher. [21, ch 10] Another way to analyze data is through the use of the Random Forest Regressor method. This method is based on the Random Forests algorithm, which was initially designed for classification but can also be adapted for regression analysis.

2.5.3 GPT

In the following section, a detailed examination of the Generative Pre-trained Transformer (GPT)[22] architecture is provided, along with an explanation of its functionality. The discussion also covers the utilization of this advanced language model in the thesis, its importance for the research, and the potential benefits it can contribute to the study.

The GPT, created by OpenAI[1], is an advanced language model capable of producing text with human-like qualities. It generates text by considering the context of preceding text. The main part of GPT are Transformers[23], a type of semi-supervised learning, it is composed of an encoder (which receives input) and a decoder (which generates output). Featuring an attention mechanism, Transformers can process data concurrently. The overall architecture of Transformers can be seen in Figure 1 and consists of following parts:

1. **Input Embeddings** - when text is fed into the model, it first breaks the text into smaller pieces called tokens (words or parts of words). Each token is turned into a vector (a list of numbers), and the position of each token in the text is also given a numerical representation. These two sets of numbers are combined to create the input for the model.
2. **Self-Attention Mechanism** - this part helps the model understand the importance of different words in the text when processing a specific word. It has three main components: Query, Key, and Value matrices. These are created from the input numbers obtained earlier. The model calculates attention scores, which are used to weigh the importance of each word in the text.
3. **Multi-Head Attention** - GPT uses multiple parallel self-attention layers, called "heads," to capture different aspects of the text, like grammar and meaning. All the heads work together and combine their outputs to generate a final result.
4. **Feed-Forward Neural Networks** - Each layer of GPT also has a small neural

[1] OpenAI: https://openai.com/

network that processes the output from the multi-head attention. This network has two layers with a special function (ReLU) in between to help the model learn more complex patterns.

5. **Layer Normalization and Residual Connections** - these techniques are used in each layer of GPT to make training more stable and efficient. Layer normalization ensures the input features are standardized, and residual connections help keep the original input information by adding it back to the output of the layer.
6. **Output Layer** - the final layer's output is passed through another layer and a function (Softmax) that turns it into probabilities for each token in the vocabulary. The token with the highest probability is chosen as the prediction.

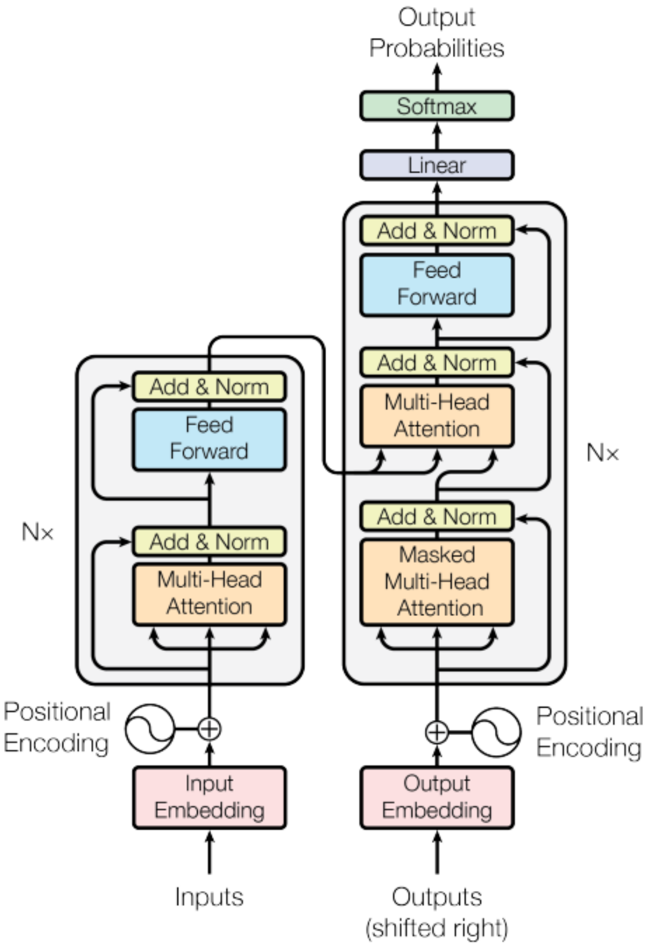


Figure 1. Transformer model architecture.

Pre-training and Fine-tuning

Initially, GPT is trained on a massive amount of text data without supervision, learning to predict the next word in a sentence. After that, it is fine-tuned for specific tasks using labeled data. This two-step process helps GPT perform well on a wide range of tasks. [24]

1. **Pre-training** - in this step, GPT is trained on a massive amount of text data from diverse sources, like websites, books, and articles. The model doesn't know anything about the specific task it will be used for during this stage. The goal here is to learn the structure, grammar, and context of the language and to capture the general knowledge hidden within the text. During pre-training, the model learns to predict the next word in a sentence, given the previous words (this is called masked language modeling). For example, if the input text is "The cat is on the ___," the model learns to predict the word "mat." GPT adjusts its internal parameters to minimize the error between its predictions and the actual words in the sentences. By doing this over a vast amount of text, the model learns various language patterns and gains a broad understanding of the language.
2. **Fine-tuning** - after the pre-training phase, GPT becomes a powerful language model but still needs to be adapted for specific tasks, like sentiment analysis, which is the use of natural language processing, text analysis, computer linguistics and biometrics to systematically identify, extract, quantify and study affective states and subjective information. [7, p. 1093] This is where fine-tuning comes in. During fine-tuning, GPT is trained on a smaller, task-specific dataset with labeled examples. The labeled data helps the model understand the desired output for the specific task.

The combination of pre-training and fine-tuning is essential for GPT's success. Pre-training helps the model learn language patterns and general knowledge, while fine-tuning adapts the model to perform well on specific tasks. This two-step process is what allows GPT to be a versatile and powerful language model, capable of handling a wide range of natural language processing tasks.

This thesis utilizes GPT to extract further insights from the underlying data by analyzing text fields gathered from both grand surveys and questionnaires completed by students.

The aim is to conduct sentiment analysis on these text fields, converting them into numerical values that represent the overall sentiment expressed. These numerical values can then be employed as supplementary data points in a suitable format for machine learning models to predict either students' dropout rates or final scores.

2.6 Application

This section presents the development of an application designed to address the challenge of high student dropout rates in university courses. The application aims to predict students' academic performance, including their semester and exam scores, as well as whether they are likely to pass or fail the course. The following subsections outline the key requirements and the architecture of the application.

2.6.1 Requirements

In order for the application to achieve practicality and usability, it is essential that it fulfills a number of key requirements:

1. **Handle serving predictions during different stages of the course** - The application should be designed to handle serving predictions at different stages of the course, such as the beginning, middle, and end of the semester. This would enable educators and administrators to intervene and provide necessary support to students who may be at risk of dropping out or failing the course.
2. **Predict students' semester score** - The application should be able to predict the final semester score of students based on their academic performance throughout the course. This would allow educators to identify students who may be struggling and provide timely interventions to help them improve their grades.
3. **Predict students' exam score** - The application should be able to predict students' exam scores based on their academic performance throughout the semester. This would allow educators to identify students who may need additional support and help them prepare for upcoming exams.
4. **Predict if a student passes or not** - The application should be able to predict whether a student is likely to pass or fail the course based on their overall academic performance throughout the semester. This would enable educators to identify

students who may be at risk of failing the course and provide them with the necessary support to improve their grades.

5. **Give predictions in a human-readable format** - The application should be able to provide predictions in a human-readable format, such as a CSV file, to enable educators to easily interpret the data and take appropriate actions to support students.

2.6.2 Architecture

The architecture theory for this application is centered around the use of Docker and Docker Compose to facilitate containerization of the application and its dependencies. This allows for seamless deployment of the application across multiple platforms and environments. SSH is also utilized to enable secure remote access to the application and facilitate remote management and updates.

In addition, WatchTower is implemented to enable automated updates of the Docker containers, ensuring that the application is always up-to-date with the latest versions of its dependencies. The GitLab CI/CD pipeline is also utilized to automate the deployment and testing of the application code. Python, along with Pandas and NumPy, is utilized to perform the data analysis required for the application to generate predictions about students' dropout rates and final scores. The trained machine learning models are then saved on the virtual machine to enable quick and easy access to the models for future use.

Overall, the architecture for this application leverages a range of cutting-edge technologies and frameworks to ensure that the application is robust, scalable, and secure. By utilizing containerization, automation, and remote management technologies, the application can be deployed across multiple platforms with ease, while Python and its associated data analysis libraries enable efficient and effective analysis of student performance data. Ultimately, this architecture ensures that the application is optimized for performance, security, and scalability, enabling educators and administrators to make informed decisions and provide timely support to students throughout their academic journey.

3. Data

In this study, data was gathered from the ITI0102 Introduction To Programming Moodle courses as CSV files during the 2021 and 2022 academic years. This data encompassed a variety of student characteristics, including demographics and academic details such as prior programming experience.

Moreover, the dataset included students' feelings before and during the course, gathered through surveys and weekly questionnaires. The goal of this information was to provide a better understanding of students' motivation, confidence, and worries related to the course, highlighting the impact of emotions on their academic success.

Apart from the originally acquired data, new insights could be generated by combining and analyzing existing data. Through extracting additional information, a more extensive understanding of the factors influencing students' performance in programming courses can be achieved.

3.1 Data protection

To combine data from various CSV files, it was necessary to assign names to individual records. However, after the integration of data, both the name and ID number were removed to protect students' privacy. This ensured that when analyzing the data or making predictions, it was impossible to associate any particular record with a particular student.

Furthermore, to maintain anonymity, the author and supervisor were the only individuals who were granted access to the students' names. This policy helped to prevent any accidental or intentional misuse of personal information.

Additionally, the data used for analysis was never present in TalTech GitLab or any other external platform. This was done to ensure that there was no risk of unauthorized access or data breaches. By adopting these measures, the privacy of students' personal information was protected while also enabling the author to conduct a thorough and reliable analysis.

3.2 Detailed Course Structure

The course lasts for 16 weeks and is structured with a comprehensive grading system that allows for a maximum semester score of 600, which may increase if students choose to complete extra exercises. The various components of the course structure are as follows:

1. **Weekly Exercises (EX)** - these exercises take place from weeks 1 to 15 and contribute significantly to the overall point tally. Each exercise awards 32 points, except for the 8th exercise, which grants 40 points. To address inconsistencies, the points are scaled from 0 to 100.
2. **Smaller Test (TK)** - this test provide 5 points, with a minimum of 2.5 points required to pass.
3. **Larger Test (KT)** - this test offer 20 points, with a passing threshold of 10 points.
4. **Grand Survey** - taken at the start of the course and participation in the survey earns students 2 points.
5. **Weekly Questionnaires** - students gain 0.5 points for completing these questionnaires.
6. **Weekly Quizzes** - these quizzes award up to 0.5 points each.
7. **Smaller Bonus Exercises (MX)** - these exercises provide fewer points and require students to fully solve the exercise for additional practice and point allocation.
8. **Extra Challenging Exercises (XP)** - these tasks demand independent study of material beyond the course's scope, offering an opportunity for students to push their limits.
9. **Additional Bonus Tasks** - supplementary tasks, such as WAT, provide extra opportunities for students to earn points and deepen their understanding.

The final exam constitutes 400 points. Consequently, the maximum possible final score is 1000 or more, depending on the completion of extra exercises. This comprehensive structure ensures a well-rounded learning experience, accommodating various student interests and skill levels.

Data sources used for analyzing and predictions:

- General information about students
- Grand survey
- Students' weekly exercises results
- Weekly questionnaires completed by students

3.2.1 General information about students

The data structure for general student information captures various aspects of their educational background and personal demographics. This includes the following features:

1. **Study Form** - Categorized as daily study, session study, micro-degree, or voluntary outside of school. This information allows for a better understanding of the student's learning environment and commitment.
2. **Micro-Degree Program** - Indicates whether a student is actively pursuing a micro-degree, which provides insights into their academic goals and interests.
3. **Study Program** - A code representing the student's chosen field of study, primarily focused on IT-related programs. This information helps to identify the specific knowledge and skills being acquired by each student.
4. **Age and Gender** - These demographic details are derived from ID numbers when available. In cases where ID numbers are not provided, it is not possible to obtain this information. Including age and gender in the data structure allows for a more comprehensive analysis of student backgrounds and potential trends in academic performance.

By incorporating these components into the data structure, a more nuanced understanding of students' academic and personal profiles can be achieved, ultimately facilitating more effective analysis and predictive modeling.

3.2.2 Grand Survey

At the beginning of the course, students were provided with a comprehensive form comprising of 86 questions, covering various topics such as their beliefs, motivations, study techniques, susceptibility to burnout, problem-solving abilities, and prior experience in programming. The grand survey aimed to help lecturer gain an understanding of each student's unique learning style and abilities, which could, in turn, be used to enhance their learning experience.

Most of the questions in the survey were structured in a way that allowed students to select from six different options, which were assigned a numeric value ranging from 1 (not agreeing at all) to 6 (completely agreeing). Additionally, some questions had text fields where students could provide more detailed responses. To analyze text field responses, GPT was employed to convert the text-based answers into numeric values, it can be seen in more detail at Section 4.3. The complete set of survey questions is available in Appendix 7.1.

3.2.3 Students' weekly exercises results

The course consists of 15 assignments, one for each week, with the exception of the final study week, which did not have any assigned work. Students were expected to complete the weekly assignments and submit them before the deadline. The submission process allowed students to submit their work as many times as they wished before the deadline, but to redeem the points, students also had to defend their submissions. This process encouraged students to improve their work continuously and gave them an opportunity to learn from their mistakes. In addition to the weekly assignments, there was also a smaller test (`TK`) during the 5th week of the course, a larger test (`KT`) in the 10th week, but these two could be done later as well.

To get the notion of weekly exercises, the entirety of the first week weekly assignment can be seen in Appendix 7.1.

3.2.4 Weekly questionnaires

Weekly questionnaires were conducted weekly, the response format was either one choice from multiple options or text field. The choice questions can be easily mapped to numeric values, but text field processing needs different approach. The questionnaire had following questions:

1. **(Feeling) What was your mood last week regarding the course?** - response was one choice from 5 options
2. **Did you learn something useful about programming last week?** - response was one choice from 4 options
3. **How would you describe the pace of the course?** - response was one choice from 3 options
4. **How much time did you spend on the previous week's course assignments (in hours)?** - response was in text field format.
5. **How would you rate the previous week's assignment on a 10 point scale?** - response was one choice from 10 options.
6. **Positive thoughts and emotions regarding the previous week's topic and assignment** - response was in text field format.
7. **Negative thoughts and emotions regarding the previous week's topic and assignment** - response was in text field format.

In the final processed version, numeric values are assigned to questions 1, 2, 3, and 5 based on the chosen options. The 4th question has been excluded as it is a free-form text field, allowing any information to be entered, making it impractical for use. For example, there are many responses in alphanumeric format, or even responses which indicate that some students spent more hours last week on the course than there are hours in a week. The questions 6 and 7 were suitable for sentiment analysis with the help of GPT, to put numeric values from 0 to 10 for each text field, so they are in a suitable format for subsequent data processing steps. It can be seen in more detail at Section 4.2.

In the final stages of data preparation, the collected information was merged into a unified dataset using the Pandas library, a Python-based tool for data manipulation. To streamline the analysis and prediction process, only the most relevant columns were retained in the dataset.

Additional columns were introduced, derived from students' semester, exam, and final course scores. These new columns indicated whether a student passed the semester, exam, or the entire course, offering a more transparent perspective on their overall performance. To maintain data consistency, data field types were converted to their appropriate formats. When empty cells were encountered, a value of -1 was assigned to facilitate smooth processing and analysis of the dataset. This enhanced and prepared dataset can now be employed for thorough investigation and predictive modeling.

3.3 Available but not fully usable data

In this section, we explore available data that may hold potential value but is currently not being utilized or being utilized only partially for various reasons. The following data sources are worth looking into more deeply for future analysis:

- **Time Spent on the Course in Moodle** - the Moodle platform offers a summary of the duration students engage with the course, as well as the average number of connections per day during a designated time frame. Furthermore, it enables customization of time limits between clicks to ascertain session continuity. However, this data is only available for the 2022 course iteration and is accessible via the user interface in a usable format. Additionally, the tool is not ideally suited for large-scale data analysis, as it requires manual fetching of statistics for each time period.
- **Days Inactive** - although potentially valuable for gauging student engagement, the data on days inactive is exclusively presented in cumulative form, restricting its usefulness for more detailed examination.
- **Weekly Questionnaires Data for 2021** - due to the anonymous collection of these results, they cannot be attributed to individual students, rendering their integration into the current analysis unattainable.

By recognizing these data sources, future research may devise methods to surmount the limitations and integrate them into more refined analyses, ultimately enhancing our comprehension of the factors that contribute to student success in programming courses.

3.4 Overview of 2021 and 2022 students data

In this study, a subset of students, which still amounted to over 95% of the original data, was carefully selected to ensure the accuracy and reliability of the results. The initial dataset included several students who, although enrolled, did not participate in the course at all and received a final score of zero points. These students were systematically filtered out of the analysis to maintain the integrity of the data and obtain more insightful conclusions. This decision was based on a set of well-founded reasons, which are enumerated below:

1. **Improve model accuracy** - Including students who did not participate may introduce noise in data, which could lead to a less accurate predictive model. By focusing on students who were actively engaged in the course, the model will be better equipped to identify patterns and trends in the data that are relevant to student performance.
2. **Better representation of the target population** - Since the goal is to predict the performance of students who will actively participate in a course, then the training data should reflect that population. Including students who did not participate at all may distort the relationship between the input features and the target variable (e.g., course performance).
3. **Prevent overfitting** - Including non-participating students may introduce outliers in the data, which could cause the model to overfit to these specific cases. By filtering out students who did not participate, the likelihood of overfitting will be reduced and model's ability to generalize to new data will be improved.

In the 2021 edition of the course, there were a total of 367 students. The subsequent 2022 edition of the course saw an increase in participation, with 492 students attending. This section presents a summary of student data from the 2021 and 2022 editions of the course, highlighting patterns in demographics, educational backgrounds, and final grades. The information serves as a valuable resource for examining and comprehending the prevailing tendencies during these specific years.

3.4.1 Demographics

This section compares the demographics of students in the 2021 and 2022 course editions, focusing on trends and key differences in gender distribution and age ranges.

Referenced in Figure 2, in the 2021 edition, there were 251 male students and 116 female students. In the 2022 edition, the number of male students increased to 317, while the number of female students rose to 175. These changes indicate a substantial increase in female participation between the two editions, although the proportion of male students remained higher in both years.

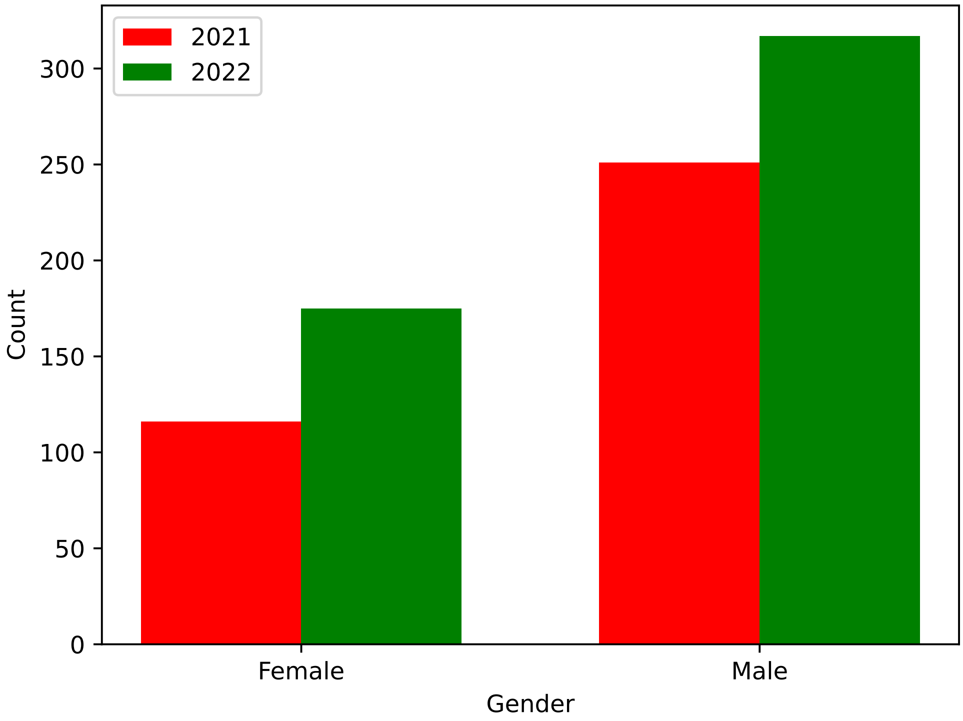


Figure 2. Gender distribution.

As illustrated in Figure 3, compared to the 2021 edition, the 2022 edition, exhibited some differences in age distribution. The number of students aged 18-20 increased to 202, and the number of students aged 21-24 rose to 99. The other age ranges remained relatively stable.

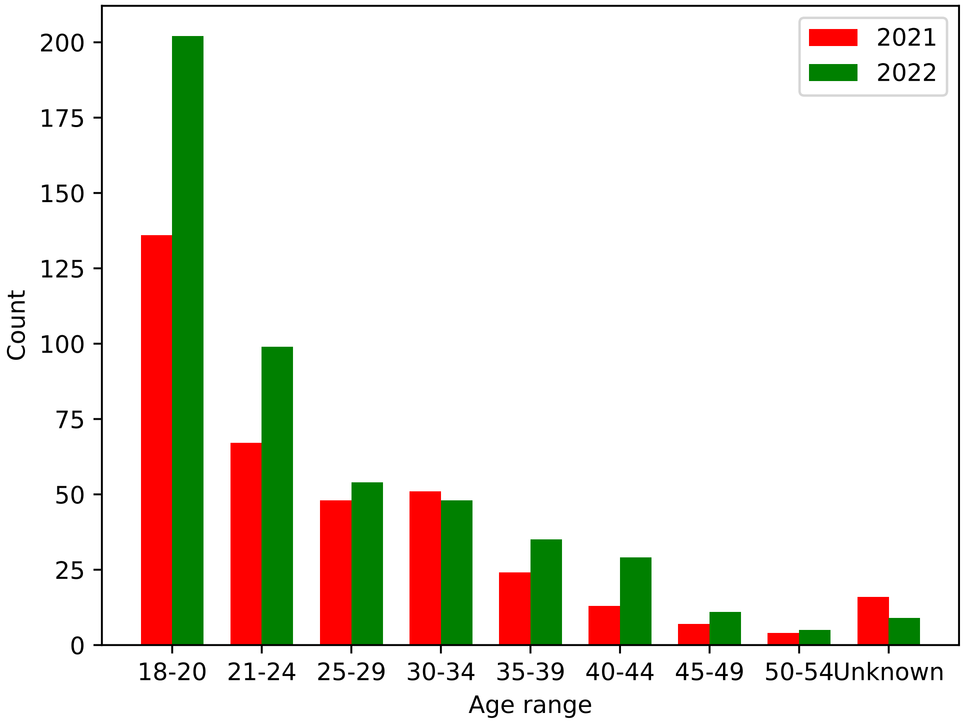


Figure 3. Age ranges.

In summary, the comparison of demographics between the 2021 and 2022 course editions reveals an increase in female representation and a general trend towards younger participants. These changes in demographic composition may provide valuable insights for tailoring course content and support systems to better serve the evolving student population.

3.4.2 Educational Background

This section compares the educational backgrounds of students in the 2021 and 2022 course editions, focusing on trends and key differences in study form, study program, and micro degree enrollment.

As depicted in Figure 4, 241 students participated in the 2021 edition in daily study, while 58 students engaged in session study. In the 2022 edition, the number of students participating in daily study increased to 325, whereas the number of students in session study decreased slightly to 49. These changes indicate a growing preference for daily study among students.

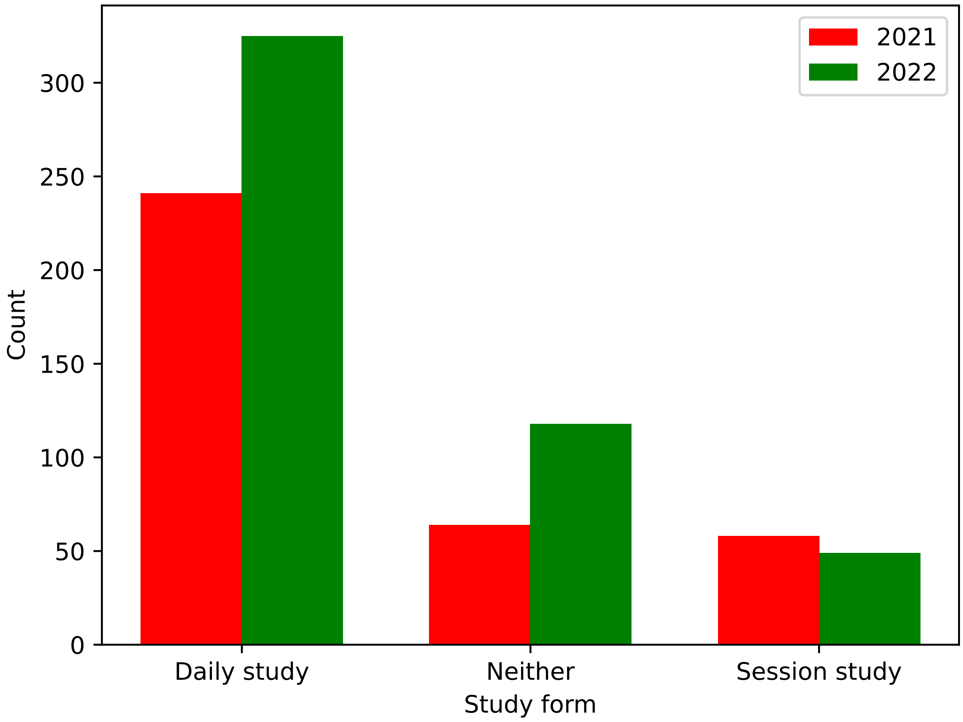


Figure 4. Study forms.

As shown in Figure 5, in 2021, the most common study programs were IADB with 166 students, followed by IAIB with 59 students, TAF with 58 students, and IAAB with 50 students. Other study programs had fewer students enrolled. In 2022, the most common study programs were IADB with 186 students, followed by TAF with 109 students, IAIB with 85 students, and IAAB with 69 students. Other study programs experienced minor fluctuations in enrollment numbers.

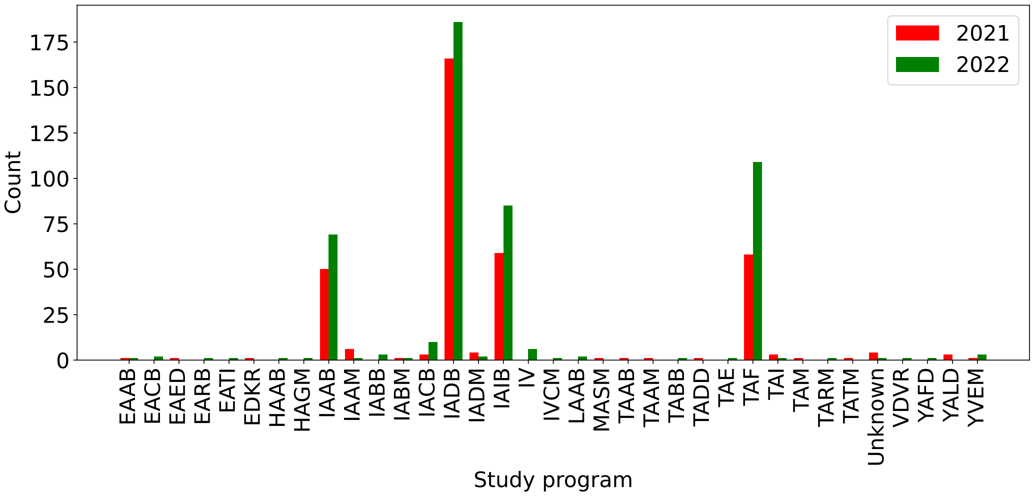


Figure 5. Study programs.

Referenced in Figure 6, in the 2021 edition, 339 students were not enrolled in a micro degree program, while 24 students were enrolled in one. In the 2022 edition, the number of students not enrolled in a micro degree program rose to 413 students, while the number of students enrolled in a micro degree program increased to 79. This increase in micro degree enrollment indicates a growing interest in specialized educational paths among students.

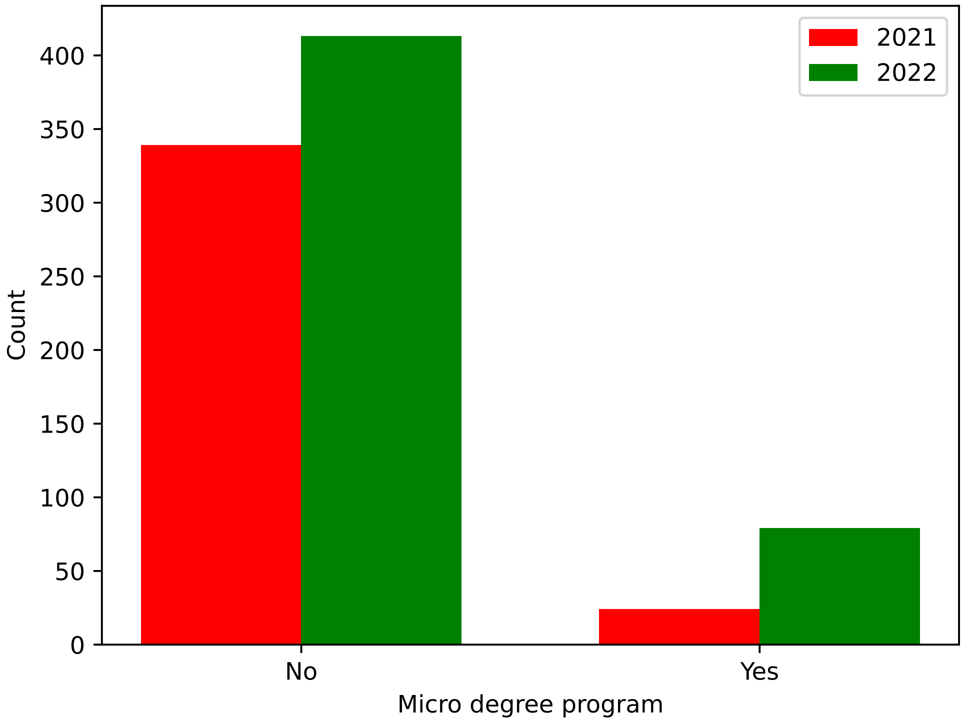


Figure 6. Micro degree programs.

In summary, the comparison of educational backgrounds between the 2021 and 2022 course editions reveals a preference for daily study, a consistent popularity of certain study programs, and an increase in micro degree enrollment. Understanding these trends can help inform the development of course materials and support structures tailored to the needs of students with diverse educational backgrounds.

3.4.3 Academic performances

As shown in Figure 7, in the 2021 edition of the course, the pass rate was 61.04%. The maximum semester score achieved by a student was 623.8, with an average of 321.08. The highest exam score recorded was 400.0, with an average of 178.47. Lastly, the maximum final score was 1023.8, and the average final score was 499.56.

On the other hand, the 2022 edition witnessed a higher pass rate of 65.24%, which may be attributed to changes in internal grading practices. The top semester score increased to 667.82, and the average semester score rose to 334.55. The maximum exam score remained at 400.0, but the average exam score improved to 182.74. The highest final score climbed to 1057.82, while the average final score reached 517.3.

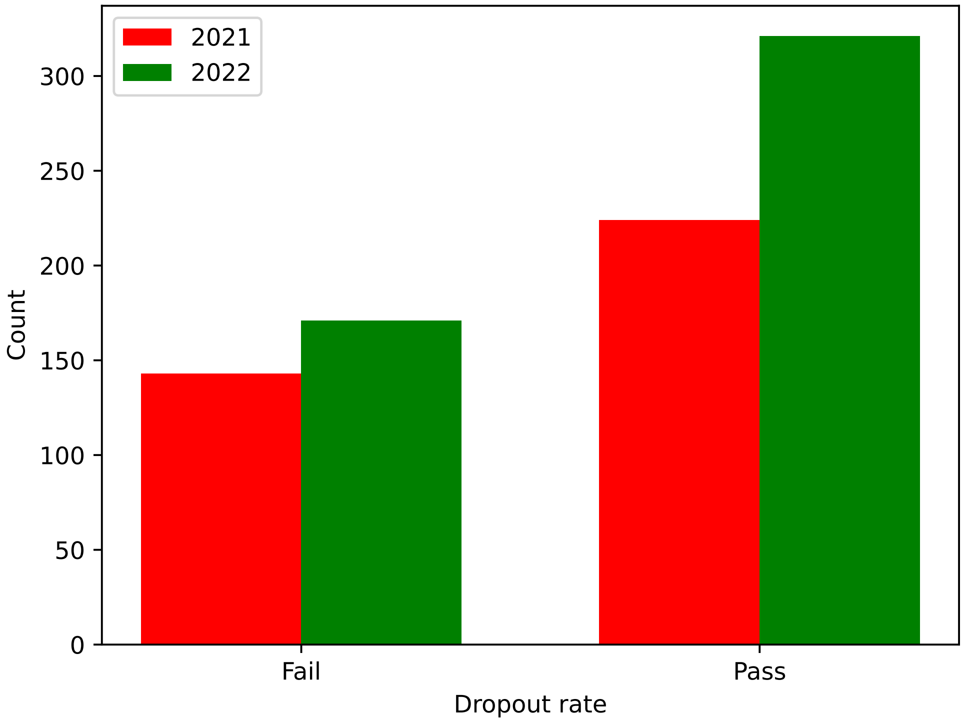


Figure 7. Dropout rates.

In order to distinguish between students who attempted the exam and received a final grade of 0 and those who did not take the exam at all, separate categories were assigned to each group, with 0 representing a grade of zero for those who took the exam and -1 for those who did not. It was observed that a considerable number of students did not attempt the exam; however, those who did were more likely to pass the course.

These grade distribution is visualized in Figure 8, comparing the grade distribution as percentages between the 2021 and 2022 editions, which had 367 and 492 students, respectively, several differences can be observed. The 2022 edition had a slightly lower proportion of students who did not attempt the exam (represented by -1), which means that there were less dropouts during the semester. When examining the grades of students who did take the exam, there was a noticeable increase in the proportion of students receiving higher grades in the 2022 edition, particularly in grades 2 and 4. This suggests an overall improvement in student performance in the 2022 edition, despite the larger number of enrolled students.

This highlights the importance of closely monitoring student progress throughout the course to ensure they do not fall behind. Implementing effective support mechanisms and early interventions can help identify and address challenges, ultimately contributing to improved student outcomes and higher pass rates.

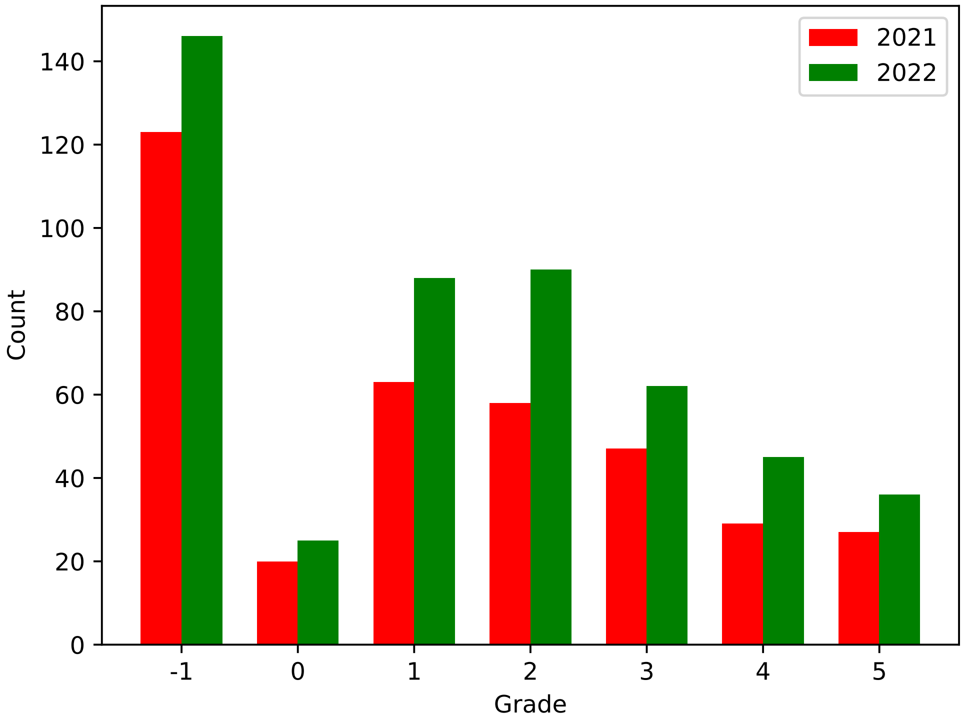


Figure 8. Grade distribution, -1 meaning grade of 0, but the student did not take the exam.

3.4.4 Grand survey and weekly questionnaires

In 2021, 66.49% of students took part in the grand survey, and among them, 70.08% successfully completed the course. In contrast, in 2022, the participation rate rose to 87.60%, with 68.91% of those students passing the course. The data demonstrates that participating in the grand survey positively correlates with an increased probability of course completion. Notably, the passing rate remains relatively stable—only experiencing a minor decrease—despite the significant increase in the number of students participating in the survey. This outcome suggests that the survey's impact on student success remains consistent even as participation grows.

The 2022 statistics for weekly questionnaires, which were not anonymized unlike the 2021 responses, show the following results: 94.51% of students completed at least one weekly questionnaire, while 21.75% took all weekly questionnaires. Of those who participated in at least one questionnaire, 66.88% passed the course, whereas a remarkable 88.79% of students who completed all weekly questionnaires successfully passed. This data analysis suggests that consistent engagement in weekly questionnaires positively correlates with a higher likelihood of course completion, highlighting the importance of regular participation for academic success.

4. Sentiment analysis and correlations

In this chapter, a comprehensive examination of the students' feedback is conducted through a multi-faceted approach. This chapter consists of three main sections, each focusing on distinct aspects of the feedback data. First, the weekly questionnaires are explored, where a sentiment analysis is performed using the GPT model, and the model-generated scores are compared with human-assigned labels to assess the model's accuracy and effectiveness. In the second section, the grand survey is analyzed, employing the GPT model to perform sentiment analysis and examine the overall sentiment trends among the student population. Finally, correlations between various factors, such as sentiment scores, academic performance, and other relevant variables, are investigated to uncover potential patterns and relationships that may offer valuable insights into the students' experiences and perspectives. To assess these relationships, Pearson correlation was utilized, providing a measure of the linear association between the variables and helping to identify the strength and direction of their relationship. [25]

4.1 Exploration of different approaches

In the early stages of sentiment analysis I started with weekly questionnaires, the initial approach I considered involved labeling positive and negative text fields, fine-tuning a language model with the corresponding data and labels, and then validating the model using previously unseen data.

To begin with, I attempted to manually assign sentiment scores to the texts. I utilized a model from HuggingFace[1], a renowned U.S.-based firm specializing in the development of machine learning resources for building applications. The company is primarily known for its transformer library tailored for natural language processing tasks, as well as its platform that enables users to share machine learning models and datasets. I employed the XLM-RoBERTa model for sentiment analysis, which had also been trained on the Estonian language, encompassing 843 million tokens and a total text size of 6.1 GB [26].

[1] HuggingFace: https://huggingface.co/

I merged positive and negative text fields from weekly questionnaires and assigned labels to them: 0 for neutral, 1 for negative, and 2 for positive. Subsequently, I attempted to fine-tune the model using this data and predict labels on unseen data. However, this approach proved to be impractical - extensive manual labor was required for labeling the texts, and the accuracy ranged between 45% and 60%, which was essentially equivalent to random guessing. Moreover, there was no discernible correlation between the text sentiment scores and students' academic performance.

Fine-tuning more powerful language models using specific prompts proved to be a faster and more predictable alternative, as it did not necessitate providing an additional dataset. This approach was also less complex and time-consuming. I used GPT-3, in addition during the development of this project, more advanced options like GPT-3.5 and GPT-4 have become available. These models are highly adept at handling such tasks, rendering it unnecessary to train a model from scratch.

4.2 Weekly questionnaires

To transform both negative and positive texts into sentiment scores, I prompted the GPT model to provide sentiment ratings on a scale of 0-10. I selected this particular scale as it conveys more detailed information about the text compared to a 0-2 scale. Additionally, when experimenting with a 0-20 scale, the results seemed less precise. On a 0-100 scale, the scores tended to cluster around multiples of 10, such as 0, 10, 20, 30, 40, and so on, which reduced the granularity of the sentiment analysis and gave basically the same amount of diversity as 0-10 scale.

My aim was to assign a label which reflects the degree of positivity in both positive and negative texts. This would facilitate linearity when utilizing the data for predictive purposes. Determining the optimal approach for sentiment analysis is a challenging endeavor. I experimented with providing text fields to the model both separately and together. The results were more reliable when both positive and negative texts were submitted within the same prompt, ensuring that the context of both texts was taken into account, as they may be interconnected in some way.

If the text fields were labeled in isolated contexts, the following scenario could occur: Student 1 leaves both fields empty, resulting in neutral scores: 5 + 5 = 10. Student 2 leaves the first field empty but submits a highly negative text in the second field: 5 + 2 = 7. In this case, it appears that the gap in positive sentiment between the two students should be greater than 3. Therefore, when assigning labels to text fields in the same context, the second student's combined score would likely be lower than 7.

In the scenario where both text fields are considered together in the same context, the scores might be adjusted to better reflect the difference in sentiment between the two students. Here is an example of how the scores could be modified: Student 1 leaves both fields empty, resulting in neutral scores: 5 + 5 = 10. Student 2 leaves the first field empty but submits a highly negative text in the second field. Since the model now takes into account the context of both fields, it may assign a lower score for the empty positive field and a more extreme negative score for the second field: 4 + 1 = 5.

With this adjustment, the gap in positive sentiment between the two students becomes 5, which more accurately reflects the difference in their feedback. By considering the context of both fields in the same prompt, the model can better gauge the sentiment difference between the students and provide more meaningful predictions and insights.

The specific model I utilized for this task was `text-davinci-003` [27] through the API, with the following parameters[28]:

- `temperature=0`: higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.
- `top_p=1.0`: an alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered.
- `frequency_penalty=0.0`: number between -2.0 and 2.0. Positive values penalize new tokens based on their existing frequency in the text so far, decreasing the model's likelihood to repeat the same line verbatim.
- `presence_penalty=0.0`: number between -2.0 and 2.0. Positive values penalize new tokens based on whether they appear in the text so far, increasing the model's likelihood to talk about new topics.

In the development process, I experimented with various prompt structures to effectively capture the sentiment in student weekly feedback texts. The primary objective was to account for instances where the negative text field indicated "nothing" or a lack of negative sentiment, which should be interpreted as a highly positive sentiment.

The final prompt used for sentiment analysis is as follows, with `<POSITIVE TEXT>` replaced by the positive text field and `<NEGATIVE TEXT>` replaced by the negative text field:

Classify the sentiment in these student weekly feedback Estonian texts into scores between 0 and 10, where 0 is very negative and 10 is very positive. The first text describes what is positive about the school week, and the second text describes what is negative about the school week. If the second text implies that there is nothing to add, it is very positive. If the second text implies that there are issues regarding school work, it is very negative. 1. `<POSITIVE_TEXT>` *2.* `<NEGATIVE_TEXT>`*. Provide the sentence sentiment ratings only as integers, separated with newlines.*

This prompt structure instructs the model to evaluate the sentiment of two separate student feedback texts: the first one highlighting the positive aspects of the school week, and the second one detailing the negative aspects. The model is then asked to classify the sentiment on a scale of 0 to 10, with 0 representing very negative and 10 representing very positive. The prompt explicitly states that if the second text implies no negative sentiment or issues concerning school work, it should be considered highly positive. The model is instructed to provide sentiment ratings as integers, separated by newlines for easier response parsing. By incorporating these instructions, the prompt effectively addresses the unique contextual aspects of the feedback texts, allowing for a more accurate sentiment analysis.

Table 1 presents an example of sentiment analysis performed on two students' weekly feedback using the GPT model. It showcases the original feedback texts in Estonian, their English translations, and the corresponding sentiment scores assigned by the GPT model. It is important to note that the model's input was the Estonian text, not the English translations.

Positive text in Estonian	**Positive text in English**	**Negative text in Estonian**	**Negative text in English**	**GPT pos**	**GPT neg**
Aitäh esimese õppenädala eest! Aine tempo on paras ning väljakutseid on piisavalt :)	Thank you for the first week of study! The pace of the subject is good and there are enough challenges :)	Ei ole	None	10	10
sain lõpuks tehtud.	I finally got it done.	Üsna masendav on see, et tegelikult võttis kõik väga palju aega.	It's quite depressing that everything actually took a very long time.	8	2

Table 1. Example of 2 students' weekly feedback on week 1 with GPT assigned sentiment scores.

In the first example, the student expressed gratitude for the first week of study and mentioned that the subject's pace and challenges were appropriate. The GPT model assigned a sentiment score of 10 for the positive text, indicating a highly positive sentiment. The negative text field contained "Ei ole," which translates to "None," implying the absence of any negative sentiment. The model correctly assigned a score of 10 for the negative text, reflecting a highly positive sentiment.

In the second example, the student mentioned completing a task, which the model recognized as a moderately positive sentiment, resulting in a score of 8. The negative text, however, described a sense of depression due to the time-consuming nature of the work. The model accurately captured the negative sentiment and assigned a score of 2, reflecting a predominantly negative sentiment.

These examples demonstrate the effectiveness of the GPT model in analyzing sentiment from students' weekly feedback texts in Estonian and assigning sentiment scores that accurately reflect the underlying emotions.

4.2.1 Comparing Human and GPT Sentiment Analysis Labeling

To evaluate the accuracy and validity of the sentiment scores assigned by GPT, a subset of 94 rows featuring positive and negative text fields from the first week of weekly questionnaires was chosen. These text fields were then compared with their corresponding human-assigned labels. The human responsible for labeling in this experiment is a lecturer involved in the course that is under examination in this thesis, ensuring the labels are as realistic as possible due to the lecturer's comprehensive understanding of the text fields' context. Moreover, correlations between sentiment scores and semester scores were determined, as semester scores serve as a critical metric for monitoring academic performance.

The human-labeled positive text field sentiment analysis scores displayed a moderate correlation of 0.25 with semester scores, while the human-labeled negative text field sentiment analysis scores showed a weak correlation of 0.1, as referenced in Table 2.

Correlation	**Positive Text Fields**	**Negative Text Fields**
Semester Score vs Human	0.25	0.1

Table 2. Human Correlations.

GPT-3 Evaluation

For this evaluation `text-davinci-003`, a model from the GPT-3 family was used. The correlations between GPT-3 and human-labeled sentiment analysis scores were assessed for both positive text fields, as seen in Figure 9 and negative text fields, which is visualized in Figure 10. The GPT-3 positive text field sentiment analysis scores showed a low correlation of 0.06 with the human-labeled scores as presented in Table 3. In contrast, the GPT-3 negative text field sentiment analysis scores demonstrated a strong correlation of 0.82 with the human-labeled scores.

Furthermore, the correlations between semester scores and various sentiment analysis scores were evaluated. Interestingly, the GPT-3 positive text field sentiment analysis scores exhibited a negative correlation of -0.05 with semester scores, which indicates a potential inverse relationship. Lastly, the GPT-3 negative text field sentiment analysis scores revealed a weak positive correlation of 0.04 with semester scores.

Correlation	**Positive Text Fields**	**Negative Text Fields**
GPT-3 vs Human	0.06	0.82
Semester Score vs GPT-3	-0.05	0.04

Table 3. GPT-3 Correlations.

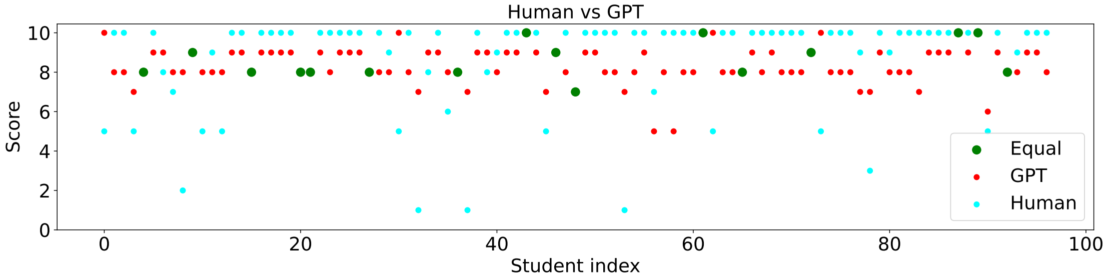


Figure 9. Scatter plot of human vs GPT-3 positive text field sentiment analysis scores.

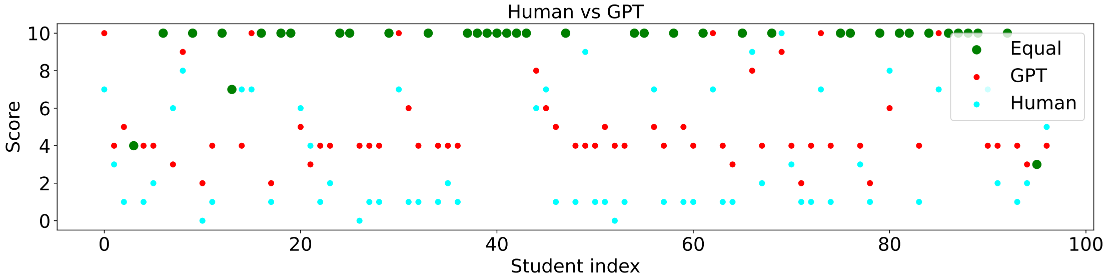


Figure 10. Scatter plot of human vs GPT-3 negative text field sentiment analysis scores.

These results suggest that the GPT-3 model performs well in capturing the sentiment of negative text fields, closely aligning with human-assigned labels. However, the model's performance in identifying the sentiment of positive text fields is less reliable, as evidenced by the low correlation with human labels.

In Table 4, which showcases the largest positive sentiment score differences between human and GPT-3, two cases stand out. The first case, discussing universal tasks and the use of English, has a human score of 1 and a GPT-3 score of 8. GPT-3 possibly sees using English positively, while the human scorer notes the neutral tone. The second case involves initial programming struggles and expected future difficulties. GPT-3 scores it at 8, and the human at 1, as GPT-3 might view overcoming challenges positively, while the human scorer perceives them as negative or neutral.

Estonian	English	Human score	GPT score	Difference
Universaalsed ulesanded mida peaks igaksuks vahemalt proovima. Dokustaat kui ka opetamine/terminoloogia voiks ikkagist inglise keeles olla.	Universal tasks that everyone should try from a distance. The document as well as the teaching/terminology could still be in English.	1	8	+7
Esimesel nädalal oli suurem pusimine just programmide töölesaamisega, mitte nii väga ülesannete lahendamisega. Aga eks need juba järgmine nädal raskemad ole.	In the first week, the main focus was on getting the programs working, not so much on solving tasks. But I guess they will be harder next week.	1	8	+7

Table 4. GPT-3 and human largest positive sentiment score differences.

For the largest differences in negative sentiment scores, which can be seen in Table 5, the first case involves the text "same thing," where the human scorer assigns a score of 0, indicating no negative sentiment, while GPT-3 assigns a score of 7. It is possible that GPT-3 inferred a negative sentiment from the repetition or lack of variation in the text, while the human scorer did not perceive any negativity. In the second case, the text describes initial confusion with a poem exercise, but eventually understanding it. The human scorer assigns a negative sentiment score of 7, while GPT-3 assigns a score of 1. This discrepancy may be due to GPT-3 focusing on the eventual understanding of the exercise as a positive outcome, while the human scorer took into account the initial confusion as a negative aspect.

Estonian	English	Human score	GPT score	Difference
sama asi	same thing	0	7	+7
Poeemi ül oli algul natuke segane aga sain lõpuks aru.	The poem ex was a bit confusing at first, but I finally understood.	7	1	-6

Table 5. GPT-3 and human largest negative sentiment score differences.

Table 9 presents cases where GPT-3 and the human scorer agreed on positive sentiment scores, the texts generally described positive experiences or learning aspects, such as finding Python syntax intuitive or enjoying a coin challenge. Both GPT-3 and the human scorer assigned high positive sentiment scores, indicating agreement on the positive nature of the experiences described.

Estonian	English	Human score	GPT score	Difference
Pythoni süntaks veidi harjumatu, aga tundub, ett üsna intuitiivne tegelikult.	The Python syntax is a bit unfamiliar, but it seems quite intuitive actually.	8	8	0
Väga meeldis müntide ülesanne!	Really liked the coin challenge!	10	10	0

Table 6. GPT-3 and human positive sentiment score agreements.

Similarly, in cases where GPT-3 and the human scorer agreed on negative sentiment scores, the texts mentioned the absence of negative emotions or experiences, as seen in Table 7. Both GPT-3 and the human scorer assigned high scores (10), indicating that they both perceived the lack of negativity as a good thing.

Estonian	English	Human score	GPT score	Difference
Hetkel ei ole negatiivseid emotsioone, võib olla järgmine nädal.	There are no negative emotions at the moment, maybe next week.	10	10	0
Pole midagi halba öelda. Minu jaoks oli selle nädala teema pigem juba selgete asjade kordamine, seega kõik arusaadav.	Nothing bad to say. For me, this week's theme was rather a repetition of already clear things, so everything is understandable.	10	10	0

Table 7. GPT-3 and human negative sentiment score agreements.

These examples demonstrate the challenges in assessing sentiment, as different aspects of the text can be emphasized or interpreted differently. While GPT-3 is a powerful language model, it is essential to consider that it may not always align with human perception when it comes to sentiment analysis.

GPT-4 Evaluation

In this evaluation, the model employed is `gpt-4`[29] which, as name suggests, is from the GPT-4 family. For `gpt-4`, it was not possible to specify extra parameters but the same prompt was used as was with `text-davinci-003`. The correlations between GPT-4 and human-labeled sentiment analysis scores were examined for both positive and negative text fields. The GPT-4 positive text field sentiment analysis scores displayed a moderate correlation of 0.31 with the human-labeled scores, which is visualized in Figure 11. In comparison, the GPT-4 negative text field sentiment analysis scores exhibited a strong correlation of 0.91 with the human-labeled scores, as seen in Figure 12.

Additionally, the correlations between semester scores and GPT-4 sentiment analysis scores were assessed. The GPT-4 positive text field sentiment analysis scores revealed a weak positive correlation of 0.12 with semester scores as presented in Table 8. Similarly, the GPT-4 negative text field sentiment analysis scores demonstrated a weak positive correlation of 0.08 with semester scores.

Correlation	Positive Text Fields	Negative Text Fields
GPT-4 vs Human	0.31	0.91
Semester Score vs GPT-4	0.12	0.08

Table 8. GPT-4 Correlations.

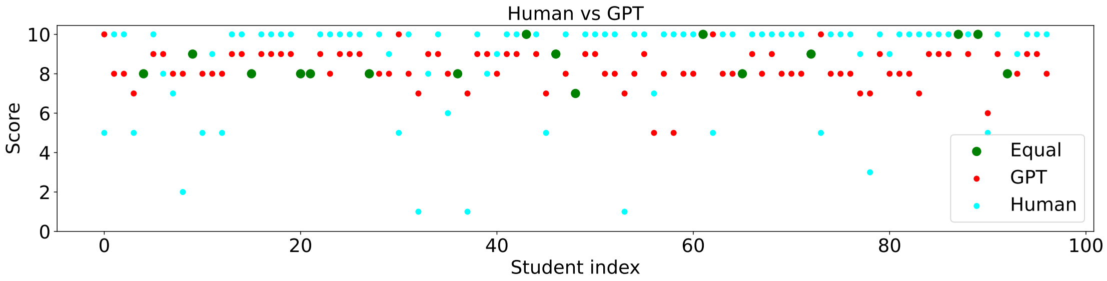


Figure 11. Scatter plot of human vs GPT-4 positive text field sentiment analysis scores.

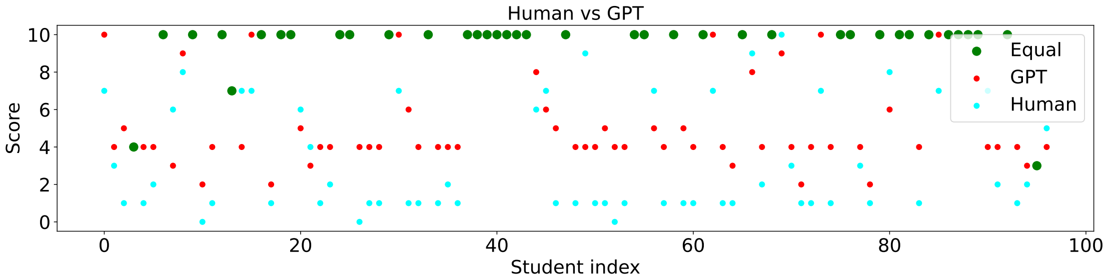


Figure 12. Scatter plot of human vs GPT-4 negative text field sentiment analysis scores.

These findings indicate that the GPT-4 model performs exceptionally well in determining the sentiment of negative text fields, aligning closely with human-assigned labels. However, the model's performance in identifying the sentiment of positive text fields could be improved, as evidenced by the moderate correlation with human labels.

In the instances where the difference in sentiment scores between the GPT models and human scorers was most significant, both GPT-3 and GPT-4 shared the same examples, as illustrated in Table 9. However, GPT-4's scores tended to be marginally closer to the human scores than GPT-3's. This could be indicative of GPT-4's improved ability to recognize and analyze nuances in sentiment, as it is a more advanced model.

Estonian	English	Human score	GPT score	Difference
Universaalsed ulesanded mida peaks igaksuks vahemalt proovima. Dokustaat kui ka opetamine/terminoloogia voiks ikkagist inglise keeles olla.	Universal tasks that everyone should try from a distance. The document as well as the teaching/terminology could still be in English.	1	7	+6
Esimesel nädalal oli suurem pusimine just programmide töölesaamisega, mitte nii väga ülesannete lahendamisega. Aga eks need juba järgmine nädal raskemad ole.	In the first week, the main focus was on getting the programs working, not so much on solving tasks. But I guess they will be harder next week.	1	7	+6

Table 9. GPT-4 and human positive sentiment score largest differences.

In the largest negative sentiment score differences, shown in Table 10, the first example discusses the lack of negative emotions and a difficult task. The human score is 4, while GPT-4 scores it 9, possibly focusing on the task's difficulty as negative. The second example, appearing in both GPT-3 and GPT-4 comparisons, shows GPT-4's score closer to the human score. This suggests GPT-4's improved architecture offers a more refined understanding of sentiment, indicating the potential for continuous advancements in natural language processing and sentiment analysis.

Estonian	English	Human score	GPT score	Difference
Praegu veel negatiivseid emotsioone teema ja ülesanded ei tekitanud, kuid neljas ülesanne tundus natuke raske enne abiõppejõu abi.	Currently, the topic and tasks did not cause negative emotions, but the fourth task seemed a bit difficult before the help of the assistant teacher.	4	9	+5
Poeemi ül oli algul natuke segane aga sain lõpuks aru.	The poem ex was a bit confusing at first, but I finally understood.	6	1	-5

Table 10. GPT-4 and human negative sentiment score largest differences.

Table 11 shows cases where GPT-4 and the human scorer agreed on positive sentiment scores, the texts generally described positive experiences or learning aspects, such as the excitement of watching programs work or enjoying solving tasks. Both GPT-4 and the human scorer assigned similar positive sentiment scores, indicating agreement on the positive nature of the experiences described.

Estonian	**English**	**Human score**	**GPT score**	**Difference**
Mulle meeldis, kui minu programmid töötasid, seda on väga põnev jälgida.	I liked, when my programs worked, it is very exciting to watch.	9	9	0
Huvitav oli kontrollida oma teadmisi praktikal ning meeldis pusida ülesandeid. Kõik läks hästi v.a. koodistiil, sellega peab natuke harjuma.	It was interesting to check my knowledge in practice and I liked to do tasks. Everything went well except code style, it takes some getting used to.	8	8	0

Table 11. GPT-4 and human positive sentiment score agreements.

Similarly, in cases where GPT-4 and the human scorer agreed on negative sentiment scores, the texts mentioned liking everything or used a winking emoticon, which typically conveys a positive sentiment, which can be seen in Table 12. Both GPT-4 and the human scorer assigned high scores (10), indicating that they both perceived the lack of negativity as a good thing.

Estonian	**English**	**Human score**	**GPT score**	**Difference**
;)	;)	10	10	0
Kõik meeldis.	Liked everything.	10	10	0

Table 12. GPT-4 and human negative sentiment score agreements.

These examples demonstrate that, while GPT-4 is an advanced language model, it may not always align with human perception in sentiment analysis. There can be challenges in assessing sentiment, as different aspects of the text can be emphasized or interpreted differently by the model and the human scorer.

Summary

The performance of GPT models in sentiment analysis is relatively impressive, despite the limited training data available in Estonian. As more data becomes accessible and the model evolves, its capabilities in sentiment analysis could further improve. GPT models effectively identify text fields without negative content, but it would be advantageous to focus more on detecting texts with a high degree of negative sentiment.

An observation across both GPT models is their hesitancy to assign very low numeric scores, which might suggest a need to modify the prompt for improved detection of low sentiment scores. GPT models display decreased accuracy in evaluating positive sentiment, possibly due to an overly optimistic interpretation of the text. Further investigation is necessary to determine the cause of this discrepancy. The lower accuracy in positive sentiment analysis could also be attributed to the presence of less constructive content in positive texts, despite them typically containing more content on average.

In most cases, GPT-4 exhibits enhanced performance in sentiment analysis compared to GPT-3, particularly when recognizing negative text fields. However, this improvement comes with a higher implementation cost. The rapid progress in the field of large language models, coupled with further prompt optimization, could result in a wider range of precise sentiment scores, substantially boosting the model's overall efficacy for sentiment analysis applications.

4.3 Grand survey analysis

After analyzing weekly questionnaires and exploring various labeling methods for text fields, the most effective approach was identified. In the grand survey analysis, GPT was employed from the beginning. Due to the complexity of the labeling tasks, which extended beyond merely determining the positivity of the content, the `gpt-3.5-turbo` model[27] was selected, although it does not support parameter setting. This model, an enhanced version of the `text-davinci-003`, is optimized for chat interactions, resulting in more consistent and accurate outcomes.

Of the 86 survey questions, 8 were in a text field response format, which GPT labeled. A scoring scale from 0 to 100 was utilized, with the national exam text field being the basis for this decision, since the scale for that is from 0 to 100. In addition to ensure superior detail, consistency, and accuracy in the labeling process. This scale facilitated a refined evaluation of the text fields, ultimately enhancing the overall analysis of the survey responses.

The methodology used to analyze text fields in the grand survey involved a systematic approach that comprised a list of prompts, which were subsequently processed through the GPT model. These prompts were specifically designed to elicit responses from GPT in a concise and structured format, while ensuring that GPT produced numeric scores that could be standardized and accurately evaluated. To this end, particular instructions were incorporated into the prompts to guide the responses, with the aim of improving the consistency, detail, and accuracy of the generated scores.

The list presented in Appendix 7.1 consists of questions to which the students provided text field responses and the corresponding prompts that were used as inputs to GPT. It is worth noting that GPT was primarily designed for chat applications, which could result in excessively lengthy responses. As such, the prompts were structured to encourage GPT to provide succinct and precise answers that would produce accurate and meaningful scores. Furthermore, GPT will not provide a score unless it has a complete understanding of the context, because of this, there were instances where it was unable to accurately determine a score. To address this, instructions were included in the prompts to approximate a score when GPT was unsure or could not determine one.

To facilitate the analysis of the text fields in the survey, the provided text and the text that GPT produced were enclosed in # symbols. This helped to standardize the data processing and analysis, which in turn allowed for a quantitative evaluation of the results. Each prompt has been designed to rate the students' responses on a specific metric, such as their experience with programming languages, their motivation for studying IT, or their ability to write code.

Examples

It is intriguing to note that the student's national mathematics exam result is initially recorded as a text field rather than a numeric value. This is occasionally justified, as some students may have taken the exam a long time ago and cannot recall their exact score or can only provide a range. In certain instances, students may not remember their math result but share other exam results instead. In these situations, GPT can extract at least some valuable information, making it a superior alternative to employing a hardcoded parser due to the numerous edge cases that may arise from human input. For instance, it would be difficult to cover all edge cases with a hardcoded parser when analyzing the responses depicted in Table 13:

Estonian	English	GPT score
$43 \leq x \leq 47$	$43 \leq x \leq 47$	45
hinne 5 (1995a)	grade 5 (in 1995)	90
"Ei mäleta, 80-90"	"Don't remember, 80-90"	85
"Keemias 58, Bioloogias 65"	"58 in Chemistry, 65 in Biology"	62

Table 13. Math exam GPT scores.

Table 14 offers a sample of students' prior programming experience derived from the grand survey. The data is organized into three columns: the original Estonian text, the English translation, and the corresponding GPT score. The GPT score aims to quantify each student's experience with programming languages, with higher scores signifying greater proficiency.

For instance, a student who has never worked with programming languages is assigned a GPT score of 0, while another who has completed a Python course in high school receives a score of 70. A student with minimal Python experience earns a score of 20, and one who has experience with Python, computer games and web pages is given a score of 80. By analyzing these GPT scores, a clearer understanding of the students' programming backgrounds can be achieved.

Estonian	English	GPT score
Python- Arvutimängu, veebilehti	Python- Computer games, web pages	80
Mitte kunagi	Never	0
Eriliselt ei ole, kui siis veidi Pythonit	Nothing special, except for a little Python	20
Python, läbisin gümnaasiumis ühe Pythoni kursuse.	Python, I took a Python course in high school.	70

Table 14. Experience with programming languages GPT scores.

4.4 Correlations

In this section, the correlations between the 2021 and 2022 features with respect to the semester score will be investigated, as it is seen as the most important factor for academic success. The reasoning for choosing the semester score as the target variable is its crucial role in meeting course requirements and being eligible for exams.

To support this choice, the correlations between the semester score and exam score for both years are following: 0.880 for 2021 and 0.853 for 2022. These numbers show a strong link between a student's performance during the semester and their exam results, suggesting that doing well in one often leads to doing well in the other. Additionally, the 2022 correlations related to students' weekly questionnaire feedback scores compared to the GPT-labeled scores will be looked in more detail. Finally, a summary of the findings will be offered to capture the understanding gained from this analysis.

4.4.1 2021 Features correlations

All 2021 features correlations with respect to the semester score can be seen in Appendix 7.1. As anticipated, weekly exercises play a significant role in determining the semester score since they constitute a large part of it. However, it is intriguing to observe the high importance of week 7 weekly assignment and week 6 weekly assignment, with correlations of 0.903 and 0.897 respectively. But higher placement for the larger test (KT) (correlation of 0.861) and the smaller test (TK) (correlation of 0.787) was expected.

Examining the grand survey questions, several noteworthy correlations emerged. These include students' perception of their performance compared to their classmates (Q41_14) (correlation of 0.505), students' expected grades (Q84_12) (correlation of 0.497), do students' think that the lecturer gives adequate time for understanding difficult tasks (Q55_28), and students' confidence in comprehending the most challenging assignments given by the lecturer (Q32_5).

In the context of the grand survey GPT-labeled text fields, the following features exhibited the strongest relationships, with their respective correlations in parentheses: coding tasks (Q81_9_GPT) (0.491), national mathematics exam results (Q73_1_GPT) (0.472), students' motivation for studying IT (Q77_5_GPT) (0.420), and students' prior experience with programming languages (Q74_3_GPT) (0.407).

An analysis of student attributes revealed that those in session study, on average, performed better. However, this outcome is likely due to the smaller number of students in this category.

No noteworthy correlations were found in relation to gender, micro-degree program, or study program code. Interestingly, age exhibited a negative correlation.

4.4.2 2022 Features correlations

All 2022 features correlations with respect to the semester score can be seen in Appendix 7.1. Similar to the analysis of 2021 features, weekly exercises in 2022 continue to play a significant role, with week 7 weekly assignment remaining the highest again.

Among the grand survey questions, three were the same as the previous year but appeared in a different order. The question which indicates whether or not students' think that the lecturer gives adequate time for understanding difficult tasks (`Q55_28`) was replaced with a question which pertains to students' expectations of their performance in the school year (`Q48_21`), which had a correlation of 0.297.

In the GPT-labeled grand survey text fields, the same top four questions were observed, with only the first two switching places. These responses may serve as good indicators for the future, since the list of most important features is identical for both years.

Analysis of student attributes again revealed high correlation with study form, students in session study again performed better on average. No correlation was found for gender and study program code. In addition to age, a negative correlation was observed for micro-degree program, suggesting that students pursuing a micro degree performed worse on average.

Since in the 2022 edition it was possible to connect of weekly questionnaires with student names, it was possible to analyze them. In the correlation table, each item is mapped with its corresponding week (e.g., week 10 is W10) as follows:

- **Feeling** - What was your mood last week regarding the course?
- **Learning useful topics** - Did you learn something useful about programming last week?
- **Pace** - How would you describe the pace of the course?
- **Exercise difficulty** - How would you rate the previous week's assignment on a 10 point scale?

- **GPT positive score** - GPT-assigned sentiment score for the response to: Positive thoughts and emotions regarding the previous week's topic and assignment.
- **GPT negative score** - GPT-assigned sentiment score for the response to: Negative thoughts and emotions regarding the previous week's topic and assignment.

Week 10 of weekly questionnaire feedback was particularly influential - questions regarding feeling and pace both had a strong correlation of about 0.64. In addition the consistently high GPT scores were impressive, for example week 10 GPT labelled positive and negative text fields had correlations of 0.625 and 0.604 respectively.

The 2022 data included information on the time students devoted to the course and their daily connection count, allowing for supplementary analysis. As expected, the later weeks proved more impactful, with the number of connections per day exhibiting a stronger correlation than the duration spent on the course. Furthermore, it was possible to determine the number of days a student has remained inactive, which exhibited a strong negative correlation with their semester score.

A more comprehensive examination entails comparing the student manual scores with the GPT scores to assess their alignment with the text fields provided by the students. Table 15 presents the correlations between students' weekly questionnaire feedback scores and GPT text field scores. The table displays the correlations for various aspects: how the student is feeling, whether the student learned something useful, the pace of the course and the exercise difficulty, along with the numeric scores that the GPT assigned to positive and negative text fields as target variables.

Feeling	Learning useful topics	Pace	Exercise difficulty	Target
0.103	-0.034	0.032	0.073	GPT_pos
0.349	0.033	0.238	0.308	GPT_neg

Table 15. Correlations of students weekly questionnaires feedback scores and GPT text field scores.

The analysis uncovers several insights:

- **Feeling** demonstrates a strong correlation with their written responses in the text field. This suggests that students who are feeling better tend to provide more positive feedback, whereas those who are not feeling as well are more likely to express negative sentiments.

- **Learning useful topics** does not seem to be substantially reflected in the text fields. This may indicate that students do not explicitly mention the usefulness of the content in their feedback, or it may suggest that other factors are more influential in shaping their sentiments.
- **Pace** exhibits some correlation with students' feedback. This could imply that students who find the pace of the course suitable are more likely to have positive sentiments, while those who struggle with the pace may express negativity in their responses.
- **Exercise difficulty** also plays a significant role in students' feedback. It is plausible that students who find exercises more challenging are more likely to express negative feelings in their responses.

Interestingly, the correlation is notably higher for negative text fields. This could be expected, as these fields provide students with an opportunity to express their frustrations and concerns, which might be more salient when they experience difficulties or dissatisfaction.

4.5 Sentiment Analysis Efficiency: Costs and Time Factors

To summarize, the 2021 grand survey included 259 entries. In 2022, by not anonymizing the weekly questionnaires data, it enabled an analysis that combined the weekly questionnaires with the grand survey. Over several weeks, these questionnaires accumulated 4,594 entries. A GPT-4-based experiment was conducted to label the first week's questionnaires, which had 546 entries. The 2022 grand survey had 505 entries. GPT was used to label 8 text fields in each grand survey entry and 2 text fields in each weekly questionnaire entry.

Identifying the most suitable prompts and models initially required some time, resulting in an experimentation cost of approximately $40 for assigning numeric values to text fields using various GPT models. Nevertheless, if the optimal prompts and models had been utilized from the beginning, scores would need to be allocated to the following:

259×8 ***(2021 Grand Survey)*** *+* 505×8 ***(2022 Grand Survey)*** *+* 4594×2 ***(2022 Weekly Questionnaires)*** *= 15,300 text fields.*

This would cost roughly $25 (approximately €22, depending on the exchange rate) and take about 1 hour to complete, even though GPT APIs occasionally face high traffic and produce errors. A retry mechanism effectively handles these issues. The efficiency of this process is especially notable when compared to the significant human effort required to manually perform the same task.

5. Predictions

In this chapter, the aim is to validate the prediction model using historical data from the fall semesters of 2021 and 2022, simulating realistic situations such as predicting student performance at the beginning or in the middle of the semester. Furthermore, the effectiveness of the model in a practical setting will be explored if it is implemented for the fall semester of 2023.

The focus is on predicting three key aspects of student performance:

1. **The likelihood of a student passing the course**, throughout the weeks
2. **The probability of a student passing the final exam**, assuming they will take it, at the end of semester
3. **The student's semester score**, throughout the weeks, given its strong correlation with the final exam score (as discussed in Section 4.4).

Accurate predictions of semester scores are crucial, as they enable the identification of students at risk of underperforming. Timely intervention with these students can lead to improved academic outcomes.

5.1 Exploration of different approaches

Several data sampling techniques were explored to address potential imbalances in the dataset and improve model performance. These techniques include:

- **Downsampling** - this technique involves reducing the number of majority-class samples to match the minority-class sample count. While downsampling can help balance the dataset, it may result in the loss of valuable information from the majority class.
- **Upsampling** - upsampling involves increasing the number of minority-class samples to match the majority-class sample count. This can be achieved through replication or interpolation of existing samples. While upsampling can lead to a

more balanced dataset, it might also introduce overfitting due to the repetition of minority-class samples.

- **Synthetic Minority Over-sampling Technique (SMOTE)**[30] - an advanced method that generates synthetic samples for the minority class by interpolating between existing minority-class instances. This technique enhances the dataset's balance without the drawbacks of simple replication seen in upsampling. However, SMOTE can still result in overfitting, especially if synthetic samples are generated too close to the decision boundary.

After evaluating these techniques, the decision was made to use the original, unsampled data in order to simulate a more realistic scenario. This approach acknowledges that real-world datasets may not always be balanced, and it is essential to develop a model that is robust and capable of handling such imbalances. Furthermore, utilizing the original data ensures that no information is lost or artificially introduced during the sampling process, leading to more reliable predictions.

The predictions experimentation began by employing the scikit-learn[31], which is a simple Python library and includes efficient tools for predictive data analysis. However, PyCaret[32], which is an open-source, low-code machine learning library in Python that automates machine learning workflows, was used. It was ultimately selected because of its remarkable capacity, user-friendly nature, and comprehensive functionality, which sets it apart from other alternatives, it also uses scikit-learn internally.

Neural networks[33] were also considered as a potential approach for predicting student performance, with the help of PyTorch[34], which is a machine learning framework and also has support for using it in Python. While they can be effective, there are some limitations to using neural networks in this context. The available dataset is relatively small, which can hinder model performance. Additionally, training neural networks can be time-consuming, and the process of tuning hyperparameters can be complex.

5.2 Model and feature selection

The process of model and feature selection for predicting students' academic performance requires finding a delicate balance between several factors. These factors include training accuracy, testing accuracy, and other performance metrics. Achieving this balance ensures that the model is neither overfitting nor underfitting the data, thereby optimizing its predictive capabilities.

5.2.1 Model selection

The Random Forest algorithm was chosen as the primary model for both classification and regression for this study, as it demonstrates exceptional performance in handling non-linear data. This ensemble learning method constructs multiple decision trees, ultimately yielding more accurate and stable predictions. Key advantages of utilizing the Random Forest algorithm include [19]:

- **Ultra-Scalability** - Random Forests are inherently parallelizable, making them well-suited for large-scale applications or high-performance computing environments.
- **Robustness Against Overfitting** - by averaging the results of numerous decision trees, Random Forests mitigate the risk of overfitting, thus enhancing model generalization.
- **Descriptive Power** - Random Forests provide insightful interpretations of the relationships between features and target variables, facilitating a deeper understanding of the underlying data.
- **Minimal Hyperparameter Tuning** - compared to other machine learning models, Random Forests require relatively little hyperparameter tuning, allowing for efficient model optimization without sacrificing valuable data resources.

While the Random Forest algorithm offers numerous benefits, it is essential to consider alternative models that may exhibit comparable performance in predicting students' academic performance. The following models were evaluated as potential candidates:

- **Extra Trees[35] (both regression and classifier)** - similar to Random Forests, the Extremely Randomized Trees algorithm constructs multiple decision trees. However, it employs a more randomized feature selection process, potentially resulting in improved generalization capabilities.
- **Linear Regression[21] (regression)** - as a simple yet powerful model, Linear

Regression can provide a solid baseline for performance comparison. It assumes a linear relationship between input features and the target variable, which may be adequate for certain applications.

- **Boosting Techniques (both regression and classifier)** - boosting methods, such as XGBoost[36] and AdaBoost[37], combine multiple weak learners into a single strong learner. These algorithms can often achieve high predictive accuracy, albeit at the cost of increased complexity and potential overfitting.

5.2.2 Feature selection

In this study, all features were included in the analysis, leveraging the Random Forest algorithm's capacity to handle potential noise independently. This choice was based on the algorithm's inherent ability to discern and concentrate on more informative features, while reducing the influence of less pertinent ones. Including all data yielded superior results compared to using only weekly assignments or a combination of weekly assignments and non-GPT labelled data.

The treatment of categorical data as either categorical or continuous was also examined. Ultimately, the decision was made to treat categorical data as continuous for the following reasons:

1. **Feature interactions** - when features are encoded, the algorithms may lose their ability to capture these interactions, resulting in decreased performance.
2. **Increased dimensionality** - Encoding categorical features, especially using one-hot encoding, significantly increased the dataset's dimensionality, which led to the "curse of dimensionality." This made it harder for the model to learn and generalize well, which caused reduction in performance.
3. **Noisy or less informative features** - Some categorical features did not contribute significantly to the target variable's prediction and it just introduced noise in the model.

More comprehensive analyses will be primarily conducted during the fourth week, as that period represents an optimal balance between obtaining sufficient high-quality data and providing timely assistance to students. This strategic timing allows for meaningful insights to be collected before the administration of smaller tests.

To enhance the model's reliability, a K-Fold Cross-Validation technique with k=10 was utilized, dividing the dataset into ten subsets for iterative training and validation, thereby minimizing overfitting risks. After the initial training, the model was fine-tuned. If this process improved performance, the updated model was adopted. Once training and fine-tuning were complete, the final model was trained on the entire training data and used to make predictions on unseen data. The accuracy and reliability of these predictions were then evaluated.

For regression, the following metrics were employed in the context of predicting students' semester scores:

1. **Mean Absolute Error (MAE)** - represents the average magnitude of absolute errors. In this context, it indicates the average deviation of predicted semester scores from the actual semester scores.
2. **MAE as %** - expresses the MAE value as a percentage. In this context, it reveals the proportion of the average error in predictions relative to the maximum possible semester score. For example when MAE is 60 and maximum possible semester score is 600, it implies that the average percentage deviation is 10%.
3. **Root Mean Square Error (RMSE)** - calculates the average of squared errors. In this context, it serves as a valuable metric because larger errors are especially undesirable.
4. **Coefficient of Determination (R2)** - measures the strength of the relationship between the model and the dependent variable. In this context, it demonstrates how well the model fits the data and can accurately predict semester scores.

In the context of predicting whether a student passes the exam or course, the following classification metrics are used, with examples based on a scenario where 80 out of 100 students pass the course:

1. **Accuracy** - represents the proportion of correctly classified instances out of the total instances. For example, if a model accurately predicts the pass rate of 80 out of 100 students, the accuracy is 0.8.
2. **Recall** - measures the ability of the model to correctly identify the positive cases

out of all the actual positive cases. For instance, if 80 students out of 100 passed and the model correctly identified 60 of them, the recall is 0.75.

3. **Precision** - measures the proportion of true positive predictions among all positive predictions made by a model. For example, if the model predicted 60 students to pass and 60 of them actually passed, the precision is 1.0, even though the model failed to identify 20 other students who passed the course.
4. **F1 Score** - offers a harmonic mean of precision and recall, providing a single metric that balances both aspects of the model's performance. In the context of the previous examples, the F1 score would be approximately 0.857.

5.3 Predictions of Academic Performance in 2021 and 2022 separately

Initially, predictions were conducted separately for the years 2021 and 2022. For each year, a Random Forest model was trained on 80% of the data using cross-validation, followed by academic performance predictions on the remaining 20% of the data. These predictions were made at various points during the course, including before the course started and at weeks 1, 4, 8, and 15. Additionally, at week 15, a prediction was made regarding whether a student would pass the exam or not. This specific prediction is particularly relevant at this stage because it helps assess the likelihood of success just before the final evaluations, enabling timely interventions if needed.

5.3.1 Predicting semester score

Table 16 displays the regression metrics used for evaluating the models' performance in predicting students' semester scores.

Dataset	Week	MAE	MAE as %	RMSE	R2
Training data: 80% of 2021 **Testing data:** 20% of 2021	**0**	103.3347	17.8379	126.0836	0.2038
	1	99.4929	17.1747	119.0058	0.2907
	4	73.2754	12.6490	89.4965	0.5989
	8	31.6381	5.4614	39.5548	0.9216
	15	15.2860	2.6334	19.5792	0.9590
Training data: 80% of 2022 **Testing data:** 20% of 2022	**0**	122.4135	21.4122	146.0559	0.2136
	1	105.4018	18.4365	125.6834	0.4176
	4	59.2445	10.3629	84.9282	0.7341
	8	33.5847	5.8745	51.4081	0.9026
	15	25.0026	4.2799	41.2802	0.8089

Table 16. Separate years semester score prediction metrics.

Overall, the predictions for the 2021 academic year exhibit better performance than those for the 2022 academic year, as evidenced by the lower MAE, MAE as %, and RMSE values and higher R2 scores. Several factors could contribute to this difference.

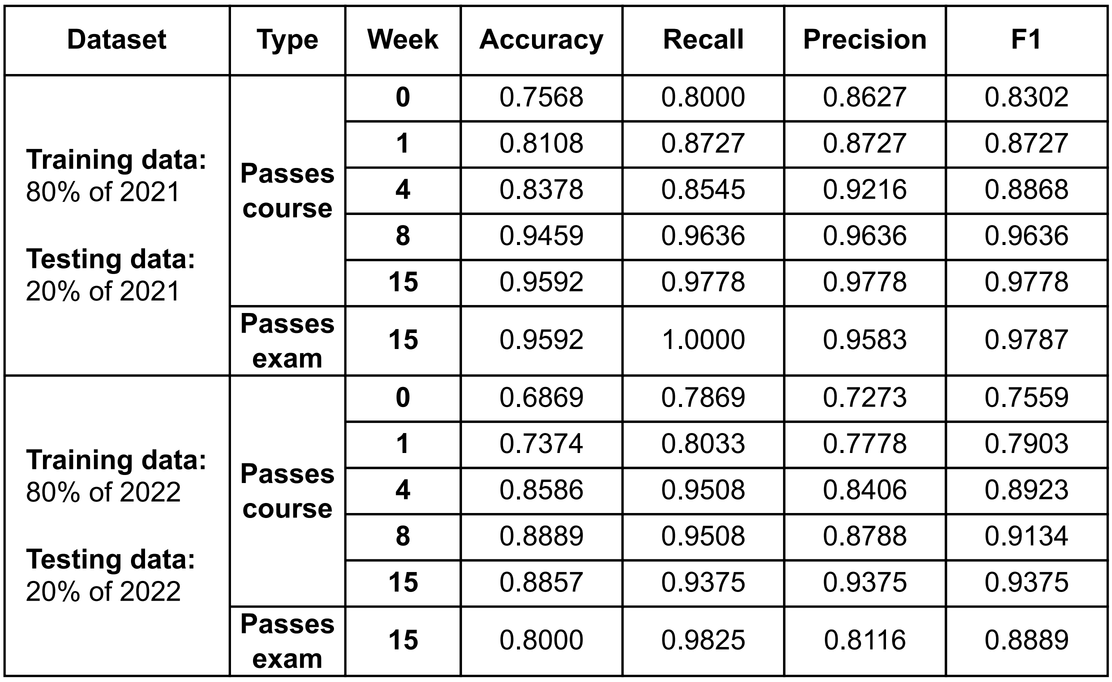
First, the 2022 dataset may inherently be more challenging due to differences in student populations, course structures, or other external factors that could impact student performance.

Second, the curse of dimensionality might play a role. Although the 2022 metrics were better at week 4, possibly due to the inclusion of weekly questionnaires and time spent on the course, the performance declined in week 15 compared to week 8. This may indicate that adding more features could have led to overfitting or increased complexity, which negatively impacted the model's performance in the later weeks.

Finally, the dataset for the 2022 academic year is considerably larger than that of 2021, which may also influence the results. Larger datasets often require more complex models or increased hyperparameter tuning to achieve optimal performance. Additionally, a larger dataset may include more noise or variations in the data, making it more challenging to capture the underlying patterns that contribute to accurate predictions.

5.3.2 Predicting passes the exam/course or not

Table 17 presents the classification metrics for predicting whether a student passes the course or not. An earlier thesis focusing on the same course during the 2021 academic year achieved an accuracy of 0.718 at the beginning of the course [5]. In contrast, the current work attained an accuracy of 0.7568. This improvement might be attributed to the utilization of PyCaret rather than scikit-learn, as PyCaret enables a more sophisticated training process and facilitates finer model tuning. Furthermore, predictions regarding whether a student would pass the exam were carried out using 15 weeks of data.



Dataset	Type	Week	Accuracy	Recall	Precision	F1
Training data: 80% of 2021 **Testing data:** 20% of 2021	**Passes course**	**0**	0.7568	0.8000	0.8627	0.8302
		1	0.8108	0.8727	0.8727	0.8727
		4	0.8378	0.8545	0.9216	0.8868
		8	0.9459	0.9636	0.9636	0.9636
		15	0.9592	0.9778	0.9778	0.9778
	Passes exam	**15**	0.9592	1.0000	0.9583	0.9787
Training data: 80% of 2022 **Testing data:** 20% of 2022	**Passes course**	**0**	0.6869	0.7869	0.7273	0.7559
		1	0.7374	0.8033	0.7778	0.7903
		4	0.8586	0.9508	0.8406	0.8923
		8	0.8889	0.9508	0.8788	0.9134
		15	0.8857	0.9375	0.9375	0.9375
	Passes exam	**15**	0.8000	0.9825	0.8116	0.8889

Table 17. Separate years pass rate prediction metrics.

The classification metrics, comprising Accuracy, Recall, Precision, and F1 score, exhibited a trend analogous to the regression analysis. Generally, the 2021 academic year demonstrated superior performance compared to the 2022 academic year, with the exception being week 4 again.

Predicting exam pass rate, the 2021 academic year produced impressive metrics, achieving an accuracy of 0.9592. In contrast, the 2022 academic year displayed a relatively weaker performance, with an accuracy of 0.8000. This discrepancy might be due to the accumulation of numerous features over the 15-week period in the 2022 dataset, leading to increased noise and hindering the model's accuracy.

5.4 Bidirectional Predictions of Academic Performance in 2021 and 2022

In this section, a bidirectional prediction approach is employed, similar to the workflow in Section 5.3. The entire 2021 dataset was used for training, while the complete 2022 dataset served as the testing data. Subsequently, the roles were reversed, with the full 2022 dataset used for training and the entirety of the 2021 dataset designated as testing data. Predictions were made during the same time periods, and results were evaluated using the same metrics. Additionally, a more in-depth analysis was conducted for week 4, providing feature importances, plots, and confusion matrices. This analysis was chosen for week 4 because, by this point, students have typically settled in, and there is sufficient data to make fairly accurate predictions. Furthermore, it is still early enough in the course to intervene and assist students who may be struggling.

5.4.1 Predicting semester score

In Table 18, it is evident that when using the 2022 dataset for training the model and predicting the 2021 semester scores, the performance metrics are generally better, with the exception of week 4.

Dataset	Week	MAE	MAE as %	RMSE	R2
Training data: 100% of 2021 **Testing data:** 100% of 2022	**0**	118.2232	17.7028	144.0387	0.1637
	1	111.8137	16.7431	135.0168	0.2652
	4	67.4304	10.0971	90.6169	0.6690
	8	36.2164	5.4231	54.8662	0.8787
	15	25.1258	3.7624	43.2379	0.7819
Training data: 100% of 2022 **Testing data:** 100% of 2021	**0**	116.9522	18.7483	144.2660	0.1944
	1	105.7065	16.9456	129.7028	0.3489
	4	70.2080	11.2549	92.9278	0.6657
	8	38.5580	6.1811	52.0387	0.8952
	15	22.6096	3.6245	30.3573	0.8969

Table 18. Bidirectional years semester score prediction metrics.

These results are somewhat anticipated, given that the dataset for the year 2022 is considerably larger, allowing the model to learn the data more comprehensively. One noteworthy observation is that by the 4th week, the predicted results exhibit a MAE as a percentage of approximately 10%, which corresponds to a difference of about one grade.

It is worth noting the improvement in MAE and RMSE values as the weeks progress. These reductions indicate that the model's predictions become more accurate as more information becomes available throughout the course. Although the R2 value increases from week 0 to week 8, it slightly decreases in week 15 when predicting the 2021 semester scores using the 2022 dataset for training. This could suggest that the model's predictive power slightly weakens towards the end of the course, possibly due to variations in grading systems.

Table 19 illustrates most important features for predicting students' semester scores on week 4.

2021 as training data					
Feature	EX04	EX03	Q76_4_GPT	EX02	Age
Importance	0.452904	0.389499	0.016368	0.015896	0.012613

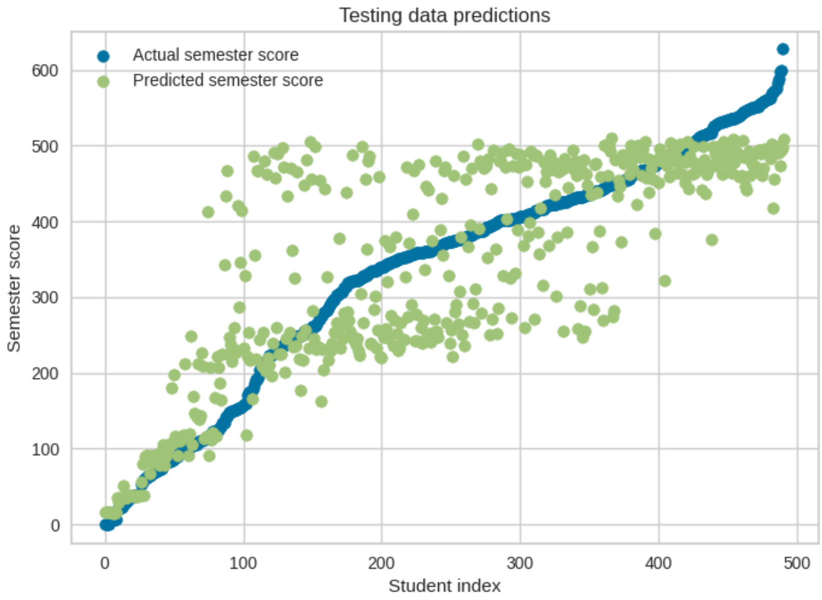
2022 as training data					
Feature	EX04	EX03	Q73_1_GPT	Q81_9_GPT	EX02
Importance	0.774731	0.071098	0.017776	0.011062	0.007163

Table 19. Predicting semester score on week 4 feature importances.

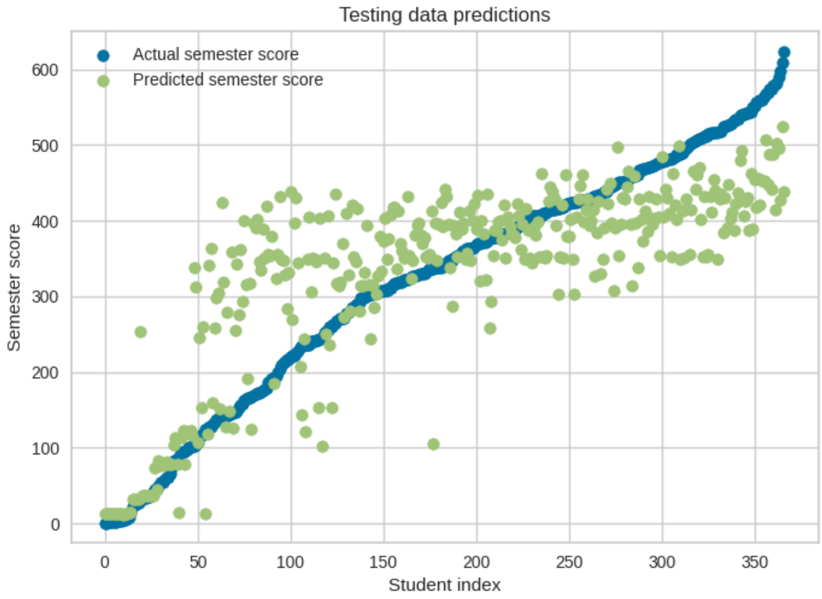
With year 2021 as training data, most important features are primarily the weekly exercises. Moreover, the table indicates that students' prior participation in olympiads (`Q76_4_GPT`) and their age are also significant factors. Although the last two are less significant than the weekly exercises, they should not be overlooked, as they may still provide valuable insights into a student's background and capabilities.

When using year 2022 as training data, the most crucial features are still primarily the weekly assignment results. However, now, the coding task (`Q81_9_GPT`) and national mathematics exam result (`Q73_1_GPT`) are significant factors. The increased presence in importance of grand survey features is likely due to the higher number of students participating in the grand survey in 2022 compared to 2021, as discussed in Subsection 3.4.4.

Figure 13 offers a detailed view of students' semester scores predictions with 4 weeks of data.



Using 2021 as training data and predicting 2022 semester scores.



Using 2022 as training data and predicting 2021 semester scores.

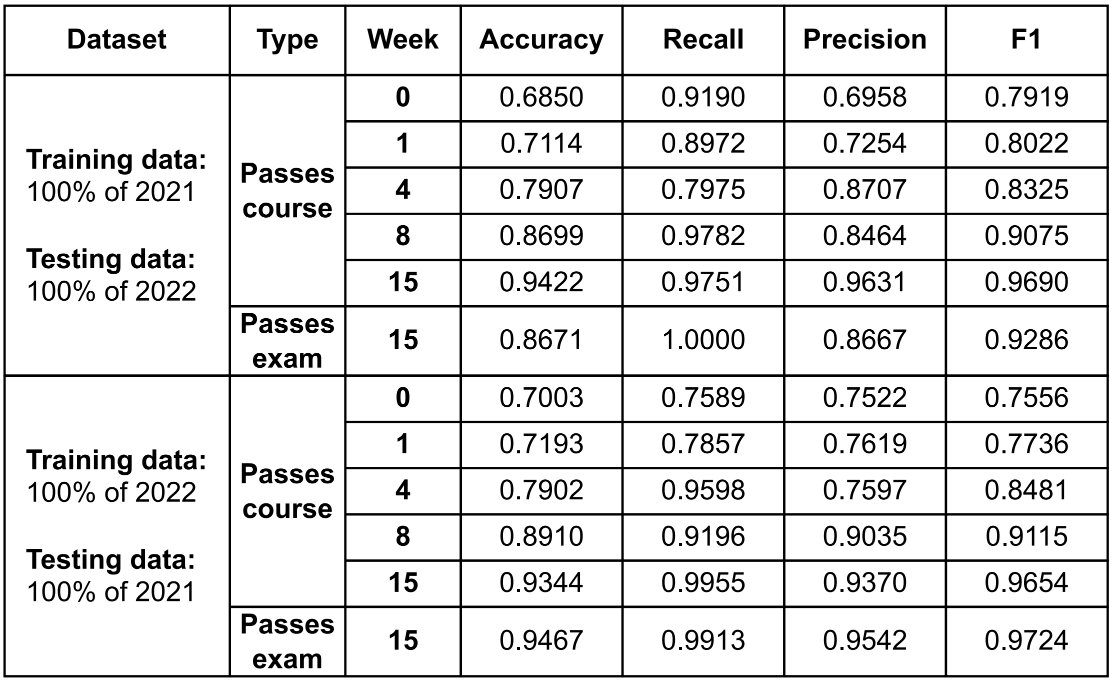
Figure 13. Bidirectional predictions semester scores plots with 4 weeks of data.

The plots exhibit general stability. When using the 2021 dataset for training, the most significant residuals are found in the 100-300 point range. Although the visual representation appears more precise with the 2022 dataset as training data and residuals don't seem as significant, the metrics still indicate a minor decrease in performance. This decline is likely due to the smaller size of the 2021 dataset, which is being used for testing, allowing outliers to exert a more substantial influence on the predictions.

5.4.2 Predicting passes the exam/course or not

In order to assess the performance of the classification model, a comparison is made with a baseline model that presumes all students pass the course. In the year 2021, there was a pass rate of 61.04%, which corresponds to a dummy classifier accuracy of 0.6104. The following year, 2022, saw a rise in the pass rate to 65.24%, and the dummy classifier accuracy increased accordingly to 0.6524.

In Table 20, classification performance metrics can be seen.



Dataset	Type	Week	Accuracy	Recall	Precision	F1
Training data: 100% of 2021 **Testing data:** 100% of 2022	**Passes course**	**0**	0.6850	0.9190	0.6958	0.7919
		1	0.7114	0.8972	0.7254	0.8022
		4	0.7907	0.7975	0.8707	0.8325
		8	0.8699	0.9782	0.8464	0.9075
		15	0.9422	0.9751	0.9631	0.9690
	Passes exam	**15**	0.8671	1.0000	0.8667	0.9286
Training data: 100% of 2022 **Testing data:** 100% of 2021	**Passes course**	**0**	0.7003	0.7589	0.7522	0.7556
		1	0.7193	0.7857	0.7619	0.7736
		4	0.7902	0.9598	0.7597	0.8481
		8	0.8910	0.9196	0.9035	0.9115
		15	0.9344	0.9955	0.9370	0.9654
	Passes exam	**15**	0.9467	0.9913	0.9542	0.9724

Table 20. Bidirectional predictions pass rate metrics.

Upon examining the provided classification performance metrics, it is evident that all predictions demonstrate a considerably improved performance compared to the dummy classifiers, even at the initial stages of the course. Furthermore, using 2022 as the base year yields superior results compared to using 2021 as training data. Interestingly, when using 2021 data, performance metrics drop significantly for predicting whether a student passes the exam or not, which can be attributed to the high recall value - indicating an overly optimistic model that predicts all students will pass.

Comparing the recall values for both years, there is a noticeable difference in the model's ability to correctly identify passing students. In 2021, the model is overly optimistic, especially at the beginning of the course (with a recall of 0.9190 in week 0).

Feature importances for both years when predicting students' pass rate with 4 weeks of data are presented in Table 21.

2021 as training data					
Feature	EX03	EX04	Age	Study program	EX02
Importance	0.349935	0.30351	0.067704	0.043085	0.017902
2022 as training data					
Feature	EX04	EX03	EX02	Study program	Q24_10
Importance	0.471593	0.156961	0.075398	0.019997	0.014199

Table 21. Predicting course pass rate on week 4 feature importances.

Noteworthily for year 2021 the grand survey features don't have any presence in top 5 most important features. In addition to weekly assignment results, age and study program play a huge factor. When using 2022 as training data, weekly assignment results are still very important, but, interestingly, age is replaced with students' belief that basic mathematical ability is practically unchangeable (`Q24_10`).

Confusion matrices for predicting students' pass rate on week 4 are shown in Table 22.

2021 as training data

		Prediction	
		Passes	Fails
Actual	Passes	133	38
	Fails	65	256

2022 as training data

		Prediction	
		Passes	Fails
Actual	Passes	75	68
	Fails	9	215

Table 22. Confusion matrices for predicting course pass rate on week 4.

When using 2021 as training data, the model exhibits a rather optimistic tendency, which has both positive and negative implications. On the one hand, it results in fewer false alarms, reducing the likelihood of mistakenly identifying students as needing help when they are actually performing well. On the other hand, this optimistic bias may lead to overlooking students who genuinely require assistance, leaving them unattended and potentially hindering their academic progress.

When using 2022 as training data, the model exhibits a more pessimistic tendency, which offers certain advantages over the optimistic approach observed with 2021 data. The pessimistic model is more effective in identifying students who are at risk of failing, as it successfully detects most of these students while generating false alarms for those who pass. Consequently, this approach ensures that all students in need of help are discovered, allowing for timely intervention and support. However, it is important to acknowledge that the false alarms generated for well-performing students may create unnecessary concern and consume resources that could be better allocated to those who genuinely require assistance. Nevertheless, in the current context, pessimistic tendency is generally preferred.

6. Application

The culmination of this master's thesis is the application, where all the components come together to deliver the final output. The source code for this application can be accessed on GitLab at henutt/master-thesis.

6.1 Overview

The application is unconventional, primarily comprising of endpoints and the PyCaret library for training models and predicting academic performance. Designed to remain lean and serve as a minimum viable product, the application does not rely on any databases or external resources. This streamlined approach simplifies maintenance and facilitates the decision-making process for future development, if necessary. For example, it would be possible to integrate some kind of external GPT resource into the application and implement sentiment analysis for text fields.

The application consists of three endpoints: send students data to get corresponding academic performance predictions, get supported features, get supported prediction types. The input data, containing information about students, can be submitted to the application in the form of CSV files. These files must adhere to the specific file structure identical to those downloaded from Moodle. The application handles the merging of different CSV files and performs necessary data transformations. At present, the application supports grand survey data, weekly questionnaires data, and weekly assignment results. In addition, the type of prediction must be specified in the request, such as semester score, whether the student passes the exam, or whether the student passes the course. Even though the application does not perform sentiment analysis on the text fields, it still supports text fields sentiment score features. Afterward, the application will generate a CSV file with merged input data and an additional column containing the predictions.

One of the strengths of this application is its flexibility; lecturers can submit various combinations of data, as long as the submitted data contains available features. Because the application is not designed for real-time processing, it can train the model on the fly within a minute. The versatility of the application can be achieved by keeping anonymised students' data CSV files from the 2021 and 2022 editions of the course on the disk as training data.

6.2 Workflow

Figure 14 provides a visual representation of the workflow process, which can be summarized in these general steps:

1. **Data Collection** - CSV files containing students' data will be downloaded from Moodle.
2. **Data Submission** - CSV files will be sent to the application via the designated endpoint for processing.
3. **Data Processing and Training** - The application will identify the features present in the incoming CSV files, filter the original training data to include only the features found in the incoming files, and train the Random Forest model using the filtered training data.
4. **Prediction** - The application will apply the trained model to predict students' academic performance based on the input data.

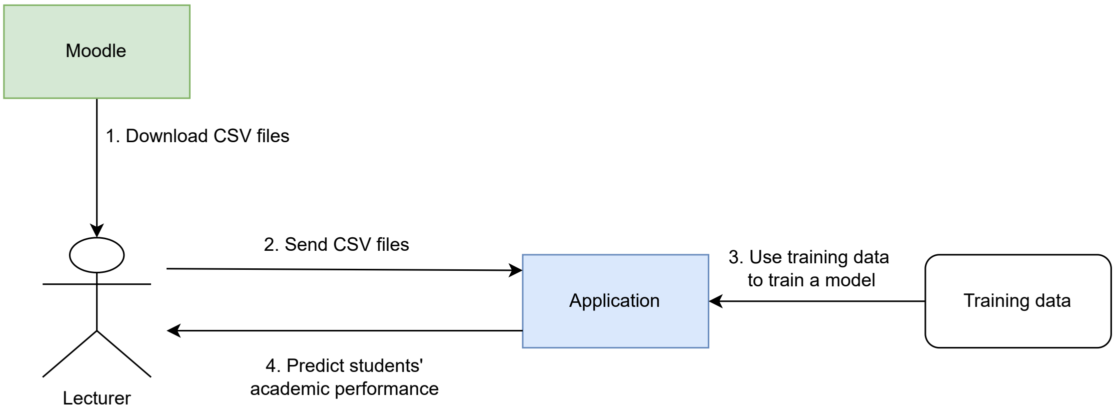

Moodle
1. Download CSV files
2. Send CSV files
Application
3. Use training data to train a model
Training data
4. Predict students' academic performance
Lecturer

Figure 14. Application workflow.

7. Summary

In this master's thesis, a major focus was placed on examining text fields from grand survey and weekly questionnaire responses, specifically by assigning a numeric value to represent the sentiment expressed in these text fields. The main goal was to turn the sentiment in these text fields into a quantifiable value and investigate its connection with students' academic success. To make sure the numeric scores were valid, the study compared them with academic scores and scores given by humans, showing that they were consistent and aligned.

Additionally, a thorough analysis of correlations was conducted to identify the factors with the most significant impact on students' academic performance. It's important to note that a strong correlation between a feature and academic success didn't always imply high importance. For example, age wasn't strongly linked to performance, but it still played an important role in the models' predictions.

The numeric values gathered from the sentiment analysis of text fields were combined with other features and used as input for models predicting academic success. Incorporating these sentiment scores, grand survey and weekly questionnaires responses contributed to more accurate predictions. However, at later parts of the course the number of weekly questionnaires variables accumulated, and the large quantity of features introduced additional complexity and noise. Overall, the consistency and performance measurements were outstanding across various types of predictions, such as semester score and dropout rate, as well as at different points in the course timeline. This high performance was observed in different years, including 2021 and 2022, and also when making bidirectional predictions for different years.

As a practical solution aimed at reducing the dropout rate in the course addressed in this thesis, an application was developed. The goal of this application is to assist lecturers in future iterations of the course by identifying students who may require additional support early in the learning process. By leveraging the insights gained from this study, educators can proactively address students' needs, and create a better learning environment.

The study aimed to examine three main hypotheses, each focusing on different aspects of the relationship between grand survey and weekly questionnaires responses, sentiment analysis using GPT models, and the prediction of students' academic performance based on limited data.

The first hypothesis proposed that there would be a strong correlation between the grand survey and weekly questionnaires responses and academic performance. The findings supported this hypothesis, as features from both the grand survey and weekly questionnaires placed quite high in terms of their correlation with academic performance.

The second hypothesis suggested that GPT models would assign sentiment analysis scores to texts that exhibit a stronger correlation with students' academic performance compared to human assessments. Although the results did not confirm this hypothesis, considering that the texts were in Estonian it performed reasonably well. Additionally, with improvements to the prompt and the continued development of these models, it is expected that their performance will only improve in the future.

The final hypothesis suggested that it would be possible to predict whether a student passes the course or not with over 80% accuracy using just a few weeks of data. The study found that with four weeks of data, the accuracy of predictions for year 2021 was 83% and for 2022 it was 85%. When conducting bidirectional predictions, the accuracy was slightly lower but still close, at around 79%. This indicates that the hypothesis was generally supported by the results.

The study aimed to address several research questions, exploring various aspects of predicting academic performance using limited data, the accuracy of bidirectional predictions when forecasting academic performance across different years, and the relationship between GPT sentiment analysis results and students' academic performance in comparison to human assessments.

The first research question focused on determining the accuracy of predicting academic performance using only a few weeks of data. The analysis showed that the prediction accuracy varied depending on the number of weeks considered. Starting from week 0, the accuracy was 68% and increased up to 85% by week 4.

The second research question aimed to assess the accuracy of bidirectional predictions when forecasting one year's academic performance using data from another year. The results revealed that the predictions were quite reliable. For instance, when predicting semester scores for the first four weeks, the deviation ranged from 118 to 67 points. Similarly, when predicting whether a student would pass or not for the same time period, the accuracy started at 68% on week 0 and increased to 79% by week 4.

The final research question investigated the connection between GPT sentiment analysis results and students' academic performance, as well as their correlation with human assessments. The findings showed that the connections were not particularly significant for either case. However, when GPT-4 labeled negative text fields, the scores were closely aligned with human assessments, exhibiting a correlation of 0.91.

7.1 Recommendations for the future

Regarding data collection, several improvements could be made to facilitate analysis. Firstly, simplifying the process can be achieved by presenting the questions in the grand survey and weekly questionnaires as numeric fields or lists of options rather than text fields, whenever it is possible and feasible. For instance, representing national math exam results and weekly time spent on the course in hours as numeric fields would be more efficient. Second, streamlining the data export process by enabling data export from a single source instead of merging multiple CSV files would save time and reduce errors. Third, it is crucial to ensure consistency across all data sources. In some cases, a student's final grade in the generated CSV file was inconsistent with the grade entered into the university system; addressing this issue would enhance data reliability.

Several important features present in the 2022 dataset were missing in the 2021 dataset. These features encompass days inactive, non-anonymous weekly questionnaires, time spent on Moodle and daily connections. Maintaining these features in future iterations and potentially incorporating additional relevant metrics from the course can result in more comprehensive and precise analyses.

Extra recommendations for future research and development include exploring the possibility of having human evaluators assign sentiment scores to all text fields, allowing for a more comprehensive comparison with machine-generated scores. Also, given the abundance of available features, particularly those from grand survey and weekly questionnaire fields, as they contain numerous attributes, it is essential to prioritize identifying and using only the most important ones. This prioritization will be helpful in enhancing academic performance predictions in future model developments. Finally, enhancing the developed application by integrating GPT model into the application also boosts its completeness and usefulness, eliminating the need for external labeling.

References

[1] K. Mardo. “Info- ja kommunikatsioonitehnoloogia erialade tudengite õpingud kõrgkoolis: esimesel aastal väljalangemine ja õpingute jätkamine”. MA thesis. 2016, pp. 11–12.

[2] K. Ehala. “Kontekstipõhine õppeedukuse monitoorimise mudel”. MA thesis. 2018, pp. 11–12.

[3] K. Sults. *Õpingute katkestamise põhjused ja õpingutega jätkamise motivatsioon*. 2015.

[4] H. Ots. “Õpisoorituse ennustamine Moodle’i logiandmete ja enesehinnanguliste õppimisega seotud psühholoogiliste tegurite põhjal”. MA thesis. 2020.

[5] E. S. Puudist. “Andmeaida ja masinõppe mudelite loomine üliõpilaste rühmitamiseks ja akdeemilise suutlikkuse ennustamiseks”. Bachelor’s Thesis. 2022.

[6] B. Uga. “TTÜ tudengite väljalangemise ennustamine: tõenäosuse arvutamine masinõppe meetodite abil ning tulemuste kuvamine veebirakenduses”. Bachelor’s Thesis. 2017.

[7] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. “Sentiment analysis algorithms and applications: A survey”. In: *Ain Shams engineering journal* 5.4 (2014), pp. 1093–1113.

[8] Ziniu Hu et al. “Gpt-gnn: Generative pre-training of graph neural networks”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 1857–1867.

[9] Charles Lang et al. *Handbook of learning analytics*. SOLAR, Society for Learning Analytics and Research New York, 2017.

[10] Doug Clow. “The learning analytics cycle: closing the loop effectively”. In: *Proceedings of the 2nd international conference on learning analytics and knowledge*. 2012, pp. 134–138.

[11] Zafar Iqbal et al. “Machine learning based student grade prediction: A case study”. In: *arXiv preprint arXiv:1708.08744* (2017).

[12] Lovenoor Aulck et al. "Predicting student dropout in higher education". In: *arXiv preprint arXiv:1606.06364* (2016).

[13] Danny Yen-Ting Liu et al. "An enhanced learning analytics plugin for Moodle: student engagement and personalised intervention". In: *ASCILITE 2015-Australasian Society for Computers in Learning and Tertiary Education, Conference Proceedings*. 2019.

[14] *Moodle Statistics*. https://huggingface.co/. Accessed: 2023-04-09.

[15] Miguel Ángel Conde et al. "Exploring student interactions: Learning analytics tools for student tracking". In: *Learning and Collaboration Technologies: Second International Conference, LCT 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2–7, 2015, Proceedings 1*. Springer. 2015, pp. 50–61.

[16] Zhi-Hua Zhou. *Machine learning*. Springer Nature, 2021.

[17] Vladimir Nasteski. "An overview of the supervised machine learning methods". In: *Horizons. b* 4 (2017).

[18] Sebastian Raschka. *Python machine learning*. Packt publishing ltd, 2015.

[19] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[20] Oliver Kramer and Oliver Kramer. "K-nearest neighbors". In: *Dimensionality reduction with unsupervised nearest neighbors* (2013), pp. 13–23.

[21] Alvin C. Rencher and William F. Christensen. "Chapter 10, Multivariate regression – Section 10.1, Introduction". In: *Methods of Multivariate Analysis*. 3rd ed. Vol. 709. Wiley Series in Probability and Statistics. John Wiley & Sons, 2012, p. 19. ISBN: 978-1-118-39167-9.

[22] Luciano Floridi and Massimo Chiriatti. "GPT-3: Its nature, scope, limits, and consequences". In: *Minds and Machines* 30 (2020), pp. 681–694.

[23] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[24] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[25] Israel Cohen et al. "Pearson correlation coefficient". In: *Noise reduction in speech processing* (2009), pp. 1–4.

[26] Alexis Conneau et al. "Unsupervised cross-lingual representation learning at scale". In: *arXiv preprint arXiv:1911.02116* (2019).

[27] *OpenAI*. https://platform.openai.com/docs/model-index-for-researchers. Accessed: 2023-02-28.

[28] *API Reference*. https://platform.openai.com/docs/api-reference/completions/create. Accessed: 2023-03-27.

[29] *OpenAI*. https://platform.openai.com/docs/models/gpt-4. Accessed: 2023-04-15.

[30] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[31] *scikit-learn*. https://scikit-learn.org/stable/. Accessed: 2023-04-20.

[32] *PyCaret*. https://pycaret.org/. Accessed: 2023-04-20.

[33] Kevin Gurney. *An introduction to neural networks*. CRC press, 2018.

[34] *PyTorch*. https://pytorch.org/. Accessed: 2023-04-22.

[35] Pierre Geurts, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees". In: *Machine learning* 63 (2006), pp. 3–42.

[36] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[37] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Hendrik Ütt

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Integrating Sentiment Analysis and Machine Learning to Predict Students’ Academic Performances in an Introductory Programming Course”, supervised by Ago Luberg
   1.1. 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;
   1.2. 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.

09.05.2023

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student’s application for restriction on access to the graduation thesis that has been signed by the school’s dean, except in case of the university’s right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - Survey structure

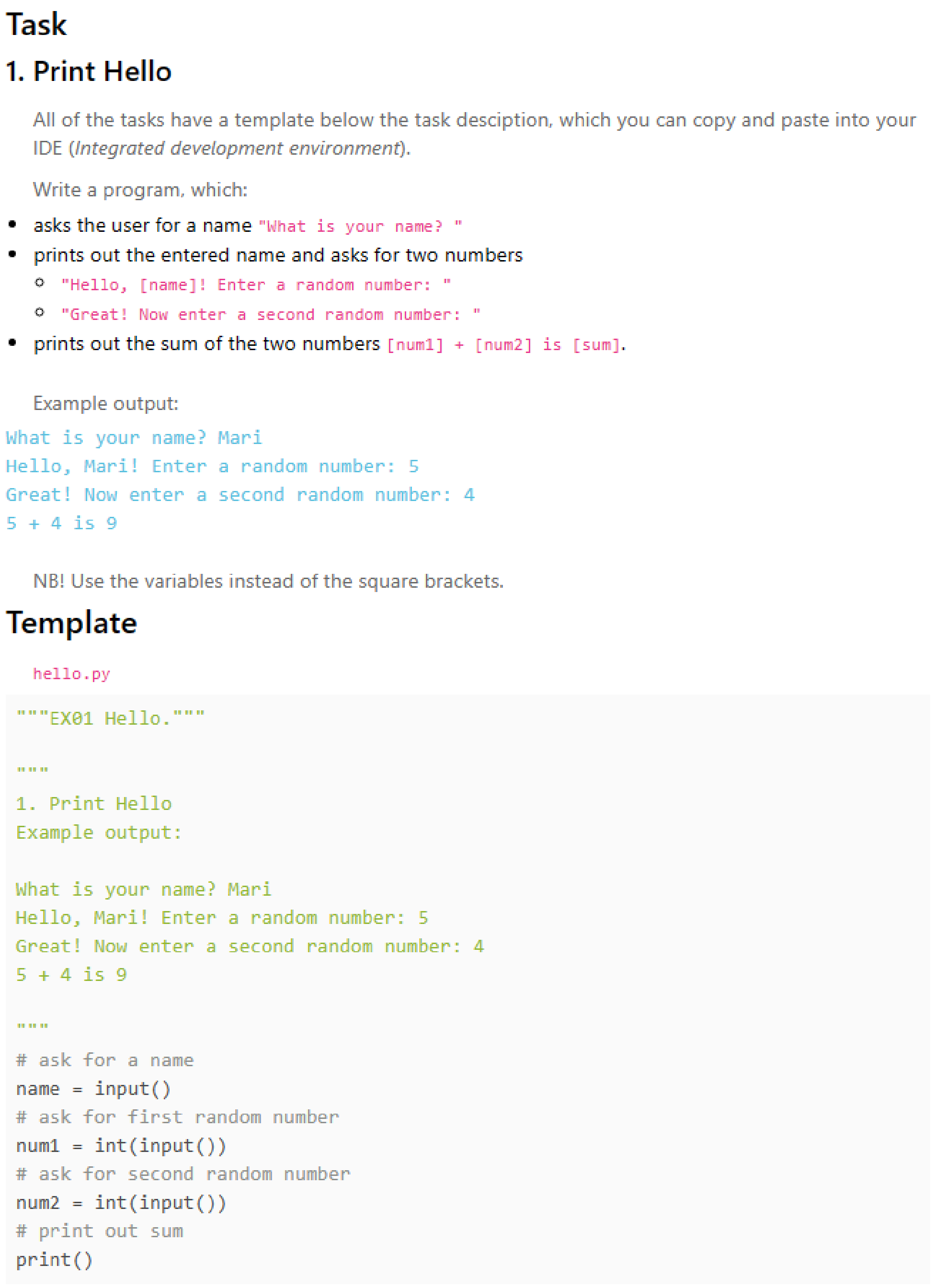
Question	Estonian	English
Prefix start	Kui olen aines tavapärasest rohkem pingutanud (vähe või üldse mitte puudunud, õigeaegselt kohal olnud, ka igavamate koduste töödega tegelenud jne), on see olnud seepärast, et ...	If I have put more effort into the subject than usual (not missing classes at all or being late, doing boring homework, etc.), it has been because ...
Q00_1	olen tahtnud õppejõule/vanematele rõõmu valmistada	I have wanted to make my lecturer/parents happy
Q00_2	olen tahtnud neist teemadest võimalikult palju teada saada	I have wanted to know as much as possible about these topics
Q01_3	need tunnid on olnud minu jaoks huvitavad	these lessons have been interesting for me
Q02_4	minu jaoks on olnud oluline saavutada häid õpitulemusi	it has been important for me to achieve good results
Q03_5	muidu oleks mul teiste ees (õppejõud/õpingukaaslased/vanemad) häbi olnud	otherwise I would have been ashamed in front of others (lecturer/students/parents)
Q04_6	see on lihtsalt olnud minu kui õppija ülesanne	it has just been my duty as a student
Q05_7	minu eesmärk on olnud olla hea õpilane	my goal has been to be a good student
Q06_8	mulle on meeldinud õppejõu/kaaslaste positiivne tagasiside	I have liked the positive feedback from my lecturer/students
Q07_9	muidu oleks ma tundnud piinlikkust, et ma piisavalt ei pinguta	otherwise I would have felt embarrassed that I am not trying hard enough
Q08_10	väärtustan õppimist (mis iganes õppeainega poleks tegemist)	I value learning (regardless of the subject)
Q09_11	olen neid tunde tõeliselt nautinud	I have really enjoyed these lessons
Q10_12	muidu oleksid teised (õppejõud/kaaslased/vanemad) minus pettunud olnud	otherwise others (lecturer/students/parents) would have been disappointed in me
Q11_13	arvan, et hea hariduse nimel tasub pingutada	I think that it is worth putting effort into a good education
Q12_14	muidu oleks ma tundnud piinlikkust, et olen teistest rumalam	otherwise I would have felt embarrassed that I am dumber than others
Q13_15	mulle on meeldinud, kui mind on teistest targemaks peetud	I have liked being considered smarter than others
Q14_16	olen pidanud seda õppeainet edasiste õpingute jaoks väga vajalikuks	I have considered this subject to be very necessary for my future studies
Prefix end		
Q15_1	Kui päris aus olla, ei saa inimene oma matemaatilise võimekuse taset muuta.	If you are really honest, a person cannot change his/her mathematical ability level
Q16_2	Inimene võib küll uusi oskuseid omandada, kuid suhtlemispädevuse baastase on praktiliselt muutmatu	A person can acquire new skills, but the basic level of communication skills is practically unchangeable
Q17_3	Inimese ärevuse tase on midagi, mida ei saa eriti palju muuta	A person's anxiety level is something that cannot be changed much
Q18_4	Igale inimesele on antud kindel annus matemaatilist võimekust ning seda on praktiliselt võimatu muuta	Every person has a certain amount of mathematical ability and it is practically impossible to change it
Q19_5	Kui õppija ei suuda lahendada programmeerimise algkursuse ülesannet sama kiiresti kui teised, on see ülesanne tema jaoks liiga raske.	If a student cannot solve programming course assignments as quickly as others, the assignment is too difficult for him/her
Q20_6	Matemaatiline võimekus on omadus, mida ei saa eriti muuta	Mathematical ability is a quality that cannot be changed much
Q21_7	Kui õppija ei suuda lahendada programmeerimise algkursuse ülesandeid sama kiiresti kui teised, tuleks talle anda lihtsamaid ülesandeid.	If a student cannot solve programming course assignments as quickly as others, he/she should be given easier assignments
Q22_8	Kui päris aus olla, ei saa inimene oma suhtlemispädevust muuta	If you are really honest, a person cannot change his/her communication skills
Q23_9	Igal inimesel on teatud ärevuse tase ja selle muutmiseks ei saa eriti midagi teha	Every person has a certain level of anxiety and it is not possible to change it much
Q24_10	Inimene võib küll uusi asju õppida, kuid matemaatilise võimekuse baastase on praktiliselt muutmatu	A person can learn new things, but the basic level of mathematical ability is practically unchangeable
Q25_11	Igal inimesel on teatud suhtlemispädevuse tase ja selle muutmiseks ei saa eriti midagi teha	Every person has a certain level of communication skills and it is not possible to change it much
Q26_12	Kui palju inimene ka ei pingutaks, ta ei saa oma ärevuse taset eriti muuta	How much ever a person tries, he/she cannot change his/her anxiety level much
Q27_13	Väga keeruliste programmeerimise algkursuse ülesannete puhul on lahenduse kallal pikemat aega pusimisest kasu vaid tõeliselt võimekatel õppijatel.	Very difficult programming course assignments are only useful for really talented students if they spend a long time on them
Q28_1	Kui mõni asi on algkursuse ülesannetes raske, teen parema meelega midagi muud	If something is difficult in the course, I would rather do something else
Q29_2	Tunnen end ärevana, kui teen raskete ülesannetega algkursuse kodutööd, mis tuleb järgmisel päeval esitada	I feel nervous when I do difficult assignments in the course, which I have to present the next day
Q30_3	Kui ma teen algkursuse ülesannetes mõne vea, kardan ma, et õppejõud ja õpingukaaslased peavad mind rumalaks	If I make a mistake in the course, I am afraid that my lecturer and classmates will think that I am stupid

Question	Estonian	English
Q31_4	Isegi kui tean, et mingist koolitööst pole pääsu, ei hakka ma sellega kunagi kohe pihta	Even if I know that there is no way out of some schoolwork, I will never start it right away
Q32_5	Olen kindel, et saan algkursusel aru ka keerulisematest ülesannetest, mis õppejõud meile annab	I am sure that I will understand even the most difficult assignments in the course that the lecturer will give us
Q33_6	Tunnen, et hakkan kaotama huvi oma koolitöö vastu tervenisti	I feel that I will lose interest in my schoolwork completely
Q34_7	Kui mõni asi on algkursuse aines raske, jätan selle pooleli	If something is difficult in the course, I leave it unfinished
Q35_8	Lahendan hea meelega algkursuse ülesandeid	I like solving assignments in the course
Q36_9	Programmeerimise algkursus on minu jaoks huvitav	Programming course is interesting to me
Q37_10	Ülikooliga seotud probleemide tõttu magan öösel sageli halvasti	I have trouble sleeping because of problems related to my university
Q38_11	Kui mul algkursuse ülesannetes midagi valesti läheb, muretsen ma, et olen õpingukaaslaste arvates rumal	If I make a mistake in the course assignments, I worry that I'm stupid compared to my classmates
Q39_12	Mulle meeldib lahendada algkursuse ülesandeid, millele leian kiiresti õiged vastused	I like solving assignments in the course that I can find the right answers quickly
Q40_13	Tunnen end ärevana, kui teen algkursusel hindelist tööd	I feel anxious when I do a graded assignment in my course
Q41_14	Arvan, et õpingukaaslastega võrreldes läheb mul algkursusel hästi.	I think I'm doing well in the course compared to my classmates
Q42_15	Tunnen end ärevana, kui mõtlen mõne algkursuse saabuva hindelise töö peale	I feel anxious when I think about an upcoming graded assignment in my course
Q43_16	Lükkan algkursuse koduste töödega alustamist nii pikalt edasi, et ei jõua neid tähtajaks valmis	I postpone starting my homework so long that I don't have time to finish it on time
Q44_17	Kui mul on raskusi, siis saan oma õppejõu abile loota	If I have difficulties, I can rely on my lecturer
Q45_18	Kahtlen pidevalt, kas mu koolitööl tervenisti on mingit mõtet	I constantly doubt whether my schoolwork has any meaning at all
Q46_19	Mõtisklen vabal ajal sageli oma ülikooliga seotud probleemide üle	I often think about my university-related problems when I have free time
Q47_20	Mulle meeldib lahendada algkursuse ülesandeid, mis on väga lihtsad, nii et saan palju õigeid vastuseid	I like solving easy tasks so that I can get many correct answers
Q48_21	Arvan, et mul läheb sel õppeaastal hästi	I think I will do well this school year
Q49_22	Õppejõud austab minu arvamust	The lecturer respects my opinion
Q50_23	Ma muretsen, et õppejõu meelest läheb mul halvemini kui õpingukaaslastel	I worry that the lecturer thinks I am doing worse than my classmates
Q51_24	Õppejõud mõistab tõeliselt, mida ma tunned	The lecturer understands what I feel
Q52_25	Koolitööst tulenev surve põhjustab mulle probleeme minu lähisuhetes	The pressure from schoolwork causes problems in my close relationships
Q53_26	Mulle meeldib lahendada algkursuse ülesandeid, mis on väga rasked, nii et saan ülesannete lahendamise kohta rohkem teada	I like solving difficult tasks in the course, so that I can learn more about solving the tasks
Q54_27	Tunnen end ärevana, kui pean algkursusel iseseisvalt keerukaid ülesandeid lahendama	I feel anxious when I have to solve difficult tasks on my own in the course
Q55_28	Õppejõud annab meile piisavalt aega, et ka keerukamatest ülesannetest aru saada	The lecturer gives us enough time to understand the more difficult tasks
Q56_29	Õppejõud näitab üles siirast huvi õppijate käekäigu suhtes	The lecturer shows sincere interest in the progress of the students
Q57_30	Tunnen end ärevana, kui algkursusel alustatakse uue teemaga	I feel anxious when the course starts a new topic
Q58_31	Mulle ei meeldi algkursusel osaleda	I don't like participating in the course
Q59_32	Programmeerimise algkursus on minu jaoks raske	The introduction to programming course is difficult for me
Q60_33	Mulle meeldib lahendada algkursuse ülesandeid, mille puhul vastuseni jõudmine võtab aega	I like solving the tasks of the course, which take time to get to the answer
Q61_34	Viivitan algkursuse koduste töödega alustamisega viimase hetkeni	I postpone the start of the homework of the course until the last moment
Prefix start	Olen programmeerimise algukursuseks õppides…	While studying for the introduction to programming course I have…
Q62_1	materjali korduvalt läbi lugenud	read the material several times
Q63_2	etteantud materjalis olulised kohad ära märkinud	marked the important parts of the material
Q64_3	vahetult enne eksamit hästi intensiivselt materjaliga tegelenud	studied intensively just before the exam
Q65_4	materjalist olulisemaid kohti teistele seletanud	explained the important parts of the material to others
Q66_5	püüdnud väga täpselt õppida just selle materjali järgi, mis on ette antud	tried to learn very precisely just the material that is given
Q67_6	püüdnud luua endale õpitava kohta seostatud terviku	tried to create a whole about the material that I am learning
Q68_7	erinevatelt kursuselt saadud teadmisi omavahel seostanud	connected the knowledge from different courses
Q69_8	ülesandeid näidise järgi korduvalt läbi lahendanud	solved the exercises several times according to the example
Q70_9	õpitava materjali õpikaaslasega läbi arutanud	I have discussed the material with my study partner
Q71_10	enne uue materjaliga tööle asumist üle vaadanud, kuidas see on osadeks jaotatud	before starting to work with new material, I have checked how it is divided into parts
Prefix end		
Q73_1_GPT	Matemaatika riigieksami tulemus	National mathematics exam result
Q74_3_GPT	Kas oled eelnevalt kasutanud mõnda programmeerimiskeelt? Nimeta need (ning mida oled nendega teinud)	Have you used any programming languages before? If yes, which ones and what have you done with them?
Q75_3_GPT	Kas oled eelnevalt läbinud mõne programmeerimiskursuse? Nimeta need (online kursused loevad ka)	Have you taken any programming courses before? If yes, which ones?

Question	Estonian	English
Q76_4_GPT	Kas oled osalenud olümpiaadidel (ükskõik mis erialal)? Nimeta need + tulemused.	Have you participated in any olympiads? If yes, which ones?
Q77_5_GPT	Miks sa tulid IT-d õppima?	Why did you come to study IT?
Q78_6_GPT	Kas mängid arvutimänge? Kui jah, siis milliseid?	Do you play computer games? If yes, which ones?
Q79_7	Nuputamisülesanne 1	Quiz task 1
Q80_8	Nuputamisülesanne 2	Quiz task 2
Q81_9_GPT	Koodi kirjutamise ülesanne	Coding task
Q82_10	Loogikaülesanne 1	Logic task 1
Q83_11	Loogikaülesanne 2	Logic task 2
Q84_12	Palun siin veel anda tagasiside oma oodatava tulemuse kohta. Mis hinde saad?	Please give feedback on your expected grade. What grade do you expect to get?
Q85_13_GPT	Tundub, et üks vabatekstiga küsimus sobiks ka. Igasugune kommentaar on siia oodatud.	Looks like one free text question would fit. Any kind of comment is welcome here.

Table 23. Grand Survey structure. Rows shaded in cyan are headers, in light gray are corresponding questions prefix start and ends, in yellow are text fields, which have been processed using GPT to extract numerical values from the textual data.

Appendix 3 - First Weekly Exercise



Task

1. Print Hello

All of the tasks have a template below the task desciption, which you can copy and paste into your IDE (*Integrated development environment*).

Write a program, which:

- asks the user for a name `"What is your name? "`
- prints out the entered name and asks for two numbers
  - `"Hello, [name]! Enter a random number: "`
  - `"Great! Now enter a second random number: "`
- prints out the sum of the two numbers `[num1] + [num2] is [sum]`.

Example output:

```
What is your name? Mari
Hello, Mari! Enter a random number: 5
Great! Now enter a second random number: 4
5 + 4 is 9
```

NB! Use the variables instead of the square brackets.

Template

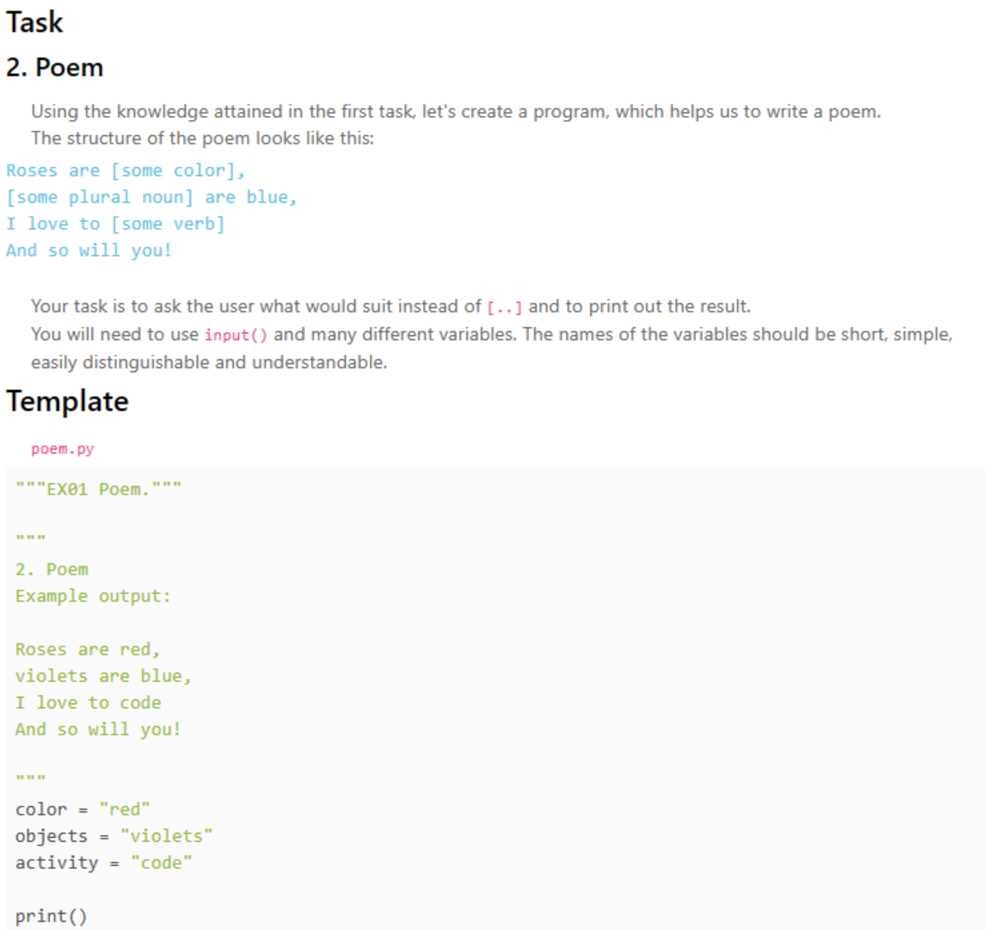
`hello.py`

```
"""EX01 Hello."""

"""
1. Print Hello
Example output:

What is your name? Mari
Hello, Mari! Enter a random number: 5
Great! Now enter a second random number: 4
5 + 4 is 9

"""
# ask for a name
name = input()
# ask for first random number
num1 = int(input())
# ask for second random number
num2 = int(input())
# print out sum
print()
```



Task

2. Poem

Using the knowledge attained in the first task, let's create a program, which helps us to write a poem.
The structure of the poem looks like this:

```
Roses are [some color],
[some plural noun] are blue,
I love to [some verb]
And so will you!
```

Your task is to ask the user what would suit instead of `[..]` and to print out the result.
You will need to use `input()` and many different variables. The names of the variables should be short, simple, easily distinguishable and understandable.

Template

poem.py

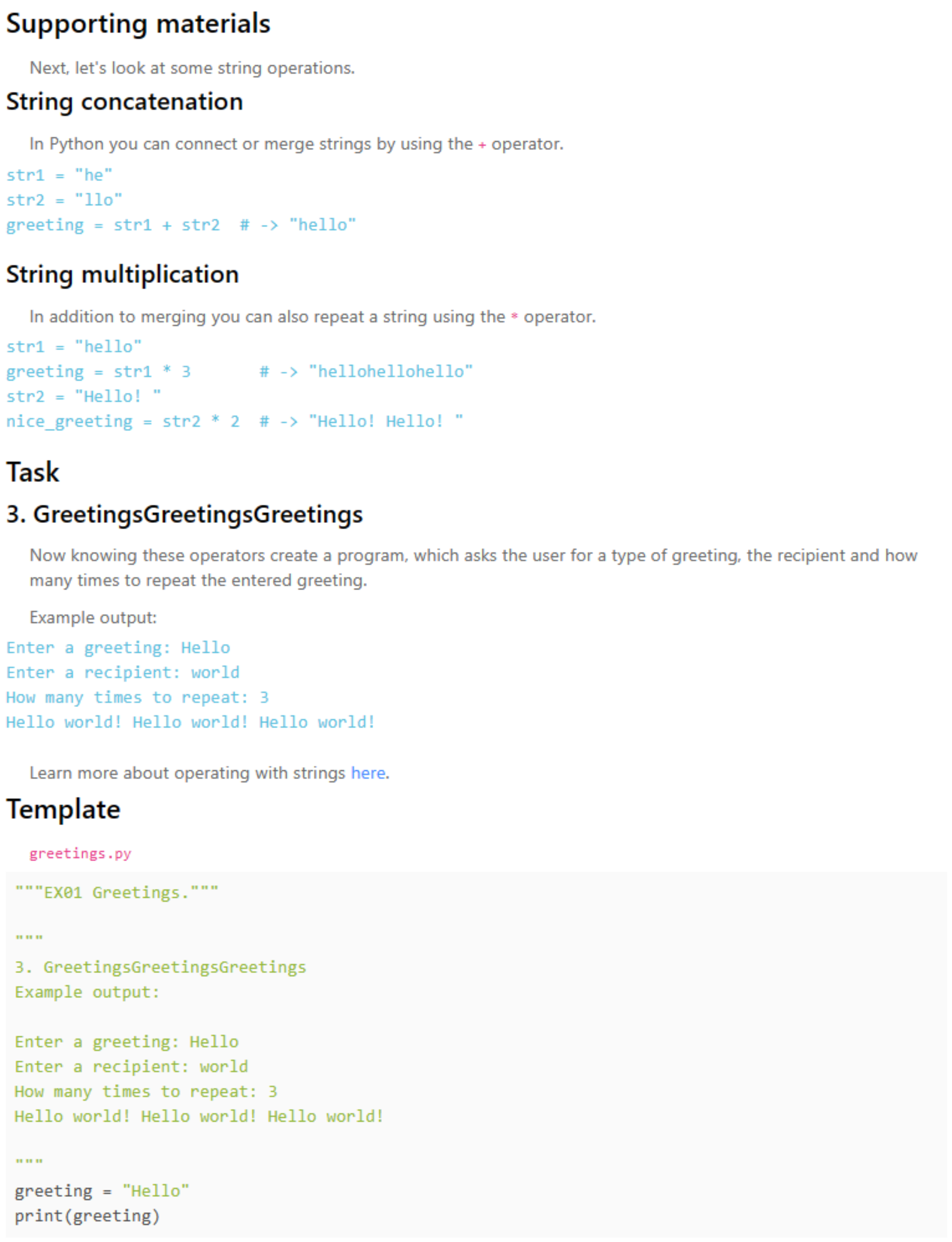
```python
"""EX01 Poem."""

"""
2. Poem
Example output:

Roses are red,
violets are blue,
I love to code
And so will you!

"""
color = "red"
objects = "violets"
activity = "code"

print()
```



Supporting materials

Next, let's look at some string operations.

String concatenation

In Python you can connect or merge strings by using the + operator.

```
str1 = "he"
str2 = "llo"
greeting = str1 + str2  # -> "hello"
```

String multiplication

In addition to merging you can also repeat a string using the * operator.

```
str1 = "hello"
greeting = str1 * 3       # -> "hellohellohello"
str2 = "Hello! "
nice_greeting = str2 * 2  # -> "Hello! Hello! "
```

Task

3. GreetingsGreetingsGreetings

Now knowing these operators create a program, which asks the user for a type of greeting, the recipient and how many times to repeat the entered greeting.

Example output:

```
Enter a greeting: Hello
Enter a recipient: world
How many times to repeat: 3
Hello world! Hello world! Hello world!
```

Learn more about operating with strings here.

Template

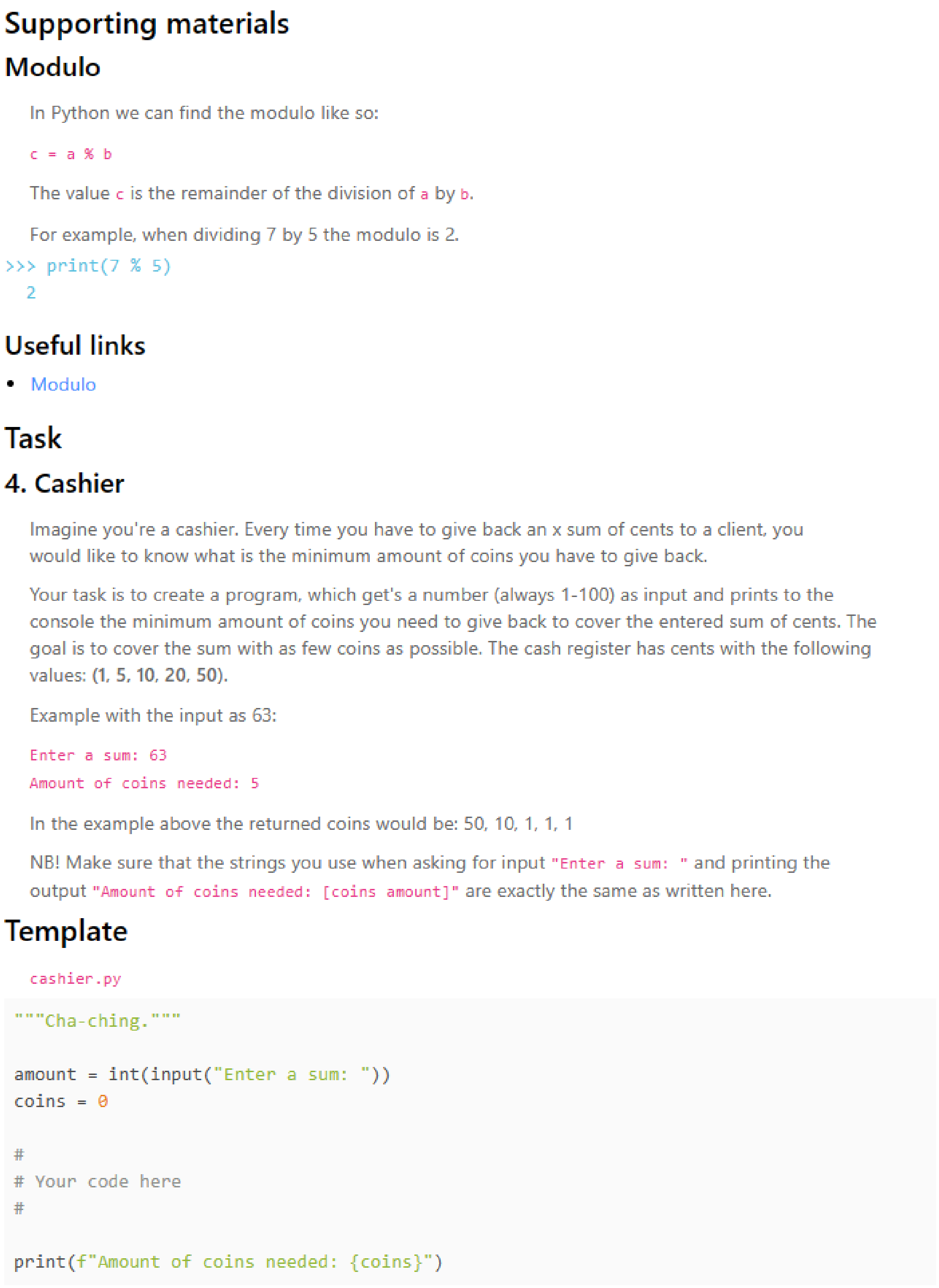
greetings.py

```
"""EX01 Greetings."""

"""
3. GreetingsGreetingsGreetings
Example output:

Enter a greeting: Hello
Enter a recipient: world
How many times to repeat: 3
Hello world! Hello world! Hello world!

"""
greeting = "Hello"
print(greeting)
```



Supporting materials

Modulo

In Python we can find the modulo like so:

```
c = a % b
```

The value `c` is the remainder of the division of `a` by `b`.

For example, when dividing 7 by 5 the modulo is 2.

```
>>> print(7 % 5)
  2
```

Useful links

- Modulo

Task

4. Cashier

Imagine you're a cashier. Every time you have to give back an x sum of cents to a client, you would like to know what is the minimum amount of coins you have to give back.

Your task is to create a program, which get's a number (always 1-100) as input and prints to the console the minimum amount of coins you need to give back to cover the entered sum of cents. The goal is to cover the sum with as few coins as possible. The cash register has cents with the following values: **(1, 5, 10, 20, 50)**.

Example with the input as 63:

```
Enter a sum: 63
Amount of coins needed: 5
```

In the example above the returned coins would be: 50, 10, 1, 1, 1

NB! Make sure that the strings you use when asking for input `"Enter a sum: "` and printing the output `"Amount of coins needed: [coins amount]"` are exactly the same as written here.

Template

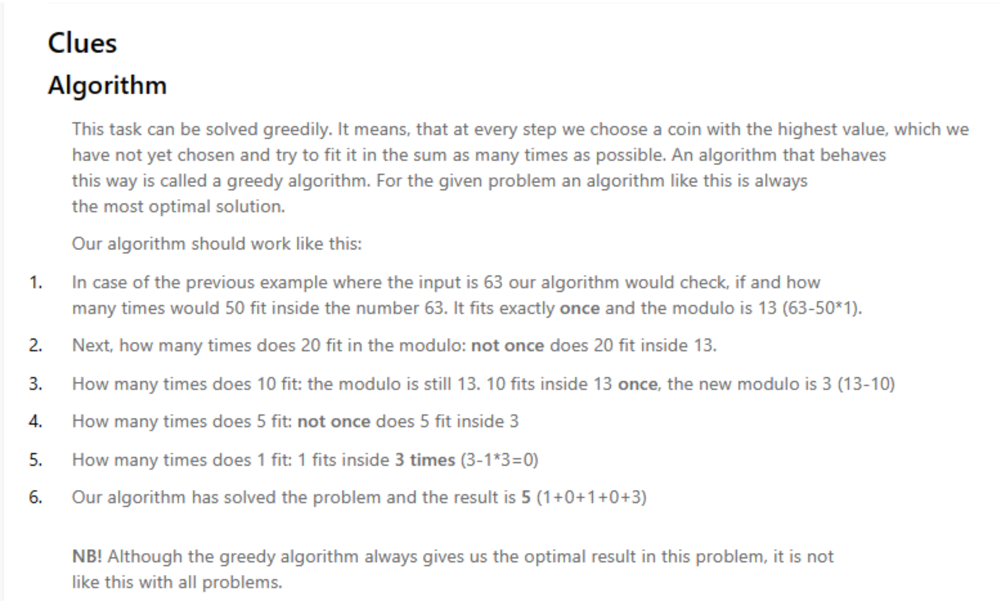
`cashier.py`

```
"""Cha-ching."""

amount = int(input("Enter a sum: "))
coins = 0

#
# Your code here
#

print(f"Amount of coins needed: {coins}")
```



Clues

Algorithm

This task can be solved greedily. It means, that at every step we choose a coin with the highest value, which we have not yet chosen and try to fit it in the sum as many times as possible. An algorithm that behaves this way is called a greedy algorithm. For the given problem an algorithm like this is always the most optimal solution.

Our algorithm should work like this:

1. In case of the previous example where the input is 63 our algorithm would check, if and how many times would 50 fit inside the number 63. It fits exactly **once** and the modulo is 13 (63-50*1).
2. Next, how many times does 20 fit in the modulo: **not once** does 20 fit inside 13.
3. How many times does 10 fit: the modulo is still 13. 10 fits inside 13 **once**, the new modulo is 3 (13-10)
4. How many times does 5 fit: **not once** does 5 fit inside 3
5. How many times does 1 fit: 1 fits inside **3 times** (3-1*3=0)
6. Our algorithm has solved the problem and the result is **5** (1+0+1+0+3)

NB! Although the greedy algorithm always gives us the optimal result in this problem, it is not like this with all problems.

Figure 15. First Weekly Exercise.

Appendix 4 – Grand Survey prompts

1. **Question:** *National mathematics exam result*

   Prompt: *Rate the student's National Math Exam result based on the text that I provide in Estonian. The text field should contain a numeric value ranging from 0 to 100, but sometimes the value may not be numeric. Assign a numeric value from 0 to 100, with 0 indicating the student scored 0 points and 100 indicating the student scored 100 points in the exam. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*
2. **Question:** *Have you used any programming languages before? If yes, which ones and what have you done with them?*

   Prompt: *Rate the student's experience with programming languages based on the text that I provide in Estonian from 0 to 100, where 0 means no experience and 100 means highly experienced. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*
3. **Question:** *Have you taken any programming courses before? If yes, which ones?*

   Prompt: *Rate the student's experience with programming courses based on the text that I provide in Estonian from 0 to 100, where 0 means no experience and 100 means highly experienced. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*
4. **Question:** *Have you participated in any olympiads? If yes, which ones?*

   Prompt: *Rate the student's performance in olympiads based on the text that I provide in Estonian from 0 to 100, where 0 means no participation and 100 means excellent results. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*
5. **Question:** *Why did you come to study IT?*

   Prompt: *Rate the student's motivation for studying IT based on the text that I provide in Estonian from 0 to 100, where 0 means not motivated and 100 means* *highly motivated. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*

6. **Question:** *Do you play computer games? If yes, which ones?*
   Prompt: *Rate the student's experience with computer games based on the text that I provide in Estonian from 0 to 100, where 0 means no experience and 100 means highly experienced. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*
7. **Question:** *Write a program that adds all even numbers in the range [0..n] (n > 0). If you can't write code, write an idea/explanation on how to solve it.*
   Prompt: *Rate the student's ability to write a program or explain the solution for summing odd numbers in the range [0..n] from 0 to 100, where 0 means no understanding and 100 means excellent understanding. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*
8. **Question:** *Looks like one free text question would fit. Any kind of comment is welcome here.*
   Prompt: *Rate the student's overall impression from the free-text comments based on the text that I provide in Estonian from 0 to 100, where 0 means no valuable input and 100 means highly valuable input. Return only the rating that you created in integer format, with # symbol being prefix and suffix for the number, when you don't know specific score, approximate. #TEXT#*

Appendix 5 - 2021 Feature correlations

Feature	Correlation	Feature	Correlation
EX07	0.903042	Q26_12	0.400022
EX11	0.898743	Q17_3	0.398383
EX06	0.897314	Q51_24	0.394130
EX08	0.894090	Q07_9	0.392350
EX09	0.892283	Q16_2	0.391419
EX12	0.882640	Q47_20	0.391240
EX13	0.865839	Q25_11	0.391166
KT	0.861013	Q12_14	0.388411
EX14	0.840706	Q10_12	0.386581
EX05	0.834335	Q67_6	0.386580
EX04	0.812202	Q18_4	0.385354
TK	0.787302	Q64_3	0.385213
EX03	0.759937	Q75_3_GPT	0.384047
EX02	0.689428	Q24_10	0.383499
EX15	0.649113	Q21_7	0.379050
EX01	0.559997	Q03_5	0.378789
Q41_14	0.504667	Q20_6	0.377416
Q84_12	0.496539	Study form	0.376876
Q81_9_GPT	0.491385	Q27_13	0.375516
Q55_28	0.475221	Q15_1	0.375003
Q32_5	0.474449	Q62_1	0.372323
Q53_26	0.473314	Q71_10	0.371046
Q73_1_GPT	0.472493	Q00_1	0.367098
Q02_4	0.471075	Q46_19	0.365212
Q48_21	0.469307	Q40_13	0.364927
Q80_8	0.462459	Q30_3	0.363601
Q60_33	0.462399	Q19_5	0.362344
Q79_7	0.456305	Q34_7	0.359791
Q35_8	0.455849	Q28_1	0.346278
Q82_10	0.454242	Q38_11	0.345752
Q05_7	0.450709	Q54_27	0.342702
Q11_13	0.450247	Q42_15	0.341945
Q56_29	0.448515	Q58_31	0.341793
Q36_9	0.446623	Q31_4	0.340523
Q00_2	0.442963	Q78_6_GPT	0.336925
Q14_16	0.434205	Q66_5	0.331206
Q44_17	0.433504	Q33_6	0.327202
Q09_11	0.431145	Q52_25	0.325751
Q49_22	0.430744	Q69_8	0.322628
Q01_3	0.430313	Q45_18	0.321956
Q04_6	0.423608	Q63_2	0.315962
Q77_5_GPT	0.420260	Q37_10	0.312462
Q70_9	0.420040	Q59_32	0.307252
Q83_11	0.419422	Q50_23	0.293688
Q39_12	0.418128	Q76_4_GPT	0.292064
Q22_8	0.410357	Q57_30	0.281344
Q68_7	0.409939	Q61_34	0.238610
Q65_4	0.408952	Q43_16	0.226528
Q08_10	0.408850	Q85_13_GPT	0.188311
Q74_3_GPT	0.407188	Gender	0.032019
Q29_2	0.406821	Micro degree program	-0.083282
Q06_8	0.406553	Study program	-0.102454
Q23_9	0.404711	Age	-0.219398
Q13_15	0.404541		

Table 24. Correlations of 2021 features with the semester score.

Appendix 6 - 2022 Feature correlations

Feature	Correlation	Feature	Correlation
EX07	0.913146	Exercise difficulty W13	0.556537
EX14	0.911876	Learning useful topics W13	0.551834
EX11	0.909445	Pace W4	0.550638
EX09	0.908758	Time spent on the course (min) W8	0.549517
EX13	0.907616	Pace W3	0.547387
EX06	0.882364	Learning useful topics W4	0.545824
EX08	0.873264	Feeling W13	0.542703
EX10	0.872776	Pace W7	0.539870
EX05	0.871250	Learning useful topics W7	0.536647
EX04	0.829531	GPT positive score W7	0.536614
KT	0.810110	Pace W14	0.533109
Connections per day W15	0.808594	Pace W15	0.533109
EX15	0.756810	GPT positive score W14	0.532132
EX03	0.732988	GPT positive score W15	0.532132
Connections per day W8	0.696555	GPT positive score W4	0.531288
Time spent on the course (min) W15	0.649450	Connections per day W4	0.527845
Feeling W10	0.648009	Feeling W3	0.525121
Pace W10	0.643740	Pace W8	0.524701
Exercise difficulty W10	0.633953	Feeling W7	0.524047
TK	0.633134	GPT positive score W8	0.522155
GPT positive score W10	0.624993	Exercise difficulty W4	0.520754
Exercise difficulty W9	0.610155	Learning useful topics W8	0.519711
GPT negative score W10	0.603617	Exercise difficulty W7	0.518978
Learning useful topics W9	0.601733	Exercise difficulty W8	0.518169
Feeling W5	0.601377	Feeling W14	0.513847
EX02	0.601174	Feeling W15	0.513847
Pace W5	0.599589	Exercise difficulty W14	0.513558
Learning useful topics W10	0.598206	Exercise difficulty W15	0.513558
Exercise difficulty W5	0.596024	Learning useful topics W14	0.512771
Exercise difficulty W11	0.594371	Learning useful topics W15	0.512771
Feeling W9	0.593763	Feeling W4	0.512579
Pace W11	0.592800	Feeling W8	0.512073
Feeling W12	0.590886	GPT negative score W11	0.501459
GPT positive score W12	0.589602	Exercise difficulty W3	0.498599
Exercise difficulty W12	0.589507	GPT negative score W9	0.485165
Pace W9	0.589108	GPT positive score W3	0.484885
Learning useful topics W11	0.587716	GPT negative score W13	0.461928
GPT positive score W5	0.587698	GPT negative score W12	0.459988
GPT positive score W9	0.586467	GPT negative score W14	0.451893
Feeling W11	0.585705	GPT negative score W15	0.451893
Learning useful topics W12	0.585443	GPT negative score W5	0.433848
Learning useful topics W5	0.585224	Learning useful topics W3	0.433316
Pace W12	0.584723	GPT negative score W6	0.432084
GPT positive score W11	0.581628	GPT negative score W8	0.424330
Pace W6	0.580584	Pace W2	0.420498
GPT positive score W6	0.577690	GPT negative score W7	0.408234
Learning useful topics W6	0.574744	Feeling W2	0.403628
Exercise difficulty W6	0.569665	EX01	0.399186
Feeling W6	0.565732	GPT negative score W4	0.387775
Pace W13	0.559969	Exercise difficulty W2	0.375600
GPT positive score W13	0.559061	GPT positive score W2	0.363197

Table 25. Correlations of 2022 features with the semester score 1/2.

Feature	Correlation	Feature	Correlation
GPT negative score W3	0.352942	Q10_12	0.210329
Pace W1	0.339981	Q08_10	0.208571
Q73_1_GPT	0.331209	Q39_12	0.208416
Q84_12	0.318419	Q75_3_GPT	0.203587
Q81_9_GPT	0.315713	Q40_13	0.202409
GPT negative score W2	0.311684	Q51_24	0.201065
Q32_5	0.304900	Q69_8	0.198268
Feeling W1	0.303266	Q47_20	0.196403
Q41_14	0.302229	Q62_1	0.192359
Q48_21	0.297291	Q00_1	0.185287
Learning useful topics W2	0.296116	Q03_5	0.184861
GPT negative score W1	0.295714	Q26_12	0.181720
Q05_7	0.295702	Q30_3	0.179938
Q79_7	0.289207	Q21_7	0.179438
Q11_13	0.276418	Q38_11	0.177747
Q82_10	0.275063	Q42_15	0.175601
Q44_17	0.274566	Q15_1	0.175498
Q80_8	0.274163	Q76_4_GPT	0.173305
Exercise difficulty W1	0.272689	Q24_10	0.165501
Study form	0.272065	Q18_4	0.165257
Q04_6	0.270813	Q46_19	0.163738
Q53_26	0.269291	Q17_3	0.163435
Q67_6	0.268411	Q23_9	0.161000
Q01_3	0.267109	Q20_6	0.159547
Q77_5_GPT	0.266582	Q52_25	0.159347
Q60_33	0.265451	Q16_2	0.158691
Q68_7	0.265239	Q27_13	0.158228
Q07_9	0.263413	Q25_11	0.157711
Q35_8	0.262431	Gender	0.155434
Q49_22	0.260581	Q22_8	0.154026
Q36_9	0.259955	Q54_27	0.152052
Q00_2	0.259292	Q37_10	0.148839
Q83_11	0.257325	Q34_7	0.147385
Time spent on the course (min) W4	0.256902	Q63_2	0.145345
Q65_4	0.256804	Learning useful topics W1	0.141915
Q14_16	0.255289	Q28_1	0.137000
Q70_9	0.253224	Q59_32	0.134042
Q02_4	0.252510	Q57_30	0.129166
Q06_8	0.244072	Q19_5	0.127437
GPT positive score W1	0.242212	Q31_4	0.122436
Q56_29	0.241625	Q50_23	0.116103
Q29_2	0.237981	Q33_6	0.104207
Q55_28	0.237752	Q58_31	0.103358
Q13_15	0.236202	Q45_18	0.088849
Q64_3	0.228625	Q43_16	0.075242
Q74_3_GPT	0.228616	Study program	0.056895
Q78_6_GPT	0.226294	Q61_34	0.050330
Q09_11	0.225535	Q85_13_GPT	0.010289
Q12_14	0.217136	Age	-0.162634
Q66_5	0.216108	Micro degree program	-0.207339
Q71_10	0.212097	Days inactive W16	-0.655054

Table 26. Correlations of 2022 features with the final score 2/2.